TOSHIBA

32bit TX System RISC
TX19A Family

TMP19A64C1DXBG

Rev1.1　2007.March.16

Contents

TMP19A64C1DXBG

32-bit RISC Microprocessor - TX19 Family

TMP19A64C1DXBG

1. Overview and Features

The TX19 family is a high-performance 32-bit RISC processor series that TOSHIBA originally developed by integrating the MIPS16™ASE (Application Specific Extension), which is an extended instruction set of high code efficiency.

TMP19A64 is a 32-bit RISC microprocessor with a TX19A processor core and various peripheral functions integrated into one package. It can operate at low voltage with low power consumption.

Features of TMP19A64 are as follows:

(1) TX19A processor core

1) Improved code efficiency and operating performance have been realized through the use of two ISA (Instruction Set Architecture) modes - 16- and 32-bit ISA modes.
   - The 16-bit ISA mode instructions are compatible with the MIPS16e-TX instructions of superior code efficiency at the object level.
   - The 32-bit ISA mode instructions are compatible with the TX39 instructions of superior operating performance at the object level.

2) Both high performance and low power consumption have been achieved.

● High performance

- Almost all instructions can be executed with one clock.
- High performance is possible via a three-operand operation instruction.
- 5-stage pipeline
- Built-in high-speed memory
- DSP function: A 32-bit multiplication and accumulation operation can be executed with one clock.

● Low power consumption

- Optimized design using a low power consumption library
- Standby function that stops the operation of the processor core

3) High-speed interrupt response suitable for real-time control

- Independency of the entry address
- Automatic generation of factor-specific vector addresses
- Automatic update of interrupt mask levels

(2) On Chip program memory and data memory

Product name	On chip ROM	On chip RAM
TMP19A64F20AXBG	2 Mbytes (Flash)	64 Kbytes
TMP19A64C1DXBG	1.5 Mbytes	56 Kbytes

- ROM correction function: 1 word × 8 blocks, 8 words × 4 blocks
- Backup RAM: 512 bytes

(3) External memory expansion

- 16-Mbyte off-chip address for code and date
- External data bus:
  Separate bus/multiplexed bus : Dynamic bus sizing for 8- and 16-bit widths ports.
- Chip select/wait controller : 6 channels

(4) DMA controller : 8 channels

- Data to be transferred to internal memory, internal I/O, external memory, and external I/O

(5) 16-bit timer : 11 channels

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG output
- Event capture function
- 2-phase pulse input counter function (1 channel assigned to perform this function):
  Multiplication-by-4 mode

(6) 32-bit timer

- 32-bit input capture register : 4 channels
- 32-bit compare register : 10 channels
- 32-bit time base timer : 1 channel

(7) Clock timer : 1 channel

(8) General-purpose serial interface: 7 channels

- Either UART mode or synchronous mode can be selected.

(9) Serial bus interface : 1 channel

- Either I^2C bus mode or clock synchronous mode can be selected

(10) 10-bit A/D converter with (S/H) : 24 channels

- Conversion speed: 54 clocks (7.85 μs@54 MHz)
- Start by an internal timer trigger
- Fixed channel/scan mode
- Single/repeat mode
- High-priority conversion mode
- Timer monitor function

(11) Watchdog timer : 1 channel

(12) Interrupt source

- CPU: 2 factors ..... software interrupt instruction
- Internal: 50 factors..... The order of precedence can be set over 7 levels (except the watchdog timer interrupt).
- External: 20 factors..... The order of precedence can be set over 7 levels (except the NMI interrupt).
  Because 8 factors are associated with KWUP, the number of interrupt factors is one.

(13) 209 pins Input/output ports

(14) Standby mode

- 4 standby modes (IDLE, SLEEP, STOP and BACKUP)

(15) Clock generator

- On-chip PLL (multiplication by 4)
- Clock gear function: The high-speed clock can be divided into 8/8, 7/8, 6/8, 5/8, 4/8, 2/8 or 1/8.
- Sub-clock: SLOW, SLEEP and BACKUP modes (32.768 kHz)

(16) Endian: Bi-endian (big-endian/little-endian)

(17) Maximum operating frequency

- 54 MHz (PLL multiplication)

(18) Operating voltage range

Core: 1.35 V to 1.65 V

I/O: 1.65 V to 3.3 V

ADC: 2.7 V to 3.3 V

Backup block : 2.3 V to 3.3 V (under normal operating conditions)
: 1.8 V to 3.3 V (in BACKUP mode)

(19) Package

- P-FBGA281 (13 mm × 13 mm, 0.65 mm pitch)

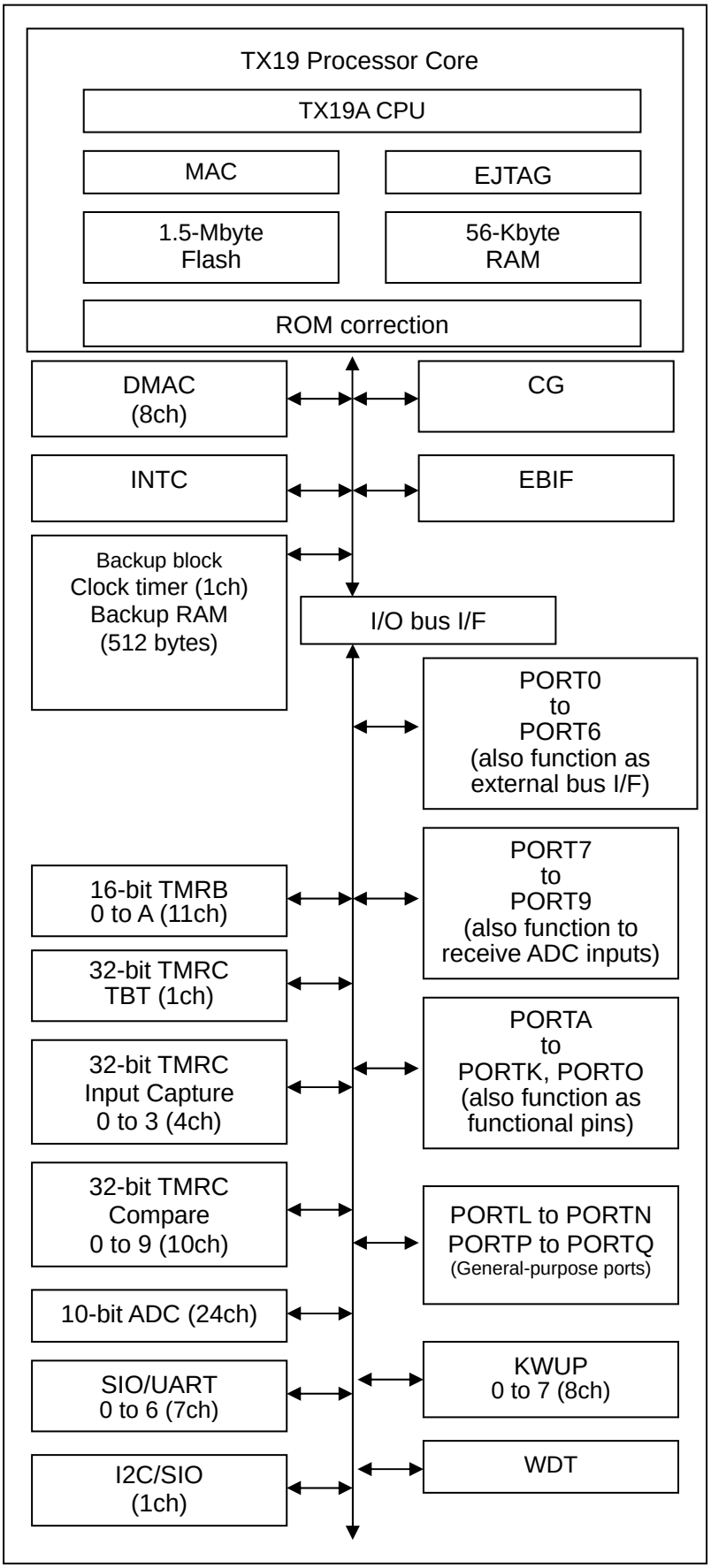


Fig. 1-1 TMP19A64C1DXBG Block Diagram

2. Pin Layout and Pin Functions

2.1 Pin Layout

Fig. 2.1.1 shows the pin layout of TMP19A64.

Fig. 2.1.1 Pin Layout Diagram (P-FBGA281)

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18
F1	F2	F3	F4	F5		F7	F8	F9	F10	F11	F12		F14	F15	F16	F17	F18
G1	G2	G3	G4	G5	G6							G13	G14	G15	G16	G17	G18
H1	H2	H3	H4	H5	H6							H13	H14	H15	H16	H17	H18
J1	J2	J3	J4	J5	J6							J13	J14	J15	J16	J17	J18
K1	K2	K3	K4	K5	K6							K13	K14	K15	K16	K17	K18
L1	L2	L3	L4	L5	L6							L13	L14	L15	L16	L17	L18
M1	M2	M3	M4	M5	M6							M13	M14	M15	M16	M17	M18
N1	N2	N3	N4	N5		N7	N8	N9	N10	N11	N12		N14	N15	N16	N17	N18
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18
	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	

Table 2.1.2 shows the pin numbers and names of TMP19A64.

Table 2.1.2 Pin Numbers and Names (1 of 2)

Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name
A1	N.C.	A13	PN2	B8	P75/AN5	C2	PCST3 (EJTAG)	C14	PM7
A2	VREFL	A14	PN0	B9	PL0	C3	P92/AN18	C15	PM3
A3	P90/AN16	A15	PM5	B10	PL3	C4	P95/AN21	C16	PK3/KEY3
A4	P93/AN19	A16	PM1	B11	PO5/TXD6	C5	P82/AN10	C17	CVCC15
A5	P80/AN8	A17	X2	B12	PO1/INT1	C6	P85/AN13	C18	XT2
A6	P83/AN11	B1	AVCC31	B13	PN3	C7	P72/AN2	D1	TDO (EJTAG)
A7	P70/AN0	B2	VREFH	B14	PN1	C8	AVSS	D2	PCST2 (EJTAG)
A8	P74/AN4	B3	P91/AN17	B15	PM4	C9	PL1	D3	$\overline{\text{DINT}}$ (EJTAG)
A9	PO7/SCLK6/CTS6	B4	P94/AN20	B16	PM0	C10	PL4	D4	DVCC15
A10	PL2	B5	P81/AN9	B17	CVSS/BVSS	C11	PO4/INT4	D5	P96/AN22
A11	PO6/RXD6	B6	P84/AN12	B18	X1	C12	PN6	D6	P86/AN14
A12	PO0/INT0	B7	P71/AN1	C1	PCST0 (EJTAG)	C13	PN4	D7	P73/AN3

Table 2.1.1 Pin Numbers and Names (2 of 2)

Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name
D8	DVCC15	F18	P46/SCOUT	K14	PI1/INT1	N18	P14/D12/AD12/A12	T8	PD4/TXD4
D9	DVSS	G1	$\overline{\text{RESET}}$	K15	PI3/INT3	P1	PE4	T9	PC0/TXD0
D10	PL5	G2	TDI (EJTAG)	K16	PI4/INT4	P2	PA2/TB0OUT	T10	PC3/TXD1
D11	PO3/INT3	G3	FVCC15	K17	DVCC30	P3	PA3/TB1IN0/INT7	T11	PH4/TCOUT8
D12	PN7	G4	DVSS	K18	PI2/INT2	P4	PA4/TB1IN1/INT8	T12	PH6
D13	PN5	G5	TOVR/TSTA (EJTAG)	L1	FVCC3	P5	PA5/TB1OUT	T13	P53/A3
D14	PM2	G6	BW0	L2	PQ1/TPD1/TPC1 (EJTAG)	P6	PB6/TBAIN0	T14	P61/A9
D15	DVCC34	G13	PK7/KEY7	L3	PQ2/TPD2/TPC2 (EJTAG)	P7	PG2/TC2IN	T15	P21/A17/A1/A17
D16	PK2/KEY2	G14	$\overline{\text{BRESET}}$	L4	PQ3/TPD3/TPC3 (EJTAG)	P8	PD6/SCLK4/$\overline{\text{CTS4}}$	T16	P23/A19/A3/A19
D17	PK4/KEY4	G15	P41/$\overline{\text{CS1}}$	L5	PE6/INTA	P9	PC2/SCLK0/$\overline{\text{CTS0}}$	T17	P00/D0/AD0
D18	XT1	G16	P37/ALE	L6	PE7/INTB	P10	PC5/SCLK1/$\overline{\text{CTS1}}$	T18	P01/D1/AD1
E1	DCLK (EJTAG)	G17	P35/$\overline{\text{BUSAK}}$	L13	P13/D11/AD11/A11	P11	P52/A2	U1	PB4/TB8OUT
E2	PCST1 (EJTAG)	G18	FVCC15	L14	P17/D15/AD15/A15	P12	P62/A10	U2	PB3/TB7OUT
E3	$\overline{\text{TRST}}$ (EJTAG)	H1	$\overline{\text{NMI}}$	L15	FVCC15	P13	P65/A13	U3	PB7/TBAIN1
E4	PCST4 (EJTAG)	H2	DVCC31	L16	PI0/INT0	P14	P26/A22/A6/A22	U4	PF1/SI/SCL
E5	ENDIAN	H3	PP7/TPD7 (EJTAG)	L17	P45/$\overline{\text{CS5}}$	P15	P02/D2/AD2	U5	PF5/$\overline{\text{DREQ3}}$
E6	P97/AN23	H4	BW1	L18	PJ3/$\overline{\text{DACK3}}$	P16	P10/D8/AD8/A8	U6	PG1/TC1IN
E7	P87/AN15	H5	$\overline{\text{PLLOFF}}$	M1	PQ0/TPD0/TPC0 (EJTAG)	P17	P12/D10/AD10/A10	U7	PD2/RXD3
E8	P76/AN6	H6	TCK (EJTAG)	M2	PQ7/TPD7/TPC7 (EJTAG)	P18	P11/D9/AD9/A9	U8	DVCC32
E9	P77/AN7	H13	TEST1	M3	PQ4/TPD4/TPC4 (EJTAG)	R1	PA0/TB0IN0/INT5	U9	PC7/RXD2
E10	PL6	H14	P31/$\overline{\text{WR}}$	M4	PE3	R2	PA1/TB0IN1/INT6	U10	PH1/TCOUT5
E11	PL7	H15	P32/$\overline{\text{HWR}}$	M5	PA7/TB3OUT	R3	PF3/$\overline{\text{DREQ2}}$	U11	PH5/TCOUT9
E12	PM6	H16	P33/$\overline{\text{WAIT}}$/$\overline{\text{RDY}}$	M6	DVCC32	R4	PF4/$\overline{\text{DACK2}}$	U12	P50/A0
E13	PK6/KEY6	H17	P30/$\overline{\text{RD}}$	M13	P06/D6/AD6	R5	PF7/TBTIN	U13	P55/A5
E14	PK5/KEY5	H18	P40/$\overline{\text{CS0}}$	M14	P07/D7/AD7	R6	PG7/TCOUT3	U14	DVCC33
E15	BVCC	J1	PP2/TPD2 (EJTAG)	M15	DVSS	R7	PG4/TCOUT0	U15	P64/A12
E16	PK1/KEY1	J2	PP3/TPD3 (EJTAG)	M16	PJ0/DREQ2	R8	PD5/RXD4	U16	P20/A16/A0/A16
E17	PK0/KEY0	J3	PP4/TPD4 (EJTAG)	M17	PJ2/$\overline{\text{DREQ3}}$	R9	PC1/RXD0	U17	P24/A20/A4/A20
E18	DVCC15	J4	PP5/TPD5 (EJTAG)	M18	PJ1/DACK2	R10	PC4/RXD1	U18	FVCC3
F1	DVSS	J5	PP6/TPD6 (EJTAG)	N1	PE5	R11	PH3/TCOUT7	V2	PB5/TB9OUT
F2	TMS (EJTAG)	J6	FVCC15	N2	PE0/TXD5	R12	P51/A1	V3	PG0/TC0IN
F3	$\overline{\text{EJE}}$ (EJTAG)	J13	DVSS	N3	PE2/SCLK5/$\overline{\text{CTS5}}$	R13	P57/A7	V4	PF0/SO/SDA
F4	BUSMD	J14	P47	N4	PE1/RXD5	R14	P66/A14	V5	PG3/TC3IN
F5	$\overline{\text{BOOT}}$	J15	N.C.	N5	PA6/TB2OUT	R15	P25/A21/A5/A21	V6	PG6/TCOUT2
F7	AVSS	J16	P44/$\overline{\text{CS4}}$	N7	DVSS	R16	P03/D3/AD3	V7	PD1/TXD3
F8	AVSS	J17	P36/ R/W	N8	PD7/INT9	R17	P04/D4/AD4	V8	PD0/SCLK2/$\overline{\text{CTS2}}$
F9	AVCC32	J18	P34/$\overline{\text{BUSRQ}}$	N9	DVCC15	R18	P05/D5/AD5	V9	PC6/TXD2
F10	DVCC34	K1	PP0/TPD0 (EJTAG)	N10	DVSS	T1	PB0/TB4OUT	V10	PH2/TCOUT6
F11	PO2/INT2	K2	PP1/TPD1 (EJTAG)	N11	P56/A6	T2	PB1/TB5OUT	V11	PH0/TCOUT4
F12	DVSS	K3	PQ5/TPD5/TPC5 (EJTAG)	N12	DVSS	T3	PB2/TB6OUT	V12	PH7
F14	$\overline{\text{BUPMD}}$	K4	PQ6/TPD6/TPC6 (EJTAG)	N14	P27/A23/A7/A23	T4	PF2/SCK	V13	P54/A4
F15	P42/$\overline{\text{CS2}}$	K5	DVSS	N15	P15/D13/AD13/A13	T5	PF6/$\overline{\text{DACK3}}$	V14	P60/A8
F16	P43/$\overline{\text{CS3}}$	K6	DVSS	N16	TEST3	T6	PG5/TCOUT1	V15	P63/A11
F17	DVCC33	K13	TEST2	N17	P16/D14/AD14/A14	T7	PD3/SCLK3/$\overline{\text{CTS3}}$	V16	P67/A15
								V17	P22/A18/A2/A18

2.2 Pin Names and Functions

Table 2.2.1 shows the names and functions of input/output pins.

Table 2.2.1 Pin Names and Functions (1 of 6)

Pin name	Number of pins	Input or output	Function
P00-P07	8	Input/output	Port 0: Input/output port that allows input/output to be set in units of bits
D0-D7		Input/output	Data (lower): Data buses 0 to 7 (separate bus mode)
AD0-AD7		Input/output	Address data (lower): Address data buses 0 to 7 (multiplexed bus mode)
P10-P17	8	Input/output	Port 1: Input/output port that allows input/output to be set in units of bits
D8-D15		Input/output	Data (upper): Data buses 8 to 15 (separate bus mode)
AD8-AD15		Input/output	Address data (upper): Address data buses 8 to 15 (multiplexed bus mode)
A8-A15		Output	Address: Address buses 8 to 15 (multiplexed bus mode)
P20-P27	8	Input/output	Port 2: Input/output port that allows input/output to be set in units of bits
A16-A23		Output	Address: Address buses 16 to 23 (separate bus mode)
A0-A7		Output	Address: Address buses 0 to 7 (multiplexed bus mode)
A16-A23		Output	Address: Address buses 16 to 23 (multiplexed bus mode)
$\overline{P30}$	1	Output	Port 30: Port used exclusively for output
$\overline{RD}$		Output	Read: Strobe signal for reading external memory
$\overline{P31}$	1	Output	Port 31: Port used exclusively for output
$\overline{WR}$		Output	Write: Strobe signal for writing data of D0 to D7 pins
$\overline{P32}$	1	Input/output	Port 32: Input/output port (with pull-up)
$\overline{HWR}$		Output	Write upper-pin data: Strobe signal for writing data of D8 to D15 pins
$\overline{P33}$	1	Input/output	Port 33: Input/output port (with pull-up)
$\overline{WAIT}$		Input	Wait: Pin for requesting CPU to put a bus in a wait state
$\overline{RDY}$		Input	Ready: Pin for notifying CPU that a bus is ready
$\overline{P34}$	1	Input/output	Port 34: Input/output port (with pull-up)
$\overline{BUSRQ}$		Input	Bus request: Signal requesting CPU to allow an external master to take the bus control authority
$\overline{P35}$	1	Input/output	Port 35: Input/output port (with pull-up)
$\overline{BUSAK}$		Output	Bus acknowledge: Signal notifying that CPU has released the bus control authority in response to BUSRQ
$\overline{P36}$	1	Input/output	Port 36: Input/output port (with pull-up)
R/$\overline{W}$		Output	Read/write: "1" shows a read cycle or a dummy cycle. "0" shows a write cycle.
P37	1	Input/output	Port 37: Input/output port
ALE		Output	Address latch enable (address latch is enabled only if access to external memory is taking place)
$\overline{P40}$	1	Input/output	Port 40: Input/output port (with pull-up)
$\overline{CS0}$		Output	Chip select 0: "0" is output if the address is in a designated address area.
$\overline{P41}$	1	Input/output	Port 41: Input/output port (with pull-up)
$\overline{CS1}$		Output	Chip select 1: "0" is output if the address is in a designated address area.
$\overline{P42}$	1	Input/output	Port 42: Input/output port (with pull-up)
$\overline{CS2}$		Output	Chip select 2: "0" is output if the address is in a designated address area.
$\overline{P43}$	1	Input/output	Port 43: Input/output port (with pull-up)
$\overline{CS3}$		Output	Chip select 3: "0" is output if the address is in a designated address area.
$\overline{P44}$	1	Input/output	Port 44: Input/output port (with pull-up)
$\overline{CS4}$		Output	Chip select 4: "0" is output if the address is in a designated address area.
$\overline{P45}$	1	Input/output	Port 45: Input/output port (with pull-up)
$\overline{CS5}$		Output	Chip select 5: "0" is output if the address is in a designated address area.
P46	1	Input/output	Port 46: Input/output port
SCOUT		Output	System clock output: Selectable between high- and low-speed clock outputs, as in the case of CPU
P47	1	Input/output	Port 47: Input/output port
P50-P57	8	Input/output	Port 5: Input/output port that allows input/output to be set in units of bits
A0-A7		Output	Address: Address buses 0 to 7 (separate bus mode)
P60-P67	8	Input/output	Port 6: Input/output port that allows input/output to be set in units of bits
A8-A15		Output	Address: Address buses 8 to 15 (separate bus mode)

Table 2.2.1 Pin Names and Functions (2 of 6)

Pin name	Number of pins	Input or output	Function
P70-P77 AN0-AN7	8	Input Input	Port 7: Port used exclusively for input Analog input: Input from A/D converter
P80-P87 AN8-AN15	8	Input Input	Port 8: Port used exclusively for input Analog input: Input from A/D converter
P90-P97 AN16-AN23	8	Input Input	Port 9: Port used exclusively for input Analog input: Input from A/D converter
PA0 TB0IN0 INT5	1	Input/output Input Input	Port A0: Input/output port 16-bit timer 0 input 0: For inputting the count/capture trigger of a 16-bit timer 0 Interrupt request pin 5: Selectable between "H" level, "L" level, rising edge, and falling edge Input pin with Schmitt trigger
PA1 TB0IN1 INT6	1	Input/output Input Input	Port A1: Input/output port 16-bit timer 0 input 1: For inputting the count/capture trigger of a 16-bit timer 0 Interrupt request pin 6: Selectable "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PA2 TB0OUT	1	Input/output Output	Port A2: Input/output port 16-bit timer 0 output: 16-bit timer 0 output pin
PA3 TB1IN0 INT7	1	Input/output Input Input	Port A3: Input/output port 16-bit timer 1 input 0: For inputting the count/capture trigger of a 16-bit timer 1 Interrupt request pin 7: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PA4 TB1IN1 INT8	1	Input/output Input Input	Port A4: Input/output port 16-bit timer 1 input 1: For inputting the count/capture trigger of a 16-bit timer 1 Interrupt request pin 8: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PA5 TB1OUT	1	Input/output Output	Port A5: Input/output port 16-bit timer 1 output: 16-bit timer 1 output pin
PA6 TB2OUT	1	Input/output Output	Port A6: Input/output port 16-bit timer 2 output: 16-bit timer 2 output pin
PA7 TB3OUT	1	Input/output Output	Port A7: Input/output port 16-bit timer 3 output: 16-bit timer 3 output pin
PB0 TB4OUT	1	Input/output Output	Port B0: Input/output port 16-bit timer 4 output: 16-bit timer 4 output pin
PB1 TB5OUT	1	Input/output Output	Port B1: Input/output port 16-bit timer 5 output: 16-bit timer 5 output pin
PB2 TB6OUT	1	Input/output Output	Port B2: Input/output port 16-bit timer 6 output: 16-bit timer 6 output pin
PB3 TB7OUT	1	Input/output Output	Port B3: Input/output port 16-bit timer 7 output: 16-bit timer 7 output pin
PB4 TB8OUT	1	Input/output Output	Port B4: Input/output port 16-bit timer 8 output: 16-bit timer 8 output pin
PB5 TB9OUT	1	Input/output Output	Port B5: Input/output port 16-bit timer 9 output: 16-bit timer 9 output pin
PB6 TBAIN0	1	Input/output Input	Port B6: Input/output port 16-bit timer A input 0: for inputting the count/capture trigger of a 16-bit timer A 2-phase pulse counter input 0
PB7 TBAIN1	1	Input/output Input	Port B7: Input/output port 16-bit timer A input 1: For inputting the count/capture trigger of a 16-bit timer A 2-phase pulse counter input 1

Table 2.2.1 Pin Names and Functions (3 of 6)

Pin name	Number of pins	Input or output	Function
PC0 TXD0	1	Input/output Output	Port C0: Input/output port Sending serial data 0: Open drain output pin depending on the program used
PC1 RXD0	1	Input/output Input	Port C1: Input/output port Receiving serial data 0
PC2 SCLK0 CTS0	1	Input/output Input/output Input	Port C2: Input/output port Serial clock input/output 0 Ready to send serial data 0 (Clear To Send): Open drain output pin depending on the program used
PC3 TXD1	1	Input/output Output	Port C3: Input/output port Sending serial data 1: Open drain output pin depending on the program used
PC4 RXD1	1	Input/output Input	Port C4: Input/output port Receiving serial data 1
PC5 SCLK1 CTS1	1	Input/output Input/output Input	Port C5: Input/output port Serial clock input/output 1 Ready to send serial data 1 (Clear To Send): Open drain output pin depending on the program used
PC6 TXD2	1	Input/output Output	Port C6: Input/output port Sending serial data 2: Open drain output pin depending on the program used
PC7 RXD2	1	Input/output Input	Port C7: Input/output port Receiving serial data 2
PD0 SCLK2 CTS2	1	Input/output Input/output Input	Port D0: Input/output port Serial clock input/output 2 Ready to send serial data 2 (Clear To Send): Open drain output pin depending on the program used
PD1 TXD3	1	Input/output Output	Port D1: Input/output port Sending serial data 3: Open drain output pin depending on the program used
PD2 RXD3	1	Input/output Input	Port D2: Input/output port Receiving serial data 3
PD3 SCLK3 CTS3	1	Input/output Input/output Input	Port D3: Input/output port Serial clock input/output 3 Ready to send serial data 3 (Clear To Send): Open drain output pin depending on the program used
PD4 TXD4	1	Input/output Output	Port D4: Input/output port Sending serial data 4: Open drain output pin depending on the program used
PD5 RXD4	1	Input/output Input	Port D5: Input/output port Receiving serial data 4
PD6 SCLK4 CTS4	1	Input/output Input/output Input	Port D6: Input/output port Serial clock input/output 4 Ready to send serial data 4 (Clear To Send): Open drain output pin depending on the program used
PD7 INT9	1	Input/output Input	Port D7: Input/output port Interrupt request pin 9: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger

Table 2.2.1 Pin Names and Functions (4 of 6)

Pin name	Number of pins	Input or output	Function
PE0 TXD5	1	Input/output Output	Port E0: Input/output port Sending serial data 5: Open drain output pin depending on the program used
PE1 RXD5	1	Input/output Input	Port E1: Input/output port Receiving serial data 5
PE2 SCLK5 CTS5	1	Input/output Input/output Input	Port E2: Input/output port Serial clock input/output 5 Ready to send serial data 5 (Clear To Send): Open drain output pin depending on the program used
PE3-PE5	3	Input/output	Ports E3 to E5: Input/output ports that allow input/output to be set in units of bits
PE6 INTA	1	Input/output Input	Port E6: Input/output port Interrupt request pin A: Selectable between "H" level, "L" level, rising edge, and falling edge Input pin with Schmitt trigger
PE7 INTB	1	Input/output Input	Port E7: Input/output port Interrupt request pin B: Selectable between "H" level, "L" level, rising edge, and falling edge Input pin with Schmitt trigger
PF0 SO SDA	1	Input/output Output Input/output	Port F0: Input/output port Pin for sending data if the serial bus interface operates in the SIO mode Pin for sending and receiving data if the serial bus interface operates in the I^2C mode Open drain output pin depending on the program used. Input with Schmitt trigger
PF1 SI SCL	1	Input/output Input Input/output	Port F1: Input/output port Pin for receiving data if the serial bus interface operates in the SIO mode Pin for inputting and outputting a clock if the serial bus interface operates in the I^2C mode Open drain output pin depending on the program used Input with Schmitt trigger
PF2 SCK	1	Input/output Input/output	Port F2: Input/output port Pin for inputting and outputting a clock if the serial bus interface operates in the SIO mode
PF3 DREQ2	1	Input/output Input	Port F3: Input/output port DMA request signal 2: For inputting the request to transfer data by DMA from an external I/O device to DMAC2
PF4 DACK2	1	Input/output Output	Port F4: Input/output port DMA acknowledge signal 2: Signal showing that DREQ2 has acknowledged a DMA transfer request
PF5 DREQ3	1	Input/output Input	Port F5: Input/output port DMA request signal 3: For inputting the request to transfer data by DMA from an external I/O device to DMAC3
PF6 DACK3	1	Input/output Output	Port F6: Input/output port DMA acknowledge signal 3: Signal showing that DREQ3 has acknowledged a DMA transfer request
PF7 TBTIN	1	Input/output Input	Port F7: Input/output port 32-bit time base timer input: For inputting the count for 32-bit time base timer
PG0-PG3 TC0IN-TC3IN	4	Input/output Input	Ports G0 to G3: Input/output ports that allow input/output to be set in units of bits For inputting the capture trigger for 32-bit timer
PG4-PG7 TCOU0-TCOUT3	4	Input/output Output	Ports G4 to G7: Input/output ports that allow input/output to be set in units of bits Outputting 32-bit timer if the result of a comparison is a match
PH0-PH5 TCOU4-TCOUT9	6	Input/output Output	Ports H0 to H5: Input/output ports that allow input/output to be set in units of bits Outputting 32-bit timber if the result of a comparison is a match
PH6-PH7	2	Input/output	Ports H6 to H7: Input/output ports that allow input/output to be set in units of bits
PI0 INT0	1	Input/output Input	Port I0: Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PI1 INT1	1	Input/output Input	Port I1: Input/output port Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PI2 INT2	1	Input/output Input	Port I2: Input/output port Interrupt request pin 2: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger

Table 2.2.1 Pin Names and Functions (5 of 6)

Pin name	Number of pins	Input or output	Function
PI3 INT3	1	Input/output Input	Port I3: Input/output port Interrupt request pin 3: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PI4 INT4	1	Input/output Input	Port I4: Input/output port Interrupt request pin 4: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PJ0 DREQ2	1	Input/output Input	Port J0: Input/output port DMA request signal 2: For inputting the request to transfer data by DMA from an external I/O device to DMAC2
PJ1 DACK2	1	Input/output Output	Port J1: Input/output port DMA acknowledge signal 2: Signal showing that DREQ2 has acknowledged a DMA transfer request
PJ2 DREQ3	1	Input/output Input	Port J2: Input/output port DMA request signal 3: For inputting the request to transfer data by DMA from an external I/O device to DMAC3
PJ3 DACK3	1	Input/output Output	Port J3: Input/output port DMA acknowledge signal 3: Signal showing that DREQ3 has acknowledged a DMA transfer request
PK0-PK7 KEY0-KEY7	8	Input/output Input	Port K: Input/output port that allows input/output to be set in units of bits KEY on wake up input 0 to 7 (with pull-up) With Schmitt trigger
PL0-PL7	8	Input/output	Port L: Input/output port that allows input/output to be set in units of bits
PM0-PM7	8	Input/output	Port M: Input/output port that allows input/output to be set in units of bits
PN0-PN7	8	Input/output	Port N: Input/output port that allows input/output to be set in units of bits
PO0 INT0	1	Input/output Input	Port O0: Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PO1 INT1	1	Input/output Input	Port O1: Input/output port Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PO2 INT2	1	Input/output Input	Port O2: Input/output port Interrupt request pin 2: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PO3 INT3	1	Input/output Input	Port O3: Input/output port Interrupt request pin 3: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PO4 INT4	1	Input/output Input	Port O4: Input/output port Interrupt request pin 4: Selectable between "H" level, "L" level, rising edge and falling edge Input pin with Schmitt trigger
PO5 TXD6	1	Input/output Output	Port O5: Input/output port Sending serial data 6: Open drain output pin depending on the program used
PO6 RXD6	1	Input/output Input	Port O6: Input/output port Receiving serial data 6
PO7 SCLK6 CTS6	1	Input/output Input/output Input	Port O7: Input/output port Serial clock input/output 6 Ready to send serial data 6 (Clear To Send): Open drain output pin depending on the program used
PP0-PP7 TPD0-TPD7	8	Input/output Output	Port P: Input/output port that allows input/output to be set in units of bits Outputting trace data from the data access address: Signal for DSU-ICE
PQ0-PQ7 TPC0-TPC7 TPD0-TPD7	8	Input/output Output Output	Port P: Input/output port that allows input/output to be set in units of bits Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE

Table 2.2.1 Pin Names and Functions (6 of 6)

Pin name	Number of pins	Input or output	Function
DCLK	1	Output	Debug clock: Signal for DSU-ICE
$\overline{\text{EJE}}$	1	Input	EJTAG enable: Signal for DSU-ICE (input with Schmitt trigger and built-in noise filter)
PCST4-0	5	Output	PC trace status: Signal for DSU-ICE
$\overline{\text{DINT}}$	1	Input	Debug interrupt: Signal for DSU-ICE (input with Schmitt trigger, pull-up and built-in noise filter)
TOVR/TSTA	1	Output	Outputting the status of PD data overflow status: Signal for DSU-ICE
TCK	1	Input	Test clock input: Signal for testing JTAG (input with Schmitt trigger and pull-up)
TMS	1	Input	Test mode select input: Signal for testing JTAG (input with Schmitt trigger and pull-up)
TDI	1	Input	Test data input: Signal for testing JTAG (input with Schmitt trigger and pull-up)
TDO	1	Output	Test data output: Signal for testing JTAG
$\overline{\text{TRST}}$	1	Input	Test reset input: Signal for testing JTAG (input with Schmitt trigger and pull-down)
$\overline{\text{NMI}}$	1	Input	Nonmaskable interrupt request pin: Pin for requesting an interrupt at the falling edge Input with Schmitt trigger and built-in noise filter
$\overline{\text{PLLOFF}}$	1	Input	Fix this pin to the "H (DVCC15) level."(Input with Schmitt trigger)
$\overline{\text{RESET}}$	1	Input	Reset: Initializing LSI (with pull-up) Input with Schmitt trigger and built-in noise filter
X1/X2	2	Input/output	Pin for connecting to a high-speed oscillator
XT1/XT2	2	Input/output	Pin for connecting to a low-speed oscillator
$\overline{\text{BUPMD}}$	1	Input	Backup mode trigger pin: This pin must be set to "L level" in backup mode.
$\overline{\text{BRESET}}$	1	Input	Backup module reset: Initializing the backup module (with pull-up) Input with Schmitt trigger
BUSMD	1	Input	Pin for setting an external bus mode: This pin functions as a multiplexed bus by sampling the "H (DVCC15) level" upon the rising of a reset signal. It also functions as a separate bus by sampling "L" upon the rising of a reset signal. When performing a reset operation, pull it up or down according to a bus mode to be used.
ENDIAN	1	Input	Pin for setting endian: This pin is used to set a mode. It performs a big-endian operation by sampling the "H (DVCC15) level" upon the rising of a reset signal, and performs a little-endian operation by sampling "L" upon the rising of a reset signal. When performing a reset operation, pull it up or down according to the type of endian to be used.
$\overline{\text{BOOT}}$	1	Input	Pin for setting a single boot mode: This pin goes into single boot mode by sampling "L" upon the rising of a reset signal. It is used to overwrite internal flash memory. By sampling "H (DVCC15) level" upon the rising of a reset signal, it performs a normal operation. This pin should be pulled up under normal operating conditions. Pull it up when resetting.
BW0-1	2	Input	Fix these pins to BW0="H (DVCC15)" and BW1="H (DVCC15)," respectively. (Input with Schmitt trigger)
VREFH	1	Input	Pin (H) for supplying the A/D converter with a reference power supply Connect this pin to AVCC31 if the A/D converter is not used.
VREFL	1	Input	Pin (L) for supplying the A/D converter with a reference power supply Connect this pin to AVSS if the A/D converter is not used.
AVCC31-32	2	–	Pin for supplying the A/D converter with a power supply. Connect it to a power supply even if the A/D converter is not used.
AVSS	3	–	A/D converter GND pin (0 V). Connect this pin to GND even if the A/D converter is not used.
TEST1-3	3	Input	TEST pin: To be fixed to GND.
CVCC15	1	–	Pin for supplying oscillators with power: 1.5 V power supply
CVSS/BVSS	1	–	GND pin (0 V) for oscillators and backup modules
DVCC15	4	–	Power supply pin: 1.5 V power supply
BVCC	1	–	Pin exclusively for supplying backup modules with power: 3 V power supply
DVCC30-34	8	–	Power supply pin: 3 V power supply
DVSS	11	–	GND pin (0 V)

Note 1: For BUSMD, ENDIAN and $\overline{\text{BOOT}}$ pins, the state designated for each pin ("H" or "L" level) must be maintained during one system clock before and after the rising of a reset signal. The reset pin must always be in a stable state at both "L" and "H" levels.

Note 2: For $\overline{\text{DREQ2}}$, $\overline{\text{DACK2}}$, $\overline{\text{DREQ3}}$ and $\overline{\text{DACK3}}$, it is necessary to go to the port function register and to select one port from two groups of ports, PF3 to PF6 and PJ0 to PJ3. Two ports cannot be operated simultaneously to use the same function.
Likewise, for pins INT0 through INT4, one port must be selected from ports PI0 to PI4 and ports PO0 to PO4.

Table 2.2.2 shows the pin names and power supply pins.

Table 2.2.2 Pin names and power supply pins

Pin name	Power supply pin	Pin name	Power supply pin
P0	DVCC33	PCST4 to 0	DVCC31
P1	DVCC33	DCLK	DVCC31
P2	DVCC33	$\overline{\text{EJE}}$	DVCC31
P3	DVCC33	$\overline{\text{TRST}}$	DVCC31
P4	DVCC33	TDI	DVCC31
P5	DVCC33	TDO	DVCC31
P6	DVCC33	TMS	DVCC31
P7	AVCC32	TCK	DVCC31
P8	AVCC32	$\overline{\text{DINT}}$	DVCC31
P9	AVCC31	TOV	DVCC31
PA	DVCC32	BUSMD	DVCC15
PB	DVCC32	$\overline{\text{BOOT}}$	DVCC15
PC	DVCC32	ENDIAN	DVCC15
PD	DVCC32	$\overline{\text{NMI}}$	DVCC15
PE	DVCC32	$\overline{\text{BRESET}}$	BVCC
PF	DVCC32	$\overline{\text{BUPMD}}$	BVCC
PG	DVCC32	X1, X2	CVCC15
PH	DVCC32	XT1, XT2	BVCC
PI	DVCC30	BW0 and 1	DVCC15
PJ	DVCC33	$\overline{\text{PLLOFF}}$	DVCC15
PK	DVCC34	$\overline{\text{RESET}}$	DVCC15
PL	DVCC34		
PM	DVCC34		
PN	DVCC34		
PO	DVCC34		
PP	DVCC31		
PQ	DVCC31		

- 2.7 V ≤ AVCC32 ≤ AVCC31

Table 2.2.3 shows the pin numbers and power supply pins.

Table 2.2.3 Pin numbers and power supply pins

Power supply pin	Pin number	Voltage range
DVCC15	D4, D8, E18, N9	1.35 V to 1.65 V
CVCC15	C17	1.35 V to 1.65 V
DVCC30	K17	1.65 V to 3.3 V
DVCC31	H2	1.65 V to 3.3 V
DVCC32	M6, U8	1.65 V to 3.3 V
DVCC33	F17, U14	1.65 V to 3.3 V
DVCC34	D15, F10	1.65 V to 3.3 V
AVCC31	B1	2.7 V to 3.3 V
AVCC32	F9	2.7 V to 3.3 V
BVCC	E15	2.3 V to 3.3 V (under normal operating conditions) 1.8 V to 3.3 V (in BACKUP mode)

3. Processor Core

The TMP19A64 has a high-performance 32-bit processor core (TX19A processor core). For information on the operations of this processor core, please refer to the "TX19A Family Architecture."

This chapter describes the functions unique to the TMP19A64 that are not explained in that document.

3.1 Reset Operation

To reset the device, ensure that the power supply voltage is in the operating voltage range, the oscillation of the internal high-frequency oscillator has stabilized at the specified frequency and that the $\overline{\text{RESET}}$ input has been "0" for at least 12 system clocks (1.78 μs during external 13.5 MHz operation).

Note that the PLL multiplication clock is quadrupled and the clock gear is initialized to the 1/8 mode during the reset period.

When the reset request is authorized,

- the system control coprocessor (CP0) register of the TX19A processor core is initialized. For further details, please refer to the chapter about architecture.
- After the reset exception handling is executed, the program branches off to the exception handler. The address to which the program branches off to (address where exception handling starts) is called an exception vector address. This exception vector address of a reset exception (for example, nonmaskable interrupt) is 0xBFC0_0000H (virtual address).
- The register of the internal I/O is initialized.
- The port pin (including the pin that can also be used by the internal I/O) is set to a general-purpose input or output port mode.

(Note 1) Set the $\overline{\text{RESET}}$ pin to "0" before turning the power on. Perform the reset after the power supply voltage has stabilized sufficiently within the operating range.

(Note 2) The reset operation can alter the internal RAM state, but does not alter data in the backup RAM.

(Note 3) Make sure that the power supply voltage has stabilized, wait for 500 μs or longer, and perform the reset.

(Note 4) In the FLASH program, the reset period of 0.5 uS or longer is required independently of the system clock.

4. Memory Map

Fig. 4.1 shows the memory map of the TMP19A64.

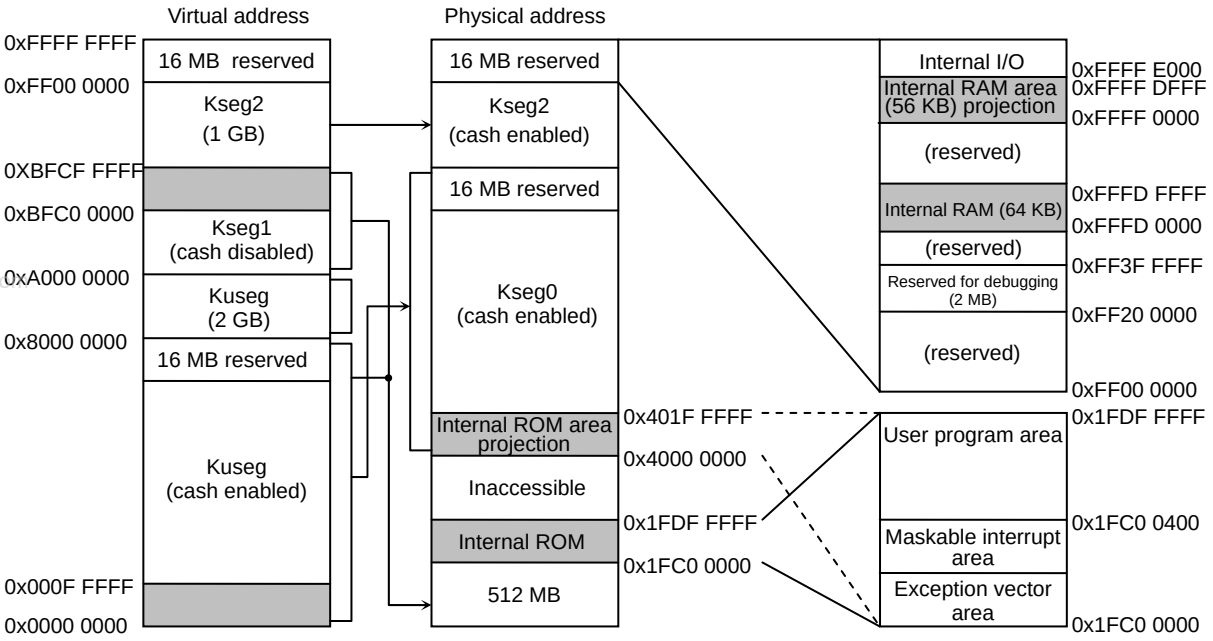


Fig. 4.1 Memory Map

(Note 1) The internal ROM is physically present in 0x1FC0_0000-0x1FDF_FFFF (2 MB). The internal RAM is physically present in 0xFFFD_0000-0xFFFD_FFFF (64 KB). 0xFFFF_0000-0xFFFF_DFFF (56 KB) becomes the projection area. You can access the internal RAM by accessing this area. The internal backup RAM area becomes 0xFFFF_E800-0xFFFF_E9FF (512 B).

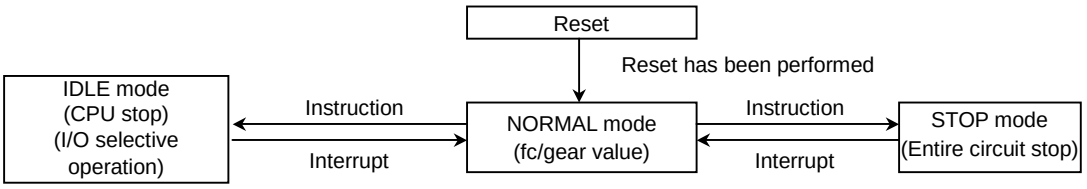
(Note 2) For the TMP19A64, a physical space of only 16 MB is available as external address space to be accessed. It is possible to place this 16-MB physical address space in a chip select area of your choice inside the 3.5-GB physical address space of the CPU. Access to internal memory, internal I/O space and reserved areas is given priority over access to the external address space. Therefore, access to the external address space is denied if any of the internal memory, internal I/O space or reserved areas are being accessed.

(Note 3) Do not place an instruction in the last four words of a physical area, specifically the last four words of an area where memory is mounted for external ROM extension (this varies depending on the system of the user).
Internal ROM: 0x1FDF_FFF0-0x1FDF_FFFF

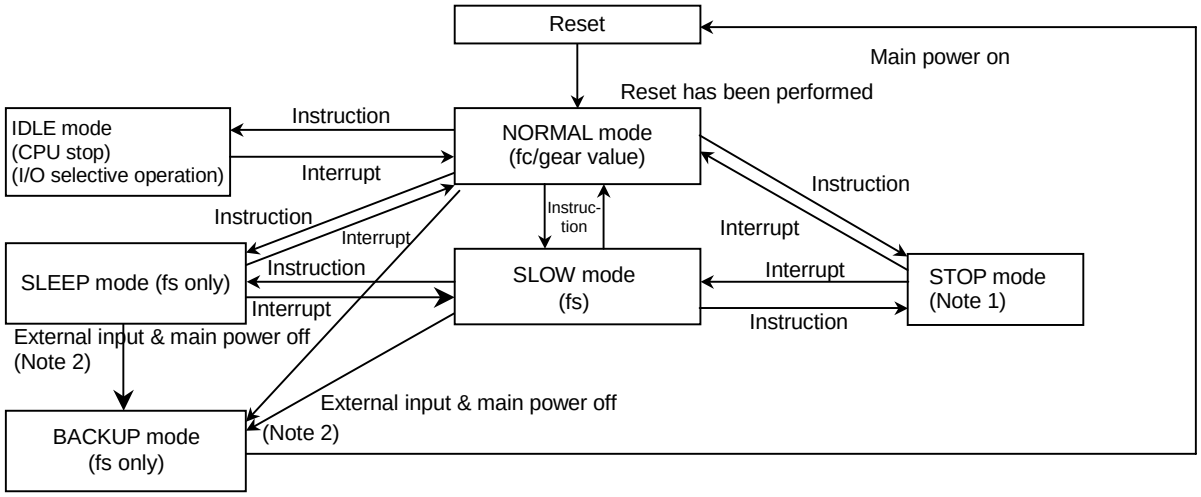
5. Clock/Standby Control

5.1 System Operation Modes

The system operation modes contain the standby modes in which the processor core operations are stopped to reduce power consumption. Fig. 5.1.1 State Transition Diagram of Each Operation Mode is shown below.



State Transition Diagram of Clock Mode When No Power is Supplied to the Backup Module



State Transition Diagram of Clock Mode When Power is Supplied to the Backup Module

(Note 1) STOP mode: All the circuits except the backup module are brought to a stop. The backup module continues operation (fs continues oscillation).

(Note 2) External input: It is necessary to activate the $\overline{\text{BUPMD}}$ pin during the $\overline{\text{RESET}}$ period. For details, see the chapter on Backup RAM.

Fig. 5.1.1 State Transition Diagram of Each Operation Mode

5.2 Default State of the System Clock

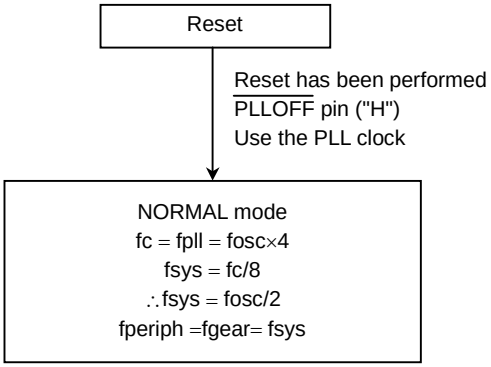


Fig. 5.2.1 Initial State of the System Clock

fosc:	**High-frequency clock frequency to be input via the X1 and X2 pins**
fpll:	**Clock frequency multiplied (quadrupled) by the PLL**
fc:	**Clock frequency when the $\overline{\text{PLLOFF}}$ pin is in the "H" state**
fs:	**Low-frequency clock frequency to be input via the XT1 and XT2 pins**
fgear:	**Clock frequency selected by the system control register SYSCR1<GEAR2:0> in the clock generator**
fsys:	**System clock frequency** **The CPU, ROM, RAM, DMAC and INTC all operate according to this clock. The internal peripheral I/O operates according to the fsys/2 clock.**
fperiph:	**Clock frequency selected by SYSCR1<FPSEL> (Clock to be input to the peripheral I/O prescaler)**

5.3 Clock System Block Diagram

5.3.1 Main System Clock

- Allows for oscillator connection or external clock input.
- Keep the $\overline{\text{PLLOFF}}$ pin (PLL (quadruple)) at the "H" level.
- Clock gear (8/8, 7/8, 6/8, 5/8, 4/8, 2/8, 1/8)
  (Default is 1/8.)
- Input frequency (high frequency)

	Input frequency range	Maximum operating frequency	Lowest operating frequency
PLL operation (for both oscillators and external input)	8-13.5 (MHz)	54 MHz	4 MHz *

* Clock gear 1/8 (default) is used when 8 MHz (MIN) is input.

- Input frequency (low frequency)

Input frequency range	Maximum operating frequency	Lowest operating frequency
30 KHz to 34 KHz	34 KHz	30 KHz

(Note) (precautions for switching the high-speed clock gear)
Switching of clock gear is executed when a value is written to the SYSCR1<GEAR2:0> register. There are cases where switching does not occur immediately after the change in the register setting but the original clock gear is used for execution of instructions. If it is necessary to use the new clock for execution of the instructions following to the clock gear switching instruction, insert a dummy instruction (to execute a write cycle).
To use the clock gear, ensure that you make the time setting such that ϕTn of the prescaler output from each block in the peripheral I/O is calibrated to ϕTn<fsys/2 (ϕTn becomes slower than fsys/2). Do not switch the clock gear during operation of the timer counter or other peripheral I/O.

(Note) Restriction on use of the clock gear
When using the clock gear to operate the peripheral I/O, use the frequency division ratio of 8/8, 4/8, 2/8 or 1/8. If other frequency division ratios are used, the peripheral I/O will not operate properly.

5.3.2 Clock Gear

- The high-speed clock is divided into 8/8, 7/8, 6/8, 5/8, 4/8, 2/8 or 1/8.
- The internal I/O prescaler clock ϕT0: fperiph/2, fperiph/4, fperiph/8 and fperiph/16

Fig. 5.3.1 shows the system clock transition diagram.

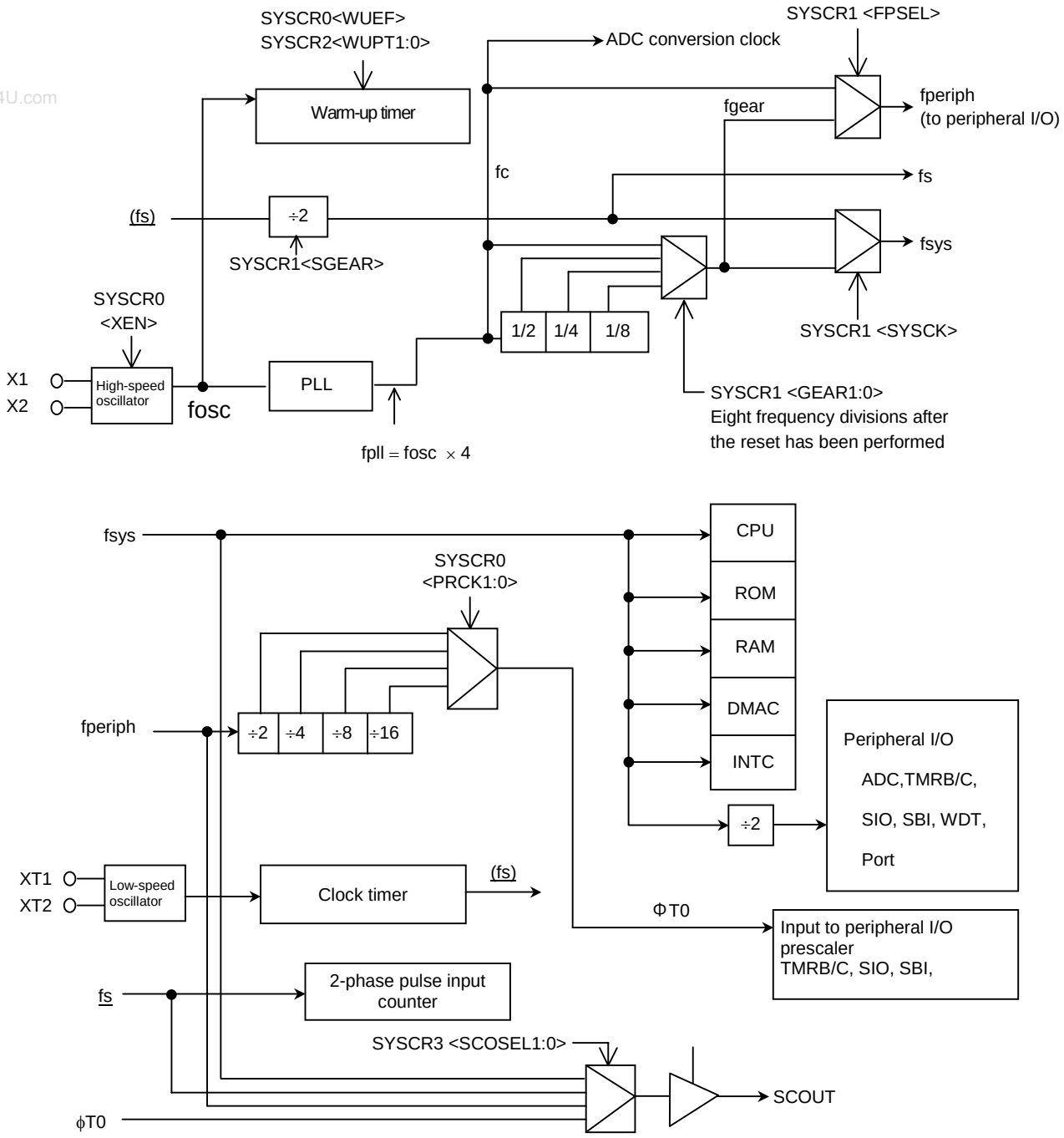


Fig. 5.3.1 System Clock Transition Diagram

5.4 CG Registers

5.4.1 System Control Registers

		7	6	5	4	3	2	1	0
SYSCR0	bit Symbol	XEN		RXEN			WUEF	PRCK1	PRCK0
(0xFFFF_EE00)	Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
	After reset	1	1	1	1	0	0	0	0
	Function	High-speed oscillator 0: Stop 1: Oscillation	Write "1."	High-speed oscillator after the STOP mode is released 0: Stop 1: Oscillation	Write "1."	This can be read as "0."	Control of warm-up timer (WUP) for oscillator 0 write: don't care 1 write: WUP Start 0 read: WUP finished 1 read: WUP operating	Select prescaler clock 00: fperiph/16 01: fperiph/8 10: fperiph/4 11: fperiph/2	

		15	14	13	12	11	10	9	8
SYSCR1	Bitsymbol		SYSCKFLG	SYSCK	FPSEL	SGEAR	GEAR2	GEAR1	GEAR0
(0xFFFF_EE01)	Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	1	1	1
	Function	This can be read as "0."	System clock status flag 0: High speed (fc) 1: Low speed (fs)	Select system clock 0: High speed (fc) 1: Low speed (fs)	Select fperiph 0: fgear 1: fc	Select gear of low-speed clock 0: fs/1 1:fs/2	Select gear of high-speed clock (fc) 000: fc 001: fc7/8 010: fc6/8 011: fc5/8 100: fc4/8 101: reserved 110: fc2/8 111: fc1/8		

		23	22	21	20	19	18	17	16
SYSCR2	Bitsymbol	DRVOSCH		WUPT1	WUPT0	STBY1	STBY0		DRVE
(0xFFFF_EE02)	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
	After reset	0	0	1	0	1	1	0	0
	Function	High-speed oscillator current control 0: High capability 1: Low capability	Write 0.	Select oscillator warm-up time 00: No WUP 01: 2^8 /Input frequency 10: 2^{14} /Input frequency 11: 2^{16} /Input frequency		Select standby mode 00:Reserved 01:STOP 10:SLEEP 11:IDLE		This can be read as "0."	1: Drive the pin even in the STOP mode.

		31	30	29	28	27	26	25	24
SYSCR3	Bitsymbol		SCOSEL1	SCOSEL0	ALESEL				
(0xFFFF_EE03)	Read/Write	R	R/W	R/W	R/W	R			
	After reset	0	0	1	1	0	0	0	0
	Function	This can be read as "0."	Select SCOUT output 00:fs 01:fperiph 10:fsys 11:φT0		Set ALE output width 0:fsys×1 1:fsys×2	This can be read as "0."			

- **Don't switch the SYSCK and the GEAR<2:0> simultaneously.**
- **If the system enters the STOP mode with SYSCR2<DRVOSCH> set at 1 (low capability), the setting will change to 0 (high capability) after the STOP mode is released. Make the setting again, as required.**
- **SYSCK can be switched when XEN is set to "1."**

(Note) Restriction on use of the clock gear

When using the clock gear to operate the peripheral I/O, set the SYSCR1<GEAR2:0> to the frequency division ratio of fc, fc4/8, fc2/8 or fc1/8. If other frequency division ratios are used, the peripheral I/O will not operate properly.

5.5 System Clock Controller

By resetting the system clock controller, the controller status is initialized to <XEN>="1" and <GEAR2:0>="111," and the system clock fsys changes to fc/8. (fc=fosc (original oscillation frequency)×4, because the original oscillation is quadrupled by PLL.) For example, when a 13.5-MHz oscillator is connected to the X1 or X2 pin, fsys becomes 6.25 MHz (=13.5×4×1/8) after the reset.

Similarly, when the oscillator is not connected and an external oscillator is used to input a clock instead, fsys becomes the frequency obtained from the calculation "input frequency×4×1/8."

(Note) Set the initial system clock frequency to 4 MHz or higher.

5.5.1 Oscillation Stabilization Time (Switching between the NORMAL and SLOW modes)

The warm-up timer is provided to confirm the oscillation stability of the oscillator when it is connected to the oscillator connection pin. The warm-up time can be selected by setting the SYSCR2<WUPT1:0> depending on the characteristics of the oscillator. The SYSCR0<WUEF> is used to confirm the start and completion of warm-up through software (instruction). After the completion of warm-up is confirmed, switch the system clock (SYSCR1<SYSCK>).

When clock switching occurs, the current system clock can be checked by monitoring the SYSCR1<SYSCKFLG>.

Table 5.5.1 shows the warm-up time when switching occurs.

(Note 1) Warm-up is not required when an oscillator is used for the clock and providing stable oscillation.

(Note 2) The warm-up timer operates according to the oscillation clock, and it can contain errors if there is any fluctuation in the oscillation frequency. Therefore, the warm-up time should be taken as approximate time.

Table 5.5.1 Warm-up Time

Warm-up time options SYSCR2<WUPT1:0>	High-speed clock (fosc)
01 (2^8/oscillation frequency)	18.963 (μs)
10 (2^{14}/oscillation frequency)	1.214 (ms)
11 (2^{16}/oscillation frequency)	4.855 (ms)

These values are calculated under the following condition:
fosc = 13.5 MHz

<Example 1> Transition from the NORMAL mode to the SLOW mode

SYSCR1<SYSCK>="1" : Switch the system clock to low speed (fs)
SYSCR1<SYSCKFLG>Read : Confirm that the current state is "1" (the current system clock is fs)
SYSCR0<XEN>="0" : Disable the high-speed oscillation (fosc)

<Example 2> Transition from the SLOW mode to the NORMAL mode

SYSCR2<WUPT1:0>="xx" : Select the warm-up time
SYSCR0<XEN>="1" : Enable the high-speed oscillation (fosc)
SYSCR0<WUEF>="1" : Start the warm-up timer (WUP)
SYSCR0<WUEF> Read : Wait until the state becomes "0" (WUP is finished)
SYSCR1<SYSCK>="0" : Switch the system clock to high speed (fgear)
SYSCR1<SYSCKFLG>Read : Confirm that the current state is "0" (the current system clock is fgear)

(Note) In the SLOW mode, the CPU operates with the low-speed clock, and the INTC, the backup block, the 2-phase pulse input counter, the KWUP, the IO port and the EBIF (external bus interface) are operable. Stop other internal peripheral functions before the system enters the SLOW mode.

5.5.2 System Clock Pin Output Function

The system clock, fsys, fsys/2 or fs, can be output from the P46/SCOUT pin. By setting the port 4 related registers, P4CR<P46C> to "1" and P4FC<P46F> to "1," the P46/SCOUT pin becomes the SCOUT output pin. The output clock is selected by setting the SYSCR3<SCOSEL1:0>.

Table 5.5.2 shows the pin states in each standby mode when the P46/SCOUT pin is set to the SCOUT output.

Table 5.5.2 SCOUT Output State in Each Standby Mode

Mode / SCOUT selection	NORMAL	SLOW	Standby mode		
			IDLE	SLEEP	STOP
<SCOSEL1:0>="00"	Output the fs clock.				
<SCOSEL1:0>="01"	Output the fpriph clock.			Fixed to "0" or "1."	
<SCOSEL1:0>="10"	Output the fsys clock.				
<SCOSEL1:0>="11"	Output the ϕT0 clock.	Fixed to "0."	Output the ϕT0 clock.	Fixed to "0."	

(Note) The phase difference (AC timing) between the system clock output by the SCOUT and the internal clock is not guaranteed.

5.5.3 Reducing the Oscillator Driving Capability

This function is intended for restricting oscillation noise generated from the oscillator and reducing the power consumption of the oscillator when it is connected to the oscillator connection pin.

Setting the SYSCR2<DRVOSCH> to "1" reduces the driving capability of the high-speed oscillator. (low capability)

This is reset to the default setting "0." When the power is turned on, oscillation starts with the normal driving capability (high capability). This is automatically set to the high driving capability state (<DRVOSCH> ="0") whenever the oscillator starts oscillation due to mode transition.

● Reducing the driving capability of the high-speed oscillator

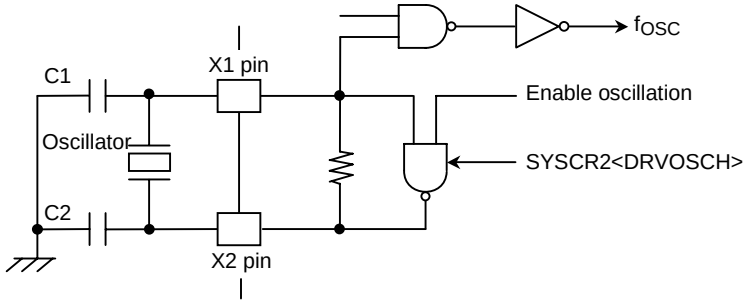


Fig. 5.5.1 Oscillator Driving Capability

5.5.4 Clock Frequency Division for Low-Speed System Clock

The low-speed clock (fs) can be divided into two by setting the system control register SYSCR1<SGEAR> to "1." This reduces the power consumption in the SLOW mode. Set the clock frequency division during high-speed oscillation.

5.6 Prescaler Clock Controller

Each internal I/O (TMRB0-A, TMRC, SIO0-6 and SBI) has a prescaler for dividing a clock. The clock ϕT0 to be input to each prescaler is obtained by selecting the "fperiph" clock at the SYSCR1<FPSEL> and the SYSCR0<PRCK1:0> and then dividing the clock according to the setting of SYSCR0<PRCK1:0>. After the controller is reset, fperiph/16 is selected as ϕT0. For details, please refer to Fig. 5.3.1 System Clock Transition Diagram.

5.7 Clock Multiplication Circuit (PLL)

Keep the PLLOFF pin at the "H" level. This pin is the circuit that outputs the fpll clock that is a quadruple of the output clock of the high-speed oscillator, fosc. This lowers the oscillator input frequency while increasing the internal clock speed.

5.8 Flash Access Control Circuit (PFB)

The PFBWAIT register can be used to select the speed of access to the flash memory.

You need to set an appropriate flash access speed for the operating frequency to be used.

		31 - 2	1	0
PFBWAIT	bit Symbol	-	PFBWAIT	
(0xFFFF_E500)	Read/Write	R	R/W	R/W
	After reset	0	1	1

PFBWAIT: WAIT number

11: 4-clock access/10: 3-clock access/01: 2-clock access

00: Setting disabled

	Operating frequency (fc) MHz		
PFBWAIT<1:0>	40-	< 45	<=54
11	○	○	○
10	○	○	○
01	○	×	×
00	–	–	–

○: Settable ×: Not settable –: Setting prohibited

Note) If an appropriate access speed is not specified, the program can operate improperly.

5.9 Standby Controller

The TX19A core has several low-consumption modes. To shift to the STOP, SLEEP or IDLE (Halt or Doze) mode, set the RP bit in the CPO status register, and then execute the WAIT instruction.

Before shifting to the mode, you need to select the standby mode at the system control register (SYSCR2).

The IDLE, SLEEP and STOP modes have the following features:

IDLE: Only the CPU is stopped in this mode.

The internal I/O has one bit of the ON/OFF setting register for operation in the IDLE mode in the register of each module. This enables operation settings for the IDLE mode. When the internal I/O has been set not to operate in the IDLE mode, it stops operation and holds the state when the system enters the IDLE mode.

Table 5.9.1 shows a list of IDLE setting registers.

Table 5.9.1 Internal I/O Setting Registers for the IDLE Mode

Internal I/O	IDLE mode setting register
TMRB0-A	TBxRUN<I2TBx>
TBT	TBTRUN<I2TBT>
SIO0-6	SCxMOD1<I2Sx>
SBI	SBIBR0<I2SBI>
A/D converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

(Note 1) The Halt mode is activated by setting the RP bit in the status register to "0," executing the WAIT command and shifting to the standby mode. In this mode, the TX19A processor core stops the processer operation while holding the status of the pipeline. The TX19A gives no response to the bus control authority request from the internal DMA, so the bus control authority is maintained in this mode.

(Note 2) The Doze mode is activated by setting the RP bit in the status register to "1" and shifting to the standby mode. In this mode, the TX19A processor core stops the processer operation while holding the status of the pipeline. The TX19A can respond to the bus control authority request given from the outside of the processor core.

SLEEP: Only the internal low-speed oscillator, the backup block, the 2-phase pulse input counter operate.

STOP: All the internal circuits are brought to a stop.

5.9.1 CG Operations in Each Mode

Table5.9.1 Status of CG in Each Operation Mode

Clock source	Mode	Oscillation circuit	PLL	Clock supply to peripheral I/O	Clock supply to CPU
Oscillator	Normal	○	○	○	○
	Slow	○	×	Partial supply (Note)	○
	Idle (Halt)	○	○	Selectable	×
	Idle (Doze)	○	○	Selectable	×
	Sleep	fs only	×	Backup block/2-phase pulse input counter	×
	Stop	×	×	×	×

○: ON or clock supply ×: OFF or no clock supply

(Note) Peripheral functions that can work in the SLOW mode: INTC, external bus interface, IO port, backup block and 2-phase pulse input counter

5.9.2 Block Operations in Each Mode

Table 5.9.2 Block Operating Status in Each Operation Mode

Block	NORMAL	SLOW	IDLE (Doze)	IDLE (Halt)	SLEEP	STOP	BACKUP
TX19A processor core	○	○	×	×	×	×	×
DMAC	○	○	○	×	×	×	×
INTC	○	○	○	○	×	×	×
External bus I/F	○	○	○	×	×	×	×
IO port	○	○	○	×	×	×	×
ADC	○	×	ON/OFF selectable for each module		×	×	×
SIO	○	×			×	×	×
I2C	○	×			×	×	×
TMRB	○	×			×	×	×
TMRC	○	×			×	×	×
WDT	○	×			×	×	×
2-phase counter	○	○			○	×	×
Backup block	○	Δ (Note 1)			○	○/× (Note 3)	○
KWUP	○	○	○	○	×	○	×
CG	○	○	○	○	○	×	×
High-speed oscillator (fc)	○	Δ (Note 2)	○	○	×	×	×
Low-speed oscillator (fs)	○	○	○	○	○	○	○

○: ON ×: OFF

- **Low-speed oscillation is active when the BVCC is applied, and not active when the BVCC is shut off.**

(Note 1) The backup RAM is inaccessible in the SLOW mode.

(Note 2) When the system enters the SLOW mode, the high-speed oscillator must be stopped by setting the SYSCR1<XEN>.

(Note 3) In the SLOW mode, the backup block operates differently depending on the $\overline{\text{BUPMD}}$ pin.

5.9.3 Releasing the Standby State

The standby state can be released by an interrupt request when the interrupt level is higher than the interrupt mask level, or by the reset. The standby release source that can be used is determined by a combination of the standby mode and the state of the interrupt mask register <IM15:8> assigned to the status register in the system control coprocessor (CPO) of the TX19A processor core. Details are shown in Table 5.9.3 Standby Release Sources and Standby Release Operations.

- Release by an interrupt request

  Operations of releasing the standby state using an interrupt request vary depending on the interrupt enabled state. If the interrupt level specified before the system enters the standby mode is equal to or higher than the value of the interrupt mask register, an interrupt handling operation is executed by the trigger after the standby is released, and the processing is started at the instruction next to the standby shift instruction (WAIT instruction). If the interrupt request level is lower than the value of the interrupt mask register, the processing is started with the instruction next to the standby shift instruction (WAIT instruction) without executing an interrupt handling operation. (The interrupt request flag is maintained at "1".)

  For a nonmaskable interrupt, an interrupt handling is executed after the standby state is released irrespectively of the mask register value.

- Release by the reset

  Any standby state can be released by the reset.
  Note that releasing of the STOP mode requires sufficient reset time to allow the oscillator operation to become stable. (Refer to Table 5.1.)

  When the standby mode is released by the reset, data in the backup RAM can maintain the state immediately before the standby state is started, but other settings will be initialized. (When the standby mode is released by an interruption, the state immediately before the standby state is started will be maintained.)

Please refer to "6. Interrupt" for details of interrupts for STOP, SLEEP and IDLE release and ordinary interrupts.

Table 5.9.3 Standby Release Sources and Standby Release Operations

(Interrupt level)>(Interrupt mask)

Interrupt accepting state			Interrupt enabled EI= "1"			Interrupt disabled EI= "0"		
Standby mode			IDLE (programmable)	SLEEP	STOP	IDLE (programmable)	SLEEP	STOP
Standby release source	Interrupt	INTWDT	◎	×	×	◎	–	–
		INT0-B	◎	◎	◎ (Note 1)	○	○	○ (Note 1)
		KWUP0-7	◎	◎	◎ (Note 1)	○	○	○ (Note 1)
		INTRTC	◎	◎	×	○	○	×
		INTTBA (Note 2)	◎	◎		○	○	
		INTTB0-9	◎	×	×	○	×	×
		INTRX0-6, TX0-6	◎	×	×	○	×	×
		INTS	◎	×	×	○	×	×
		INTAD/ADHP/ADM	◎	×	×	○	×	×

◎ : Starts the interrupt handling after the standby mode is released. (The LSI is initialized by the reset.)

○ : Starts the processing at the address next to the standby instruction (without executing the interrupt handling) after the standby mode is released.

× : Cannot be used for releasing the standby mode

– : Cannot execute masking with an interruption mask when a nonmaskable interrupt is selected.

(Note 1) The standby mode is released after the warm-up time has elapsed.

(Note 2) These operations are applicable only when the 2-phase pulse input counter mode is selected. If any other modes are selected, the operations will be the same as those for the INTTB0 to INTTB9.

- **To release the standby mode by using the level mode interrupt in the interruptible state, keep the level until the interrupt handling is started. Changing the level before then will prevent the interrupt processing from starting properly.**
- **To enter the standby mode when the CPU has disabled the acceptance of interrupts, disable interrupts other than the recovery factors in advance by using the interrupt controller (INTC). Otherwise, the standby mode can be released by any other interrupts than the recovery factors.**
- **To recover from the standby mode when the CPU has disabled the acceptance of interrupts, set the interrupt level higher than the interrupt mask (Interrupt level > Interrupt mask). If the interrupt level is equal to or lower than the interrupt mask (Interrupt level ≤ Interrupt mask), the system cannot recover from the standby mode.**

5.9.4 STOP Mode

In the STOP mode, all the internal circuits, including the internal oscillators, are brought to a stop. The pin states in the STOP mode vary depending on the setting of the SYSCR2<DRVE>. Table 5.9.6 shows the pin states in the STOP mode. When the STOP mode is released, the system clock output is started after the elapse of warm-up time at the warm-up counter to allow the internal oscillators to stabilize. After the STOP mode is released, the system returns to the operation mode that was active immediately before the STOP mode (NORMAL or SLOW), and starts the operation.

It is necessary to make these settings before the instruction to enter the STOP mode is executed. Specify the warm-up time at the SYSCR2<WUPT1:0>.

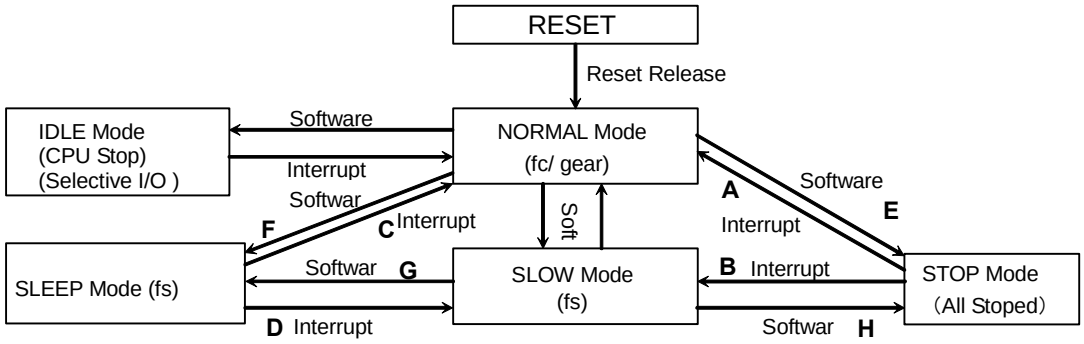
(Note) To shift from the NORMAL mode to the STOP mode on the TMP19A64, do not set the SYSCR2<WUPT1:0> to "00" or "01" for the warm-up time setting. The internal system recovery time cannot be satisfied when the system recovers from the STOP mode.

Table 5.9.4 Warm-up Settings for Transitions of Operation Modes

Transition of operation mode	Warm-up setting
NORMAL → IDLE	Not required
NORMAL → SLEEP	Not required
NORMAL → SLOW	Not required
NORMAL → STOP	Not required
IDLE → NORMAL	Not required
SLEEP → NORMAL	Required
SLEEP → SLOW	Not required
SLOW → NORMAL	Required (Note 1)
SLOW → SLEEP	Not required
SLOW → STOP	Not required
STOP → NORMAL	Required
STOP → SLOW	Not required

Note 1) When the high-speed oscillator is stopped in the SLOW mode

(NOTE)19A64 requires a recovery time from Warming up state as following

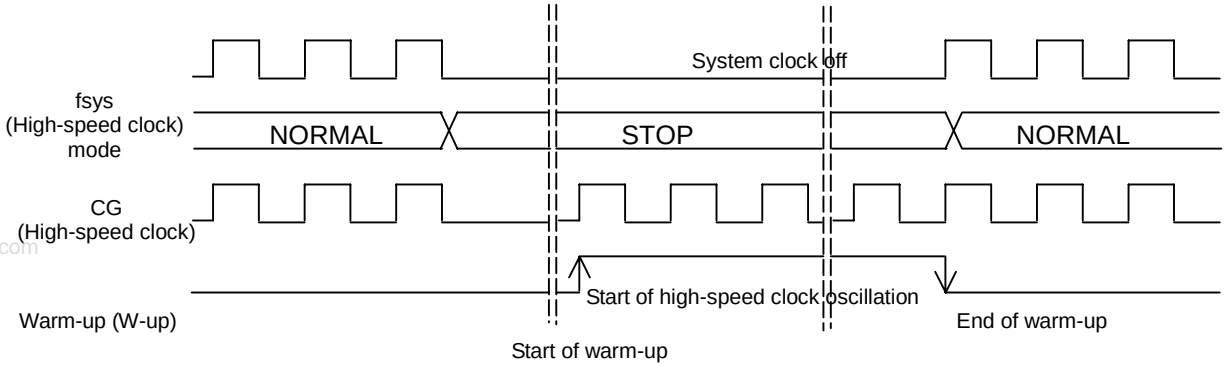


State Transition Diagram

WUP Trigger	State Transition	Running Mode after WUP	Minimum required Operation time before WAIT instruction done (sec)
STOP release	A	STOP/SLEEP	64 / (fsys) in NOMAL mode
	B	STOP/SLEEP	16 / (fsys) in SLOW mode
SLEEP release	C	STOP/SLEEP	64 / (fsys) in NOMAL mode
	D	STOP/SLEEP	-

5.9.5 Recovery from the STOP or SLEEP Mode

1. Transition of operation modes: NORMAL → STOP → NORMAL

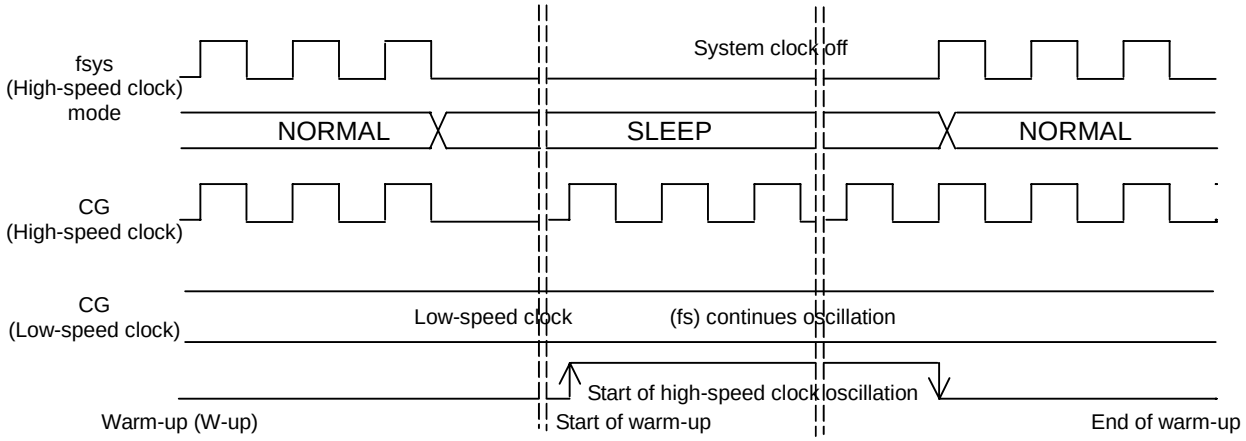


when @fosc=13.5 MHz

Selection of warm-up time SYSCR2<WUPT1:0>	Warm-up time (fosc)
01 (2^8/fosc)	Setting disabled
10 (2^{14}/fosc)	1.214 ms
11 (2^{16}/fosc)	4.855 ms

(Note) When @fosc=13.5 MHz, the internal system recovery time cannot be satisfied. Do not set <WUPT1:0> to "01."

2. Transition of operation modes: NORMAL → SLEEP → NORMAL

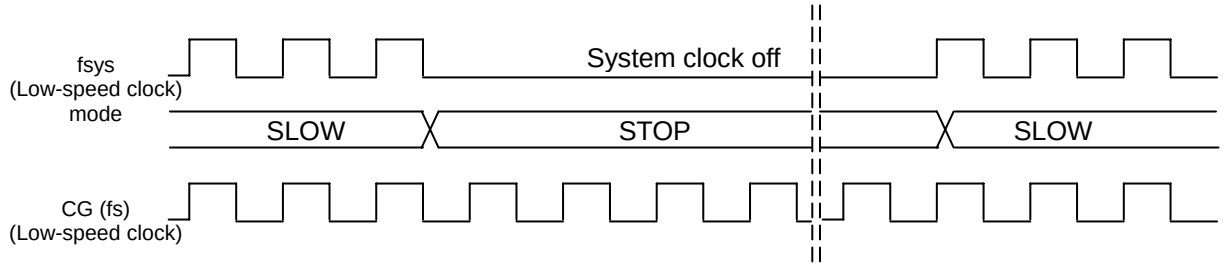


when @fosc=13.5 MHz

Selection of warm-up time SYSCR2<WUPT1:0>	Warm-up time (fosc)
01 (2^8/fosc)	Setting disabled
10 (2^{14}/fosc)	1.214 ms
11 (2^{16}/fosc)	4.855 ms

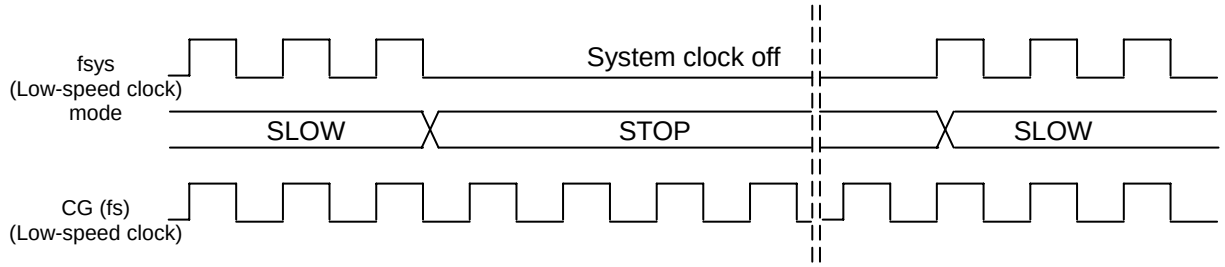
(Note) When @fosc=13.5 MHz, the internal system recovery time cannot be satisfied. Do not set <WUPT1:0> to "01."

3. Transition of operation modes: SLOW → STOP → SLOW



(Note) The low-speed clock (fs) continues oscillation. There is no need to make a warm-up setting.

4. Transition of operation modes: SLOW → SLEEP → SLOW



(Note) The low-speed clock (fs) continues oscillation. There is no need to make a warm-up setting.

Table 5.9.6 Pin States in the STOP Mode in Each State of SYSCR2<DRVE> (1/2)

Pin name	Input/output	<DRVE>=0	<DRVE>=1
P00-P07	Input mode Output mode AD0-AD7, D0-D7	— — —	— Output —
P10-P17	Input mode Output mode, A8-A15 AD8-AD15, D8-D15	— — —	— Output —
P20-P27	Input mode Output mode, A0-A7/A16-A23	— —	— Output
P30 (/RD), P31 (/WR)	Output pin	—	Output
P32, P35, P36	Input mode Output mode, /HWR, /BUSAK, R/W_	PU* PU*	— Output
P33	Input mode, /WAIT, /RDY Output mode	PU* PU*	— Output
P34	Input mode Output mode BUSRQ	PU* PU* PU*	— Output Output
P37 (ALE)	Input mode Output mode ALE (Output mode)	— — "L" level output	— Output "L" level output
P40-P45	Input mode Output mode, CS0-CS5	PU* PU*	Input Output
P46 (SCOUT)	Input mode Output mode	— —	Input Output
P47	Input mode Output mode	— —	Input Output
P50-P57	Input mode Output mode, A0-A7	— —	— Output
P60-P67	Input mode Output mode, A8-A15	— —	— Output
P7, P8, P9	Input pin, AN0-AN23	—	—
PA0, PA1, PA3, PA4	Input mode Output mode INT5-INT8 (Input mode)	— — Input	Input Output Input
cPA2, PA5, PA6, PA7	Input mode Output mode, TB0OUT,TB1-3OUT	— —	Input Output
PB0-PB7	Input mode, TBAIN1 Output mode, TB4-9OUT	— —	Input Output
PC0-PC7	Input mode, SCLK0-1, RXD0-2, /CTS0-1 Output mode, SCLK0-1,TXD0-2	— —	Input Output
PD0-PD6	Input mode, SCLK2-4, RXD3-4, /CTS2-4 Output mode, SCLK2-4,TXD3-4	— —	Input Output
PD7	Input mode Output mode INT9 (Input mode)	— — Input	Input Output Input
PE0-PE2	Input mode, SCLK5, RXD5, /CTS5 Output mode, SCLK5, TXD5	— —	Input Output
PE3-PE5	Input mode Output mode	— —	Input Output
PE6-PE7	Input mode Output mode INTA-INTB (Input mode)	— — Input	Input Output Input

Table 5.9.6 Pin States in the STOP Mode in Each State of SYSCR2<DRVE> (2/2)

Pin name	Input/Output	<DRVE>=0	<DRVE>=1
PF0-PF7	Input mode, SDA, SI, SCL, SCK, /DREQ2-, TBTIN Output mode, SO, SDA, SCL, SCK, /DACK2-3	— —	Input Output
PG0-PG7	Input mode, TC0-3IN Output mode, TCOUT0-3	— —	Input Output
PH0-PH5	Input mode Output mode, TCOUT4-9	— —	Input Output
PH6-PH7	Input mode Output mode	— —	Input Output
PI0-PI4	Input mode Output mode INT0-INT4 (Input mode)	— — Input	Input Output Input
PJ0-PJ3	Input mode, /DREQ2-3 Output mode, /DACK2-3	— —	Input Output
PK0-PK7	Input mode Output mode KEY0-KEY7 (Input mode)	— — Input	Input Output Input
PL, PM, PN	Input mode Output mode	— —	Input Output
PO0-PO4	Input mode Output mode INT0-INT4 (Input mode)	— — Input	Input Output Input
PO5-PO7	Input mode, RXD6, /CTS6 Output mode, TXD6,	— —	Input Output
PP, PQ	Input mode Output mode TPD0-7, TPC0-7	— — Output	Input Output Output
$\overline{\text{NMI}}$	Input pin	Input	Input
$\overline{\text{PLLOFF}}$	Input pin	Input	Input
$\overline{\text{RESET}}$	Input pin	Input	Input
$\overline{\text{BUPMD}}$	Input pin	Input	Input
$\overline{\text{BRESET}}$	Input pin	Input	Input
BUSMD	Input pin	Input	Input
ENDIAN	Input pin	Input	Input
$\overline{\text{BOOT}}$	Input pin	Input	Input
BW0-1	Input pin	Input	Input
TEST1-3	Input pin	Input	Input
X1	Input pin	—	—
X2	Output pin	"H" level output	"H" level output

— : Indicates that the input is disabled for the input mode and the input pin and the impedance becomes high for the output mode and the output pin. Note that the input is enabled when the port function register (PxFC) is "1" and the port control register (PxCR) is "0."

Input : The input gate is active. To prevent the input pin from floating, fix the input voltage to the "L" or "H" level.

Output : The pin is in the output state.

PU* : This is the programmable pull-up pin. The input gate is always disabled. No feedthrough current flows even if the high impedance is selected.

6. Interrupts

6.1 Overview

The features of the TX19A64 interrupts are as follows:

- 2 interrupts from the CPU itself (software interrupt instruction)
- 21 external pins ($\overline{\text{NMI}}$, INT0 to INTB, KWUP0 to 7)
- 51 interrupts from internal I/O (including WDT interrupt)
- Generation of vectors for each interrupt factor
- Seven interrupt levels for each interrupt factor
- An interrupt can be used to activate the DMAC.

(1) Preparation for interrupt settings

- Settings required before generating interrupts:

  Set the exception table base address (the base address of the table of maskable interrupt jump addresses) to IVR.

  Set the interrupt jump addresses to the "exception table base address + IVR offset address" memory.

  Set Status <IM [4:2]> of the CP0 register to "0x111."

  * For details of the Status register, refer to the material "TX19A Core Architecture."

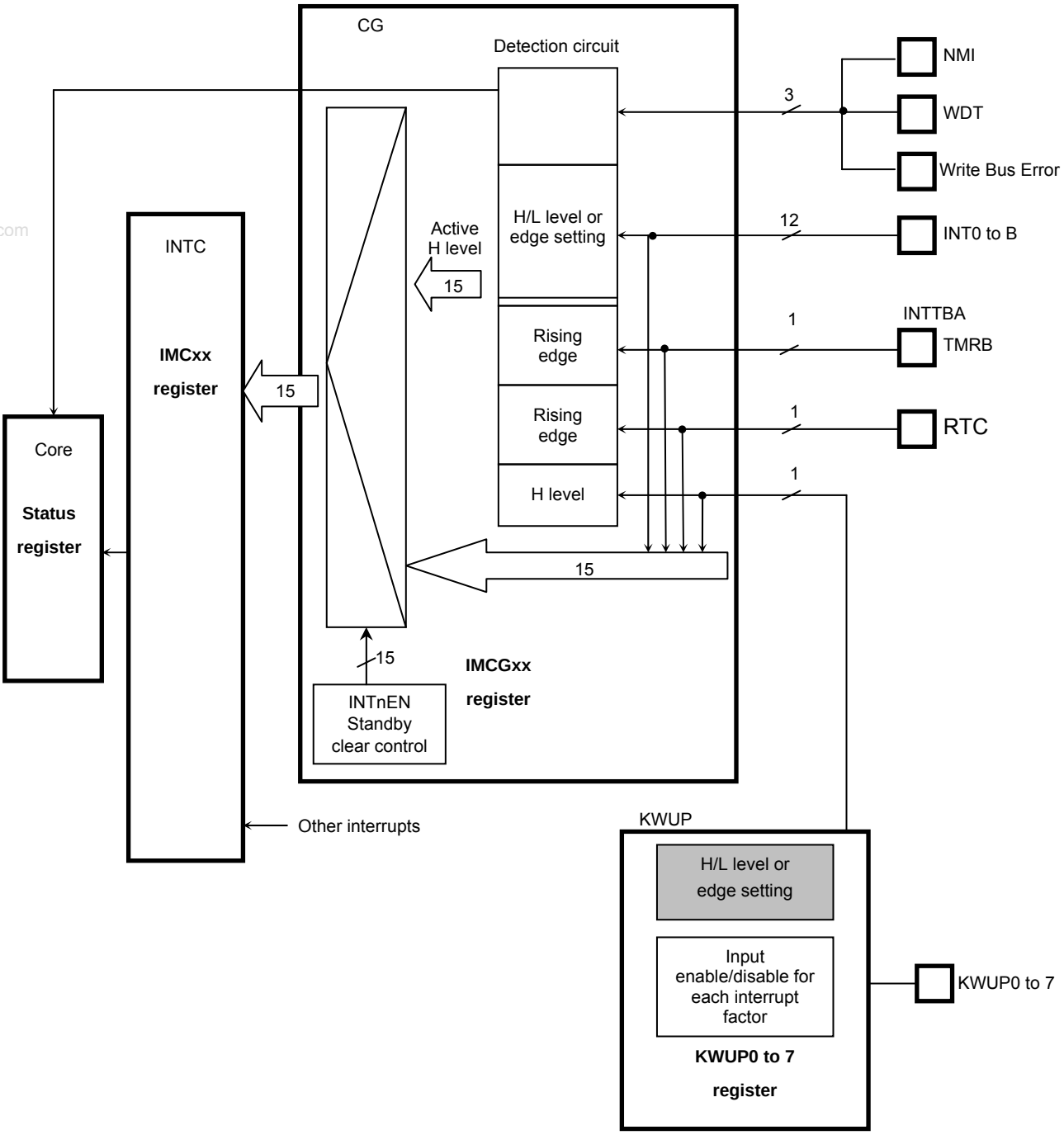


Fig. 6.1.1 Interrupt Connection Diagram

(2) Interrupts from external pins (INT0 - INTB and KWUP0-7)

When any external interrupt is to be used for setting to clear the Standby mode, use the following steps:

① Set ports

② Set functions

③ Set CG

④ Clear the EICRCG and INTCLR registers of CG

⑤ Enable interrupts with INT

a) INT0 - INTB

- If it is used to clear the Stop mode:

IMCGx<EMCGx2:0> = "xxx" : Set the standby clear request of each interrupt (INT0-B) to "active" (Refer to INTCG register).

IMCGx<INTxEN> = "1" : Set the clear input of each interrupt (INT0-B) to "enable" (Refer to INTCG register).

EICRCG<ICRCG3:0>= "xxxx" : Clear each interrupt request (INT0-B) (Refer to INTCG register).

INTCLR<EICLR8:0> = "000000100" : Clear interrupt requests INT0-B (Refer to INTCG register).

IMCx<EIMx1:0> = "01" : Set each interrupt request (INT0-B) to the H level (Refer to INTC register).

b) KWUP0-7

- If it is used to clear the Stop mode:

IMCGD<EMCGC1:0>= "01" : Set the KWUP standby clear request to "active" (Refer to INTCG register).

IMCGD<KWUPEN> = "1" : Set the KWUP clear input to "enable" (Refer to INTCG register).

IMC3<EIMD1:0> = "01" : Set KWUP interrupt request to the H level (Refer to INTC register).

IMC3<EIMD1:0> = "01" : Clear KWUP interrupt request (Refer to INTCG register).

INTCLR<EICLR8:0> = "000110100" : Clear KWUP interrupt request (Refer to INTCG register).

KWUPST<KEYINT7:0> = "1" : Set each KWUP interrupt factor to Enable (Refer to KWUP register).

Table 6.1.2 Registers to be Set for Detecting Interrupts

Interrupt		Interrupt detection levels that can be used
INT0 - INTB,KWUP		When in use, set to a rising edge in INTC (if edge detection is set for CG) or to "H" level (if level detection is set for CG). Set the active state in CG. The "L" level, "H" level, falling edge, or rising edge setting can be selected in CG register.
Internal I/O	Others	Falling edge

(Note 1) Interrupt level 0 means that the interrupt is disabled.

(3) Interrupt operation

- Basic interrupt handling

  ○ In the interrupt handler (Refer to Table 6.2.1 Interrupt Jump Address for the starting address of the interrupt handler):

  - Read the IVR value (in the figure, IVR value is 0x8000)
  - Substitute the IVR value for ICLR to clear the interrupt factor.
  - Obtain the exception handling jump address by using the IVR value (in the figure, it is 0x8000) as the corresponding address in the table (in the figure, the "jump to" address is 0x9000).
  - Jump to the exception handling routine using the "jump to" address.

  ○ In the interrupt processing routine:

  - Execute the interrupt processing
  - Set ILEV <MLEV> = 0 to return to the mask level before the exception is generated.
  - Command "ERET" to return to the routine before the exception is generated.

  Note that interrupts are disabled during the exception handling except for the case multiple interrupts are allowed.

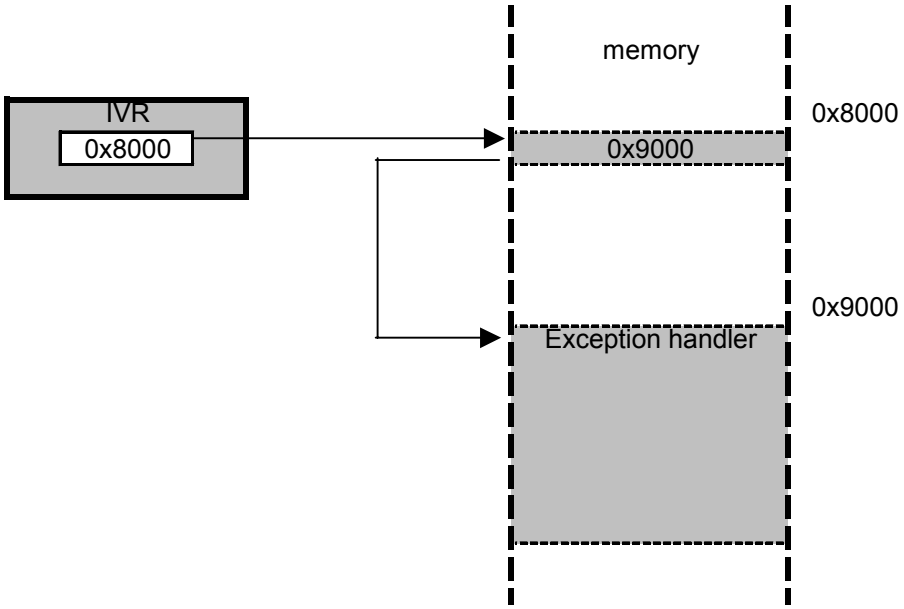


Fig. 6.1.3 Process Flow in the Interrupt Handler

6.2 Interrupt Factor

The starting address of an exception handler is defined as "exception vector address." The exception vector address for a reset exception and non-maskable interrupts is 0xBFC0_0000. The exception vector address for a debug exception is 0xBFC0_0480 (EJTAG ProbEn = 0). For other exceptions, the corresponding exception vector addresses are determined depending on the BEV bit of Status register [23] and the IV bit of the Cause register [23] of the system control coprocessor register (CP0).

Table 6.2.1 Interrupt Branch Address

Exception	BEV=0		BEV=1	
	Virtual address	Logical address	Virtual address	Logical address
Reset	0xBFC0_0000	0x1FC0_0000	0xBFC0_0000	0x1FC0_0000
EJTAG Debug (En=0)	0xBFC0_0480	0x1FC0_0480	0xBFC0_0480	0x1FC0_0480
EJTAG Debug (En=1)	0xFF20_0200	0xFF20_0200	0xFF20_0200	0xFF20_0200
Interrupt (IV=0)	0x8000_0180	0x0000_0180	0xBFC0_0380	0x1FC0_0380
Interrupt (IV=1)	0x8000_0200	0x0000_0200	0xBFC0_0400	0x1FC0_0400
All others	0x8000_0180	0x0000_0180	0xBFC0_0380	0x1FC0_0380

(Note 1) If vector addresses are to be placed in the internal ROM, set the status bit <BEV> of the system control coprocessor register (CP0) to "1."

(Note 2) The "software interrupt," which is a maskable interrupt, can be generated by setting IP [1:0] of the Cause register of CP0. This "software interrupt" is different from the "software set," which is one of the hardware interrupt factors. The "software set" interrupt is generated by setting <IL02:0> of the IMC0 register in the interrupt controller (INTC) to any value other than "0."

Table 6.2.2 List of Hardware Interrupt Factors

Interrupt Number	IVR[8:0]	Interrupt Factor	Interrupt Control Register	Address
0	0x000	Software set	IMC0	0xFFFF_E000
1	0x004	INT0 pin		
2	0x008	INT1 pin		
3	0x00C	INT2 pin		
4	0x010	INT3 pin	IMC1	0xFFFF_E004
5	0x014	INT4 pin		
6	0x018	INT5 pin		
7	0x01C	INT6 pin		
8	0x020	INT7 pin	IMC2	0xFFFF_E008
9	0x024	INT8 pin		
10	0x028	INT9 pin		
11	0x02C	INTA pin		
12	0x030	INTB pin	IMC3	0xFFFF_E00C
13	0x034	KWUP		
14	0x038	INTRX0 : Serial receiving (channel.0)		
15	0x03C	INTTX0 : Serial transmit (channel.0)		
16	0x040	INTRX1 : Serial receiving (channel.1)	IMC4	0xFFFF_E010
17	0x044	INTTX1 : Serial transmit (channel.1)		
18	0x048	INTRX2 : Serial receiving (channel.2)		
19	0x04C	INTTX2 : Serial transmit (channel.2)		
20	0x050	INTSBI : Serial bus interface 0	IMC5	0xFFFF_E014
21	0x054	INTADHP : Highest priority ADC complete interrupt		
22	0x058	INTADM : ADC monitor function interrupt		
23	0x05C	INTTB0 : 16-bit timer 0		
24	0x060	INTTB1 : 16-bit timer 1	IMC6	0xFFFF_E018
25	0x064	INTTB2 : 16-bit timer 2		
26	0x068	INTTB3 : 16-bit timer 3		
27	0x06C	INTTB4 : 16-bit timer 4		
28	0x070	INTCAPG : Input capture group	IMC7	0xFFFF_E01C
29	0x074	INTCMP0 : Compare interrupt 0		
30	0x078	INTCMP1 : Compare interrupt 1		
31	0x07C	INTCMP2 : Compare interrupt 2		
32	0x080	INTCMP3 : Compare interrupt 3	IMC8	0xFFFF_E020
33	0x084	INTCMP4 : Compare interrupt 4		
34	0x088	reserved		
35	0x08C	INTRX3 : Serial receiving (channel.3)		
36	0x090	INTTX3 : Serial transmit (channel.3)	IMC9	0xFFFF_E024
37	0x094	INTRX4 : Serial receiving (channel.4)		
38	0x098	INTTX4 : Serial transmit (channel.4)		
39	0x09C	INTRX5 : Serial receiving (channel.5)		
40	0x0A0	INTTX5 : Serial transmit (channel.5)	IMCA	0xFFFF_E028
41	0x0A4	INTRX6 : Serial receiving (channel.6)		
42	0x0A8	INTTX6 : Serial transmit (channel.6)		
43	0x0AC	INTTB5 : 16-bit timer 5		
44	0x0B0	INTTB6 : 16-bit timer 6	IMCB	0xFFFF_E02C
45	0x0B4	INTTB7 : 16-bit timer 7		
46	0x0B8	INTTB8 : 16-bit timer 8		
47	0x0BC	INTTB9 : 16-bit timer 9		
48	0x0C0	INTTBA : 16-bit timer A	IMCC	0xFFFF_E030
49	0x0C4	INTCMP5 : Compare interrupt 5		
50	0x0C8	INTCMP6 : Compare interrupt 6		
51	0x0CC	INTCMP7 : Compare interrupt 7		
52	0x0D0	INTCMP8 : Compare interrupt 8	IMCD	0xFFFF_E034
53	0x0D4	INTCMP9 : Compare interrupt 9		
54	0x0D8	INTRTC : Clock timer		
55	0x0DC	INTAD : ADC completed		
56	0x0E0	INTDMA0 : Completion of DMA transfer (channel.0)	IMCE	0xFFFF_E038
57	0x0E4	INTDMA1 : Completion of DMA transfer (channel.1)		
58	0x0E8	INTDMA2 : Completion of DMA transfer (channel.2)		
59	0x0EC	INTDMA3 : Completion of DMA transfer (channel.3)		
60	0x0F0	INTDMA4 : Completion of DMA transfer (channel.4)	IMCF	0xFFFF_E03C
61	0x0F4	INTDMA5 : Completion of DMA transfer (channel.5)		
62	0x0F8	INTDMA6 : Completion of DMA transfer (channel.6)		
63	0x0FC	INTDMA7 : Completion of DMA transfer (channel.7)		

Table 6.2.3 Interrupt Factors to Cancel Stop/Sleep/Idle Modes

Number	Interrupt Factor	Note
0	INT0	External interrupt 0
1	INT1	External interrupt 1
2	INT2	External interrupt 2
3	INT3	External interrupt 3
4	INT4	External interrupt 4
5	INT5	External interrupt 5
6	INT6	External interrupt 6
7	INT7	External interrupt 7
8	INT8	External interrupt 8
9	INT9	External interrupt 9
10	INTA	External interrupt A
11	INTB	External interrupt B
12	KWUP	Key on wake up interrupt
13	INTRTC	Clock timer interrupt
14	INTTBA	Two-phase pulse input counter interrupt
15	reserved	

* Number 0 to 13 interrupt factors can cancel Stop/Sleep modes.

* Number 14 interrupt factor can cancel the Sleep mode.

* Each factor can clear the IDLE mode.

6.3 Interrupt Detection

If any interrupt is used to cancel the Stop mode, interrupt active states of INT0 to INTB must be set in the EMCGxx field of the IMCGx register in CG and the EIMxx of the IMCx register in INTC must be set to "H" level. For KWUP0 to 7, the EMCG field of the IMCGD register in CG must be set to "H" and the EIMxx field of the IMCx register in INTC must be set to "H" level. The active state as well as enable/disable is set in KWUPSTn for each interrupt. For setting other interrupts, the EIMxx field of the IMCx register in INTC is used. Four types of active states, "H" level, "L" level, rising edge, and falling edge, are used. When the interrupt detection circuit of TMP19A64 recognizes that any input state matches with the predefined active state, it notifies the processor core or INTC of an interrupt request. If the interrupts that can be used to cancel the Stop mode are not to be used for canceling Stop mode, it is unnecessary to configure them in CG. In this case, INT0 to INTB can be set only by INTC and KWUP0 to 7 can be set in INTC and KWUPSTx.

The interrupt signal is negated by the interrupt handler after the interrupt factor is identified.

In the case of INT0 to INTB, appropriate values are written to the ICRCG field of the EICRCG register and to the EICLR field of the INTCLR register in INTC. KWUP0 to 7 are negated by setting KWUPCLR. Other interrupt signals are negated by writing a given value in the EICLR field of the INTCLR register in the INTC. To negate the interrupt factor whose active state is level-sensitive, an external circuit that has asserted the INTx signal must be operated so that it negates INTx. However, please ensure that the level input is not negated until the specified interrupt vector (IVR) has been read.

(Note) Please ensure that each setting is performed in the order of setting the active state, clearing an interrupt request, and enabling an interrupt.

(Example INT0 setting to cancel Stop mode)

IMCGA<EMCG01:00>	= "10"	:	Set INT0 active state to falling edge.
EICRCG<ICRCG3:0>	= "0000"	:	Clear the INT0 interrupt request.
IMCGA<INT0EN>	= "1"	:	Enable INT0 cancel input.
IMC0<EIM11:10>	= "01"	:	Set INT0 to "H" level.
INTCLR<EICLR8:0>	= "000000100"	:	Clear the INT0 interrupt request.
IMC0<IL12:10>	= "101"	:	Set the interrupt level of "5."
Status<IE> = "1," <IM>	= "xxx"		TX19A processor core

6.4 Interrupt Priority Arbitration

(1) Seven levels of interrupt priority

Seven levels of priority are available and each interrupt factor can be assigned to one of these levels. The interrupt level is set by the interrupt mode control register (IMCx) which has a 3-bit field (ILx) for level settings. The greater the value (interrupt level) set in IMCx <ILx2:0>, the higher the priority. If the value is set to "000" meaning the interrupt level of 0, no interrupts will be generated by the factor.

(2) Interrupt level notification

If an interrupt is generated, the INTC notifies the TX19A processor core of the interrupt level. The TX19A processor core identifies the interrupt level by reading the values in the IP field in the Cause register. If two or more interrupts (with different interrupt levels) are generated simultaneously, the INTC notifies the TX19A processor core of the highest-level interrupt factor and the lower level interrupt factors are suspended.

(3) Interrupt vector (notification of interrupt factor)

If an interrupt is generated, the INTC sets the corresponding interrupt factor vector in the vector register (IVR). The TX19A processor core identifies the interrupt factor by reading the vector register value. If two or more interrupts (with the same interrupt level) are generated simultaneously, the INTC notifies the TX19A processor core of the factor of which request number is younger. When no interrupt factors have been generated, the IVR <8:2> field is "0" (By clearing interrupt requests, the IVR register is cleared to "0.")

6.5 INTC Register

Table 6.5.1 INTC Register Map

Address	Register symbol	Register	Corresponding interrupt number
0xFFFF_E000	IMC0	Interrupt mode control register 0	3 - 0
0xFFFF_E004	IMC1	Interrupt mode control register 1	7 - 4
0xFFFF_E008	IMC2	Interrupt mode control register 2	11 - 8
0xFFFF_E00C	IMC3	Interrupt mode control register 3	15 - 12
0xFFFF_E010	IMC4	Interrupt mode control register 4	19 - 16
0xFFFF_E014	IMC5	Interrupt mode control register 5	23 - 20
0xFFFF_E018	IMC6	Interrupt mode control register 6	27 - 24
0xFFFF_E01C	IMC7	Interrupt mode control register 7	31 - 28
0xFFFF_E020	IMC8	Interrupt mode control register 8	35 - 32
0xFFFF_E024	IMC9	Interrupt mode control register 9	39 - 36
0xFFFF_E028	IMCA	Interrupt mode control register A	43 - 40
0xFFFF_E02C	IMCB	Interrupt mode control register B	47 - 44
0xFFFF_E030	IMCC	Interrupt mode control register C	51 - 48
0xFFFF_E034	IMCD	Interrupt mode control register D	55 - 52
0xFFFF_E038	IMCE	Interrupt mode control register E	59 - 56
0xFFFF_E03C	IMCF	Interrupt mode control register F	63 - 60
0xFFFF_E040	IVR	Interrupt vector register	
0xFFFF_E060	INTCLR	Interrupt request clear register	
0xFFFF_E10C	ILEV	Interrupt level register	

(Note) Unless otherwise specified, the above registers must be 32-bit accessed for both reading and writing.

6.5.1 Interrupt Vector Register (IVR)

The vector of each interrupt factor to be generated is listed below.

IVR (0xFFFF_E040)

	7	6	5	4	3	2	1	0
bit Symbol	IVR7	IVR6	IVR5	IVR4	IVR3	IVR2	IVR1	IVR0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	The vector of the interrupt factor generated is set.							

	15	14	13	12	11	10	9	8
bit Symbol								IVR8
Read/Write	R/W							R
After reset	0	0	0	0	0	0	0	0
Function								The vector of the interrupt factor generated is set.

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function								

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function								

6.5.2 Interrupt Level Register

ILEV (0xFFFF_E10C)

	7	6	5	4	3	2	1	0
bit Symbol	—	PMASK0			—	CMASK		
Read/Write	R					R/W (Note 1)		
After reset	0	000			0	000		
Function	Always reads "0."	Interrupt mask level (previous) 0			Always reads "0."	Interrupt mask level (current)		
	15	14	13	12	11	10	9	8
bit Symbol	—	PMASK2			—	PMASK1		
Read/Write	R							
After reset	0	000			0	000		
Function	Always reads "0."	Interrupt mask level (previous) 2			Always reads "0."	Interrupt mask level (previous) 1		
	23	22	21	20	19	18	17	16
bit Symbol	—	PMASK4			—	PMASK3		
Read/Write	R							
After reset	0	000			0	000		
Function	Always reads "0."	Interrupt mask level (previous) 4			Always reads "0."	Interrupt mask level (previous) 3		
	31	30	29	28	27	26	25	24
bit Symbol	MLEV	PMASK6			—	PMASK5		
Read/Write	W	R						
After reset	0	000			0	000		
Function	Interrupt level change 0: Decrement the interrupt level by 1 1: Change CMASK	Interrupt mask level (previous) 6			Always reads "0."	Interrupt mask level (previous) 5		

Note) This register must be 32-bit accessed.

Note) When a new interrupt is generated, the corresponding interrupt level is stored in CMASK and any previously stored values are shifted in their mask levels such that the previous CMASK is saved in PMASK0 and PMASK0 is saved in PMASK1 and so on.

Note 1) Upon setting MLEV to "1," set the CMASK value simultaneously. The PMASKx values are unchanged.

Note) When <MLEV> is set to "0," the interrupt mask levels in the register shift back to the previous state such that PMASK0 is moved to CMASK and PMASK1 is moved to PMASK0, and so on. The last <PMASK6> is set to "000." If it is to be used after the interrupt process, set MLEV to "0" before executing the ERET command.

6.5.3 Transition of Interrupt Mask Level

The transition sequence of the interrupt level register is illustrated below.

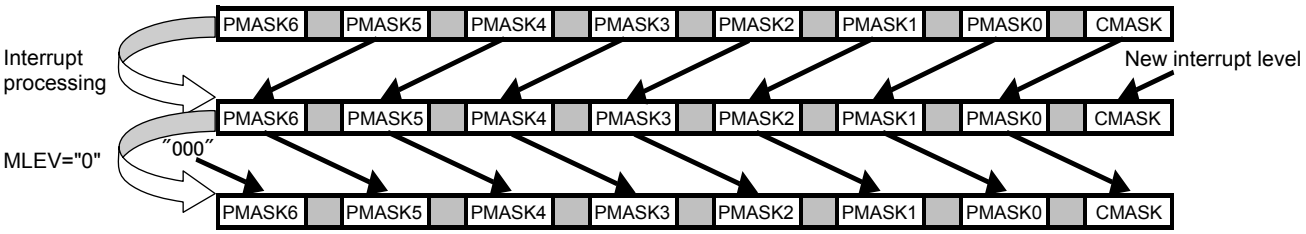


Fig. 6.5.3 Transition of Interrupt Mask Level

6.5.4 Interrupt Level Register (IMCx)

The interrupt level, active state, and whether it is a factor to activate DMAC or not are set for each interrupt factor.

IMC0 (0xFFFF_E000)

	7	6	5	4	3	2	1	0
bit Symbol		EIM01	EIM00	DM0		IL02	IL01	IL00
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 00: "L" level 01: Disable 10: Disable 11: Disable **Be sure to set "00."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 0 is set as the activation factor	Always reads "0."	If DM0 = 0, select the interrupt level for interrupt number 0 (software set). 000: Disable Interrupt 001 to 111: 1 to 7 If DM0 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM11	EIM10	DM1		IL12	IL11	IL10
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 1 to be the activation factor.	Always reads "0."	If DM1 = 0, select the interrupt level for interrupt number 1 (INT0). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM21	EIM20	DM2		IL22	IL21	IL20
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 2 to be the activation factor.	Always reads "0."	If DM2 = 0, select the interrupt level for interrupt number 2 (INT1). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM31	EIM30	DM3		IL32	IL31	IL30
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 3 to be the activation factor.	Always reads "0."	If DM3 = 0, select the interrupt level for interrupt number 3 (INT2). 000: Disable Interrupt 001 to 111: 1 to 7 If DM3 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

IMC1 (0xFFFF_E004)

	7	6	5	4	3	2	1	0
bit Symbol		EIM41	EIM40	DM4		IL42	IL41	IL40
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 4 is set as the activation factor	Always reads "0."	If DM4 = 0, select the interrupt level for interrupt number 4 (INT3) 000: Disable Interrupt 001 to 111: 1 to 7 If DM4 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM51	EIM50	DM5		IL52	IL51	IL50
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 5 to be the activation factor.	Always reads "0."	If DM5 = 0, select the interrupt level for interrupt number 5 (INT4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM5 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM61	EIM60	DM6		IL62	IL61	IL60
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 6 to be the activation factor.	Always reads "0."	If DM6 = 0, select the interrupt level for interrupt number 6 (INT5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM6 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM71	EIM70	DM7		IL72	IL71	IL70
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 7 to be the activation factor.	Always reads "0."	If DM7 = 0, select the interrupt level for interrupt number 7 (INT6). 000: Disable Interrupt 001 to 111: 1 to 7 If DM7 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

IMC2 (0xFFFF_E008)

	7	6	5	4	3	2	1	0
bit Symbol		EIM81	EIM80	DM8		IL82	IL81	IL80
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 8 is set as the activation factor	Always reads "0."	If DM8 = 0, select the interrupt level for interrupt number 8 (INT7). 000: Disable Interrupt 001 to 111: 1 to 7 If DM8 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM91	EIM90	DM9		IL92	IL91	IL90
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 9 to be the activation factor.	Always reads "0."	If DM9 = 0, select the interrupt level for interrupt number 9 (INT8). 000: Disable Interrupt 001 to 111: 1 to 7 If DM9 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIMA1	EIMA0	DMA		ILA2	ILA1	ILA0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 10 to be the activation factor.	Always reads "0."	If DMA = 0, select the interrupt level for interrupt number 10 (INT9). 000: Disable Interrupt 001 to 111: 1 to 7 If DMA = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIMB1	EIMB0	DMB		ILB2	ILB1	ILB0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 11 to be the activation factor.	Always reads "0."	If DMB = 0, select the interrupt level for interrupt number 11 (INTA) 000: Disable Interrupt 001 to 111: 1 to 7 If DMB = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

IMC3 (0xFFFF_E00C)

	7	6	5	4	3	2	1	0
bit Symbol		EIMC1	EIMC0	DMC		ILC2	ILC1	ILC0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 12 is set as the activation factor	Always reads "0."	If DMC = 0, select the interrupt level for interrupt number 12 (INTB) 000: Disable Interrupt 001 to 111: 1 to 7 If DMC = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIMD1	EIMD0	DMD		ILD2	ILD1	ILD0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 01: "H" level **Be sure to set "01."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 13 to be the activation factor.	Always reads "0."	If DMD = 0, select the interrupt level for interrupt number 13 (KWUP) 000: Disable Interrupt 001 to 111: 1 to 7 If DMD = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIME1	EIME0	DME		ILE2	ILE1	ILE0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 14 to be the activation factor.	Always reads "0."	If DME = 0, select the interrupt level for interrupt number 14 (INTRX0) 000: Disable Interrupt 001 to 111: 1 to 7 If DME = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIMF1	EIMF0	DMF		ILF2	ILF1	ILF0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 15 to be the activation factor.	Always reads "0."	If DMF = 0, select the interrupt level for interrupt number 15 (INTTX0) 000: Disable Interrupt 001 to 111: 1 to 7 If DMF = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

IMC4 (0xFFFF_E010)

	7	6	5	4	3	2	1	0
bit Symbol		EIM101	EIM100	DM10		IL102	IL101	IL100
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 16 is set as the activation factor	Always reads "0."	If DM10 = 0, select the interrupt level for interrupt number 16 (INTRX1) 000: Disable Interrupt 001 to 111: 1 to 7 If DM10 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM111	EIM110	DM11		IL112	IL111	IL110
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 17 to be the activation factor.	Always reads "0."	If DM11 = 0, select the interrupt level for interrupt number 17 (INTTX1) 000: Disable Interrupt 001 to 111: 1 to 7 If DM11 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM121	EIM120	DM12		IL122	IL121	IL120
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 18 to be the activation factor.	Always reads "0."	If DM12 = 0, select the interrupt level for interrupt number 18 (INTRX2). 000: Disable Interrupt 001 to 111: 1 to 7 If DM12 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM131	EIM130	DM13		IL132	IL131	IL130
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 19 to be the activation factor.	Always reads "0."	If DM13 = 0, select the interrupt level for interrupt number 19 (INTTX2) 000: Disable Interrupt 001 to 111: 1 to 7 If DM13 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMC5 (0xFFFF_E014)

	7	6	5	4	3	2	1	0
bit Symbol		EIM141	EIM140	DM14		IL142	IL141	IL140
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 20 is set as the activation factor	Always reads "0."	If DM14 = 0, select the interrupt level for interrupt number 20 (INTSB1). 000: Disable Interrupt 001 to 111: 1 to 7 If DM14 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM151	EIM150	DM15		IL152	IL151	IL150
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 21 to be the activation factor.	Always reads "0."	If DM15 = 0, select the interrupt level for interrupt number 21 (INTADHP) 000: Disable Interrupt 001 to 111: 1 to 7 If DM15 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM161	EIM160	DM16		IL162	IL161	IL160
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 22 to be the activation factor.	Always reads "0."	If DM16 = 0, select the interrupt level for interrupt number 22 (INTADM). 000: Disable Interrupt 001 to 111: 1 to 7 If DM16 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM171	EIM170	DM17		IL172	IL171	IL170
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 23 to be the activation factor.	Always reads "0."	If DM17 = 0, select the interrupt level for interrupt number 23 (INTTB0). 000: Disable Interrupt 001 to 111: 1 to 7 If DM17 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMC6 (0xFFFF_E018)

	7	6	5	4	3	2	1	0
bit Symbol		EIM181	EIM180	DM18		IL182	IL181	IL180
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 24 is set as the activation factor	Always reads "0."	If DM18 = 0, select the interrupt level for interrupt number 24 (INTTB1). 000: Disable Interrupt 001 to 111: 1 to 7 If DM18 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM191	EIM190	DM19		IL192	IL191	IL190
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 25 to be the activation factor.	Always reads "0."	If DM19 = 0, select the interrupt level for interrupt number 25 (INTTB2). 000: Disable Interrupt 001 to 111: 1 to 7 If DM19 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM1A1	EIM1A0	DM1A		IL1A2	IL1A1	IL1A0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 26 to be the activation factor.	Always reads "0."	If DM1A = 0, select the interrupt level for interrupt number 26 (INTTB3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1A = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM1B1	EIM1B0	DM1B		IL1B2	IL1B1	IL1B0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 27 to be the activation factor.	Always reads "0."	If DM1B = 0, select the interrupt level for interrupt number 27 (INTTB4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1B = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMC7 (0xFFFF_E01C)

	7	6	5	4	3	2	1	0
bit Symbol		EIM1C1	EIM1C0	DM1C		IL1C2	IL1C1	IL1C0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 28 to be the activation factor.	Always reads "0."	If DM1C = 0, select the interrupt level for interrupt number 28 (INTCAPG). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1C = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM1D1	EIM1D0	DM1D		IL1D2	IL1D1	IL1D0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 29 to be the activation factor.	Always reads "0."	If DM1D = 0, select the interrupt level for interrupt number 29 (INTCOMP0). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1D = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM1E1	EIM1E0	DM1E		IL1E2	IL1E1	IL1E0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 30 to be the activation factor.	Always reads "0."	If DM1E = 0, select the interrupt level for interrupt number 30 (INTCMP1). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1E = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM1F1	EIM1F0	DM1F		IL1F2	IL1F1	IL1F0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 31 to be the activation factor.	Always reads "0."	If DM1F = 0, select the interrupt level for interrupt number 31 (INTCMP2) 000: Disable Interrupt 001 to 111: 1 to 7 If DM1F = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMC8 (0xFFFF_E020)

	7	6	5	4	3	2	1	0
bit Symbol		EIM201	EIM200	DM20		IL202	IL201	IL200
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 32 to be the activation factor.	Always reads "0."	If DM20 = 0, select the interrupt level for interrupt number 32 (INTCMP3) 000: Disable Interrupt 001 to 111: 1 to 7 If DM20 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM211	EIM210	DM21		IL212	IL211	IL210
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 33 to be the activation factor.	Always reads "0."	If DM21 = 0, select the interrupt level for interrupt number 33 (INTCMP4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM21 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM221	EIM220	DM26		IL222	IL221	IL220
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	**Be sure to set "00."**		**Be sure to set "0."**	Always reads "0."	**Be sure to set "00."**		

	31	30	29	28	27	26	25	24
bit Symbol		EIM231	EIM230	DM23		IL232	IL231	IL230
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 35 to be the activation factor.	Always reads "0."	If DM23 = 0, select the interrupt level for interrupt number 35 (INTRX3) 000: Disable Interrupt 001 to 111: 1 to 7 If DM23 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMC9 (0xFFFF_E024)

	7	6	5	4	3	2	1	0
bit Symbol		EIM241	EIM240	DM24		IL242	IL241	IL240
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 36 to be the activation factor.	Always reads "0."	If DM24 = 0, select the interrupt level for interrupt number 36 (INTTX3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM24 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM251	EIM250	DM25		IL252	IL251	IL250
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 37 to be the activation factor.	Always reads "0."	If DM25 = 0, select the interrupt level for interrupt number 37 INTRX4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM25 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM261	EIM260	DM26		IL262	IL261	IL260
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 38 to be the activation factor.	Always reads "0."	If DM26 = 0, select the interrupt level for interrupt number 38 (INTTX4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM26 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM271	EIM270	DM27		IL272	IL271	IL270
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 39 to be the activation factor.	Always reads "0."	If DM27 = 0, select the interrupt level for interrupt number 39 (INTRX5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM27 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMCA (0xFFFF_E028)

	7	6	5	4	3	2	1	0
bit Symbol		EIM281	EIM280	DM28		IL282	IL281	IL280
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 40 to be the activation factor.	Always reads "0."	If DM28 = 0, select the interrupt level for interrupt number 40 (INTTX5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM28 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM291	EIM290	DM29		IL292	IL291	IL290
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 41 to be the activation factor.	Always reads "0."	If DM29 = 0, select the interrupt level for interrupt number 41 (INTRX6). 000: Disable Interrupt 001 to 111: 1 to 7 If DM29 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM2A1	EIM2A0	DM2A		IL2A2	IL2A1	IL2A0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 42 to be the activation factor.	Always reads "0."	If DM2A = 0, select the interrupt level for interrupt number 42 (INTTX6). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2A = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM2B1	EIM2B0	DM2B		IL2B2	IL2B1	IL2B0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 43 to be the activation factor.	Always reads "0."	If DM2B = 0, select the interrupt level for interrupt number 43 (INTTB5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2B = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMCB (0xFFFF_E02C)

	7	6	5	4	3	2	1	0
bit Symbol		EIM2C1	EIM2C0	DM2C		IL2C2	IL2C1	IL2C0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 44 to be the activation factor.	Always reads "0."	If DM2C = 0, select the interrupt level for interrupt number 44 (INTTB6). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2C = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM2D1	EIM2D0	DM2D		IL2D2	IL2D1	IL2D0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 45 to be the activation factor.	Always reads "0."	If DM2D = 0, select the interrupt level for interrupt number 45 (INTTB7). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2D = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM2E1	EIM2E0	DM2E		IL2E2	IL2E1	IL2E0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 46 to be the activation factor.	Always reads "0."	If DM2E = 0, select the interrupt level for interrupt number 46 (INTTB8). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2E = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM2F1	EIM2F0	DM2F		IL2F2	IL2F1	IL2F0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 47 to be the activation factor.	Always reads "0."	If DM2F = 0, select the interrupt level for interrupt number 47 (INTTB9). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2F = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMCC (0xFFFF_E030)

	7	6	5	4	3	2	1	0
bit Symbol		EIM301	EIM300	DM30		IL302	IL301	IL300
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 01: "H" level **Be sure to set "01."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 48 to be the activation factor.	Always reads "0."	If DM30 = 0, select the interrupt level for interrupt number 48 (INTTBA). 000: Disable Interrupt 001 to 111: 1 to 7 If DM30 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM311	EIM310	DM31		IL312	IL311	IL310
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 1: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 49 to be the activation factor.	Always reads "0."	If DM31 = 0, select the interrupt level for interrupt number 49 (INTCMP5) 000: Disable Interrupt 001 to 111: 1 to 7 If DM31 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM321	EIM320	DM32		IL322	IL321	IL320
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 1: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 50 to be the activation factor.	Always reads "0."	If DM32 = 0, select the interrupt level for interrupt number 50 (INTCMP6) 000: Disable Interrupt 001 to 111: 1 to 7 If DM32 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM331	EIM330	DM33		IL332	IL331	IL330
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 51 to be the activation factor.	Always reads "0."	If DM33 = 0, select the interrupt level for interrupt number 51 (INTCMP7) 000: Disable Interrupt 001 to 111: 1 to 7 If DM33 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMCD (0xFFFF_E034)

	7	6	5	4	3	2	1	0
bit Symbol		EIM341	EIM340	DM34		IL342	IL341	IL340
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 52 to be the activation factor.	Always reads "0."	If DM34 = 0, select the interrupt level for interrupt number 52 (INTCMP8) 000: Disable Interrupt 001 to 111: 1 to 7 If DM34 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM351	EIM350	DM35		IL352	IL351	IL350
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 53 to be the activation factor.	Always reads "0."	If DM35 = 0, select the interrupt level for interrupt number 53 (INTCMP9) 000: Disable Interrupt 001 to 111: 1 to 7 If DM35 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM361	EIM360	DM36		IL362	IL361	IL360
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 01: "H" level **Be sure to set "01."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 54 to be the activation factor.	Always reads "0."	If DM36 = 0, select the interrupt level for interrupt number 54 (INTRTC) 000: Disable Interrupt 001 to 111: 1 to 7 If DM36 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM371	EIM370	DM37		IL372	IL371	IL370
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 55 to be the activation factor.	Always reads "0."	If DM37 = 0, select the interrupt level for interrupt number 55 (INTAD) 000: Disable Interrupt 001 to 111: 1 to 7 If DM37 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMCE (0xFFFF_E038)

	7	6	5	4	3	2	1	0
bit Symbol		EIM381	EIM380	DM38		IL382	IL381	IL380
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 56 to be the activation factor.	Always reads "0."	If DM38 = 0, select the interrupt level for interrupt number 56 (INTDMA0) 000: Disable Interrupt 001 to 111: 1 to 7 If DM38 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM391	EIM390	DM39		IL392	IL391	IL390
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 57 to be the activation factor.	Always reads "0."	If DM39 = 0, select the interrupt level for interrupt number 57 (INTDM1) 000: Disable Interrupt 001 to 111: 1 to 7 If DM39 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM3A1	EIM3A0	DM3A		IL3A2	IL3A1	IL3A0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 58 to be the activation factor.	Always reads "0."	If DM3A = 0, select the interrupt level for interrupt number 58 (INTDMA2) 000: Disable Interrupt 001 to 111: 1 to 7 If DM3A = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM3B1	EIM3B0	DM3B		IL3B2	IL3B1	IL3B0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 59 to be the activation factor.	Always reads "0."	If DM3B = 0, select the interrupt level for interrupt number 59 (INTDMA3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM3B = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

IMCF (0xFFFF_E03C)

	7	6	5	4	3	2	1	0
bit Symbol		EIM3C1	EIM3C0	DM3C		IL3C2	IL3C1	IL3C0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 60 to be the activation factor.	Always reads "0."	If DM3C = 0, select the interrupt level for interrupt number 60 (INTDMA4) 000: Disable Interrupt 001 to 111: 1 to 7 If DM3C = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM3D1	EIM3D0	DM3D		IL3D2	IL3D1	IL3D0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 61 to be the activation factor.	Always reads "0."	If DM3D = 0, select the interrupt level for interrupt number 61 (INTDMA5) 000: Disable Interrupt 001 to 111: 1 to 7 If DM3D = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM3E1	EIM3E0	DM3E		IL3E2	IL3E1	IL3E0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 62 to be the activation factor.	Always reads "0."	If DM3E = 0, select the interrupt level for interrupt number 62 (INTDMA6). 000: Disable Interrupt 001 to 111: 1 to 7 If DM3E = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM3F1	EIM3F0	DM3F		IL3F2	IL3F1	IL3F0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 63 to be the activation factor.	Always reads "0."	If DM3F = 0, select the interrupt level for interrupt number 63 (INTDMA7). 000: Disable Interrupt 001 to 111: 1 to 7 If DM3F = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

Note: Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.

Note 1: Please ensure that the type of active state is selected before enabling an interrupt request.

Note 2: When making interrupt requests DMAC activation factors, please ensure that you put the DMAC into standby mode after setting the INTC.

6.5.5 Interrupt Request Clear Register

This register is used to clear interrupt requests. Interrupt requests are cleared by setting the IVR <IVR8:0> value.

INTCLR (0xFFFF_E060)

	7	6	5	4	3	2	1	0
bit Symbol	EICLR7	EICLR6	EICLR5	EICLR4	EICLR3	EICLR2	EICLR1	EICLR0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set the IVR <IVR8:0> value that corresponds to the interrupt request that you would like to clear.							
	15	14	13	12	11	10	9	8
bit Symbol								EICLR8
Read/Write	R							R/W
After reset	0							0
Function	Always reads "0."							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	Always reads "0."							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	Always reads "0."							

(Note 1) Do not clear interrupt requests before reading the IVR value. If an interrupt request is cleared, IVR is cleared to "0."

(Note 2) To make the interrupt controller (INTC) disable specified interrupt requests, perform the following steps in the order shown:

① Disable the processor core to accept interrupts (Status <IE> = 0).

② Disable the INTC to accept interrupts (IMCxx<ILx2:0> = 000).

③ Execute the SYNC instruction.

④ Enable the processor core to accept interrupts (Status <IE> = 1).

```
Example)  mtc0   r0,   r31    ; _DI ( );
          sb     r0,   IMC**  ; IMC**=0;
          sync                ; _SYNC( );
          mtc0   $sp,  r31    ; _EI ( );
```

(Note 3) Any internal DMA request initiated by an interrupt factor will not be cleared. When the request is to be canceled, clear the activation factor bit of (IMCx) <DMxx>.

6.5.6 INTCG Registers (Interrupts to clear Stop, Sleep and Idle modes)

- INT0 to INTB, KWUP0 to KWUP7: STOP/SLEEP/IDLE
- INTRTC, INTTBA (Two-phase pulse input counter): Sleep

IMCGA (0xFFFF_EE10)

	7	6	5	4	3	2	1	0
bit Symbol			EMCG01	EMCG00				INT0EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT0 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT0 Clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol			EMCG11	EMCG10				INT1EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT1 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT1 Clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol			EMCG21	EMCG20				INT2EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT2 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT2 Clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol			EMCG31	EMCG30				INT3EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT3 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT3 Clear input 0: Disable 1: Enable

IMCGB (0xFFFF_EE14)

	7	6	5	4	3	2	1	0
bit Symbol			EMCG41	EMCG40				INT4EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT4 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT4 Clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol			EMCG51	EMCG50				INT5EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT5 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT5 Clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol			EMCG61	EMCG60				INT6EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT6 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT6 Clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol			EMCG71	EMCG70				INT7EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT7 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT7 Clear input 0: Disable 1: Enable

IMCGC (0xFFFF_EE18)

	7	6	5	4	3	2	1	0
bit Symbol			EMCG81	EMCG80				INT8EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT8 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT8 Clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol			EMCG91	EMCG90				INT9EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT9 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT9 Clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol			EMCGA1	EMCGA0				INTAEN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INTA standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INTA Clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol			EMCGB1	EMCGB0				INTBEN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INTB standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INTB Clear input 0: Disable 1: Enable

IMCGD (0xFFFF_EE1C)

	7	6	5	4	3	2	1	0
bit Symbol			EMCGC1	EMCGC0				KWUPEN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of KWUP standby clear request. 01: "H" level Be sure to set "01."		Always reads "0."	Always reads "0."	Always reads "0."	KWUP Clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol			EMCGD1	EMCGD0				INTRTCEN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INTRTC standby clear request. 11: Rising edge Be sure to set "11."		Always reads "0."	Always reads "0."	Always reads "0."	INTRTC Clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol			EMCGE1	EMCGE0				INTTBAEN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0			0
Function	Always reads "0."	Always reads "0."	Set active state of INTTBA standby clear request. 11: Rising edge Be sure to set "11."		Always reads "0."	Always reads "0."	Always reads "0."	INTTBA Clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0			0
Function	Always reads "0."	Always reads "0."	Undefined		Always reads "0."	Always reads "0."	Always reads "0."	Write "1."

Note: In IMCGD, the initial value to request clearing of the Standby mode is different from the setting to be made in an operation condition. Be sure to set appropriate parameters before it is used to clear the Standby mode.

Be sure to set active state of the clear request if interrupt is enabled for clearing the Stop, Sleep, or Idle standby mode.

(Note1) When using interrupts, be sure to follow the following sequence of action:
① If shared with other general ports, enable the target interrupt input.
② Set active state, etc., upon initialization.
③ Clear interrupt requests.
④ Enable interrupts

(Note 2) Settings must be performed while interrupts are disabled.

(Note 3) For clearing the Stop, Sleep and Idle modes with TMP19A64, 15 factors, i.e., INT0 to INTB, INTRTC, INTTBA, and KWUP (KWUP0 to 7) are available as clearing interrupts. Whether or not INT0 to INTB are to be used as clearing interrupts as well as active state edge/level selection is set with CG. Whether or not KWUP0 to 7 are to be used as STOP/SLEEP/IDLE clearing interrupts is set with CG and active state edge/level selection is set with KWUPSTn <KEYn>. Set to High level with INTC for the above 15 factors.

Example: Enabling INT0 interrupt

IMCGA<EMCG01:00> = "10"	CG block
IMCGA<INT0EN> = "1"	(Enable input by the falling edge)
IMC0<EIM11:10> = "01"	INTC block
IMC0<IL12:10> = "101"	(Set interrupt active level to "H" and the interrupt level to 5.)

Interrupt factors other than those assigned as Stop/Sleep/Idle clear requests are set in the INTC block.

(Note 4) Among the above 15 factors to be assigned as Stop/Sleep/Idle clear request interrupts, INT0 to INTB don't have to be set with CG if they are to be used as normal interrupts. Use INTC to specify either H/L level or rising/falling edge. If KWUP0 to 7 are to be used as normal interrupts, set the active level by KWUPSTn and set High level with INTC. No CG setting is necessary. Also, if INTRTC is to be used as a normal interrupt, use CG/INTC for the setting.

Interrupt factors other than those assigned as Stop/Sleep/Idle clear requests are set in the INTC block.

EICRCG
(0xFFFF_EE20)

	7	6	5	4	3	2	1	0
bit Symbol					ICRCG3	ICRCG2	ICRCG1	ICRCG0
Read/Write	R				W/R			
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."				Always reads "0." Clear interrupt requests. 0000: INT0 0101: INT5 1010: INTA 0001: INT1 0110: INT6 1011: INTB 0010: INT2 0111: INT7 1100: KWUP 0011: INT3 1000: INT8 1101: INTRTC 0100: INT4 1001: INT9 1110: INTTBA 1111: reserved			

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							

(Note 5) To clear interrupt request of the above 15 factors that are assigned to clear Stop/Sleep/Idle modes,

① For KWUP, use KWUPST

② For INT0 to INTB, INTTBA and INTRTC use the EICRCG register in the above CG block and then use the INTCLR register in the INTC block (two locations).

③ For clearing any other interrupt requests, only INTCLR register is to be cleared.

NMIFLG (0xFFFF_EE24)

	7	6	5	4	3	2	1	0
bit Symbol						NMI	WDT	WBER
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."					NMI factor 1: NMI generated by NMI pin input	NMI factor 1: NMI generated by WDT interrupt	NMI factor 1: NMI generated by write bus error
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							

- **NMI, WDT and WBER are cleared to "0" when they are read.**

7. Input/Output Ports

7.1 Port 0 (P00 through P07)

The port 0 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register P0CR. A reset allows all bits of P0CR to be cleared to "0" and the port 0 to be put in input mode.

Besides the general-purpose input/output function, the port 0 performs other functions: D0 through D7 function as a data bus and AD0 through AD7 function as an address data bus. When external memory is accessed, the port 0 automatically functions as either a data bus or an address data bus, and all bits of P0CR are cleared to "0."

If the BUSMD pin is set to "L" level during a reset, the port 0 is put in separate bus mode (D0 to D7). If it is set to "H" level during a reset, the port 0 is put in multiplexed mode (AD0 to AD7).

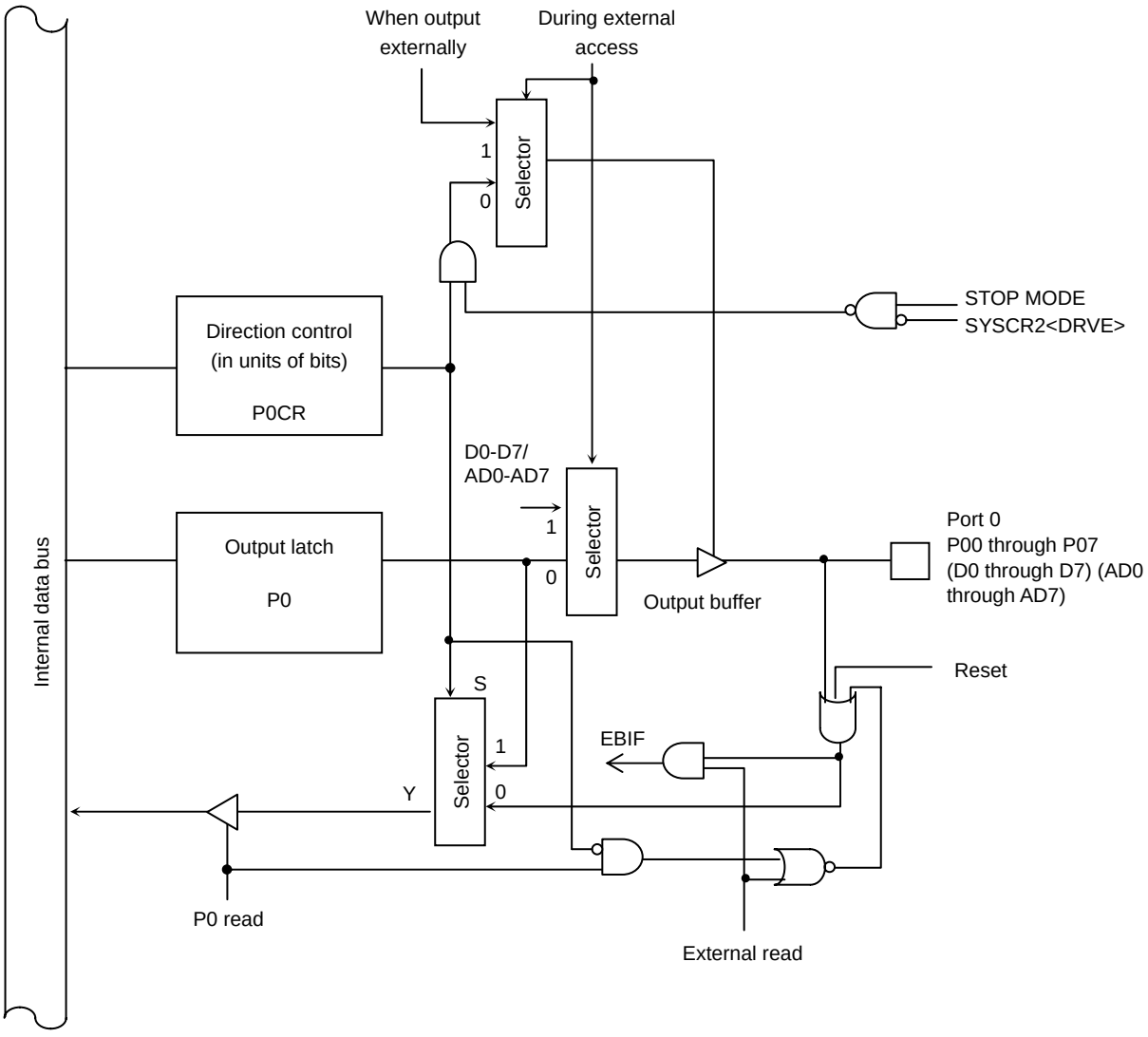


Fig. 7.1.1 Port 0 (P00 through P07)

Port 0 register

		7	6	5	4	3	2	1	0
P0	Bit Symbol	P07	P06	P05	P04	P03	P02	P01	P00
(0xFFFF_F000)	Read/Write	R/W							
	After reset	Input mode (output latch register is cleared to "0.")							

Port 0 control register

		7	6	5	4	3	2	1	0
P0CR	Bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
(0xFFFF_F002)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output (When an external area is accessed, D7-0 or AD7-0 is used and this register is cleared to "0.")							

Fig. 7.1.2 Port 0 Registers

7.2 Port 1 (P10 through P17)

The port 1 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register P1CR and the function register P1FC. A reset allows all bits of the output latch P1, P1CR and P1FC to be cleared to "0" and the port 1 to be put in input mode.

Besides the general-purpose input/output function, the port 1 performs other functions: D8 through D15 function as a data bus, AD8 through AD15 function as an address data bus, and A8 through A15 function as an address bus. To access external memory, the port 1 must be designated as an address bus or address data bus by making proper P1CR and P1FC settings.

If the BUSMD pin is set to "L" level during a reset, the port 1 is put in separate bus mode (D8 to D15). If it is set to "H" level during a reset, the port 1 is put in multiplexed mode (AD8 to AD15 or A8 to A15).

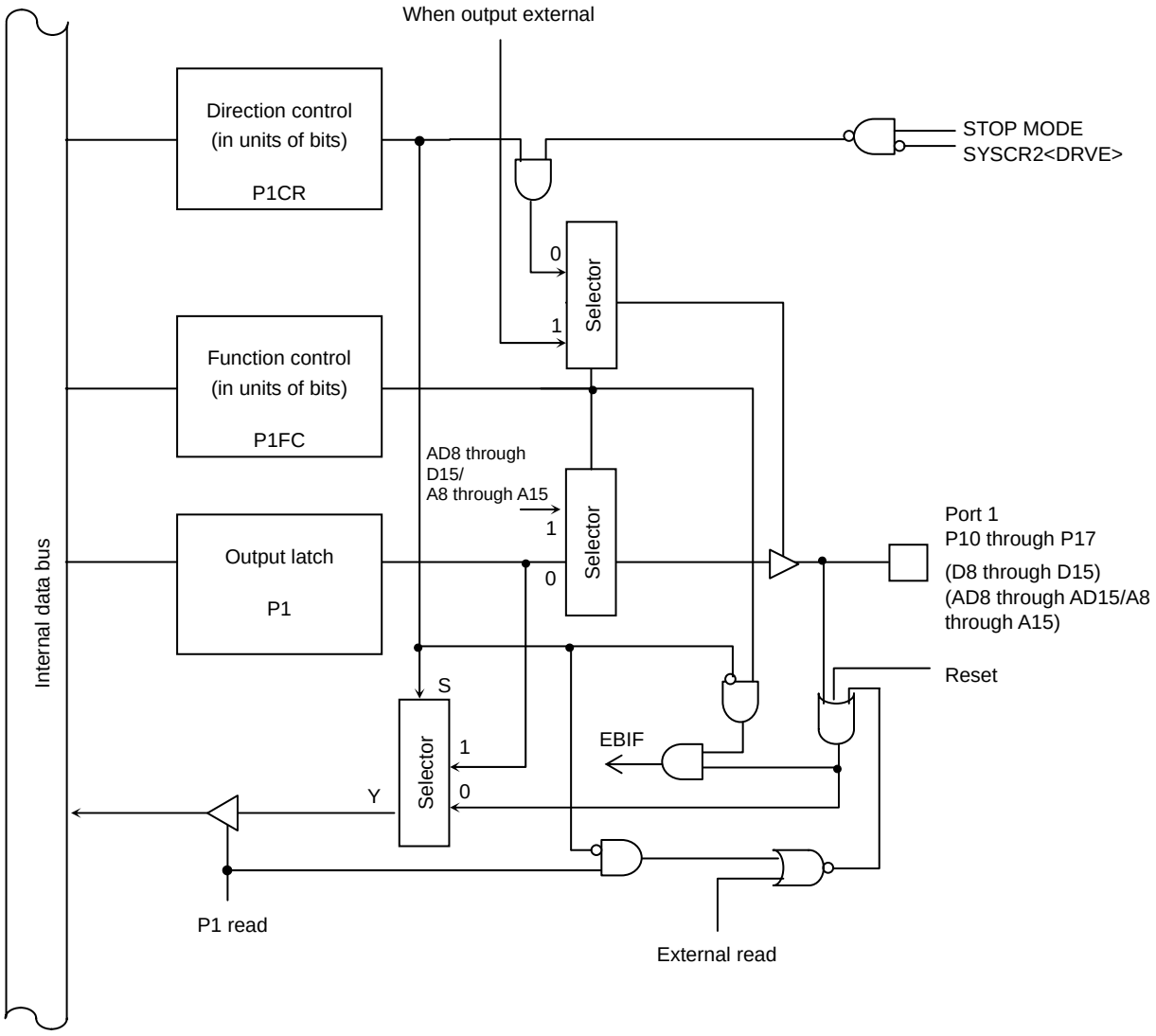


Fig. 7.2.1 Port 1 (P10 through P17)

Port 1 register

		7	6	5	4	3	2	1	0
P1	Bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
(0xFFFF_F001)	Read/Write	R/W							
	After reset	Input mode (output latch register is cleared to "0.")							

Port 1 control register

		7	6	5	4	3	2	1	0
P1CR	Bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
(0xFFFF_F004)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	<< See P1FC >>							

Port 1 function register

		7	6	5	4	3	2	1	0
P1FC	Bit Symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
(0xFFFF_F005)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	P1FC/P1CR = 00: Input, 01: Output, 10: D15 through 8 or AD15 through 8, 11: A15 through 8							

Function		**Corresponding BIT of P1FC**	**Corresponding BIT of P1CR**	**PORT to be used**
POR1 input setting		0	0	PORT1
POR1 output setting		0	1	PORT1
Separate bus mode (BUSMD="0")	Data bus (D15 through D8) input/output setting	1	0	PORT1
	Address bus (A15 through A8) output setting	1	1	
Multiplexed bus mode (BUSMD="1")	Address data bus (AD15 through AD8) input/output setting	1	0	PORT1
	Address bus (A15 through A8) output setting	1	1	

Fig. 7.2.2 Port 1 Registers

7.3 Port 2 (P20 through P27)

The port 2 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register P2CR and the function register P2FC. A reset allows all bits of the output latch P2 to be set to "1," all bits of P2CR and P2FC to be cleared to "0," and the port 2 to be put in input mode.

Besides the general-purpose input/output port function, the port 2 performs another function: A0 through A7 function as one address bus and A16 through A23 function as the other address bus. To access external memory, registers P2CR and P2FC must be provisioned to allow the port 2 to function as an address bus.

If the BUSMD pin is set to "L" level during a reset, the port 2 is put in separate mode (A16 to A23). If it is set to "H" level during a reset, the port 2 is put in multiplexed mode (A0 through A7 or A16 through A23).

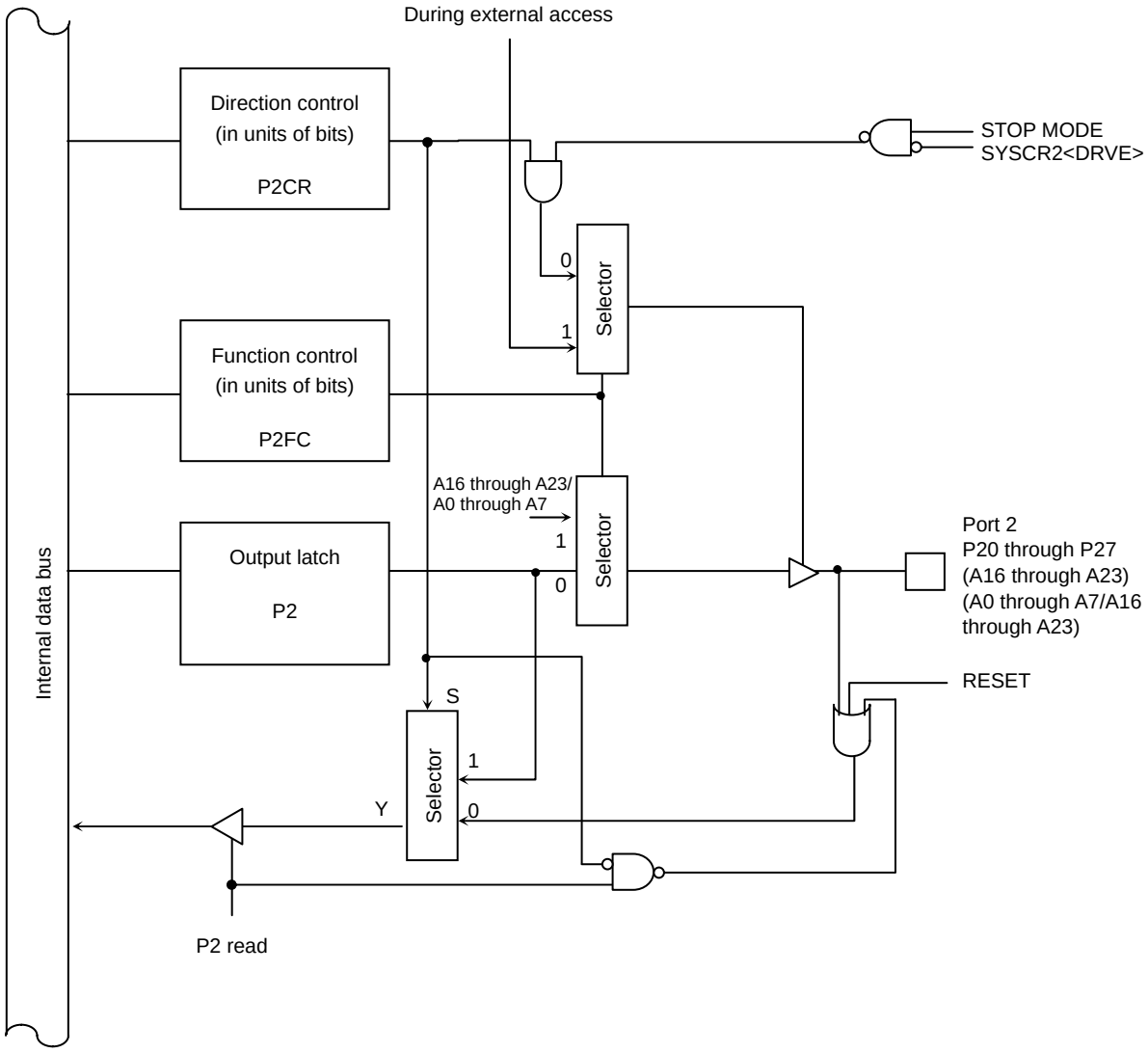


Fig. 7.3.1 Port 2 (P20 through P27)

Port 2 register

		7	6	5	4	3	2	1	0
P2	Bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
(0xFFFF_F012)	Read/Write	R/W							
	After reset	Input mode (output latch register is cleared to "1.")							

Port 2 control register

		7	6	5	4	3	2	1	0
P2CR	Bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
(0xFFFF_F014)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	<<See P2FC>>							

Port 2 function register

		7	6	5	4	3	2	1	0
P2FC	Bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
(0xFFFF_F015)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	P2FC/P2CR = 00: Input, 01: Output, 10: A7 through 0, 11: A23 through 16							

Function	Corresponding BIT of P2FC	Corresponding BIT of P2CR	PORT to be used
POR2 input setting	0	0	PORT2
POR2 output setting	0	1	PORT2
Address bus (A7 through A0) output setting (*1)	1	0	PORT2
Address bus (A23 through A16) output setting (*1)	1	1	PORT2

(*1) The same address bus (A7 through A0/A23 through A16) output settings are used in both the separate bus mode and the multiplexed bus mode (BUSMD="0," "1").

Fig. 7.3.2 Port 2 Registers

7.4 Port 3 (P30 through P37)

The port 3 is a general-purpose, 8-bit input/output port (P30 and P31 are used exclusively for output). For this port, inputs and outputs can be specified in units of bits by using the control register P3CR and the function register P3FC.

A reset allows the output latches P30 and 31 to be set to "1." If the BUSMD pin is at the "L" level when a reset is performed, P37 goes into separate bus mode, and the output latch is set to "1." If the BUSMD pin is at the "H" level when a reset is performed, P37 goes into multiplexed bus mode, and the output latch is cleared to "0." Bit 2 through bit 6 of P3CR (bits 0 and 1 are unused) are cleared to "0." Bit 7 of P3CR is cleared to "0" in separate bus mode and set to "1" in multiplexed bus mode. All bits of P3FC are cleared to "0," P30 and P31 generate "H," and P32 through P36 go into the input mode with a pull-up resistor after RESET is cleared. If the port 3 goes into separate bus mode, P37 is put into input mode. If the port 3 goes into multiplexed bus mode, P37 is put into output mode.

Besides the general-purpose input/output port function, the port 3 inputs and outputs CPU control/status signals. If the P30 pin is set to $\overline{RD}$ signal output mode (<P30F>="1"), the $\overline{RD}$ strobe is output only when an external address area is accessed. Likewise, if the P31 pin is set to $\overline{WR}$ signal output mode (<P31F>="1"), the $\overline{WR}$ strobe is output only when an external address area is accessed.

As for P32 and P36, when <P3xFC> ="1," and $\overline{BUSAK}$ ="0," Pull-up is enabled.

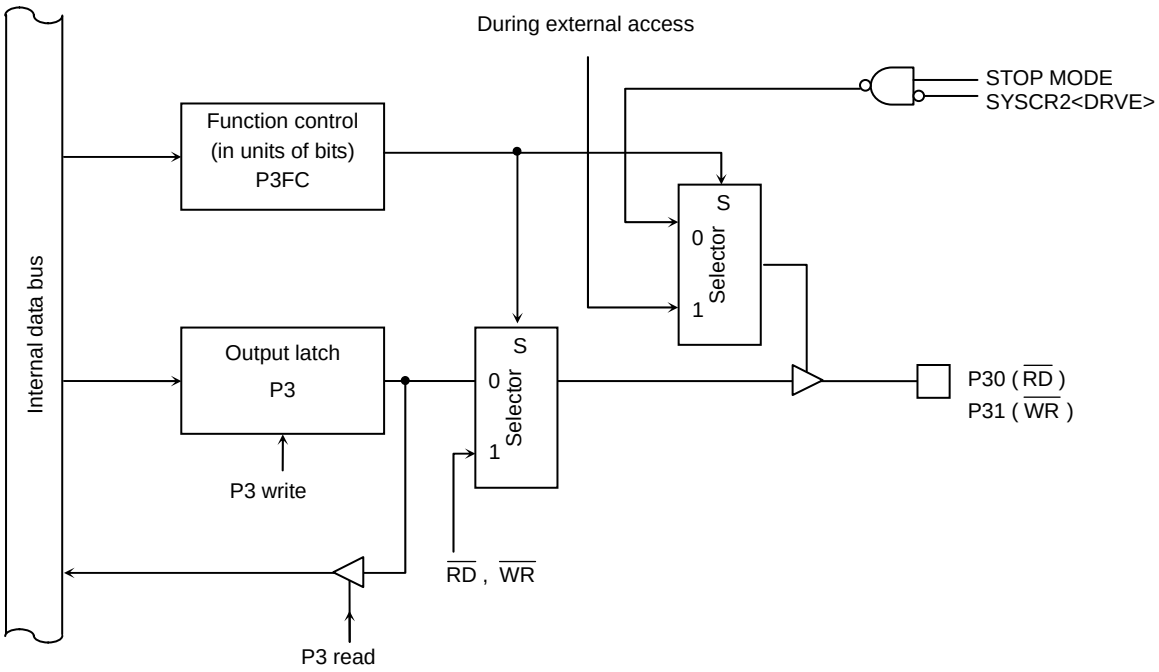


Fig. 7.4.1 Port 3 (P30, P31)

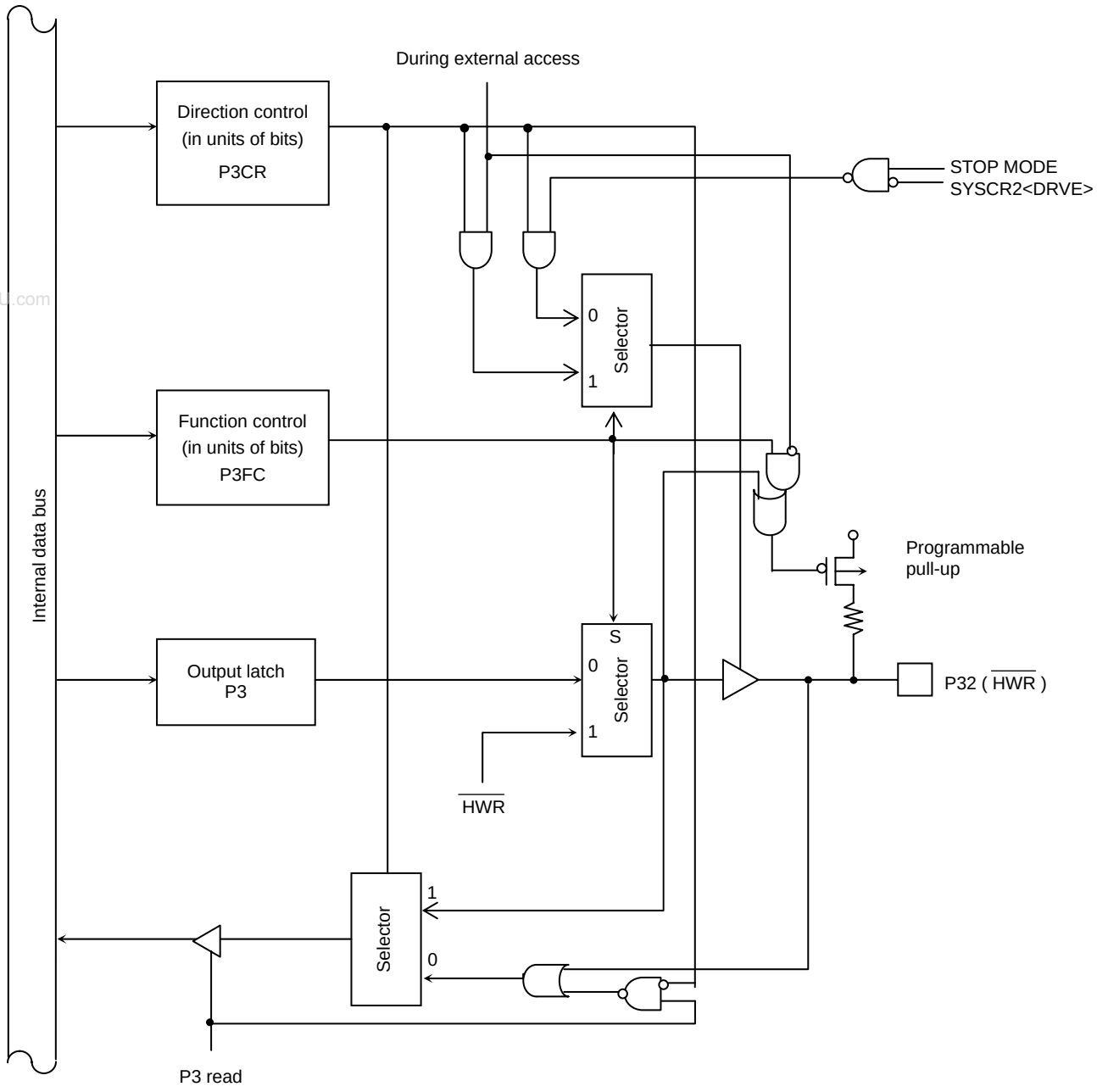


Fig. 7.4.2 Port 3 (P32)

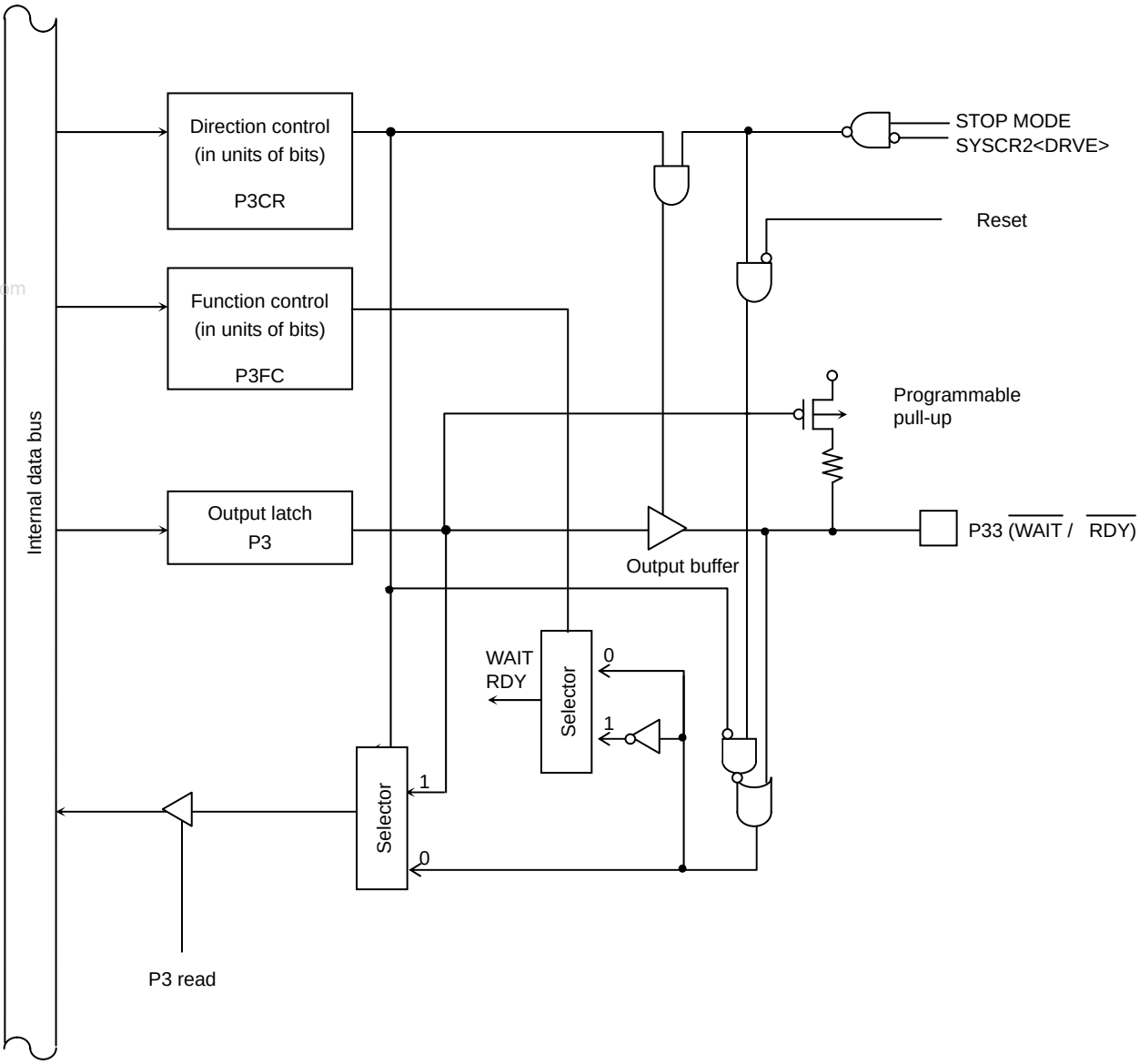


Fig. 7.4.3 Port 3 (P33)

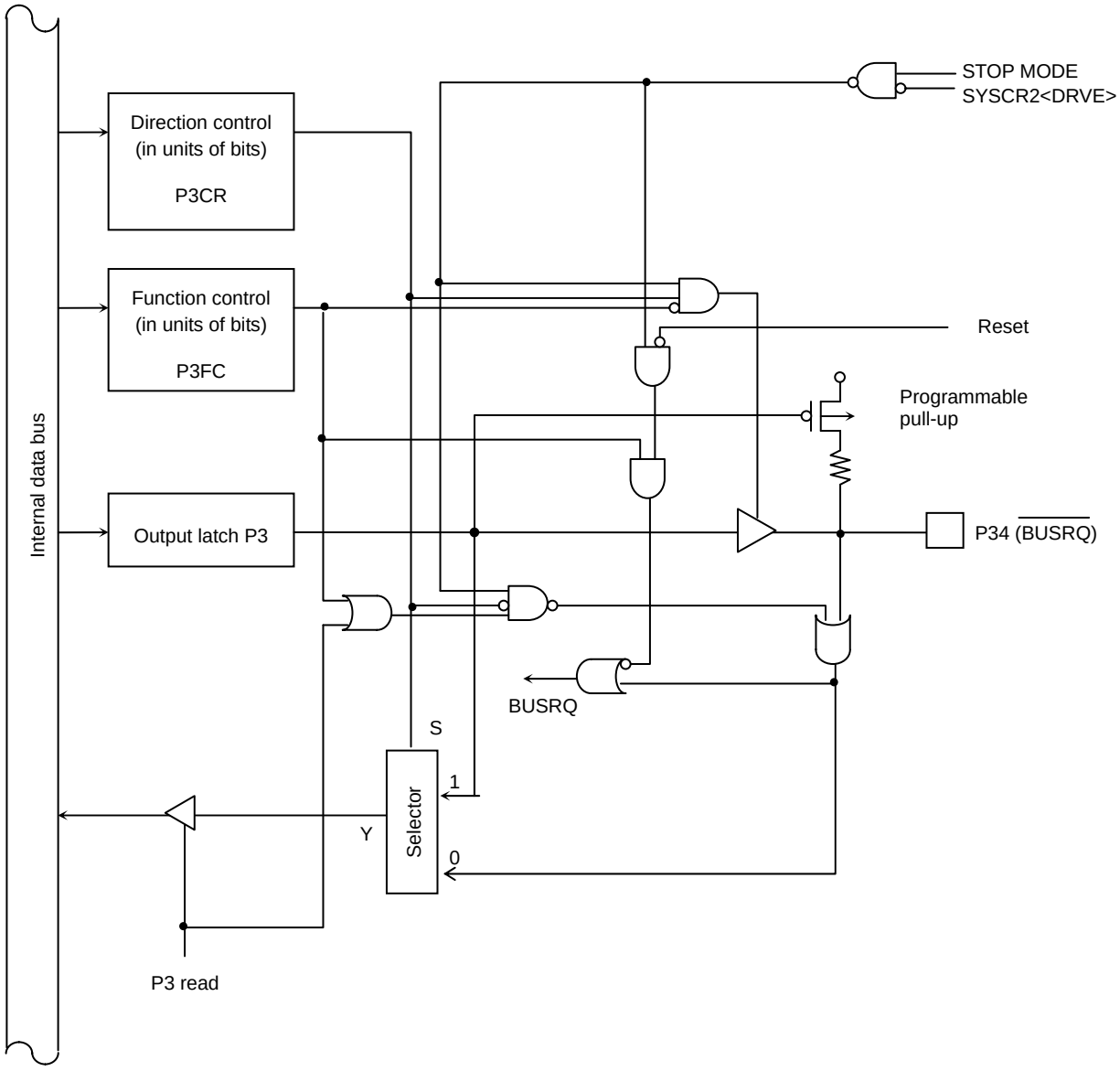


Fig. 7.4.4 Port 3 (P34)

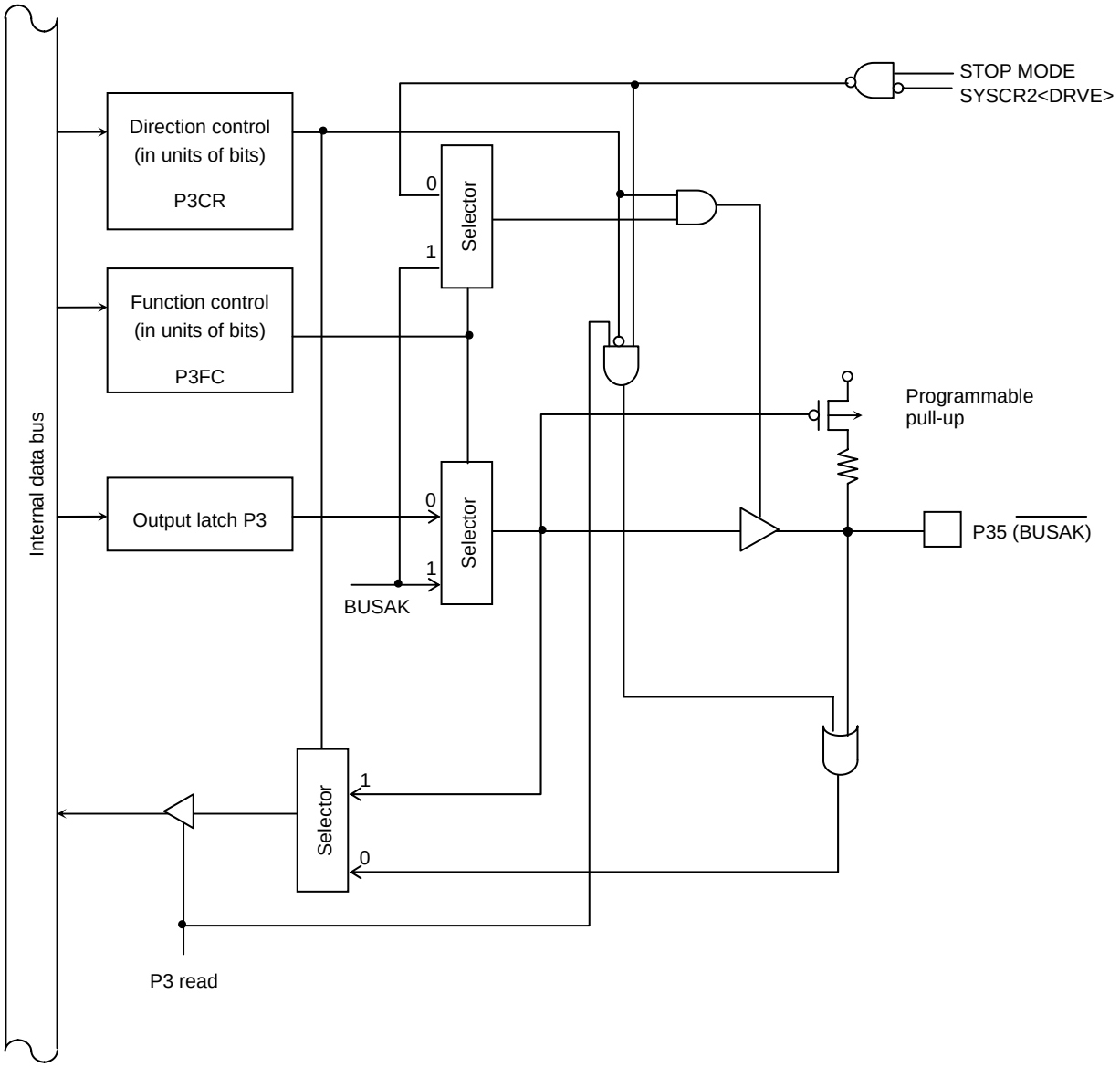


Fig. 7.4.5 Port 3 (P35)

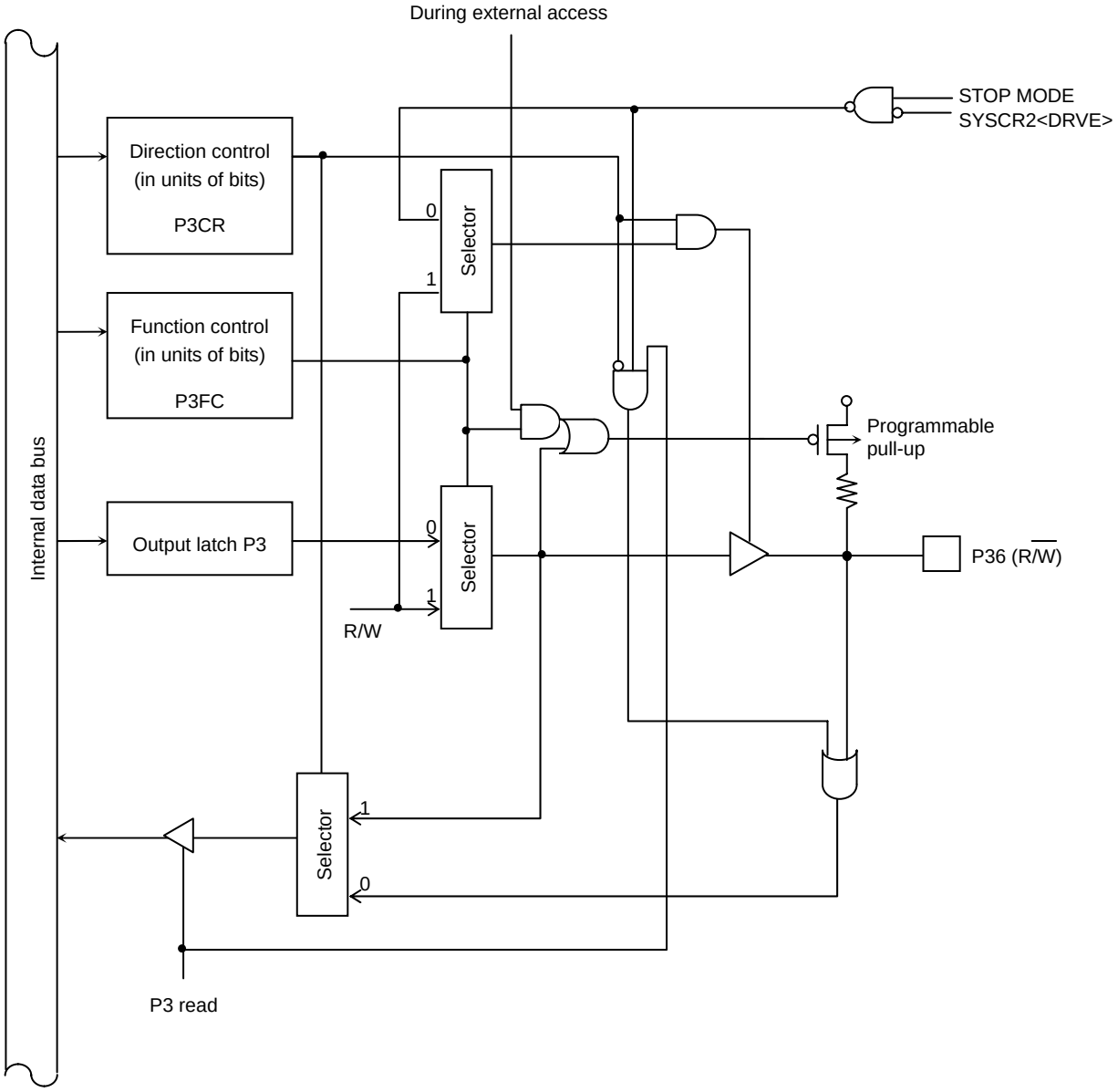


Fig. 7.4.6 Port 3 (P36)

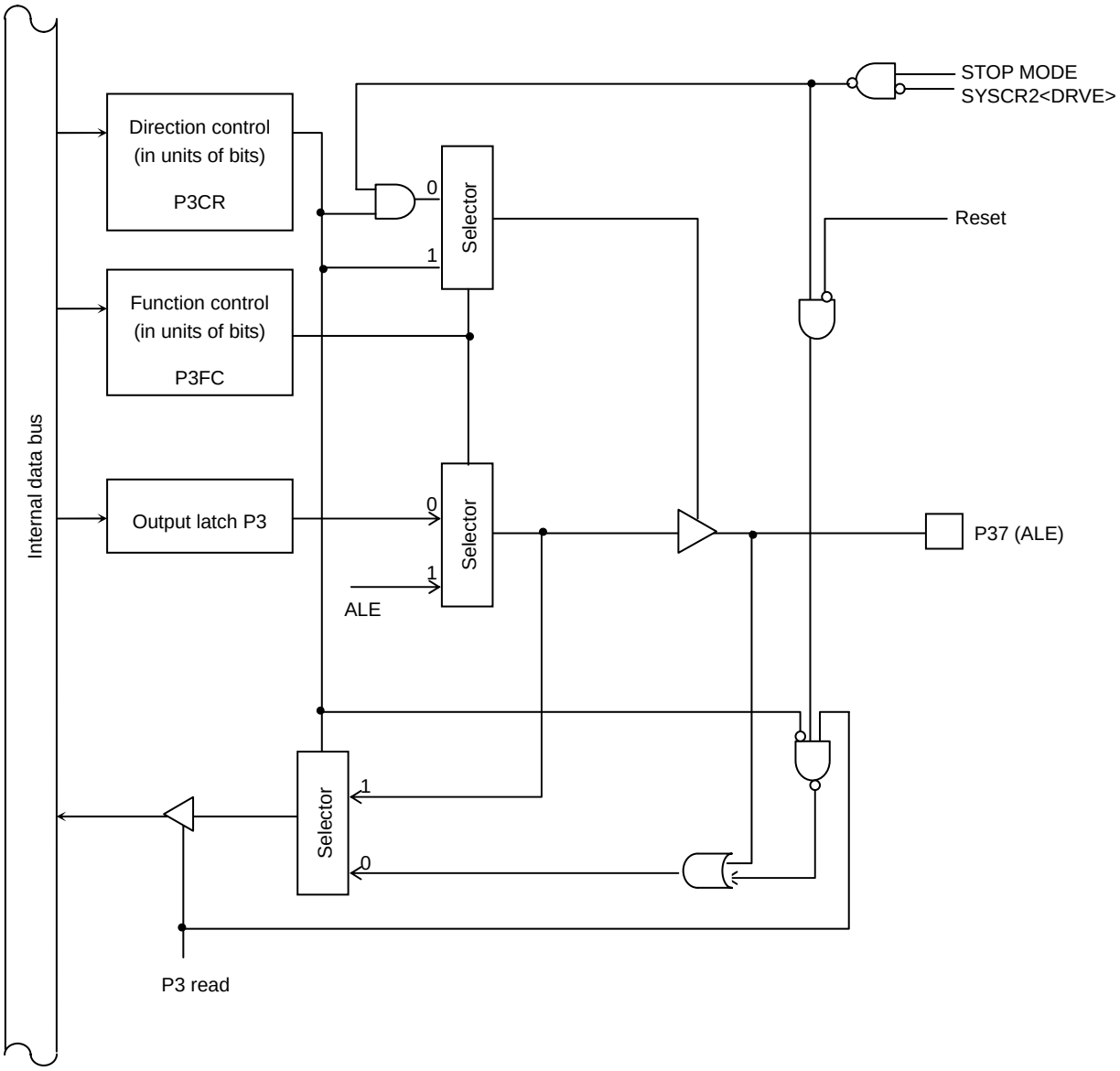


Fig. 7.4.7 Port 3 (P37)

Port 3 register

P3 (0xFFFF_F018)

	7	6	5	4	3	2	1	0
Bit Symbol	P37	P36	P35	P34	P33	P32	P31	P30
Read/Write	R/W							
After reset	To be determined according to the bus mode (*1)	Output mode						
		1	1	1	1	1	1	1

Port 3 control register

P3CR (0xFFFF_F01A)

	7	6	5	4	3	2	1	0
Bit Symbol	P37C	P36C	P35C	P34C	P33C	P32C	—	—
Read/Write	R/W						R	
After reset	To be determined according to the bus mode (*1)	0	0	0	0	0	0	0
Function		0: Input 1: Output					Output	

Port 3 function register

P3FC (0xFFFF_F01B)

	7	6	5	4	3	2	1	0
Bit Symbol	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: ALE	0: PORT 1: $\overline{R/W}$	0: PORT 1: $\overline{BUSAK}$	0: PORT 1: $\overline{BUSRQ}$	0: PORT/ $\overline{WAIT}$ 1: PORT/ $\overline{RDY}$	0: PORT 1: $\overline{HWR}$	0: PORT 1: $\overline{WR}$	0: PORT 1: $\overline{RD}$

Function	Corresponding BIT of P3FC	Corresponding BIT of P3CR	PORT to be used
RD output setting	1(*2)	–	P30
WR output setting	1(*2)	–	P31
HWR output setting	1	1	P32
WAIT input setting	0	0	P33
RDY input setting	1	0	
BUSRQ input setting	1	0	P34
BUSAK output setting	1	1	P35
R/W output setting	1	1	P36
ALE output setting (BUSMD = "1")	1(*1)	1	P37

(*1) In separate bus mode (BUSMD="0"), ALE is not output. The port 3 functions as an input/output port based on the bit setting of the control register P3CR<P37C>. After a reset, the port becomes an input port. If a reset is executed in multiplexed bus mode (BUSMD="1"), the port 3 becomes an output port at "L" level.

(*2) /RD and /WR are output only when an external area is being accessed.

Fig. 7.4.6 Port 3 Registers

7.5 Port 4 (P40 through P47)

The port 4 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register P4CR and the function register P4FC. A reset allows all bits of the output latch P4 to be set to "1" and all bits of P4CR to be reset to "0." Bits of P40FC through P46FC are reset to "0." P40 through P45 goes into the input mode with a pull-up resistor, and P46 and P47 are put into input mode.

Besides the general-purpose input/output port function, the ports 40 through 45 outputs chip select signals ($\overline{CS0}$ through $\overline{CS5}$), and the port 46 functions as a SCOUT output pin for outputting external clocks.

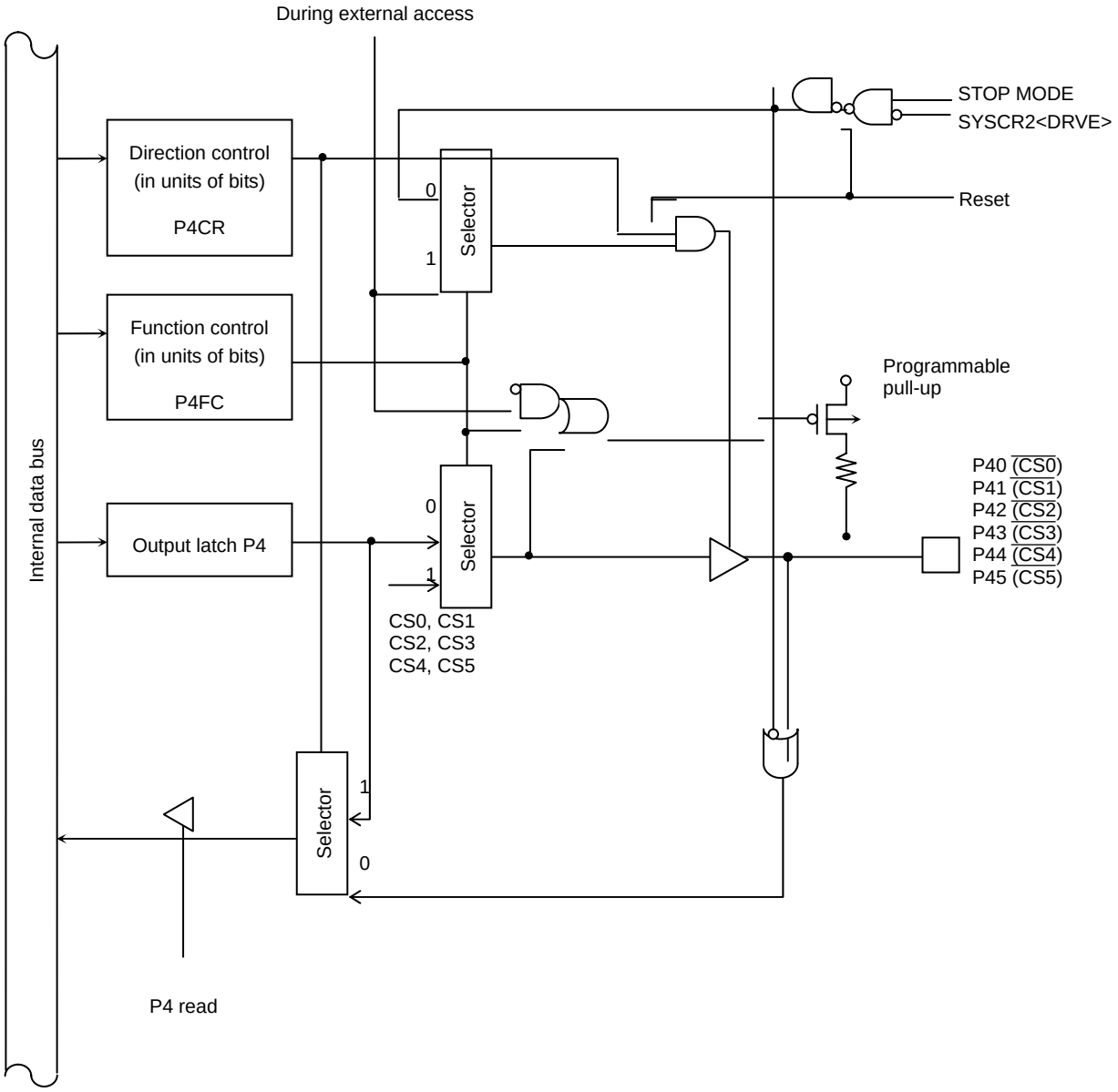


Fig. 7.5.1 Port 4 (P40 to P45)

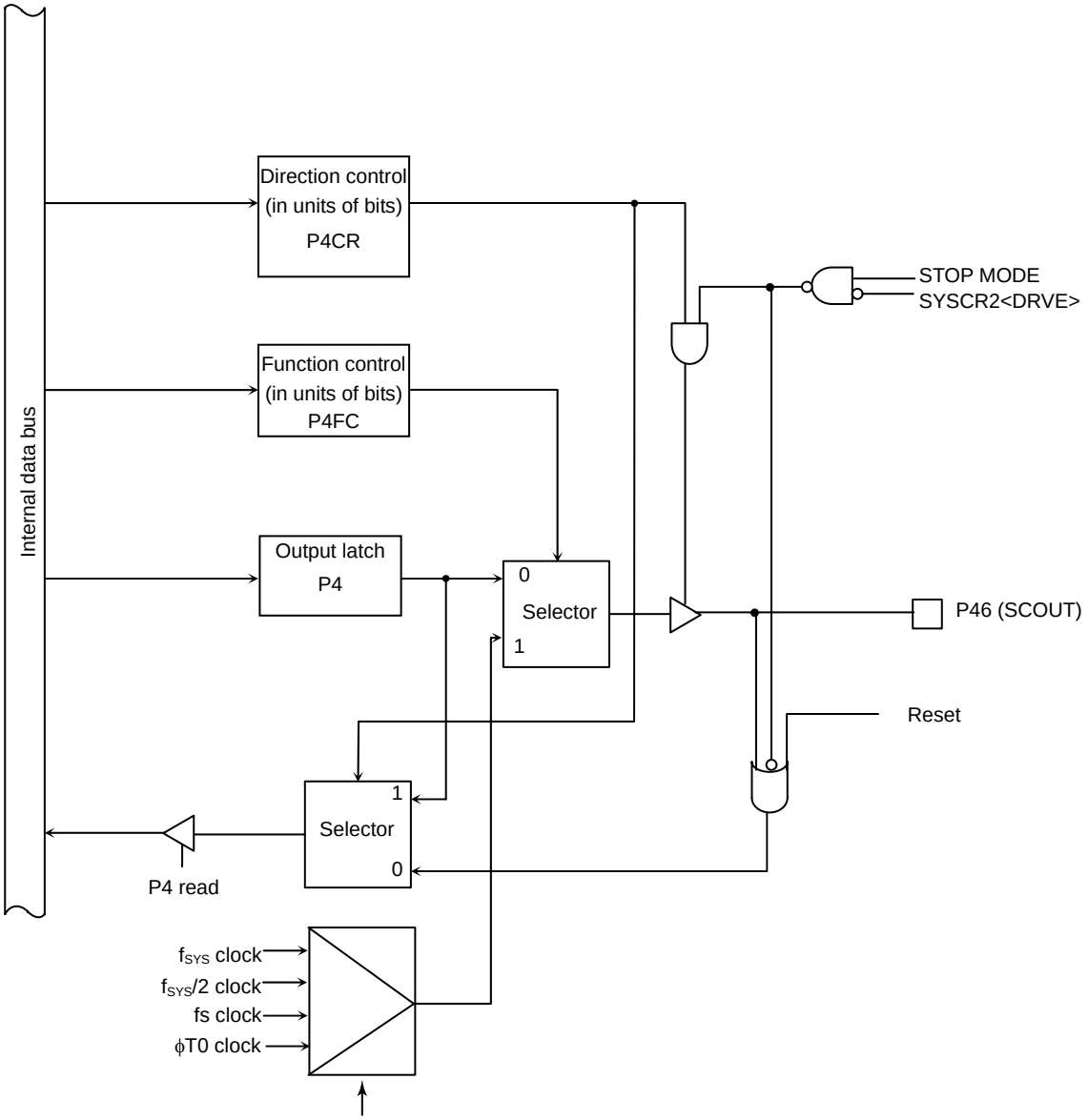


Fig. 7.5.2 Port 4 (P46)

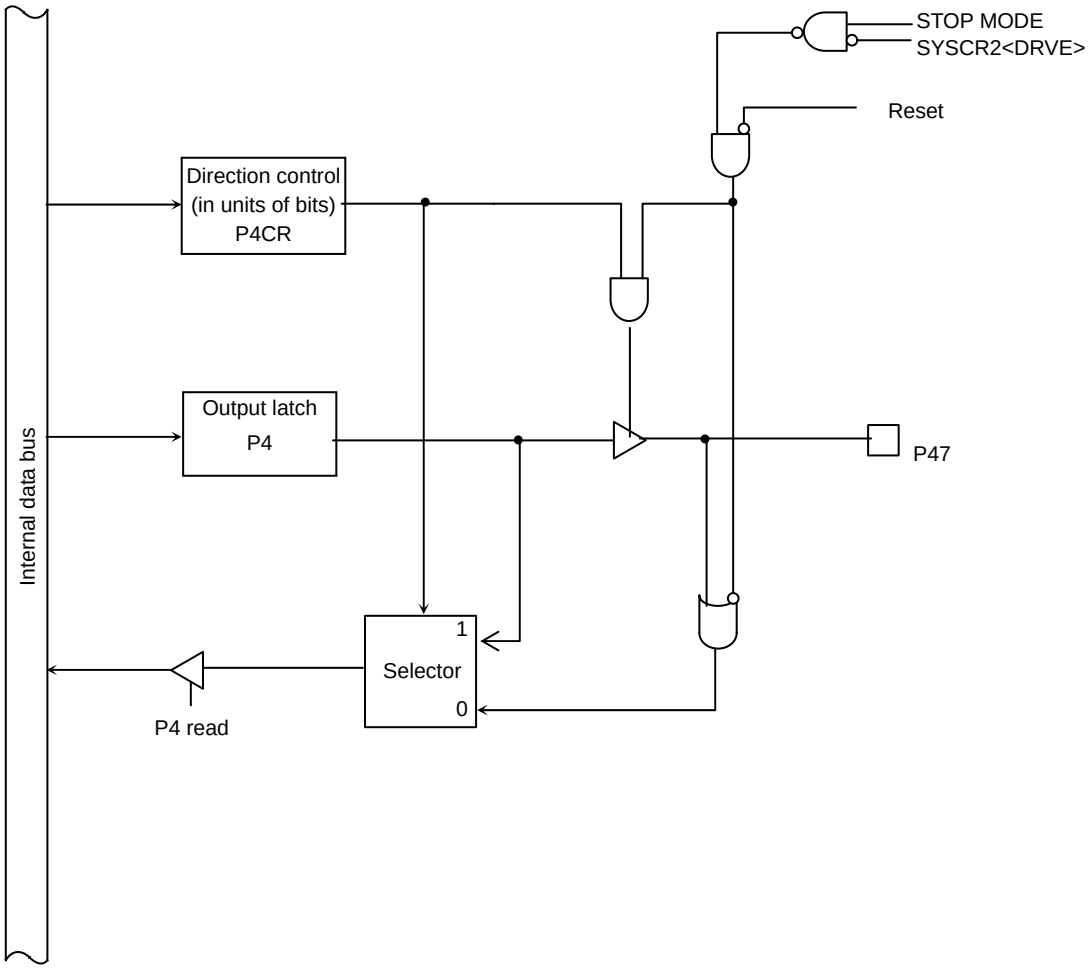


Fig. 7.5.3 Port 4 (P47)

Port 4 register

P4 (0xFFFF_F01E)

	7	6	5	4	3	2	1	0
Bit Symbol	P47	P46	P45	P44	P43	P42	P41	P40
Read/Write	R/W							
After reset	Input mode							
	1	1	1 (Pull-Up)	1 (Pull-Up)	1 (Pull-Up)	1 (Pull-Up)	1 (Pull-Up)	1 (Pull-Up)

Port 4 control register

P4CR (0xFFFF_F020)

	7	6	5	4	3	2	1	0
Bit Symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	0: Input 1: Output							

Port 4 function register

P4FC (0xFFFF_F021)

	7	6	5	4	3	2	1	0
Bit Symbol	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	0: PORT	0: PORT 1: SCOUT	0: PORT 1: $\overline{CS5}$	0: PORT 1: $\overline{CS4}$	0: PORT 1: $\overline{CS3}$	0: PORT 1: $\overline{CS2}$	0: PORT 1: $\overline{CS1}$	0: PORT 1: $\overline{CS0}$

Function	Corresponding BIT of P4FC	Corresponding BIT of P4CR	PORT to be used
CS0 output setting	1	1	P40
CS1 output setting	1	1	P41
CS2 output setting	1	1	P42
CS3 output setting	1	1	P43
CS4 output setting	1	1	P44
CS5 output setting	1	1	P45
SCOUT output setting	1	1	P46

Fig. 7.5.4 Port 4 Registers

7.6 Port 5 (P50 through P57)

The port 5 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register P5CR and the function register P5FC. A reset allows all bits of the output latch P5 to be set to "1," all bits of P5CR and P5FC to be cleared to "0," and the port 5 to be put in input mode.

The port 5 also functions as an address bus (A0 through A7). To access external memory, P5CR and P5FC must be provisioned to allow the port 5 to function as an address bus. This address bus function can be used only in separate bus mode. (To put the port 5 in separate bus mode, the BUSMD pin must be set to "L" level during a reset.)

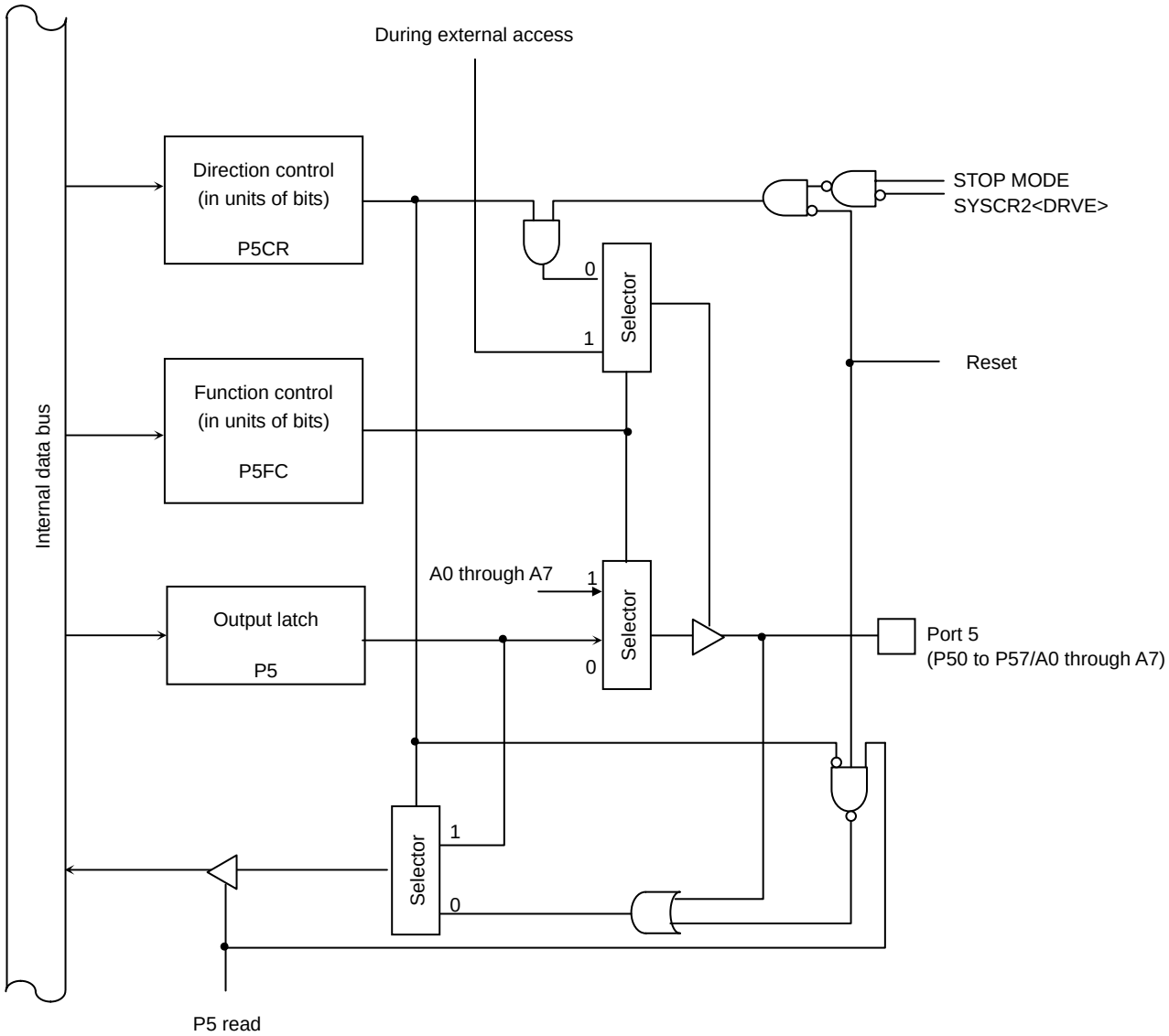


Fig. 7.6.1 Port 5 (P50 to P57)

Port 5 register

P5 (0xFFFF_F028)	7	6	5	4	3	2	1	0
Bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port 5 control register

P5CR (0xFFFF_F02C)	7	6	5	4	3	2	1	0
Bit Symbol	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port 5 function register

P5FC (0xFFFF_F02D)	7	6	5	4	3	2	1	0
Bit Symbol	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: A7	0: PORT 1: A6	0: PORT 1: A5	0: PORT 1: A4	0: PORT 1: A3	0: PORT 1: A2	0: PORT 1: A1	0: PORT 1: A0

Function	Corresponding BIT of P5FC	Corresponding BIT of P5CR	PORT to be used
POR5 input setting	0	0	PORT5
POR5 output setting	0	1	PORT5
Address bus (A7 to A0) output setting (*1)	1	1	PORT5

(*1) The same address bus (A7 through A0) output setting is used in both the separate bus mode and multiplexed bus mode (BUSMD="0," "1").

Fig. 7.6.2 Port 5 Registers

7.7 Port 6 (P60 through P67)

The port 6 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register P6CR and the function register P6FC. A reset allows all bits of the output latch P6 to be set to "1," all bits of P6CR and P6FC to be cleared to "0," and the port 6 to be put in input mode. Besides the input/output port function, the port 6 performs other functions: P60 and P63 output SIO data, P61 and P64 input SIO data, P62 and P65 input and output SIO CLK or input CTS, P61 and P64 input external interrupts, and P66 and P67 output a 16-bit timer.

The port 6 also functions as an address bus (A8 through A15). To access external memory, P6CR and P6FC must be provisioned to allow the port 6 to function as an address bus. The address bus function can be used only in separate bus mode. (To put the port 6 in separate bus mode, the BUSMD pin must be set to "L" level during a reset.)

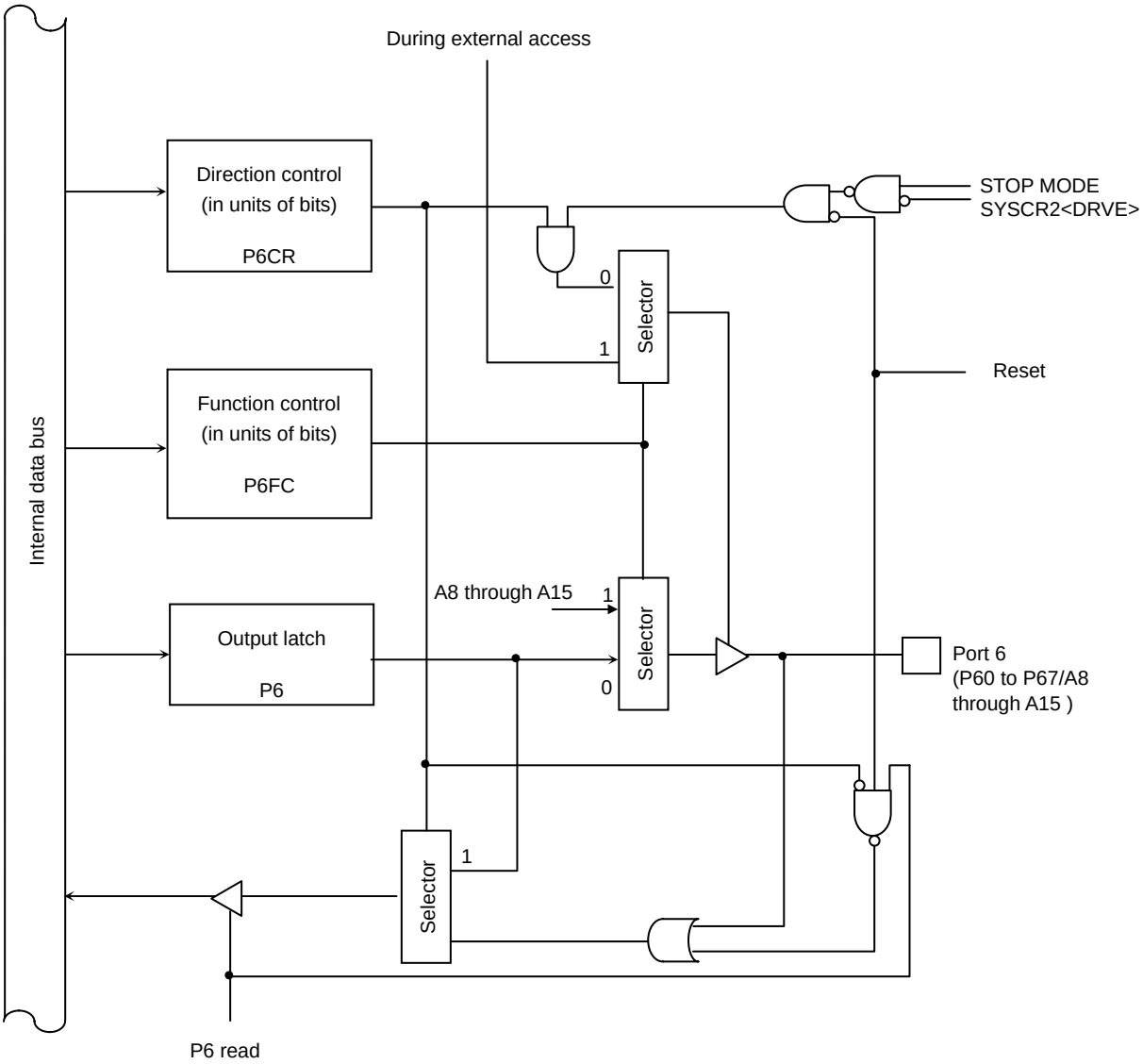


Fig. 7.7.1 Port 6 (P60 through P67)

Port 6 register

P6 (0xFFFF_F029)		7	6	5	4	3	2	1	0
	Bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port 6 control register

P6CR (0xFFFF_F02E)		7	6	5	4	3	2	1	0
	Bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 6 function register

P6FC (0xFFFF_F02F)		7	6	5	4	3	2	1	0
	Bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: A15	0: PORT 1: A14	0: PORT 1: A13	0: PORT 1: A12	0: PORT 1: A11	0: PORT 1: A10	0: PORT 1: A9	0: PORT 1: A8

Function	Corresponding BIT of P6F	Corresponding BIT of P6CR	PORT to be used
POR6 input setting	0	0	PORT6
POR6 output setting	0	1	PORT6
Address bus (A15 to A8) output setting (*1)	1	1	PORT6

(*1) The same address bus (A15 through A8) output setting is used in both the separate bus mode and multiplexed bus mode (BUSMD="0," "1").

Fig. 7.7.2 Port 6 Registers

7.8 Port 7 (P70 through P77), Port 8 (P80 through P87) and Port 9 (P90 through P97)

The ports 7, 8 and 9 are 8-bit ports and used exclusively for input. They are also used as analog input ports for the A/D converter. Inputs can be specified by using the function register PnFC. A reset allows all bits of PnFC to be cleared to "0" and the ports 7, 8 and 9 to be put in input mode.

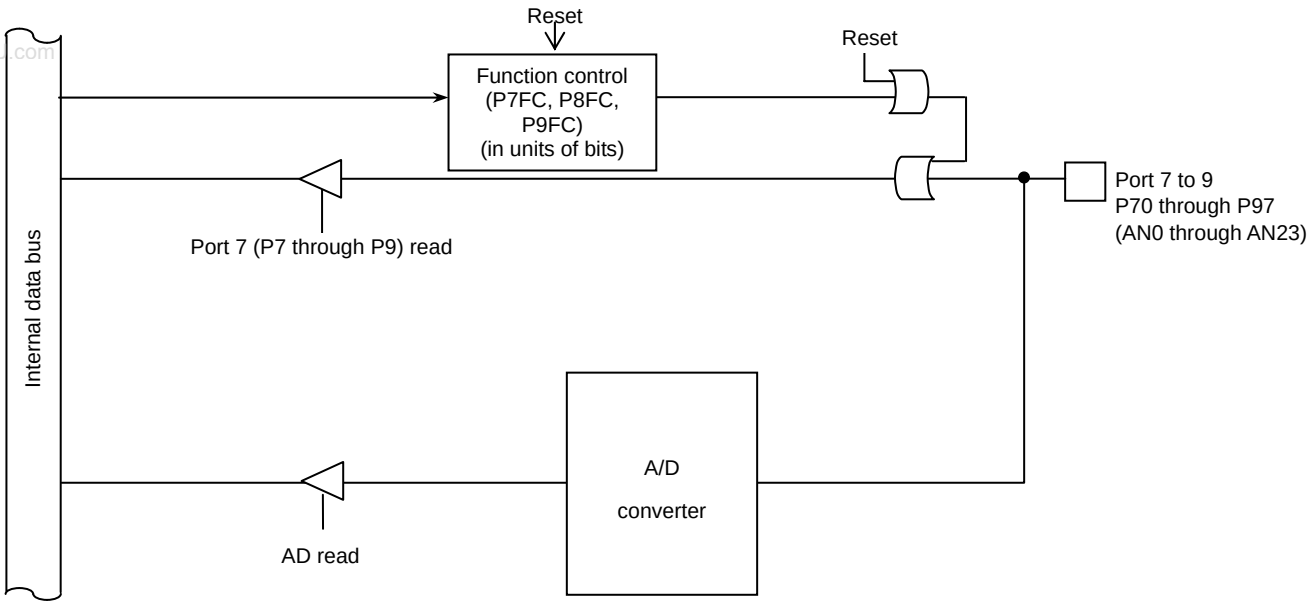


Fig. 7.8.1 Port 7 to 9 (P70 through P77, P80 through P87 and P90 through P97)

Port 7 register

P7 (0xFFFF_F040)

	7	6	5	4	3	2	1	0
Bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
Read/Write	R							
After reset	Input mode							

Port 7 function register

P7FC (0xFFFF_F048)

	7	6	5	4	3	2	1	0
Bit Symbol	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: AN7	0: PORT 1: AN6	0: PORT 1: AN5	0: PORT 1: AN4	0: PORT 1: AN3	0: PORT 1: AN2	0: PORT 1: AN1	0: PORT 1: AN0

Port 8 register

P8 (0xFFFF_F041)

	7	6	5	4	3	2	1	0
Bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R							
After reset	Input mode							

Port 8 function register

P8FC (0xFFFF_F049)

	7	6	5	4	3	2	1	0
Bit Symbol	P87F	P86F	P85F	P84F	P83F	P82F	P81F	P80F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: AN15	0: PORT 1: AN14	0: PORT 1: AN13	0: PORT 1: AN12	0: PORT 1: AN11	0: PORT 1: AN10	0: PORT 1: AN9	0: PORT 1: AN8

Port 9 register

P9 (0xFFFF_F042)

	7	6	5	4	3	2	1	0
Bit Symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R							
After reset	Input mode							

Port 9 function register

P9FC (0xFFFF_F04A)

	7	6	5	4	3	2	1	0
Bit Symbol	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: AN23	0: PORT 1: AN22	0: PORT 1: AN21	0: PORT 1: AN20	0: PORT 1: AN19	0: PORT 1: AN18	0: PORT 1: AN17	0: PORT 1: AN16

Function	**Corresponding bits of P7FC, P8FC and P9FC**
Input setting for the ports 7, 8 and 9	0
Input setting for AN23 through AN0	1

Fig. 7.8.2 Registers of the Ports 7, 8 and 9

7.9 Port A (PA0 through PA7)

The port A is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PACR. A reset allows PACR to be reset to "0" and the port A to function as an input port. Besides the input/output port function, the port A performs other functions: PA2, PA5, PA6 and PA7 output a 16-bit timer, and PA0, PA1, PA3 and PA4 input a 16-bit timer and external interrupts. These functions are enabled by setting corresponding bits of PAFC to "1." A reset allows PACR and PAFC to be cleared to "0" and the port A to be put in input mode.

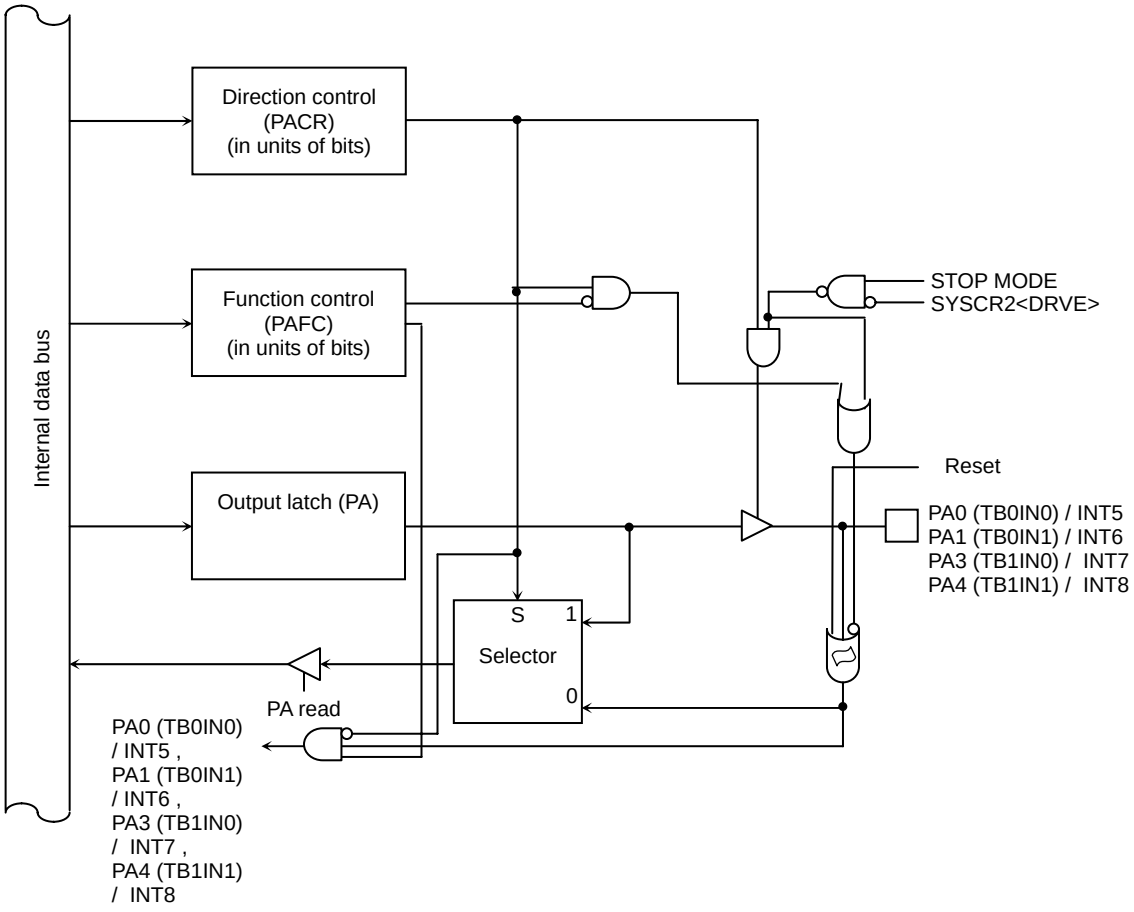


Fig. 7.9.1 Port A (PA0, PA1, PA3, PA4)

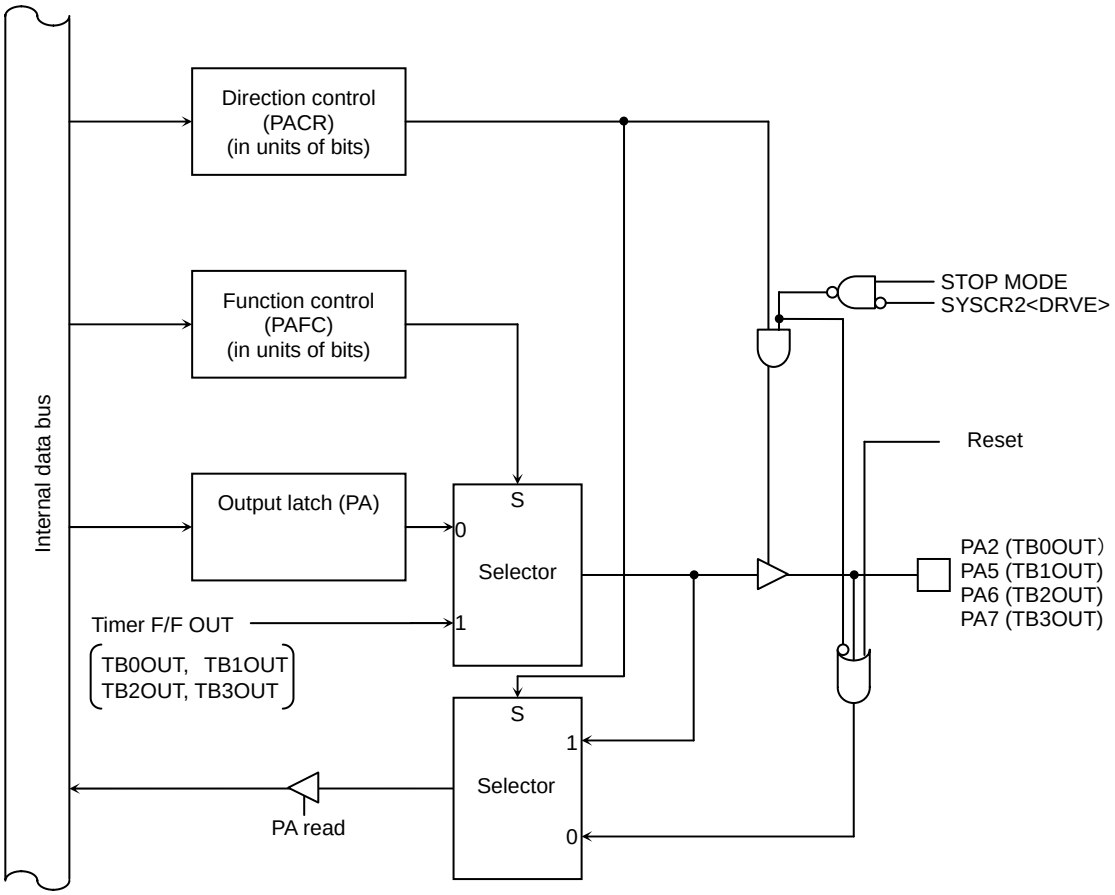


Fig. 7.9.2 Port A (PA2, PA5, PA6, PA7)

Port A register

PA (0xFFFF_F043)	7	6	5	4	3	2	1	0
Bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port A control register

PACR (0xFFFF_F047)	7	6	5	4	3	2	1	0
Bit Symbol	PA7C	PA6C	PA5C	PA4C	PA3C	PA2C	PA1C	PA0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port A function register

PAFC (0xFFFF_F04B)	7	6	5	4	3	2	1	0
Bit Symbol	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: TB3OUT	0: PORT 1: TB2OUT	0: PORT 1: TB1OUT	0: PORT 1: TB1IN1 / INT8	0: PORT 1: TB1IN0 / INT7	0: PORT 1: TB0OUT	0: PORT 1: TB0IN1 / INT6	0: PORT 1: TB0IN0 / INT5

Function	Corresponding BIT of PAFC	Corresponding BIT of PACR	PORT to be used
TB0IN0 input setting	1	0	PA0
INT5 input setting	1(*1)	0	
TB0IN1 input setting	1	0	PA1
INT6 input setting	1(*1)	0	
TB0OUT output setting	1	1	PA2
TB1IN0 input setting	1	0	PA3
INT7 input setting	1(*1)	0	
TB1IN1 input setting	1	0	PA4
INT8 input setting	1(*1)	0	
TB1OUT output setting	1	1	PA5
TB2OUT output setting	1	1	PA6
TB3OUT output setting	1	1	PA7

(*1) This bit setting is used only if an interrupt must be generated to clear the STOP status and if SYSCR<DRVE> is set to 0. In all other cases, this bit setting does not need to be used.

(Note) If two input functions in addition to the PORT function are assigned to one pin, which input function to be used shall be designated by making proper enable/disable settings provided in each function block.

Fig. 7.9.3 Port A Registers

7.10 Port B (PB0 through PB7)

Port B is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PBCR. A reset allows PBCR to be reset to "0" and the port B to function as an input port. Besides the input/output port function, the port B performs other functions: PB0 through PB5 output a 16-bit timer, and PB6 and PB7 input a 16-bit timer. These functions are enabled by setting corresponding bits of PBFC to "1." A rest allows PBCR and PBFC to be cleared to "0" and the port B to function as an input port.

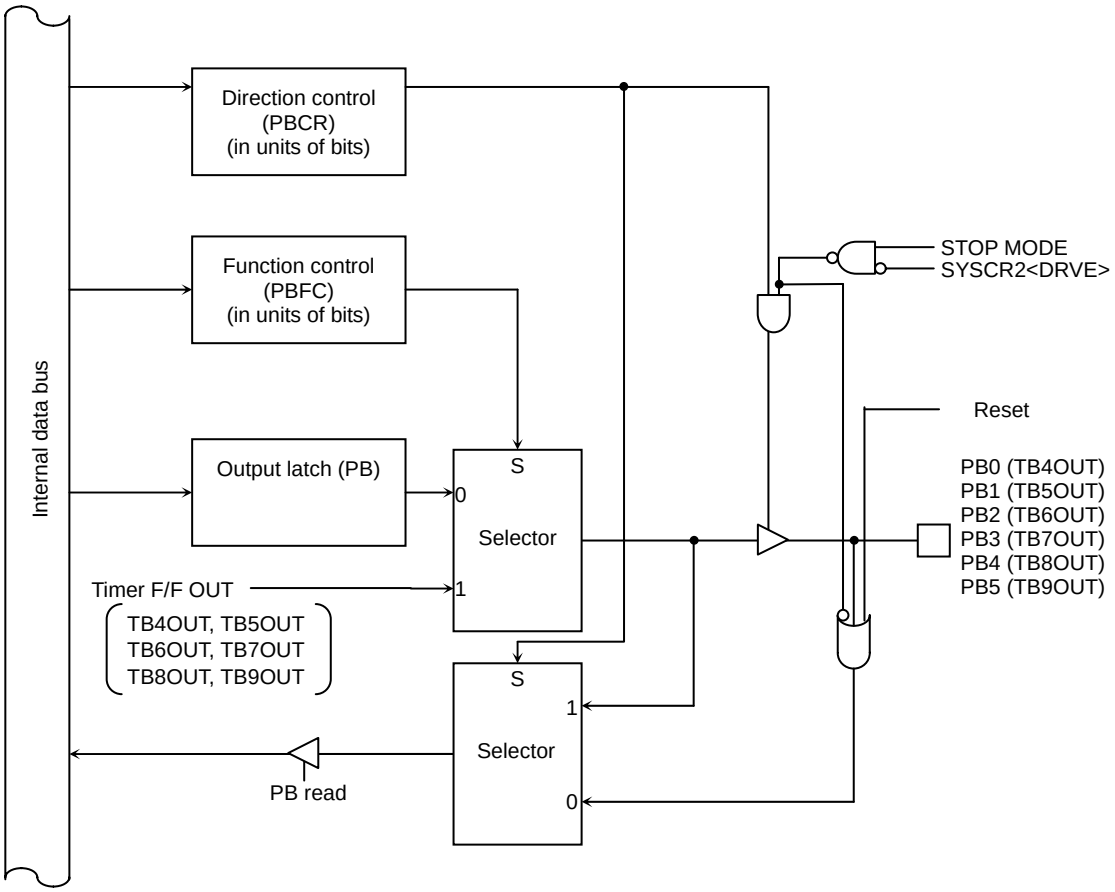


Fig. 7.10.1 Port B (PB0 through PB5)

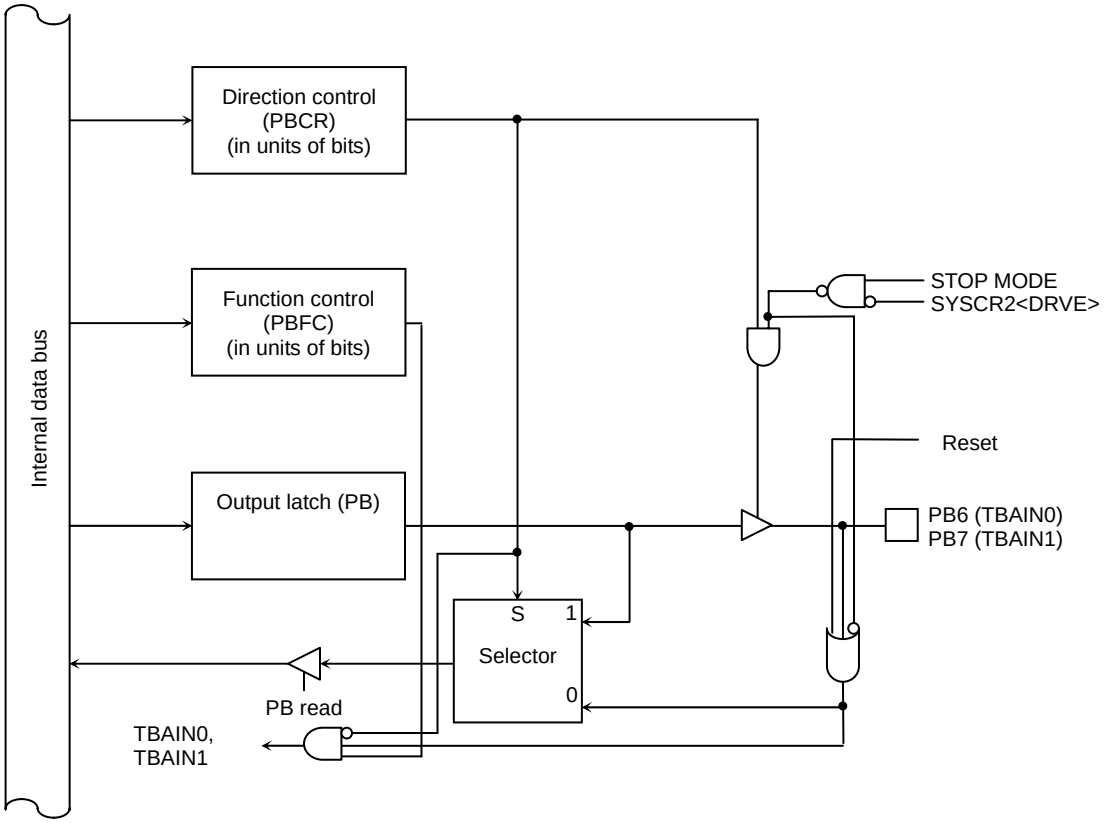


Fig. 7.10.2 Port B (PB6, PB7)

Port B register

PB (0xFFFF_F050)	7	6	5	4	3	2	1	0
Bit Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port B control register

PBCR (0xFFFF_F054)	7	6	5	4	3	2	1	0
Bit Symbol	PB7C	PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port B function register

PBFC (0xFFFF_F058)	7	6	5	4	3	2	1	0
Bit Symbol	PB7F	PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	PB0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: TBAIN1	0: PORT 1: TBAIN0	0: PORT 1: TB9OUT	0: PORT 1: TB8OUT	0: PORT 1: TB7OUT	0: PORT 1: TB6OUT	0: PORT 1: TB5OUT	0: PORT 1: TB4OUT

Function	Corresponding BIT of PBFC	Corresponding BIT of PBCR	PORT to be used
TB4OUT output setting	1	1	PB0
TB5OUT output setting	1	1	PB1
TB6OUT output setting	1	1	PB2
TB7OUT output setting	1	1	PB3
TB8OUT output setting	1	1	PB4
TB9OUT output setting	1	1	PB5
TBAIN0 input setting	1	0	PB6
TBAIN1 input setting	1	0	PB7

Fig. 7.10.3 Port B Registers

7.11 Port C (PC0 to PC7)

Port C is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PCCR. A reset allows PCCR to be reset to "0" and the port C to function as an input port. Besides the input/output port function, the port C performs other functions: PC0, PC3 and PC6 output SIO data, PC1, PC4 and PC7 input SIO data, and PC2 and PC5 input and output SIO CLK or input CTS. These functions are enabled by setting corresponding bits of PCFC to "1." A reset allows PCCR and PCFC to be cleared to "0" and the port C to function as an input port.

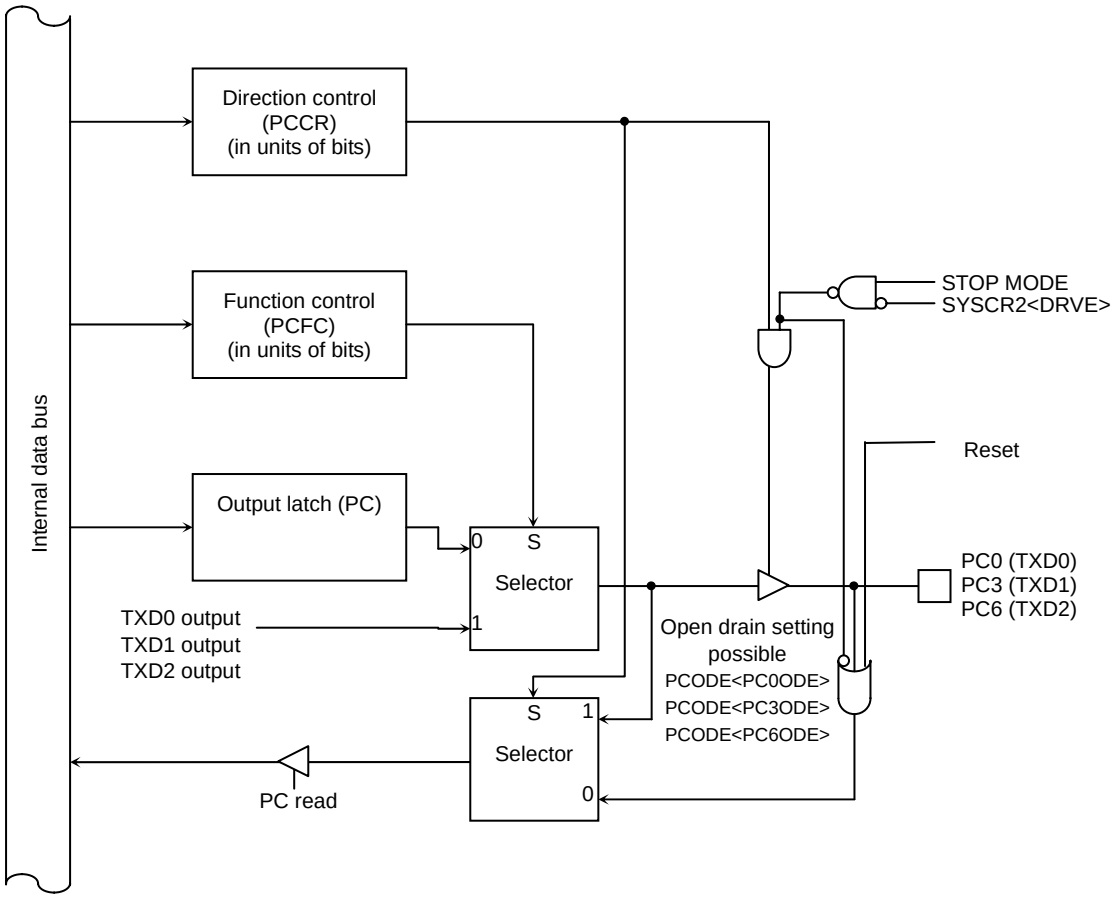


Fig. 7.11.1 Port C (PC0, PC3, PC6)

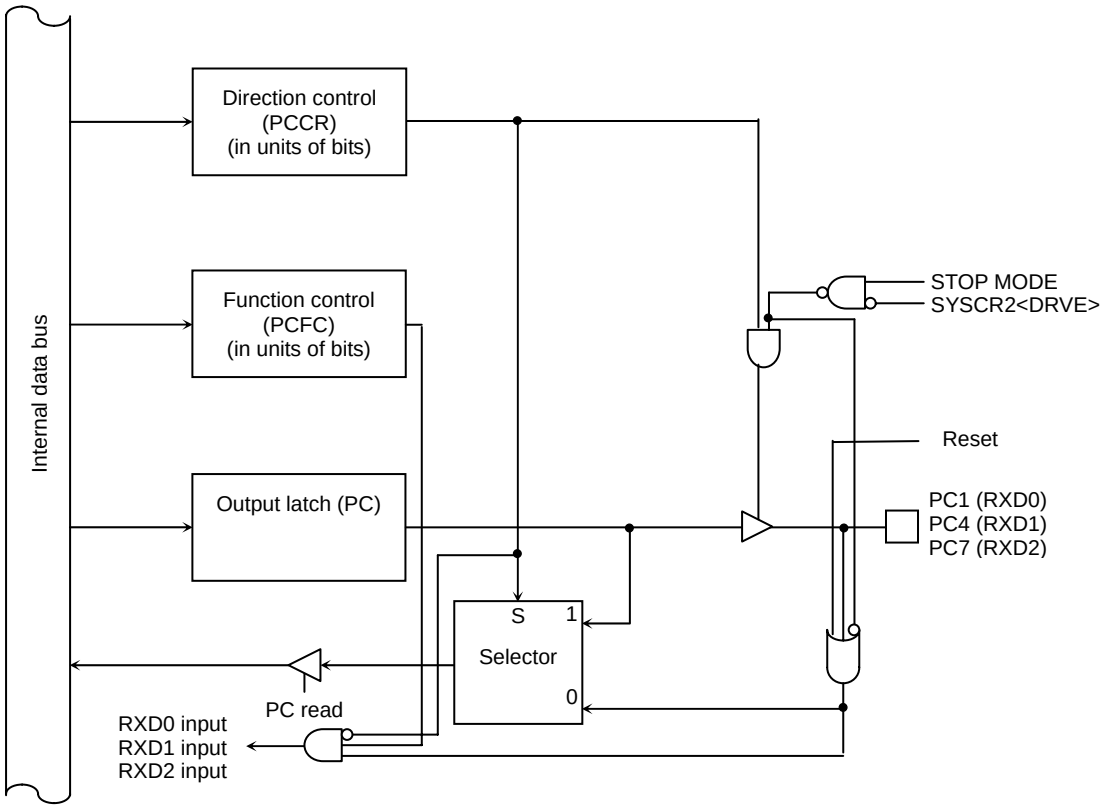


Fig. 7.11.2 Port C (PC1, PC4, PC7)

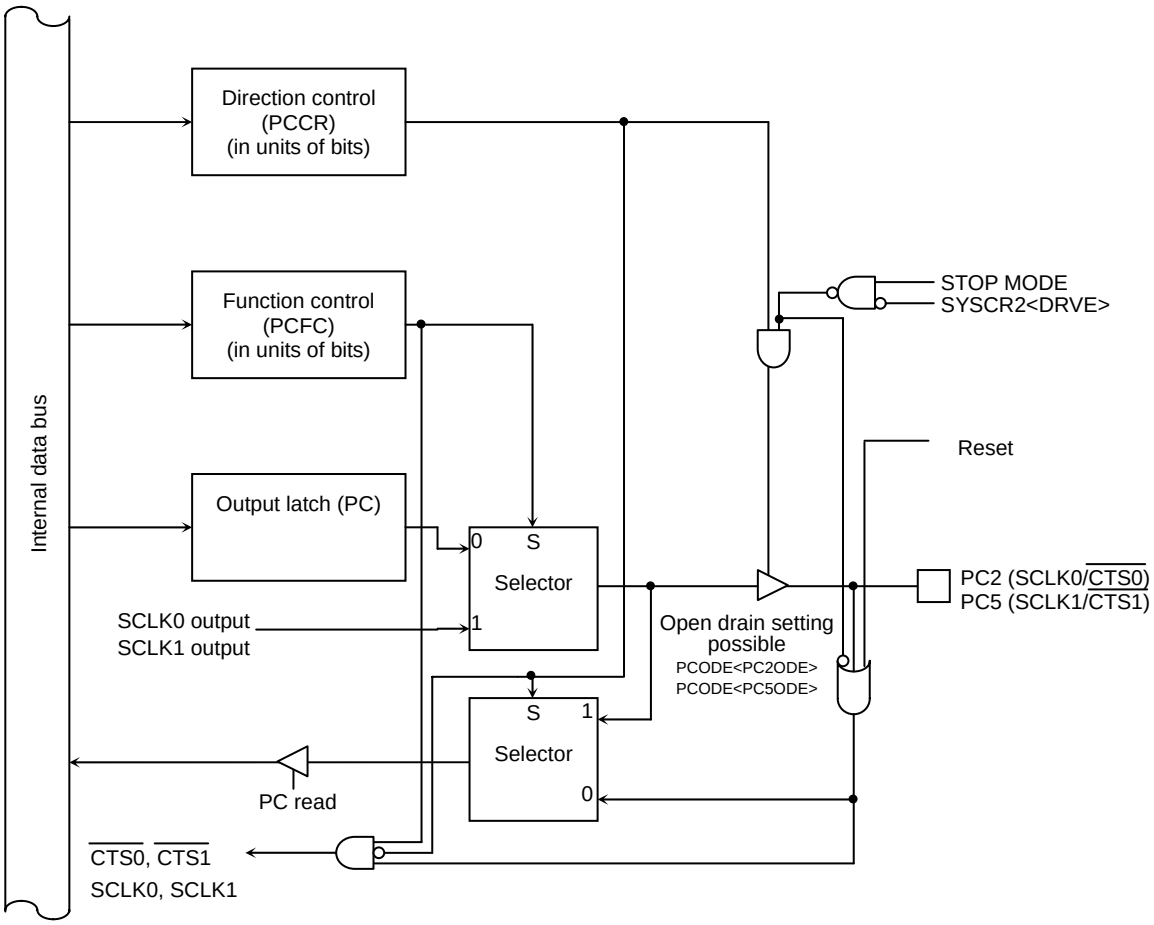


Fig. 7.11.3 Port C (PC2, PC5)

Port C register

PC (0xFFFF_F051)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port C control register

PCCR (0xFFFF_F055)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7C	PC6C	PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port C function register

PCFC (0xFFFF_F059)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7F	PC6F	PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: RXD2	0: PORT 1: TXD2	0: PORT 1: SCLK1 / $\overline{\text{CTS1}}$	0: PORT 1: RXD1	0: PORT 1: TXD1	0: PORT 1: SCLK0 / $\overline{\text{CTS0}}$	0: PORT 1: RXD0	0: PORT 1: TXD0

Port C open drain control register

PCODE (0xFFFF_F05D)

	7	6	5	4	3	2	1	0
Bit Symbol		PC6ODE	PC5ODE		PC3ODE	PC2ODE		PC0ODE
Read/Write	R	R/W		R	R/W		R	R/W
After reset	0	0	0	0	0	0	0	0
Function	0: CMOS	0: CMOS 1: Open drain	0: CMOS 1: Open drain	0: CMOS	0: CMOS 1: Open drain	0: CMOS 1: Open drain	0: CMOS	0: CMOS 1: Open drain

Function	Corresponding BIT of PCFC	Corresponding BIT of PCCR	PORT to be used
TXD0 output setting	1	1	PC0
RXD0 input setting	1	0	PC1
SCLK0 output setting	1	1	PC2
SCLK0 input setting	1	0	
CTS0 input setting	1	0	
TXD1 output setting	1	1	PC3
RXD1 output setting	1	1	PC4
SCLK1 output setting	1	1	PC5
SCLK1 input setting	1	0	
CTS1 input setting	1	0	
TXD2 output setting	1	0	PC6
RXD2 input setting	1	0	PC7

Fig. 7.11.4 Port C Registers

7.12 Port D (PD0 to PD7)

The port D is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PDCR. A reset allows PDCR to be reset to "0" and the port D to function as an input port. Besides the input/output port function, the port D performs other functions: PD0, PD3 and PD6 input and output SIO CLK or input CTS, PD1 and PD4 output SIO data, PD2 and PD5 input SIO data, and PD7 inputs external interrupts. These functions are enabled by setting corresponding bits of PDFC to "1." A reset allows PDCR and PDFC to be cleared to "0" and the port D to function as an input port.

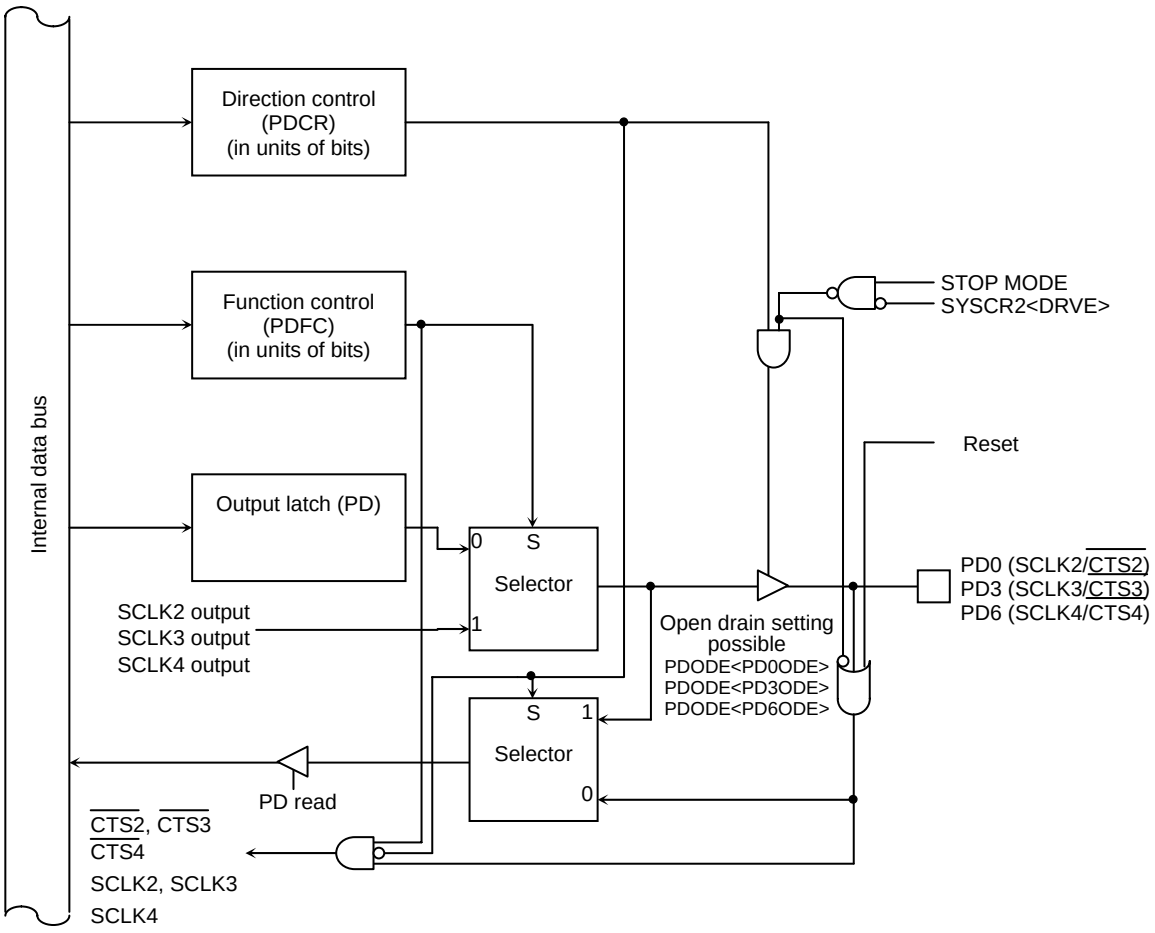


Fig. 7.12.1 Port D (PD0, PD3, PD6)

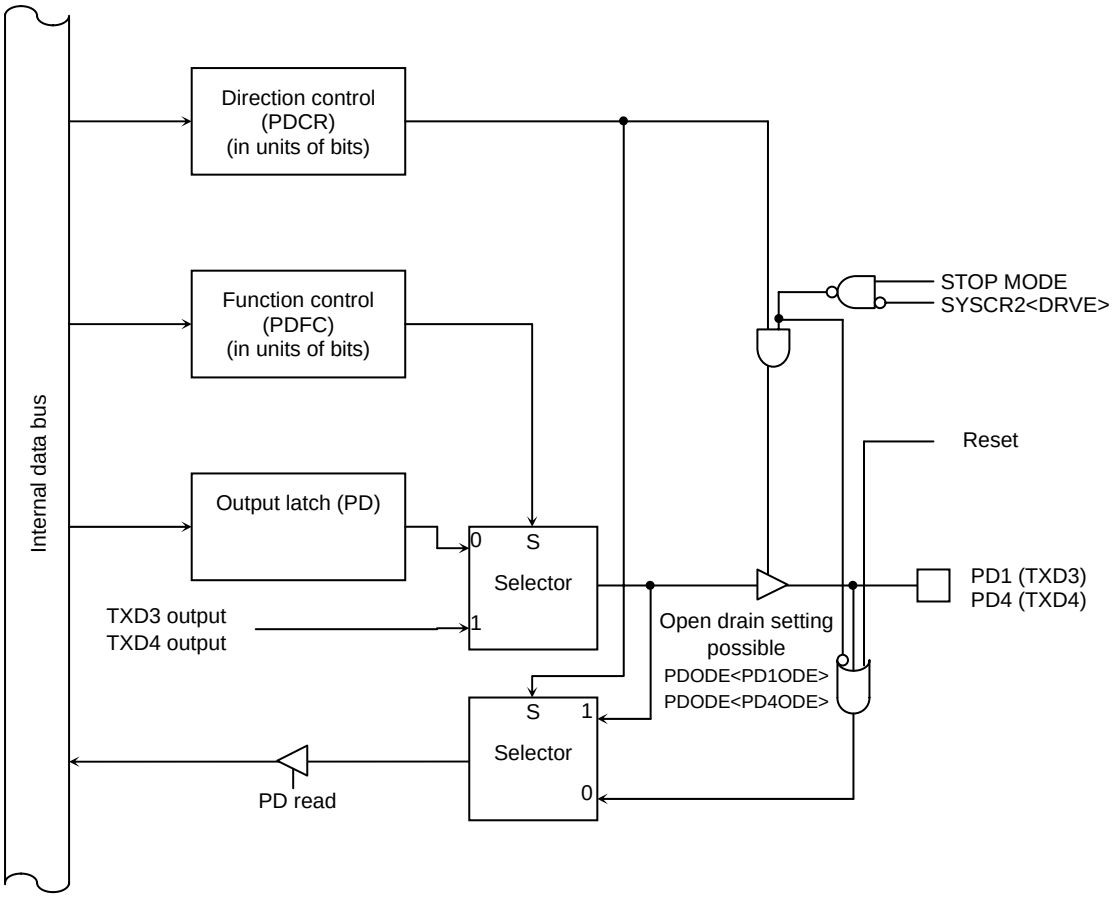


Fig. 7.12.2 Port D (PD1, PD4)

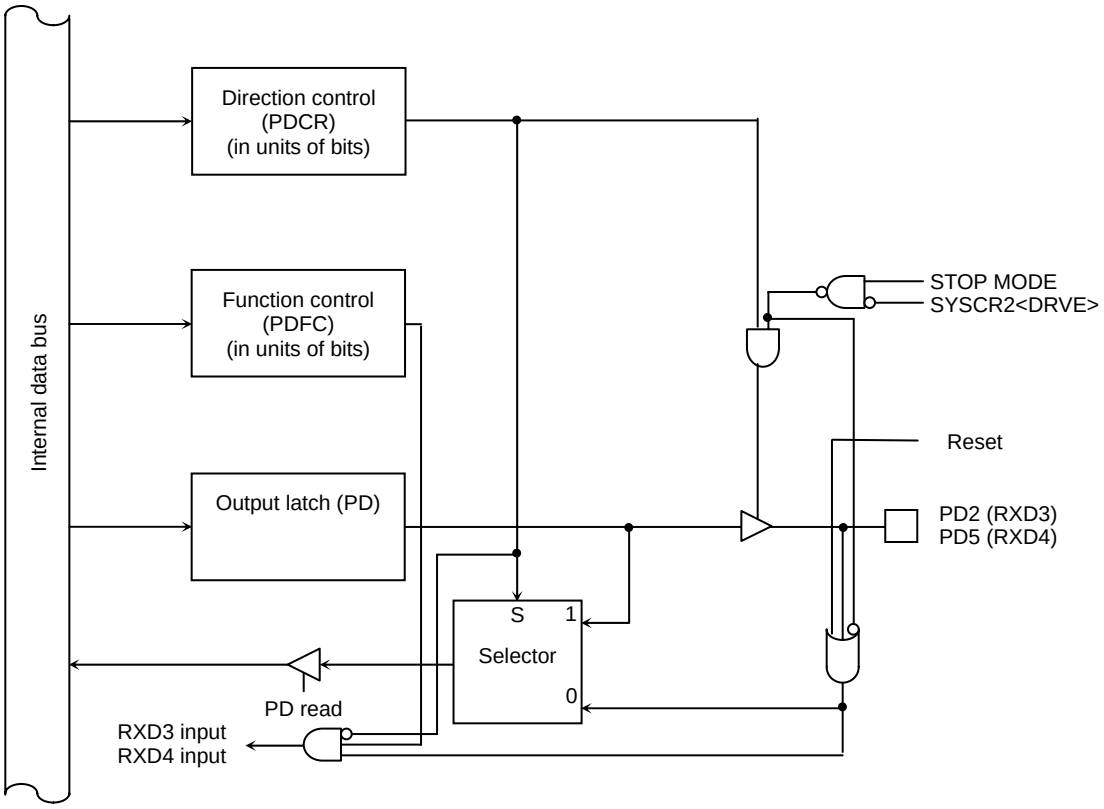


Fig. 7.12.3 Port D (PD2, PD5)

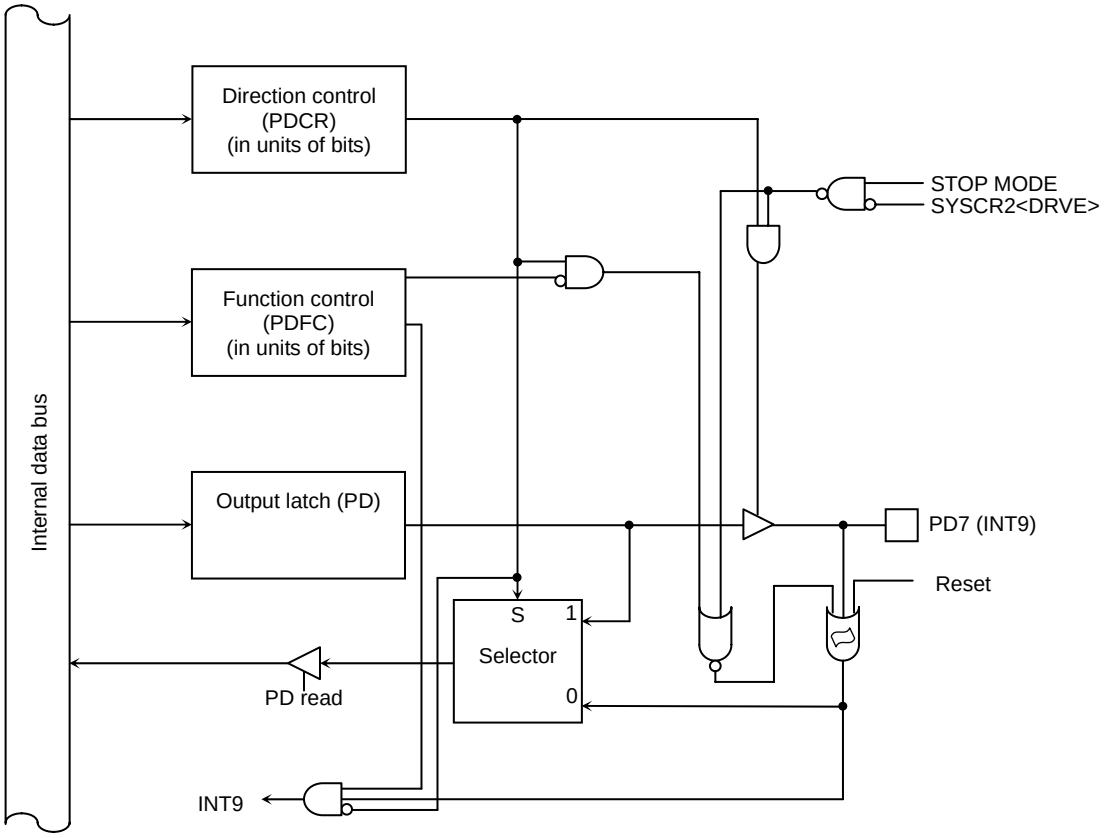


Fig. 7.12.4 Port D (PD7)

Port D register

PD (0xFFFF_F052)

	7	6	5	4	3	2	1	0
Bit Symbol	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port D control register

PDCR (0xFFFF_F056)

	7	6	5	4	3	2	1	0
Bit Symbol	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port D function register

PDFC (0xFFFF_F05A)

	7	6	5	4	3	2	1	0
Bit Symbol	PD7F	PD6F	PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: INT9	0: PORT 1: SCLK4 / $\overline{\text{CTS4}}$	0: PORT 1: RXD4	0: PORT 1: TXD4	0: PORT 1: SCLK3 / $\overline{\text{CTS3}}$	0: PORT 1: RXD3	0: PORT 1: TXD3	0: PORT 1: SCLK2 / $\overline{\text{CTS2}}$

Port D open drain control register

PDODE (0xFFFF_F05E)

	7	6	5	4	3	2	1	0
Bit Symbol		PD6ODE		PD4ODE	PD3ODE		PD1ODE	PD0ODE
Read/Write	R	R/W	R	R/W		R	R/W	
After reset	0	0	0	0	0	0	0	0
Function	0: CMOS	0: CMOS 1: Open drain	0: CMOS	0: CMOS 1: Open drain	0: CMOS 1: Open drain	0: CMOS	0: CMOS 1: Open drain	0: CMOS 1: Open drain

Function	Corresponding BIT of PDFC	Corresponding BIT of PDCR	PORT to be used
SCLK2 output setting	1	1	PD0
SCLK2 input setting	1	0	
CTS2 input setting	1	0	
TXD3 output setting	1	1	PD1
RXD3 input setting	1	0	PC2
SCLK3 output setting	1	1	PD3
SCLK3 input setting	1	0	
CTS3 input setting	1	0	
TXD4 output setting	1	1	PD4
RXD4 output setting	1	1	PD5
SCLK4 output setting	1	1	PD6
SCLK4 input setting	1	0	
CTS4 input setting	1	0	
INT9 input setting	1(*1)	0	PD7

(*1) This bit setting is used only if an interrupt must be generated to clear the STOP status and if SYSCR<DRVE> is set to 0. In all other cases, this bit setting does not need to be used.

Fig. 7.12.5 Port D Registers

7.13 Port E (PE0 through PE7)

The port E is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PECR. A reset allows PECR to be reset to "0" and the port E to function as an input port. Besides the input/output port function, the port E performs other functions: PE0 outputs SIO data, PE1 inputs SIO data, PE2 inputs and outputs SIO CLK or inputs CTS, and PE6 and PE7 input external interrupts. These functions are enabled by setting corresponding bits of PEFC to "1." A reset allows PECR and PEFC to be cleared to "0" and the port E to function as an input port.

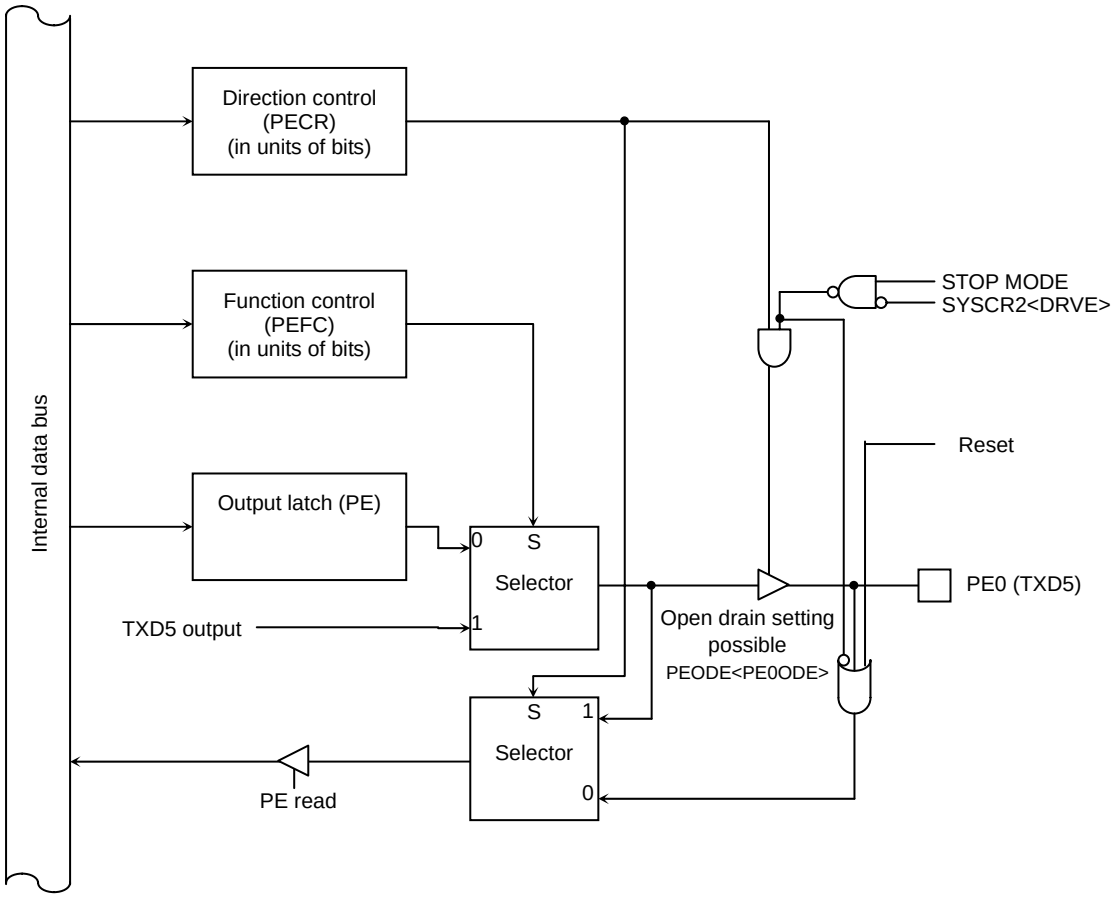


Fig. 7.13.1 Port E (PE0)

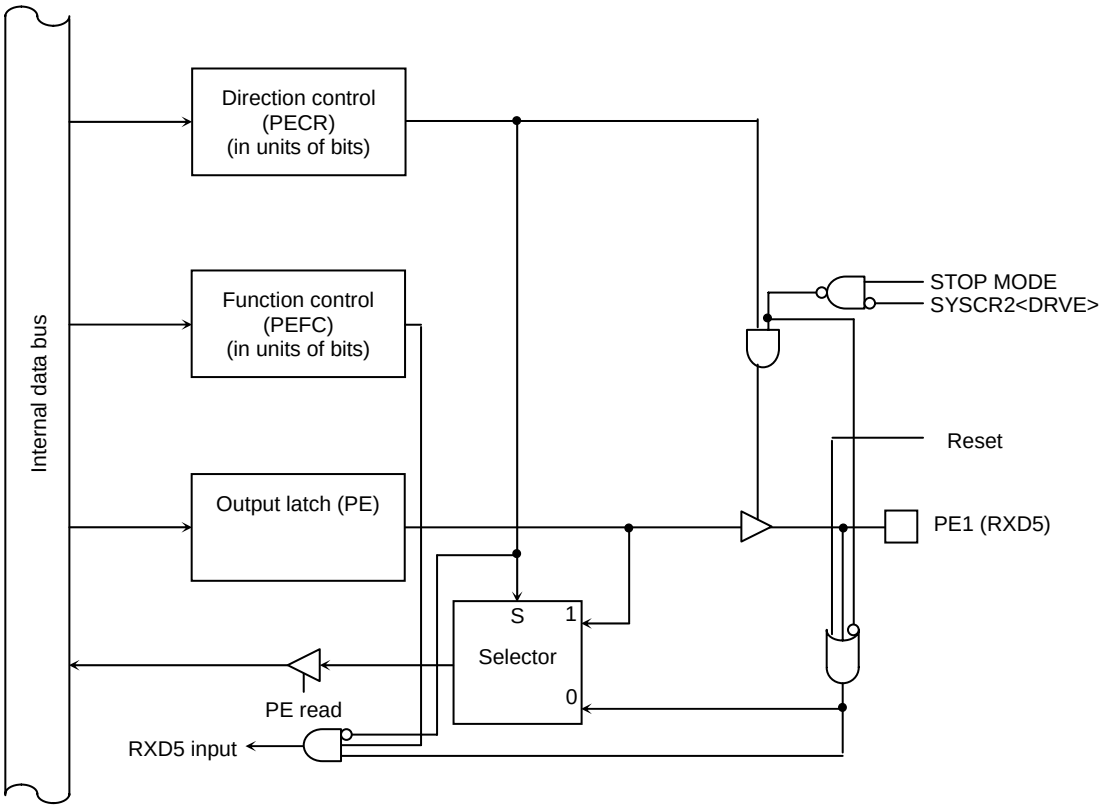


Fig. 7.13.2 Port E (PE1)

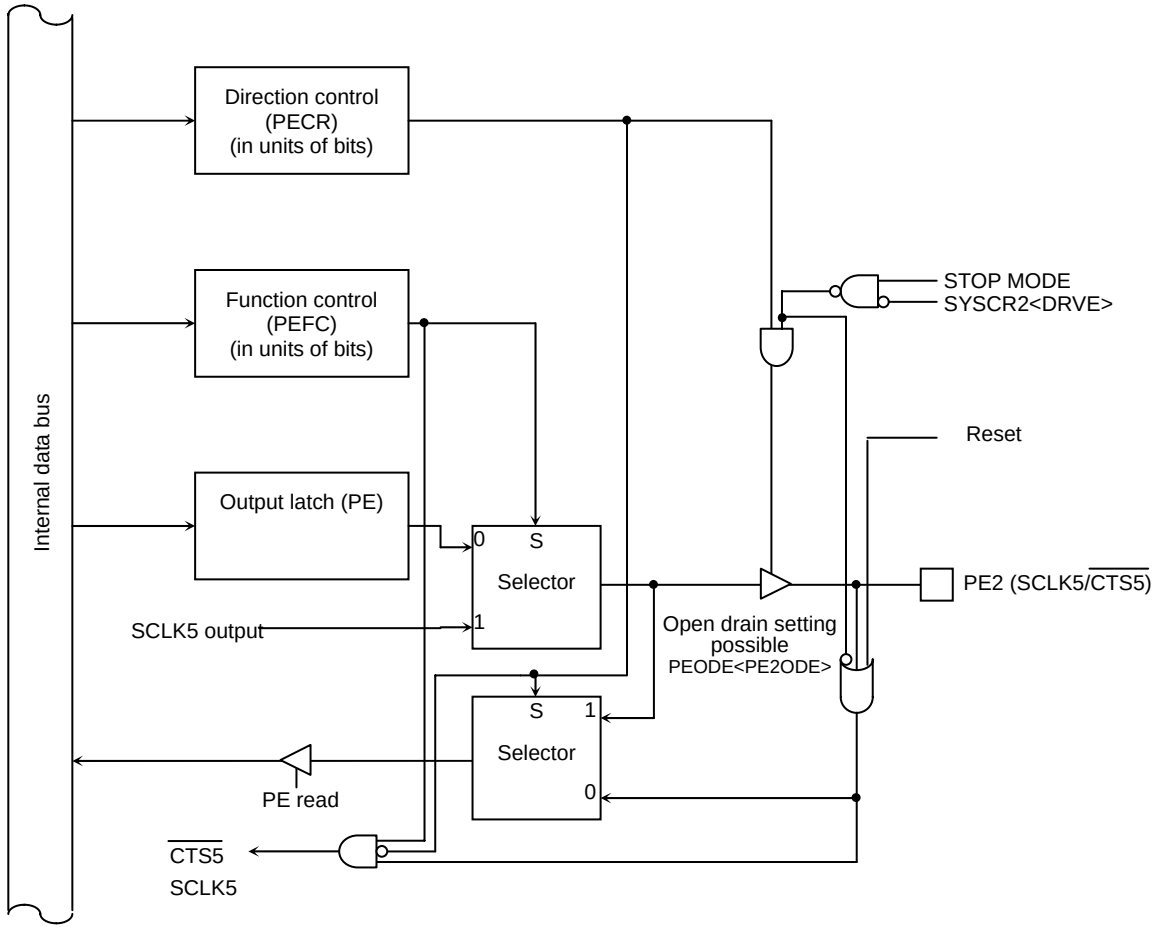


Fig. 7.13.3 Port E (PE2)

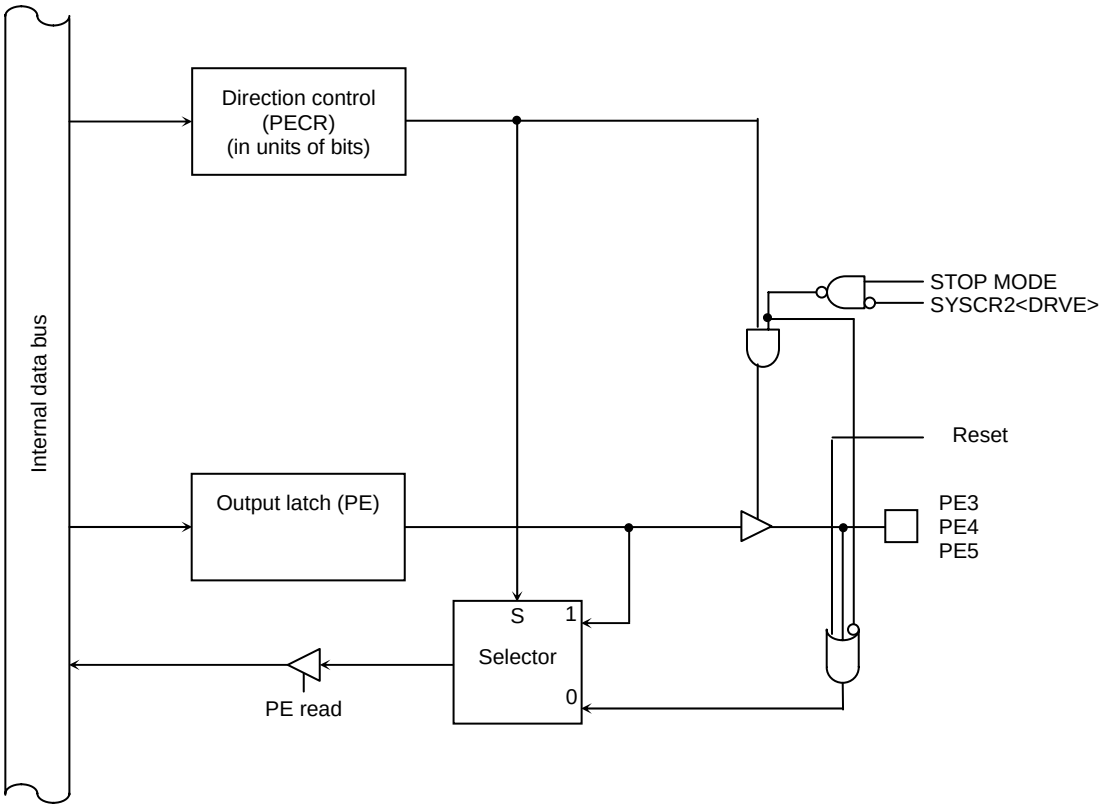


Fig. 7.13.4 Port E (PE3, PE4, PE5)

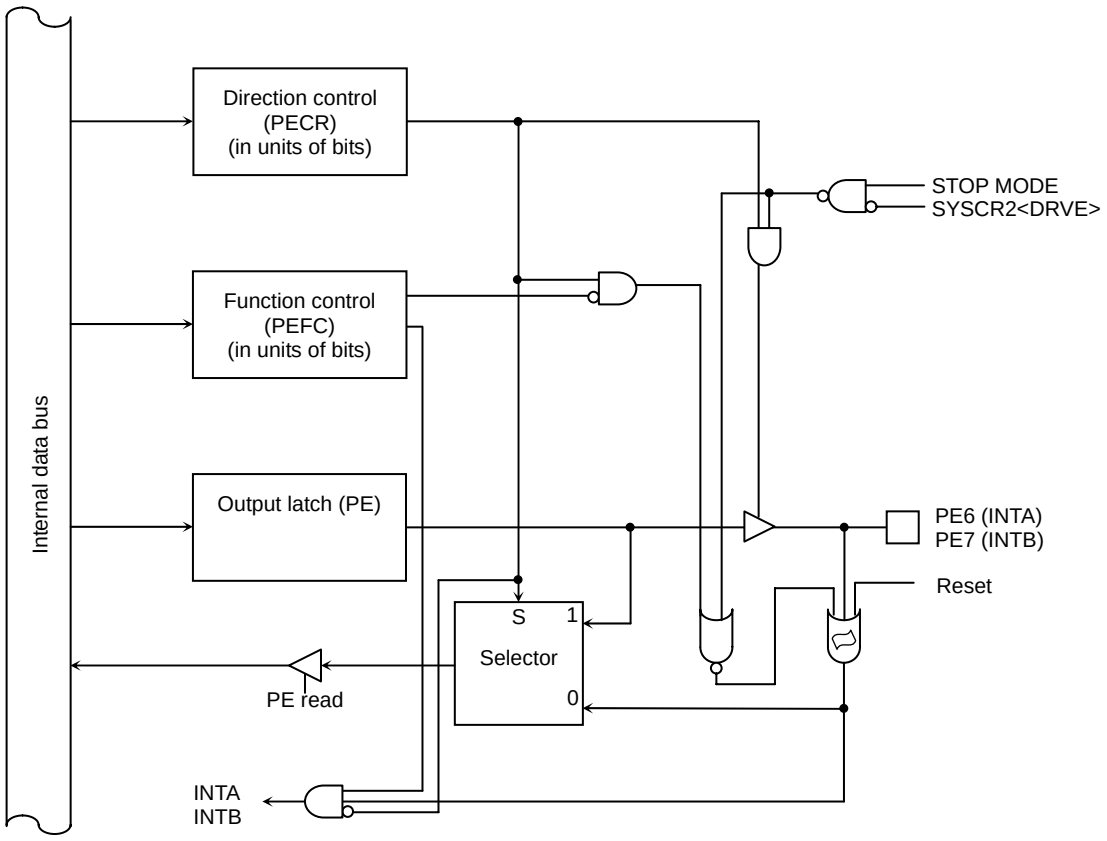


Fig. 7.13.5 Port E (PE6, PE7)

Port E register

PE (0xFFFF_F053)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port E control register

PECR (0xFFFF_F057)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port E function register

PEFC (0xFFFF_F05B)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7F	PE6F	PE5F	PE4F	PE3F	PE2F	PE1F	PE0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: INTB	0: PORT 1: INTA	0: PORT	0: PORT	0: PORT	0: PORT 1: SCLK5 / $\overline{\text{CTS5}}$	0: PORT 1: RXD5	0: PORT 1: TXD5

Port E open drain control register

PEODE (0xFFFF_F05F)

	7	6	5	4	3	2	1	0
Bit Symbol						PE2ODE		PE0ODE
Read/Write	R					R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	0: CMOS	0: CMOS	0: CMOS	0: CMOS	0: CMOS	0: CMOS 1: Open drain	0: CMOS	0: CMOS 1: Open drain

Function	Corresponding BIT of PEFC	Corresponding BIT of PECR	PORT to be used
TXD5 output setting	1	1	PE0
RXD3 output setting	1	0	PE1
SCLK5 output setting	1	1	PE2
SCLK5 input setting	1	0	
CTS5 input setting	1	0	
INTA input setting	1(*1)	0	PE6
INTB input setting	1(*1)	0	PE7

(*1) This bit setting is used only if an interrupt must be generated to clear the STOP status and if SYSCR<DRVE> is set to 0. In all other cases, this bit setting does not need to be used.

Fig. 7.13.6 Port E Registers

7.14 Port F (PF0 through PF7)

The port F is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PFCR. A reset allows PFCR to be reset to "0" and the port F to function as an input port. Besides the input/output port function, the port F performs other functions: PF0 through PF2 input and output SB1, PE3 and PE5 input the DMA request signal, PF4 and PF6 output the DMA acknowledge signal, and PF7 inputs external clock sources of a 32-bit time base timer. These functions are enabled by setting corresponding bits of PFFC to "1." A reset allows PFCR and PFFC to be cleared to "0" and the port F to function as an input port. The DMAC function is shared by PF3 through PF6 and PJ0 through PJ3. To give PF0 through PF3 the precedence in using the DMAC function, the corresponding bit of PFFC must be set to "1."

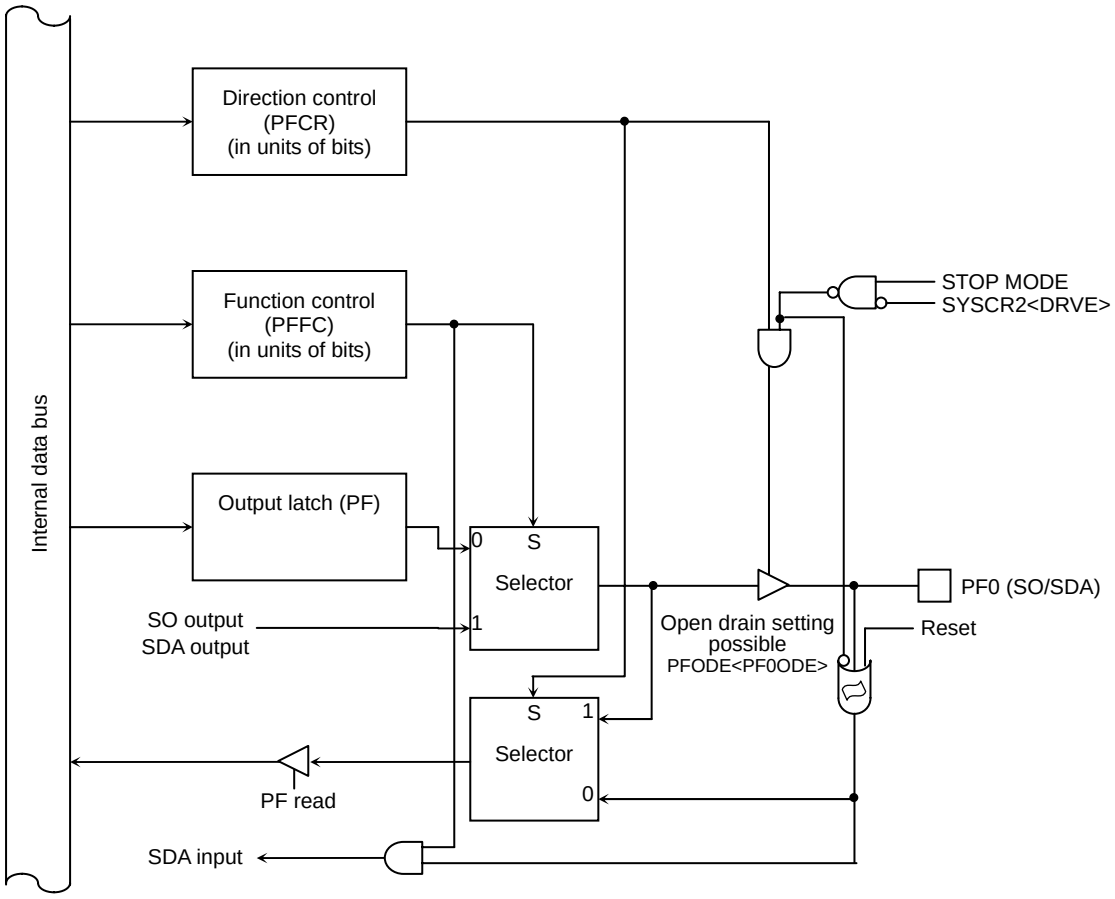


Fig. 7.14.1 Port F (PF0)

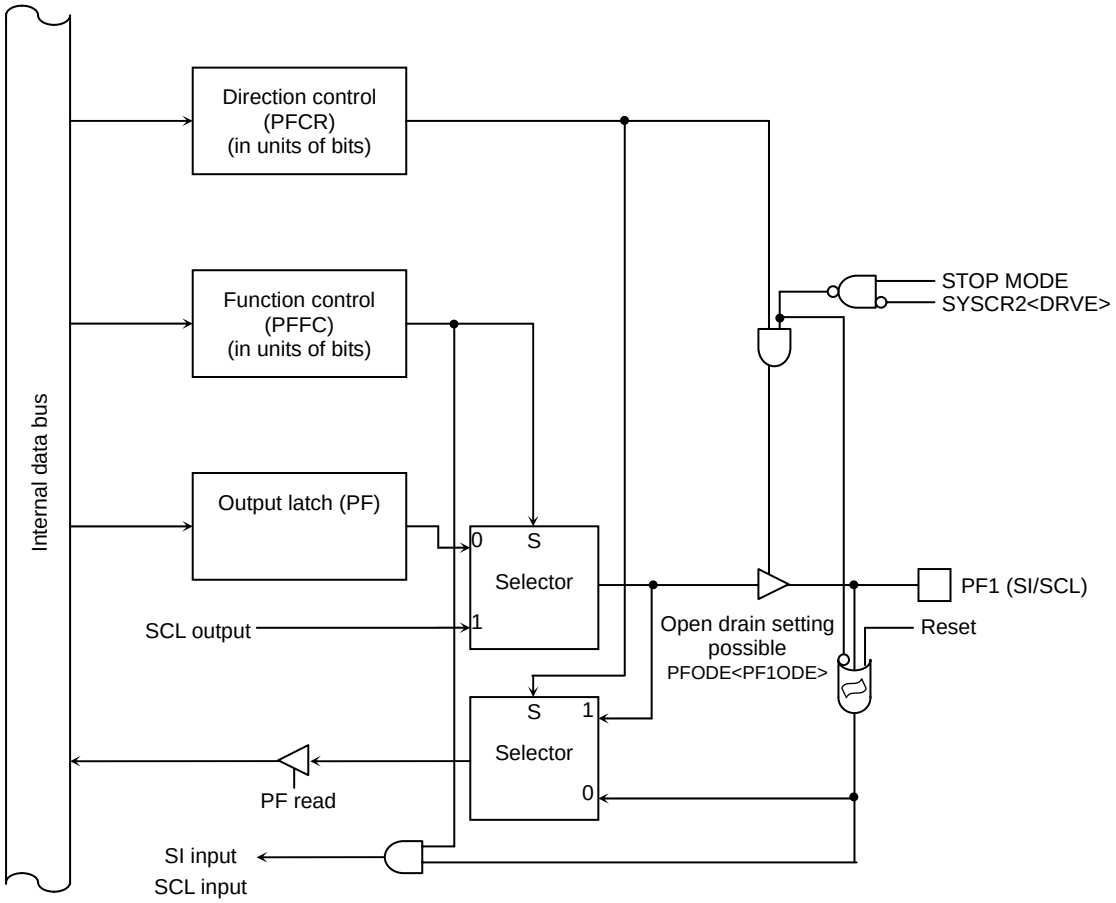


Fig. 7.14.2 Port F (PF1)

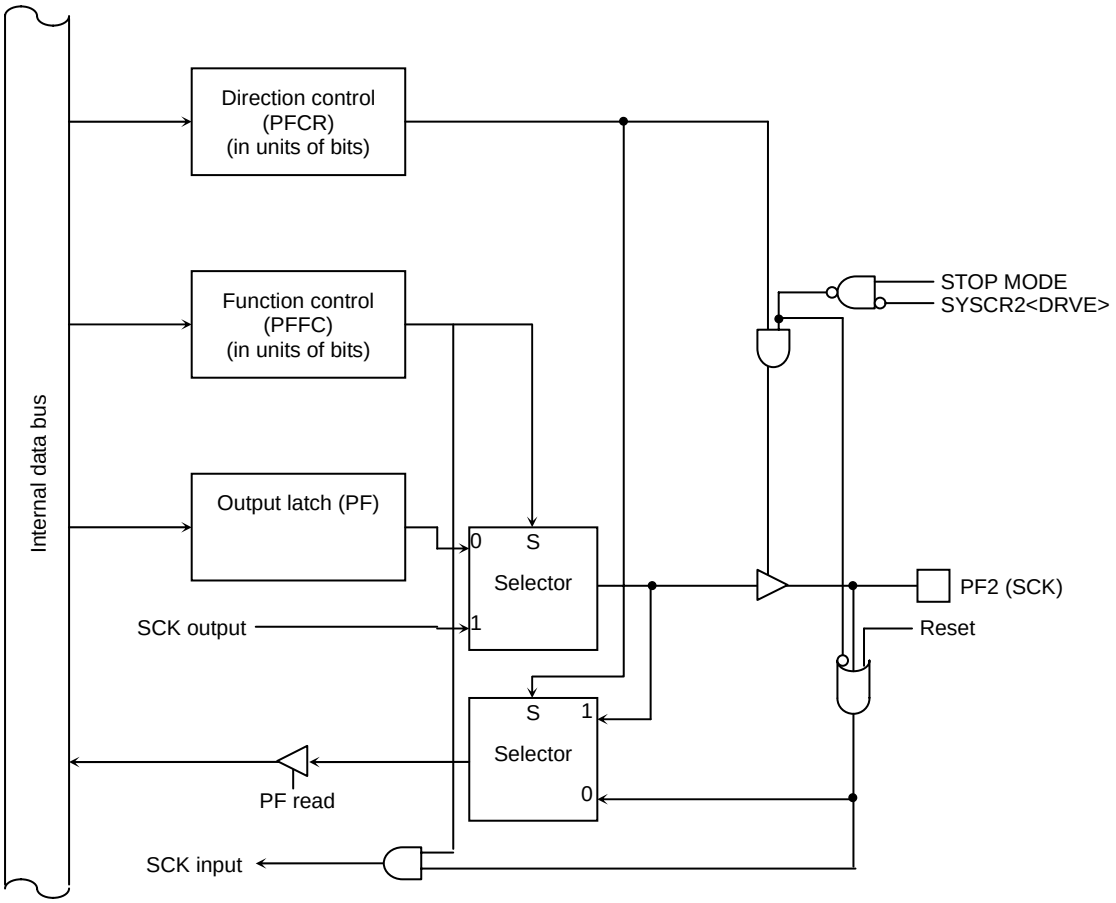


Fig. 7.14.3 Port F (PF2)

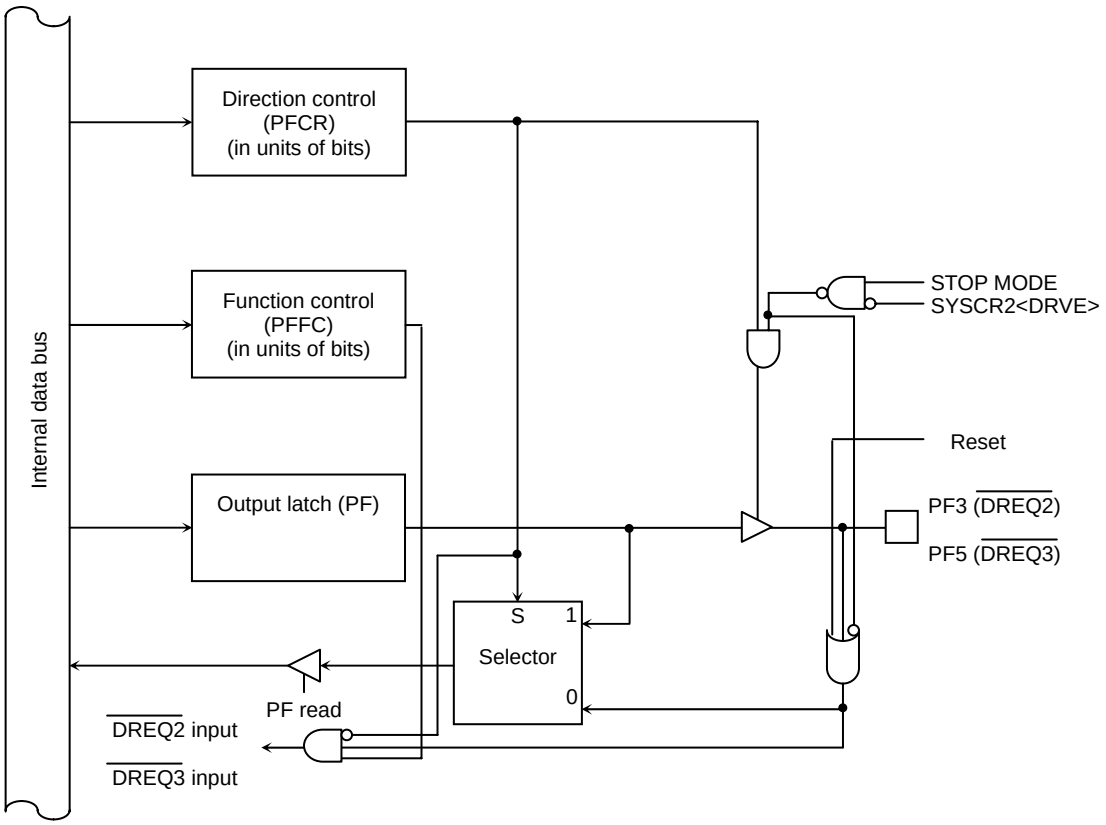


Fig. 7.14.4 Port F (PF3, PF5)

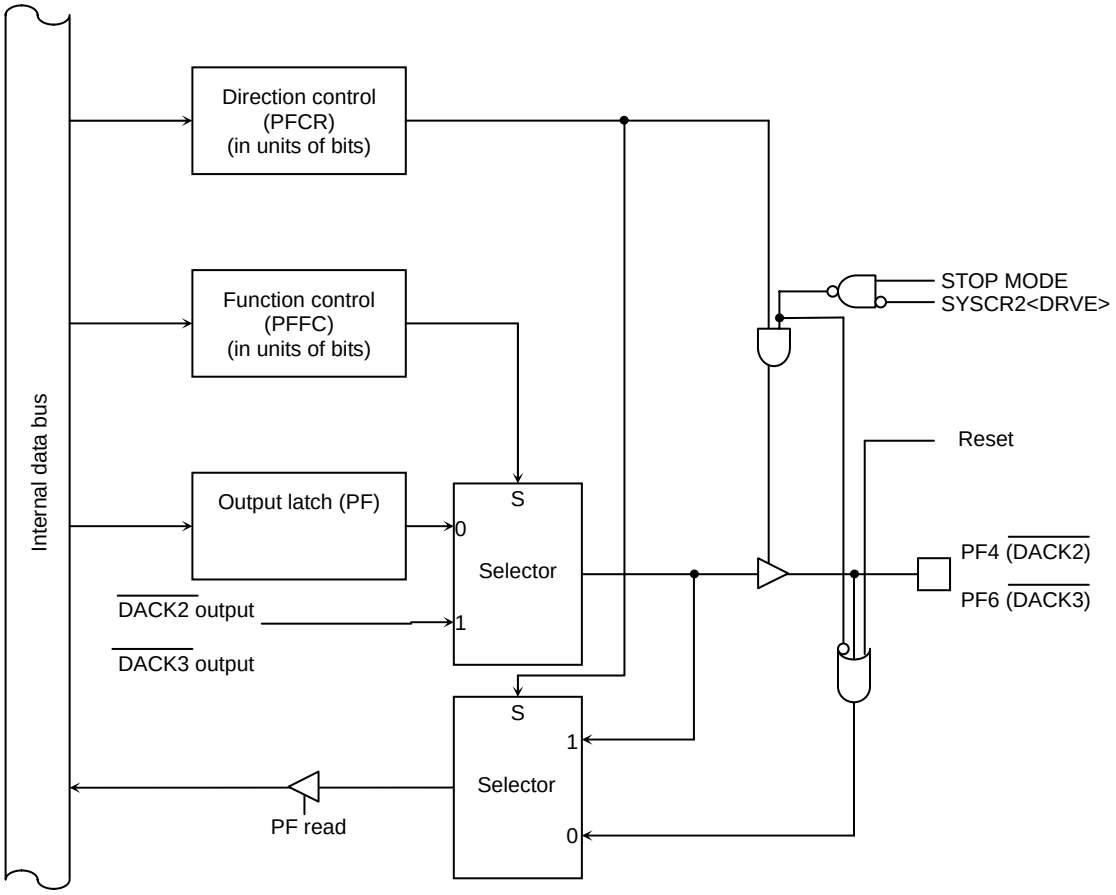


Fig. 7.14.5 Port F (PF4, PF6)

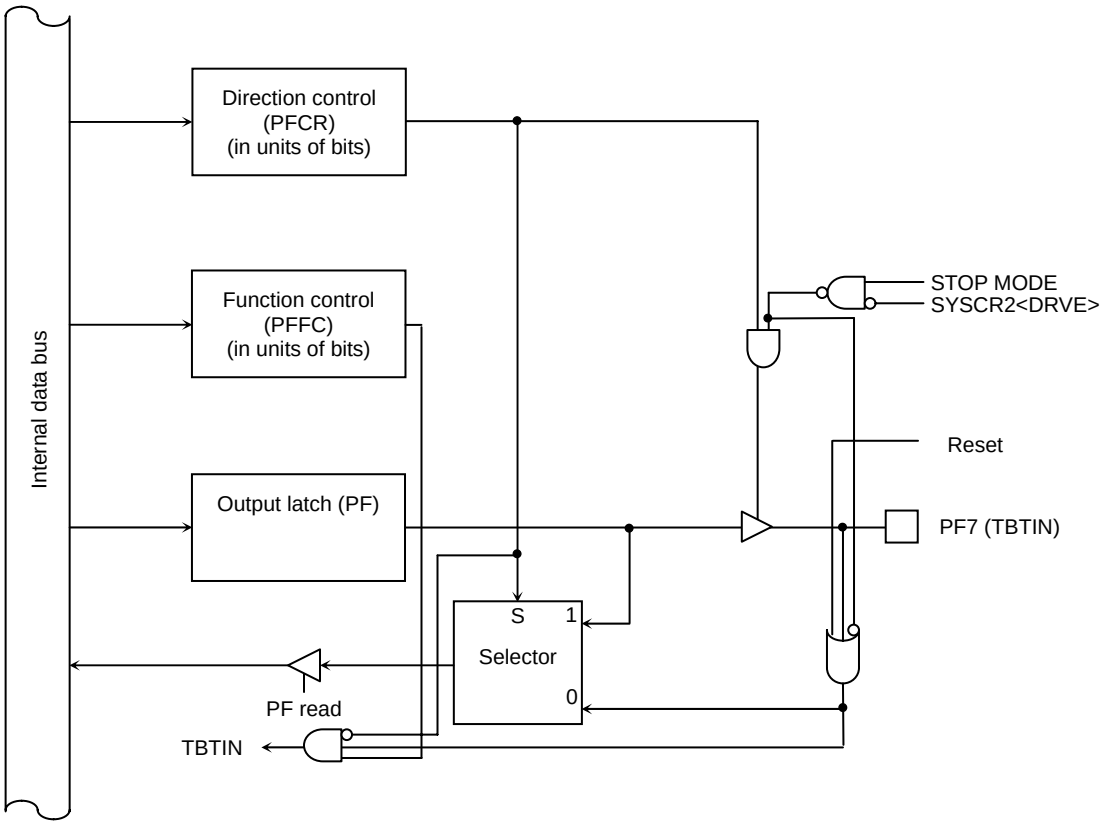


Fig. 7.14.6 Port F (PF7)

Port F register

PF (0xFFFF_F060)	7	6	5	4	3	2	1	0
Bit Symbol	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port F control register

PFCR (0xFFFF_F064)	7	6	5	4	3	2	1	0
Bit Symbol	PF7C	PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port F function register

PFFC (0xFFFF_F068)	7	6	5	4	3	2	1	0
Bit Symbol	PF7F	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: TBTIN	0: PORT 1: $\overline{\text{DACK3}}$	0: PORT 1: $\overline{\text{DREQ3}}$	0: PORT 1: $\overline{\text{DACK2}}$	0: PORT 1: $\overline{\text{DREQ2}}$	0: PORT 1: SCK	0: PORT 1: SI / SCL	0: PORT 1: SO / SDA

Port F open drain control register

PFODE (0xFFFF_F06C)	7	6	5	4	3	2	1	0
Bit Symbol							PF1ODE	PF0ODE
Read/Write	R						R/W	
After reset	0	0	0	0	0	0	0	0
Function	0: CMOS	0: CMOS	0: CMOS	0: CMOS	0: CMOS	0: CMOS	0: CMOS 1: Open drain	0: CMOS 1: Open drain

Function	Corresponding BIT of PFFC	Corresponding BIT of PFCR	PORT to be used
SO output setting	1	1	PF0
SDA output setting	1	1	
SDA input setting	1	0	
SI input setting	1	0	PF1
SCL output setting	1	1	
SCL input setting	1	0	
SCLK5 output setting	1	1	PF2
SCLK5 input setting	1	0	
DREQ2 input setting	1	0	PF3
DACK2 output setting	1	1	PF4
DREQ3 input setting	1	0	PF5
DACK3 output setting	1	1	PF6
TBTIN input setting	1	0	PF7

(Note) The DMAC function is shared by the port F and the port J. If both ports are set to use the DMAC function, the port F is given priority in using the DMAC function.

Fig. 7.14.7 Port F Registers

7.15 Port G (PG0 through PG7)

The port G is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PGCR. A reset allows PGCR to be reset to "0" and the port G to function as an input port. Besides the input/output port function, the port G performs other functions: PG0 through PG3 input a 32-bit input capture trigger, and PG4 through PG7 output a 32-bit output compare. These functions are enabled by setting corresponding bits of PGFC to "1." A reset allows PGCR and PGFC to be cleared to "0" and the port G to function as an input port.

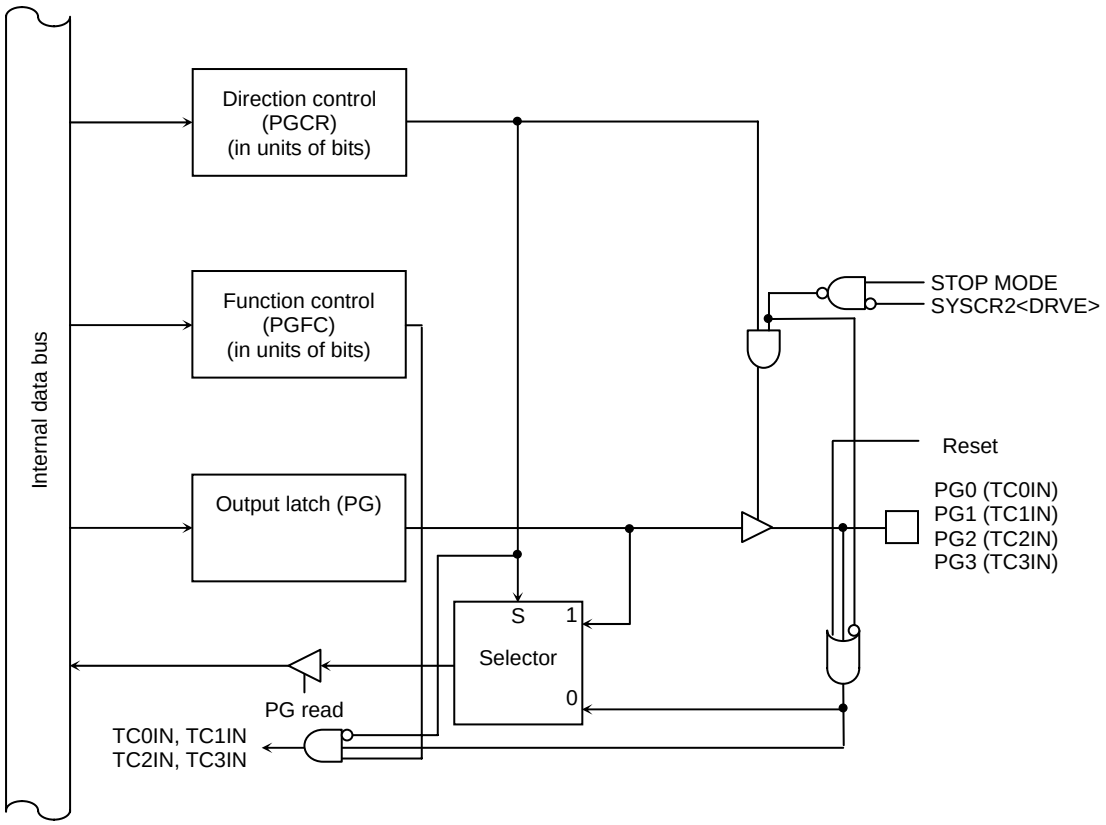


Fig. 7.15.1 Port G (PG0 through PG3)

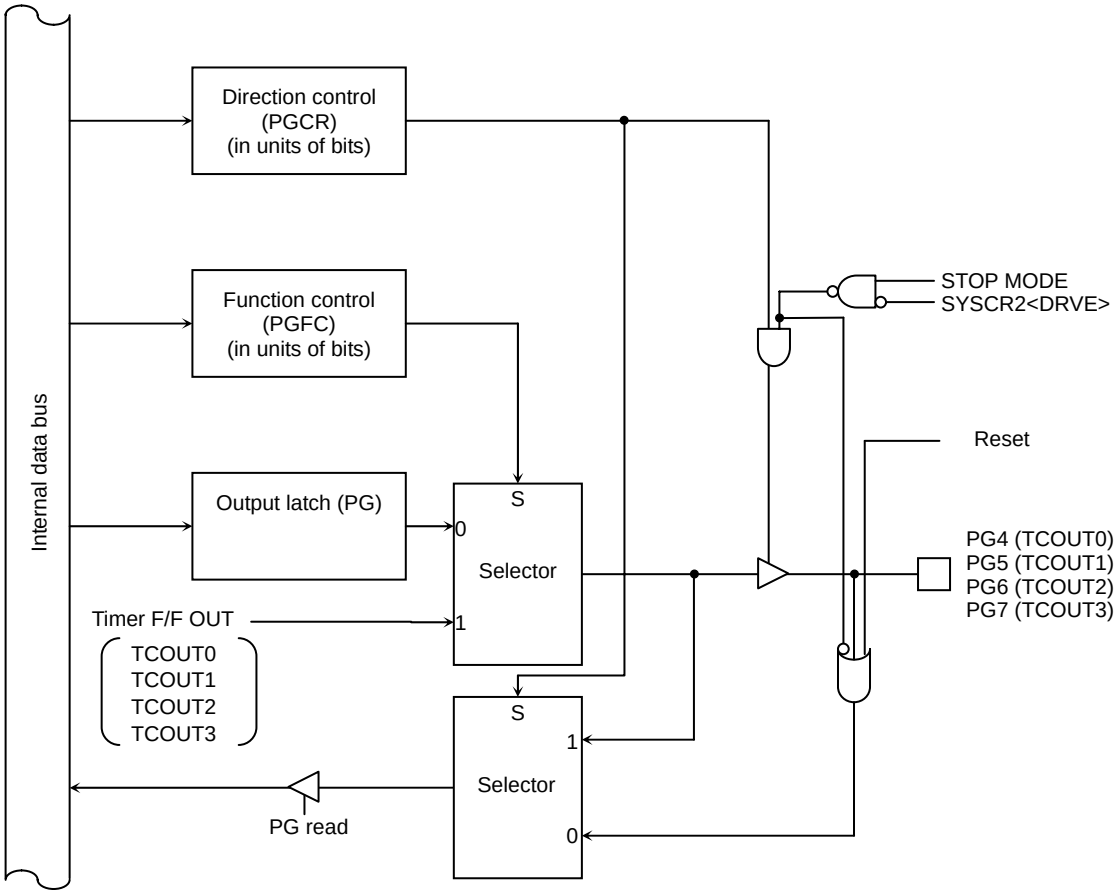


Fig. 7.15.2 Port G (PG4 through PG7)

Port G register

PG (0xFFFF_F061)

	7	6	5	4	3	2	1	0
Bit Symbol	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port G control register

PGCR (0xFFFF_F065)

	7	6	5	4	3	2	1	0
Bit Symbol	PG7C	PG6C	PG5C	PG4C	PG3C	PG2C	PG1C	PG0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port G function register

PGFC (0xFFFF_F069)

	7	6	5	4	3	2	1	0
Bit Symbol	PG7F	PG6F	PG5F	PG4F	PG3F	PG2F	PG1F	PG0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: TCOUT3	0: PORT 1: TCOUT2	0: PORT 1: TCOUT1	0: PORT 1: TCOUT0	0: PORT 1: TC3IN	0: PORT 1: TC2IN	0: PORT 1: TC1IN	0: PORT 1: TC0IN

Function	Corresponding BIT of PGFC	Corresponding BIT of PGCR	PORT to be used
TC0IN input setting	1	0	PG0
TC1IN input setting	1	0	PG1
TC2IN input setting	1	0	PG2
TC3IN input setting	1	0	PG3
TCOUT0 output setting	1	1	PG4
TCOUT1 output setting	1	1	PG5
TCOUT2 output setting	1	1	PG6
TCOUT3 output setting	1	1	PG7

Fig. 7.15.2 Port G Registers

7.16 Port H (PH0 through PH7)

The port H is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PHCR. A reset allows PHCR to be reset to "0" and the port H to function as an input port. Besides the input/output port function, the port H performs another function: PH0 through PH5 output the 32-bit output compare. This function is enabled by setting the corresponding bit of PHFC to "1." A reset allows PHCR and PHFC to be cleared to "0" and the port H to function as an input port.

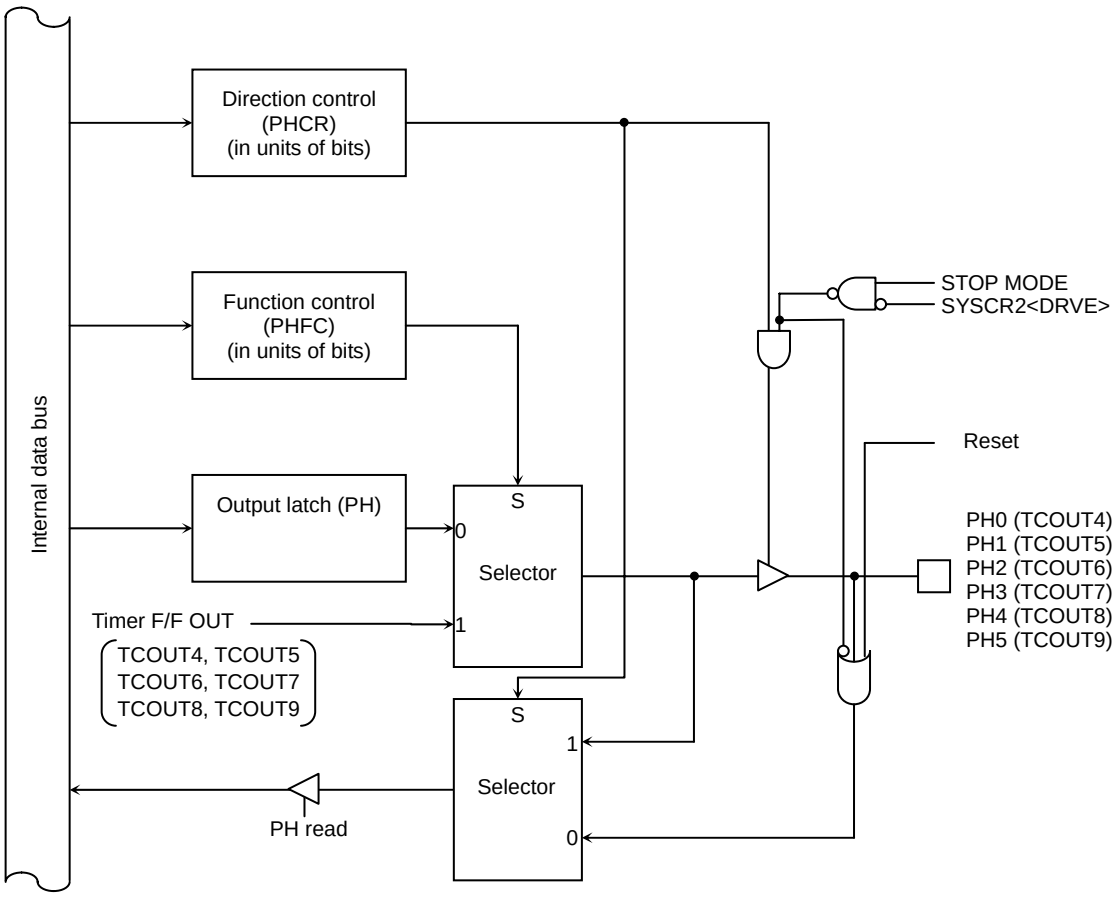


Fig. 7.16.1 Port H (PH0 through PH5)

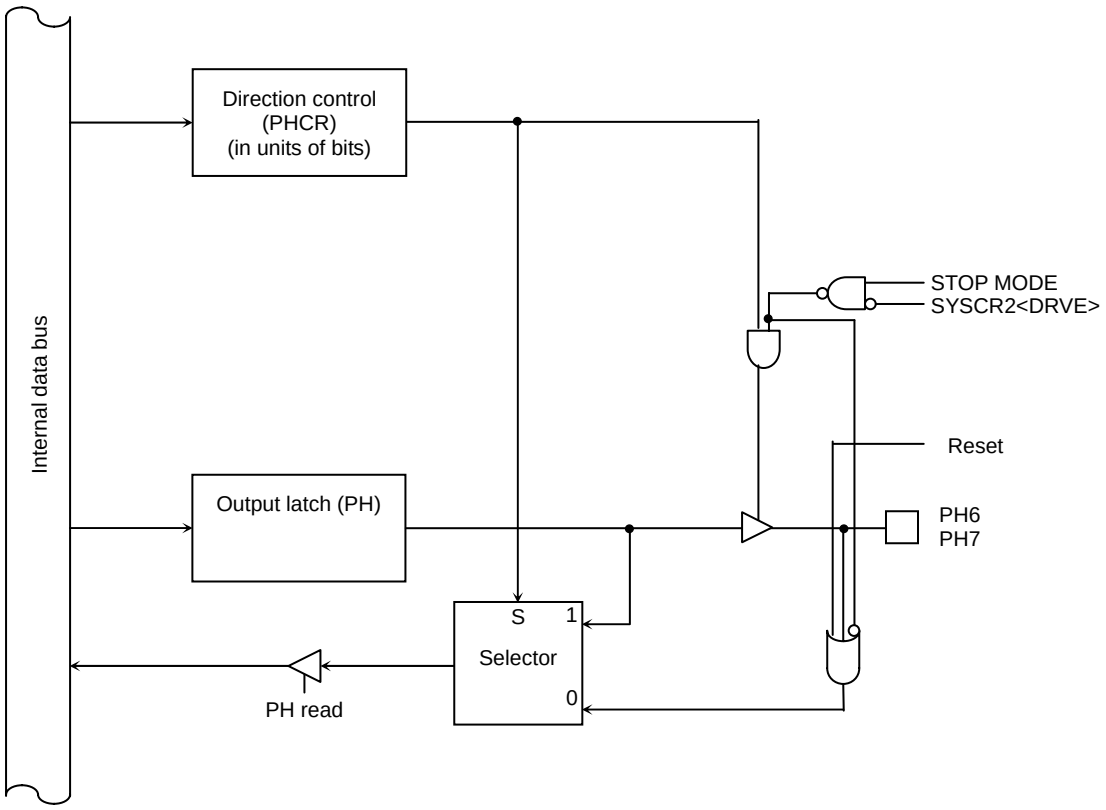


Fig. 7.16.2 Port H (PH6, PH7)

Port H register

PH (0xFFFF_F062)	7	6	5	4	3	2	1	0
Bit Symbol	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port H control register

PHCR (0xFFFF_F066)	7	6	5	4	3	2	1	0
Bit Symbol	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port H function register

PHFC (0xFFFF_F06A)	7	6	5	4	3	2	1	0
Bit Symbol			PH5F	PH4F	PH3F	PH2F	PH1F	PH0F
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	0: PORT	0: PORT	0: PORT 1: TCOUT9	0: PORT 1: TCOUT8	0: PORT 1: TCOUT7	0: PORT 1: TCOUT6	0: PORT 1: TCOUT5	0: PORT 1: TCOUT4

Function	Corresponding BIT of PHFC	Corresponding BIT of PHCR	PORT to be used
TCOUT4 output setting	1	1	PH0
TCOUT5 output setting	1	1	PH1
TCOUT6 output setting	1	1	PH2
TCOUT7 output setting	1	1	PH3
TCOUT8 output setting	1	1	PH4
TCOUT9 output setting	1	1	PH5

Fig. 7.16.3 Port H Registers

7.17 Port I (PI0 through PI4)

The port I is a general-purpose, 5-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PICR. A reset allows PICR to be reset to "0" and the port I to function as an input port. Besides the input/output port function, the port I performs another function: PI0 through PI4 input external interrupts. This function is enabled by setting the corresponding bit of PIFC to "1." A reset allows PICR and PIFC to be cleared to "0" and the port I to function as an input port. The external interrupt function is shared by PI0 through PI4 and PO0 through PO4. To give PO0 through PO4 the precedence in using the external interrupt function, the corresponding bit of POFC must be set to the interrupt function.

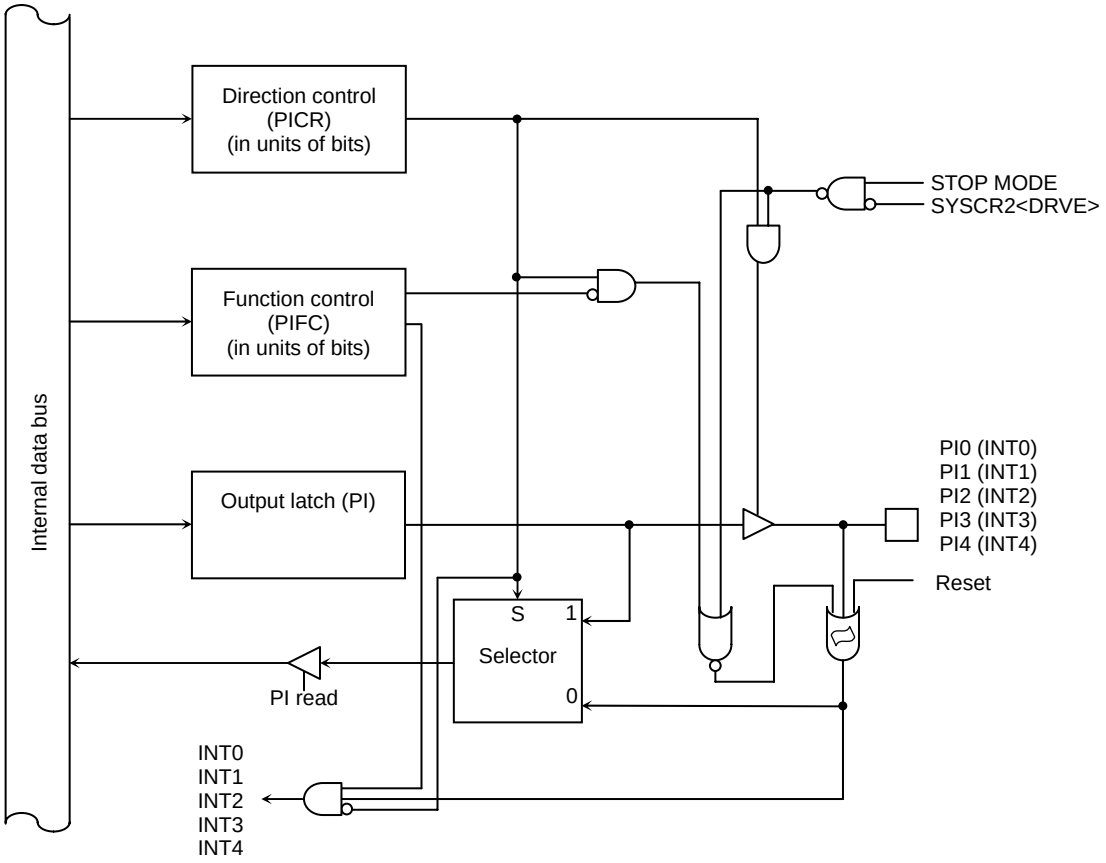


Fig. 7.17.1 Port I (PI0 through PI4)

Port I register

		7	6	5	4	3	2	1	0
PI	Bit Symbol				PI4	PI3	PI2	PI1	PI0
(0xFFFF_F063)	Read/Write	R			R/W				
	After reset	Input mode (output latch register is set to "1.")							

Port I control register

		7	6	5	4	3	2	1	0
PICR	Bit Symbol				PI4C	PI3C	PI2C	PI1C	PI0C
(0xFFFF_F063)	Read/Write	R			R/W				
	After reset	0	0	0	0	0	0	0	0
	Function				0: Input 1: Output				

Port I function register

		7	6	5	4	3	2	1	0
PIFC	Bit Symbol				PI4F	PI3F	PI2F	PI1F	PI0F
(0xFFFF_F06B)	Read/Write	R			R/W				
	After reset	0	0	0	0	0	0	0	0
	Function				0: PORT 1: INT4	0: PORT 1: INT3	0: PORT 1: INT2	0: PORT 1: INT1	0: PORT 1: INT0

Function	Corresponding BIT of PIFC	Corresponding BIT of PICR	PORT to be used
INT0 input setting	1 (*1)	0	PI0
INT1 input setting	1 (*1)	0	PI1
INT2 input setting	1 (*1)	0	PI2
INT3 input setting	1 (*1)	0	PI3
INT4 input setting	1 (*1)	0	PI4

(Note*1) This bit setting is used only if an interrupt must be generated to clear the STOP status and if SYSCR<DRVE> is set to 0. In all other cases, this bit setting does not need to be used.

(Note) The external interrupt function is shared by the port I and the port O. If both ports are set to use the external interrupt function, the port O is given priority in using the external interrupt function.

Fig. 7.17.2 Port I Registers

7.18 Port J (PJ0 through PJ3)

The port J is a general-purpose, 4-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PJCR. A reset allows PJCR to be reset to "0" and the port J to function as an input port. Besides the input/output port function, the port J performs other functions: PJ0 and PJ2 input the DMA request signal, and PJ1 and PJ3 output the DMA acknowledge signal. These functions are enabled by setting the corresponding bits of PJFC to "1." A reset allows PJCR and PJFC to be cleared to "0" and the port J to function as an input port. The DMAC function is shared by PJ0 through PJ3 and PF3 through PF6. To give PF0 through PF3 the precedence in using the DMAC function over PJ0 through PJ3, the corresponding bit of PFFC must be set to "1."

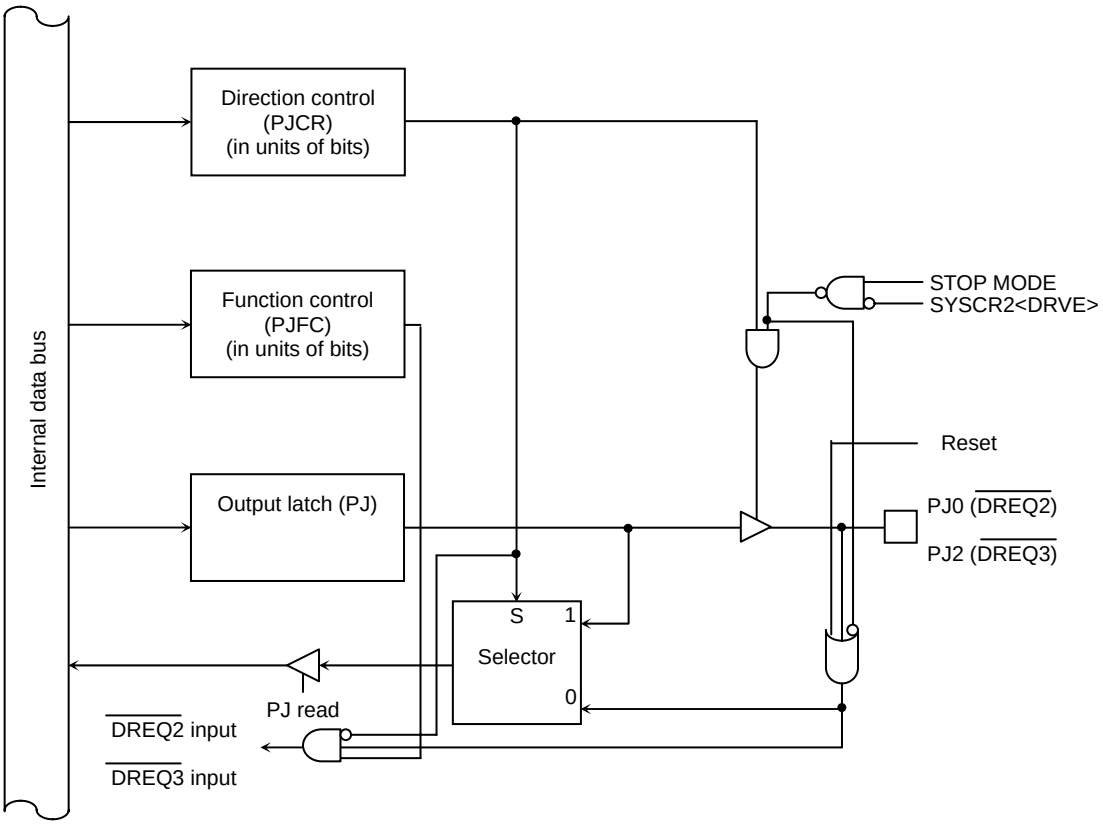


Fig. 7.18.1 Port J (PJ0, PJ2)

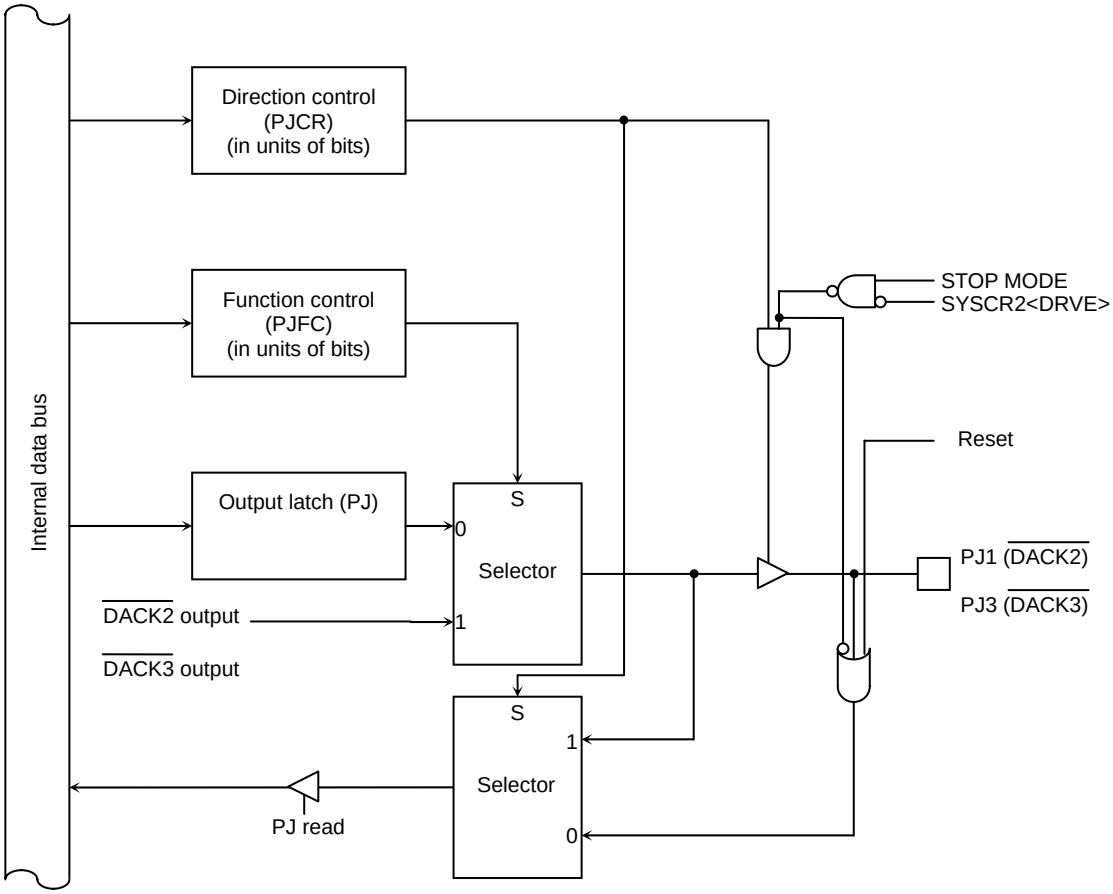


Fig. 7.18.2 Port J (PJ1, PJ3)

Port J register

PJ (0xFFFF_F070)	7	6	5	4	3	2	1	0
Bit Symbol					PJ3	PJ2	PJ1	PJ0
Read/Write	R				R/W			
After reset	Input mode (output latch register is set to "1.")							

Port J control register

PJCR (0xFFFF_F074)	7	6	5	4	3	2	1	0
Bit Symbol					PJ3C	PJ2C	PJ1C	PJ0C
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function					0: Input 1: Output			

Port J function register

PJFC (0xFFFF_F078)	7	6	5	4	3	2	1	0
Bit Symbol					PJ3F	PJ2F	PJ1F	PJ0F
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function					0: PORT 1: $\overline{\text{DACK3}}$	0: PORT 1: $\overline{\text{DREQ3}}$	0: PORT 1: $\overline{\text{DACK2}}$	0: PORT 1: $\overline{\text{DREQ2}}$

Function	Corresponding BIT of PJFC	Corresponding BIT of PJCR	PORT to be used
DREQ2 input setting	1	0	PJ0
DACK2 output setting	1	1	PJ1
DREQ3 input setting	1	0	PJ2
DACK3 output setting	1	1	PJ3

(Note) The DMAC function is shared by the port F and the port J. If both ports are set to use the DMAC function, the port F is given priority in using the DMAC function.

Fig. 7.18.3 Port J Registers

7.19 Port K (PK0 through PK7)

The port K is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PKCR. A reset allows PKCR to be reset to "0" and the port K to function as an input port. Besides the input/output port function, PK0 through PK7 perform the KEY input function. This function is enabled by setting the corresponding bit of PKFC to "1." A reset allows PKCR and PKFC to be cleared to "0" and the port K to function as an input port.

The ports K0 through K7 have a pull-up resistor function. This function is enabled only if KUPPUP<KEYPUPn> of the key-on wake-up circuit is set to "1" and if KEY input is enabled by KWUPSTn. For further details, refer to the section where key-on wake-up is discussed. If these ports are in operation, the pull-up function is disabled.

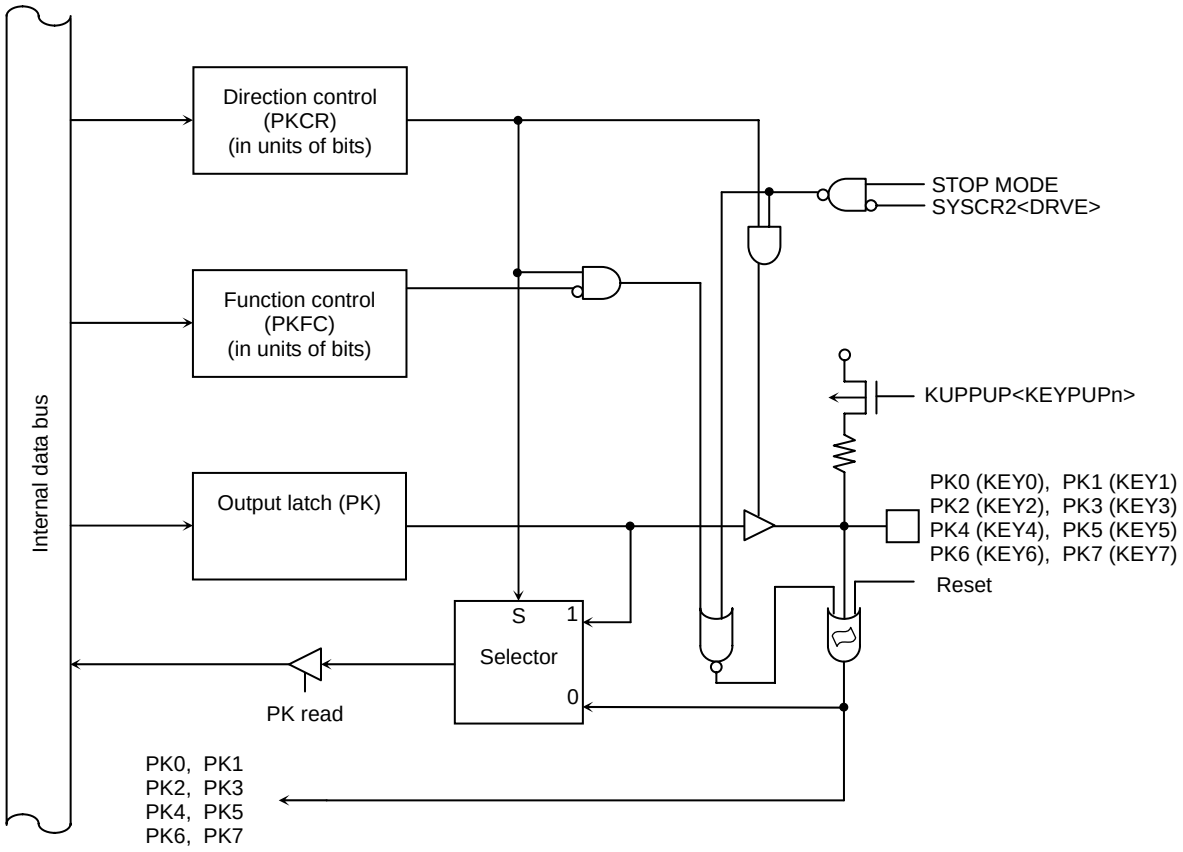


Fig. 7.19.1 Port K (PK0 through PK7)

Port K register

PK (0xFFFF_F071)	7	6	5	4	3	2	1	0
Bit Symbol	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port K control register

PKCR (0xFFFF_F075)	7	6	5	4	3	2	1	0
Bit Symbol	PK7C	PK6C	PK5C	PK4C	PK3C	PK2C	PK1C	PK0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port K function register

PKFC (0xFFFF_F079)	7	6	5	4	3	2	1	0
Bit Symbol	PK7F	PK6F	PK5F	PK4F	PK3F	PK2F	PK1F	PK0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: KEY7	0: PORT 1: KEY6	0: PORT 1: KEY5	0: PORT 1: KEY4	0: PORT 1: KEY3	0: PORT 1: KEY2	0: PORT 1: KEY1	0: PORT 1: KEY0

Function	Corresponding BIT of PKFC	Corresponding BIT of PKCR	PORT to be used
KEY0 input setting	1	0	PK0
KEY1 input setting	1	0	PK1
KEY2 input setting	1	0	PK2
KEY3 input setting	1	0	PK3
KEY4 input setting	1	0	PK4
KEY5 input setting	1	0	PK5
KEY6 input setting	1	0	PK6
KEY7 input setting	1	0	PK7

(*1) This bit setting is used only if an interrupt must be generated to clear the STOP status and if SYSCR<DRVE> is set to 0. In all other cases, this bit setting does not need to be used.

Fig. 7.19.2 Port K Registers

7.20 Port L (PL0 through PL7)

The port L is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PLCR. A reset allows PLCR to be reset to "0" and the port L to function as an input port.

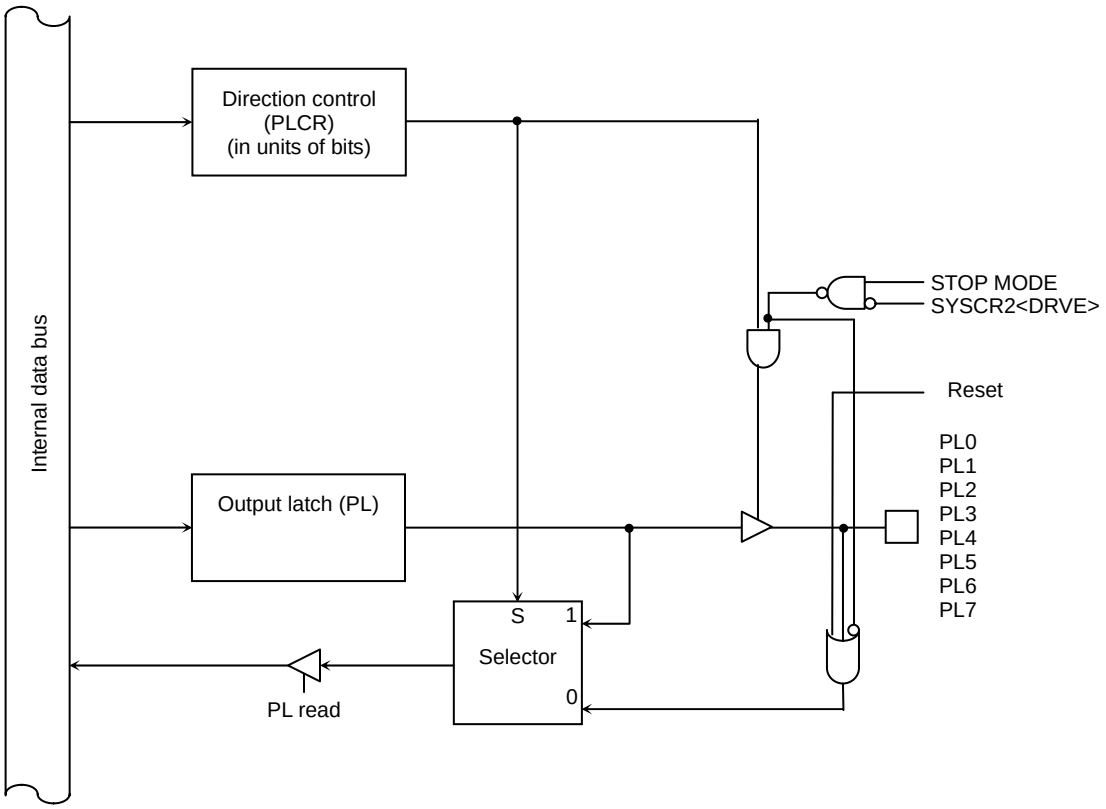


Fig. 7.20.1 Port L (PL0 through PL7)

Port L register

		7	6	5	4	3	2	1	0
PL	Bit Symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
(0xFFFF_F0C0)	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port L control register

		7	6	5	4	3	2	1	0
PLCR	Bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
(0xFFFF_F0C4)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

7.21 Port M (PM0 through PM7)

The port M is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PMCR. A reset allows PMCR to be reset to "0" and the port M to function as an input port.

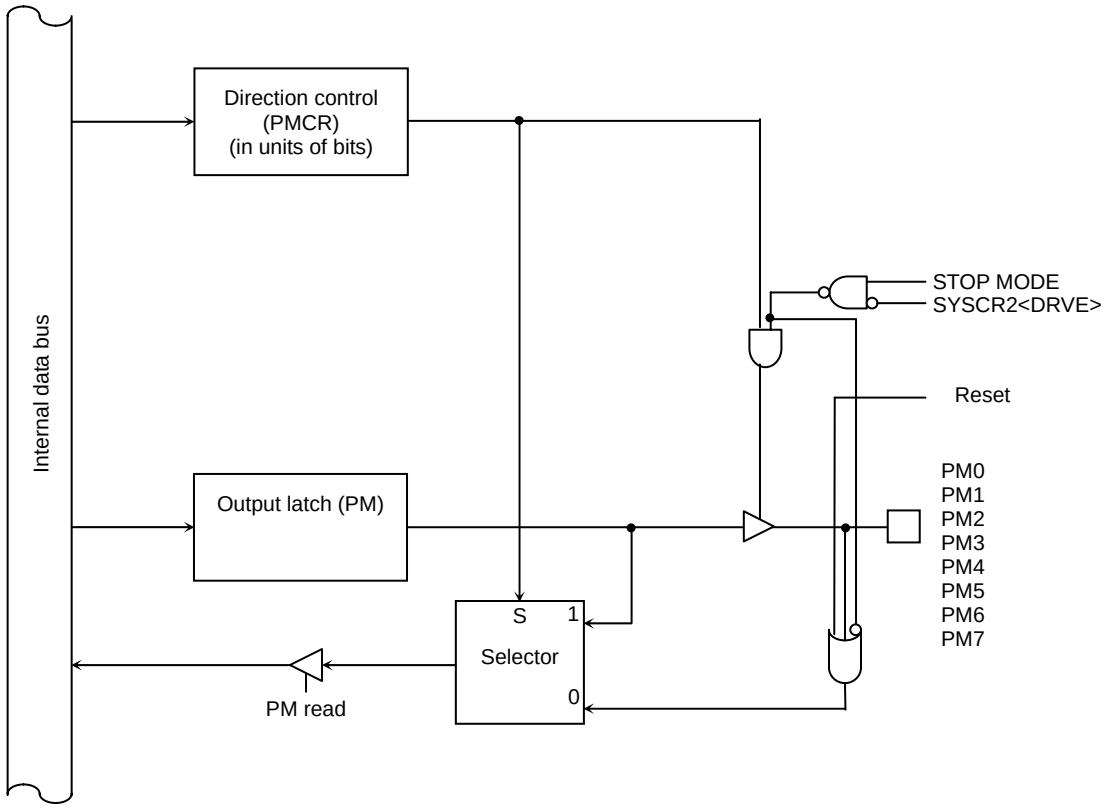


Fig. 7.21.1 Port M (PM0 through PM7)

Port M register

		7	6	5	4	3	2	1	0
PM	Bit Symbol	PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0
(0xFFFF_F0C1)	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port M control register

		7	6	5	4	3	2	1	0
PMCR	Bit Symbol	PM7C	PM6C	PM5C	PM4C	PM3C	PM2C	PM1C	PM0C
(0xFFFF_F0C5)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

7.22 Port N (PN0 through PN7)

The port N is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PNCR. A reset allows PNCR to be reset to "0" and the port N to function as an input port.

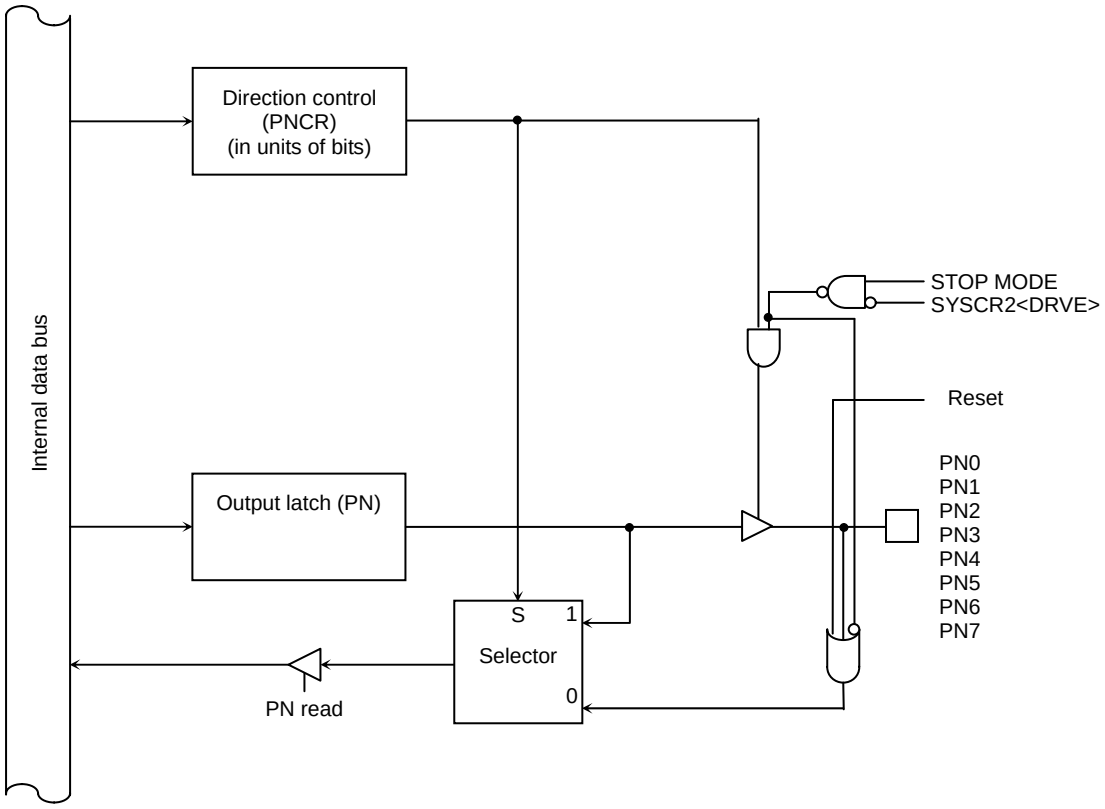


Fig. 7.22.1 Port N (PN0 through PN7)

Port N register

		7	6	5	4	3	2	1	0
PN	Bit Symbol	PN7	PN6	PN5	PN4	PN3	PN2	PN1	PN0
(0xFFFF_F0C2)	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port N control register

		7	6	5	4	3	2	1	0
PNCR	Bit Symbol	PN7C	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
(0xFFFF_F0C6)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

7.23 Port O (PO0 through PO7)

The port O is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register POCR. Besides the input/output port function, the port O performs another function: PO0 through PO4 input external interrupts. This function is enabled by setting the corresponding bit of POFC to "1." A rest allows POCR and POFC to be cleared to "0" and the port O to function as an input port. The external interrupt function is shared by PO0 through PO4 and PI0 through PI4. To give PO0 through PO4 the precedence in using the external interrupt function, the corresponding bit of POFC must be set to the interrupt function.

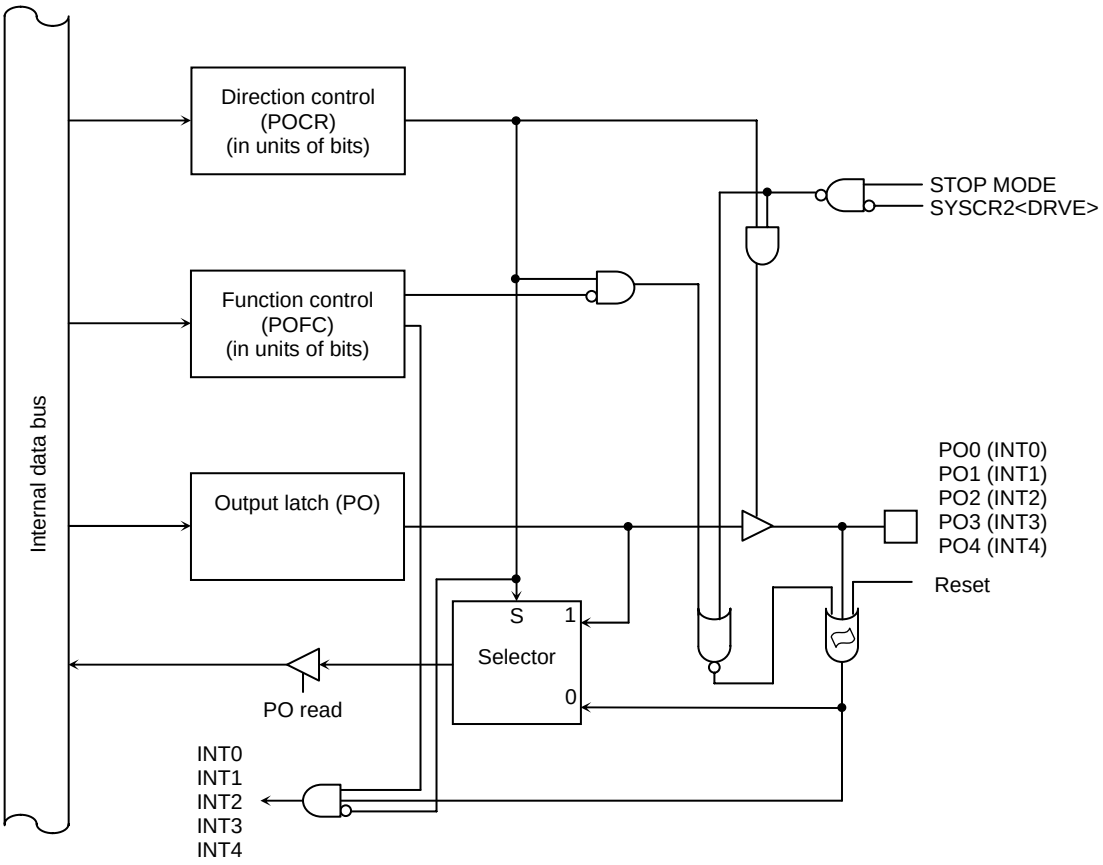


Fig. 7.23.1 Port O (PO0 through PO4)

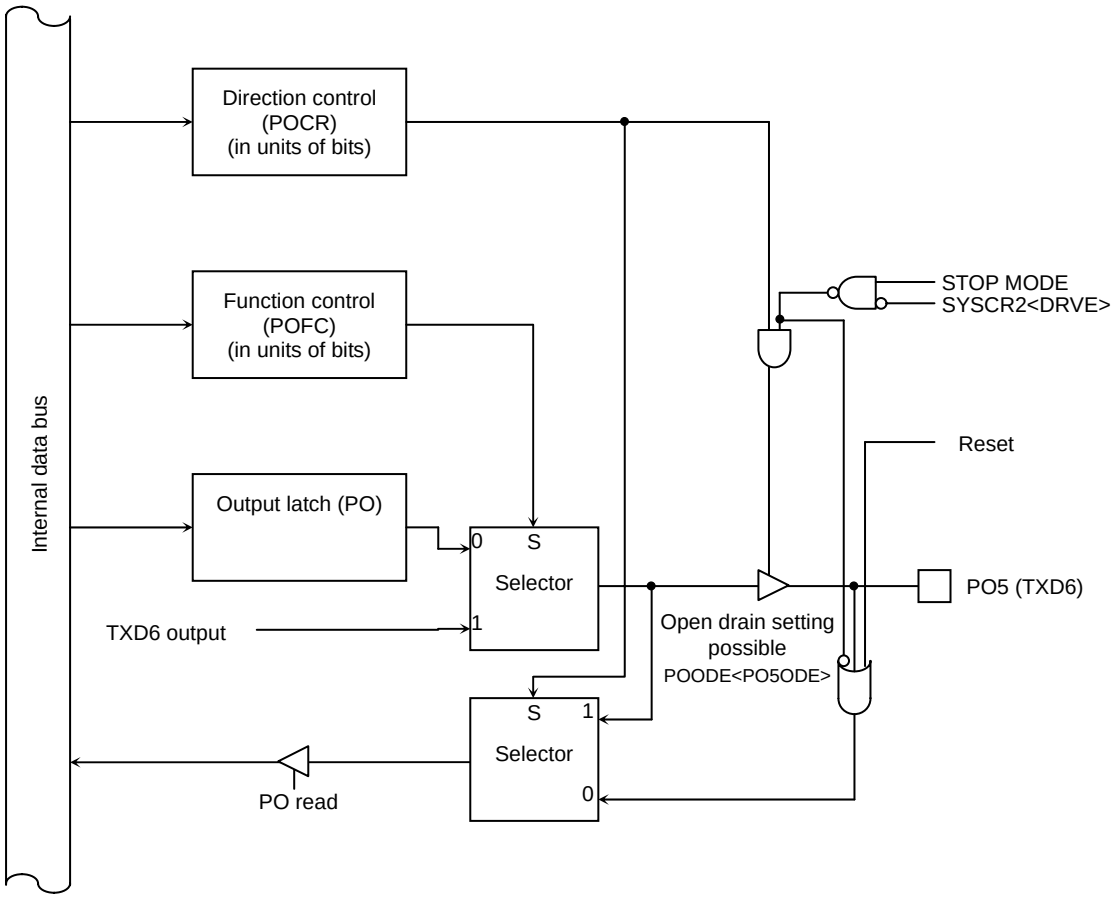


Fig. 7.23.2 Port O (PO5)

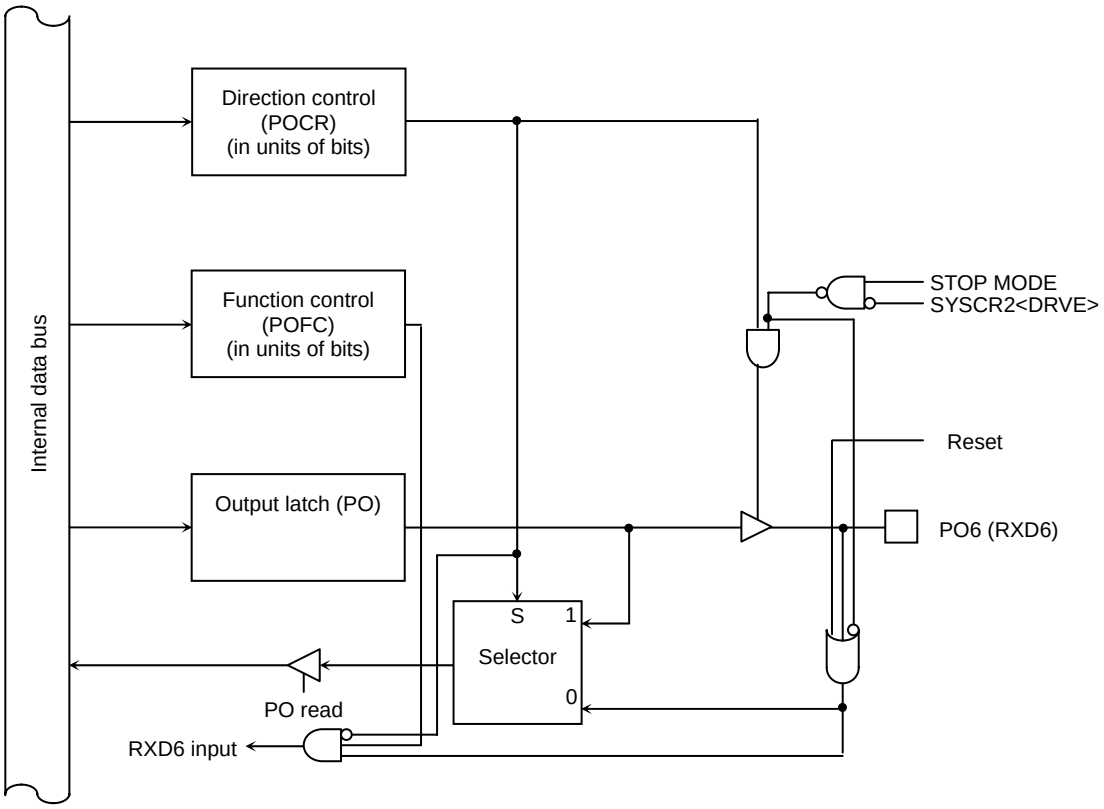


Fig. 7.23.3 Port O (PO6)

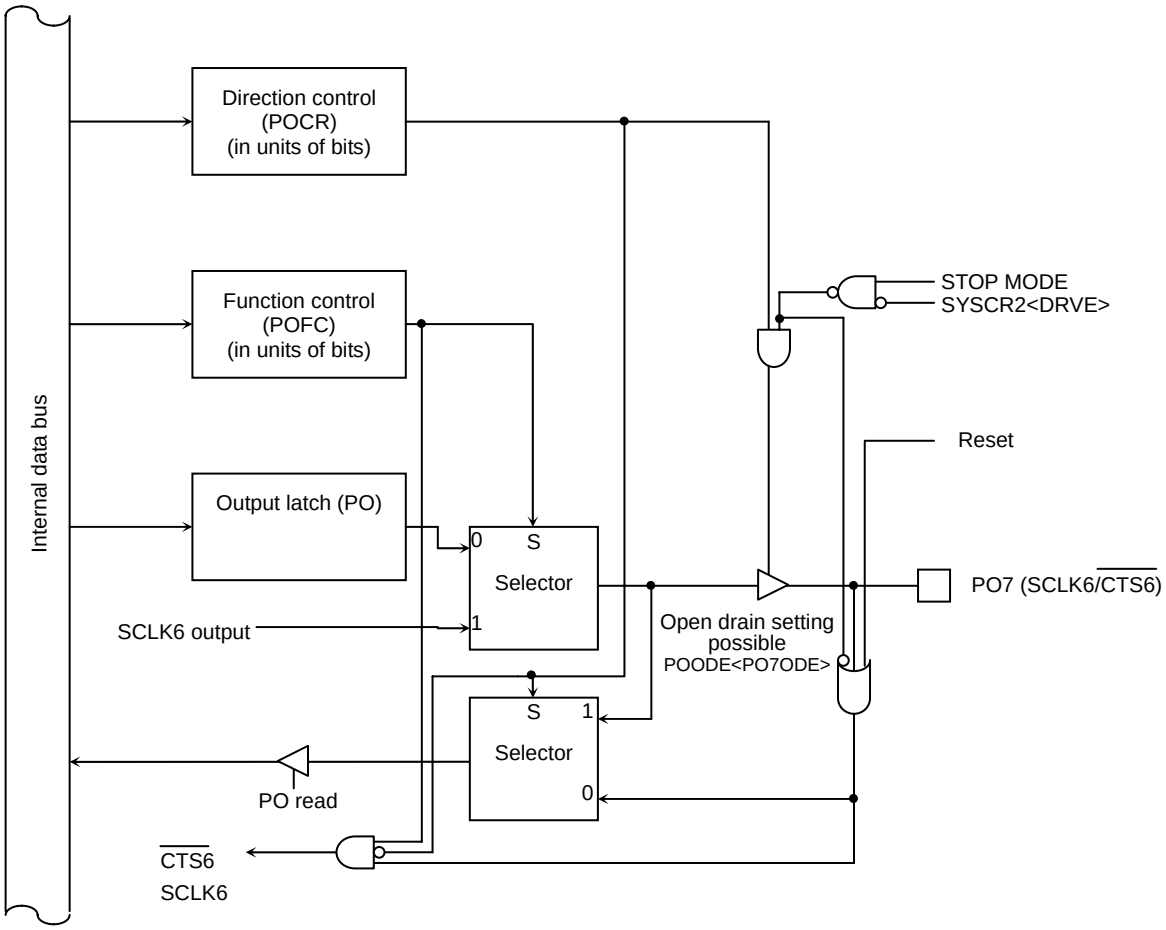


Fig. 7.23.4 Port O (PO7)

Port O register

PO (0xFFFF_F0C3)

	7	6	5	4	3	2	1	0
Bit Symbol	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port O control register

POCR (0xFFFF_F0C7)

	7	6	5	4	3	2	1	0
Bit Symbol	PO7C	PO6C	PO5C	PO4C	PO3C	PO2C	PO1C	PO0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port O function register

POFC (0xFFFF_F0CB)

	7	6	5	4	3	2	1	0
Bit Symbol				PO4F	PO3F	PO2F	PO1F	PO0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: SCLK6 CTS6	0: PORT 1: RXD6	0: PORT 1: TXD6	0: PORT 1: INT4	0: PORT 1: INT3	0: PORT 1: INT2	0: PORT 1: INT1	0: PORT 1: INT0

Port O open drain control register

POODE (0xFFFF_F0CF)

	7	6	5	4	3	2	1	0
Bit Symbol	PO7ODE		PO5ODE					
Read/Write	R/W	R	R/W	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	0: CMOS 1: Open drain	0: CMOS	0: CMOS 1: Open drain	0: CMOS	0: CMOS	0: CMOS	0: CMOS	0: CMOS

Function	Corresponding BIT of POFC	Corresponding BIT of POCR	PORT to be used
INT0 input setting	1(*1)	0	PO0
INT1 input setting	1(*1)	0	PO1
INT2 input setting	1(*1)	0	PO2
INT3 input setting	1(*1)	0	PO3
INT4 input setting	1(*1)	0	PO4
TXD6 output setting	1	1	PO5
RTD6 input setting	1	0	PO6
SCLK6 output setting	1	1	PO7
SCLK6 input setting	1	0	
CTS6 input setting	1	0	

(*1) This bit setting is used only if an interrupt must be generated to clear the STOP status and if SYSCR<DRVE> is set to 0. In all other cases, this bit setting does not need to be used.

(Note) The external interrupt function is shared by the port 1 and the port 0. If both ports are set to use the external interrupt function, the port 0 is given priority in using the external interrupt function.

Fig. 7.23.5 Port O Registers

7.24 Port P (PP0 through PP7)

The port P is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PPCR. Besides the input/output port function, the port P performs another function: PP0 through PP7 output the signal for EJTAG. This function is enabled by a combination of the EJTAG debug level and the corresponding bit of PPFC. A reset allows PPCR and PPFC to be cleared to "0" and the port P to function as an input port.

If DSU-ICE is used for debugging, the port P outputs the signal for EJTAG. Therefore, it is recommended not to use the port P as an input/output port.

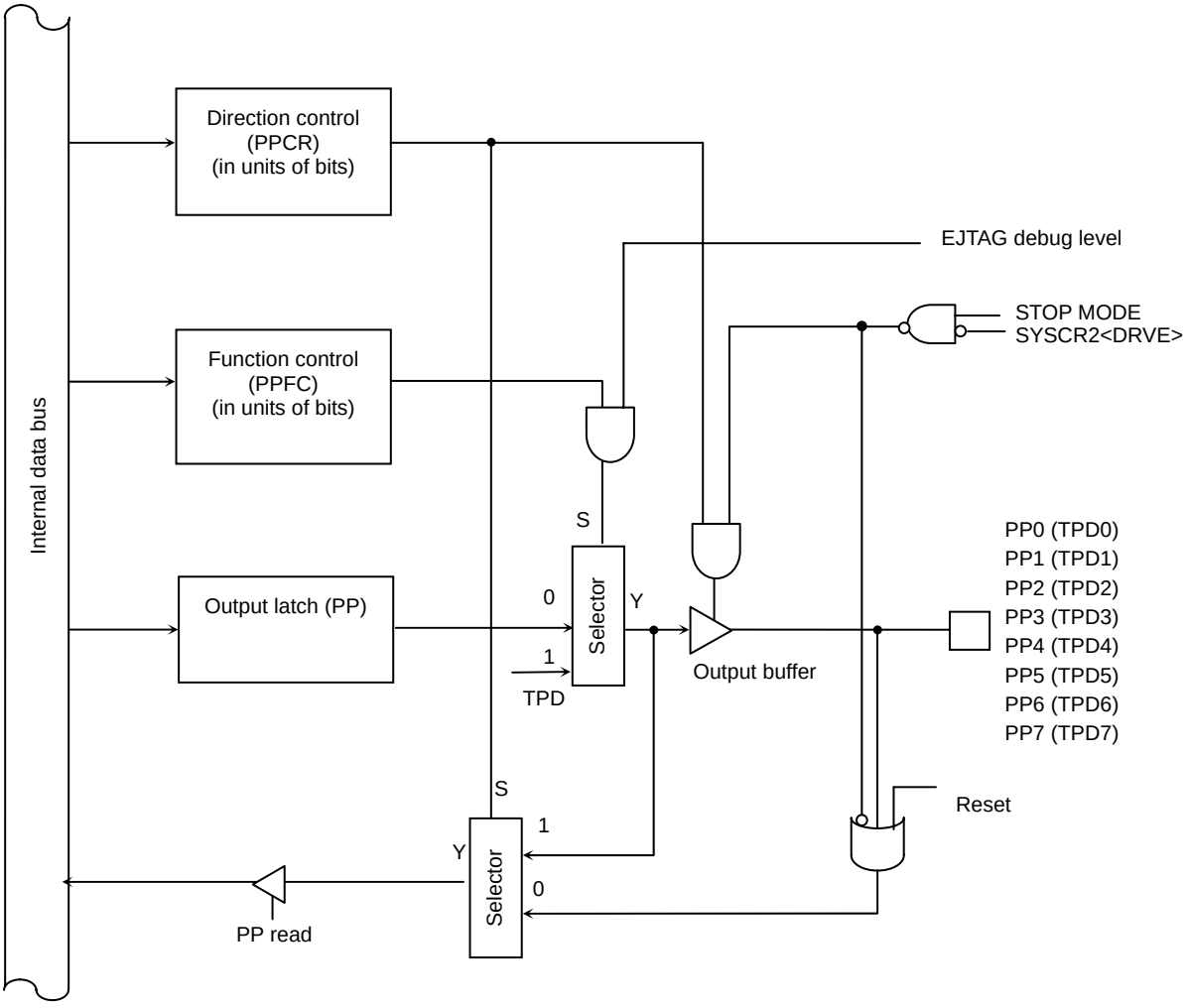


Fig. 7.24.1 Port P (PP0 through PP7)

(Note) The above system diagram does not show the debug function.

Port P register

PP (0xFFFF_F0D0)	7	6	5	4	3	2	1	0
Bit Symbol	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port P control register

PPCR (0xFFFF_F0D4)	7	6	5	4	3	2	1	0
Bit Symbol	PP7C	PP6C	PP5C	PP4C	PP3C	PP2C	PP1C	PP0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port P function register

PPFC (0xFFFF_F0D8)	7	6	5	4	3	2	1	0
Bit Symbol	PP7F	PP6F	PP5F	PP4F	PP3F	PP2F	PP1F	PP0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: TPD7/TPC7	0: PORT 1: TPD6/TPC6	0: PORT 1: TPD5/TPC5	0: PORT 1: TPD4/TPC4	0: PORT 1: TPD3/TPC3	0: PORT 1: TPD2/TPC2	0: PORT 1: TPD1/TPC1	0: PORT 1: TPD0/TPC0

Fig. 7.24.2 Port P Registers

Note) If the port P or the port Q is used to generate the output signal for EJTAG, a necessary port P or Q setting must be made using a tool. The PPFC register setting must be made in units of bites.

	Level 0	Level 1	Level 2		Level 3
			PPFC=#FF	PPFC≠#FF	
PORT P	PORT	PORT	TPD	PORT	TPD
PORT Q	PORT	TPC	PORT	TPD	TPC

Fig. 7.24.3 Ports P and Q function relative to debug levels

Note) For information on debug levels and other details, refer to the DSU Probe Handling Manual.

7.25 Port Q (PQ0 through PQ7)

The port Q is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PQCR. Besides the input/output port function, PQ0 through PQ7 output the signal for EJTAG. This function is enabled by a combination of a debug level and the corresponding bit of PPFC. A reset allows PQCR and PPFC to be cleared to "0" and the port Q to function as an input port.

If DSU-ICE is used for debugging, the port Q outputs the signal for EJTAG. Therefore, it is recommended not to use the port Q as an input/output port.

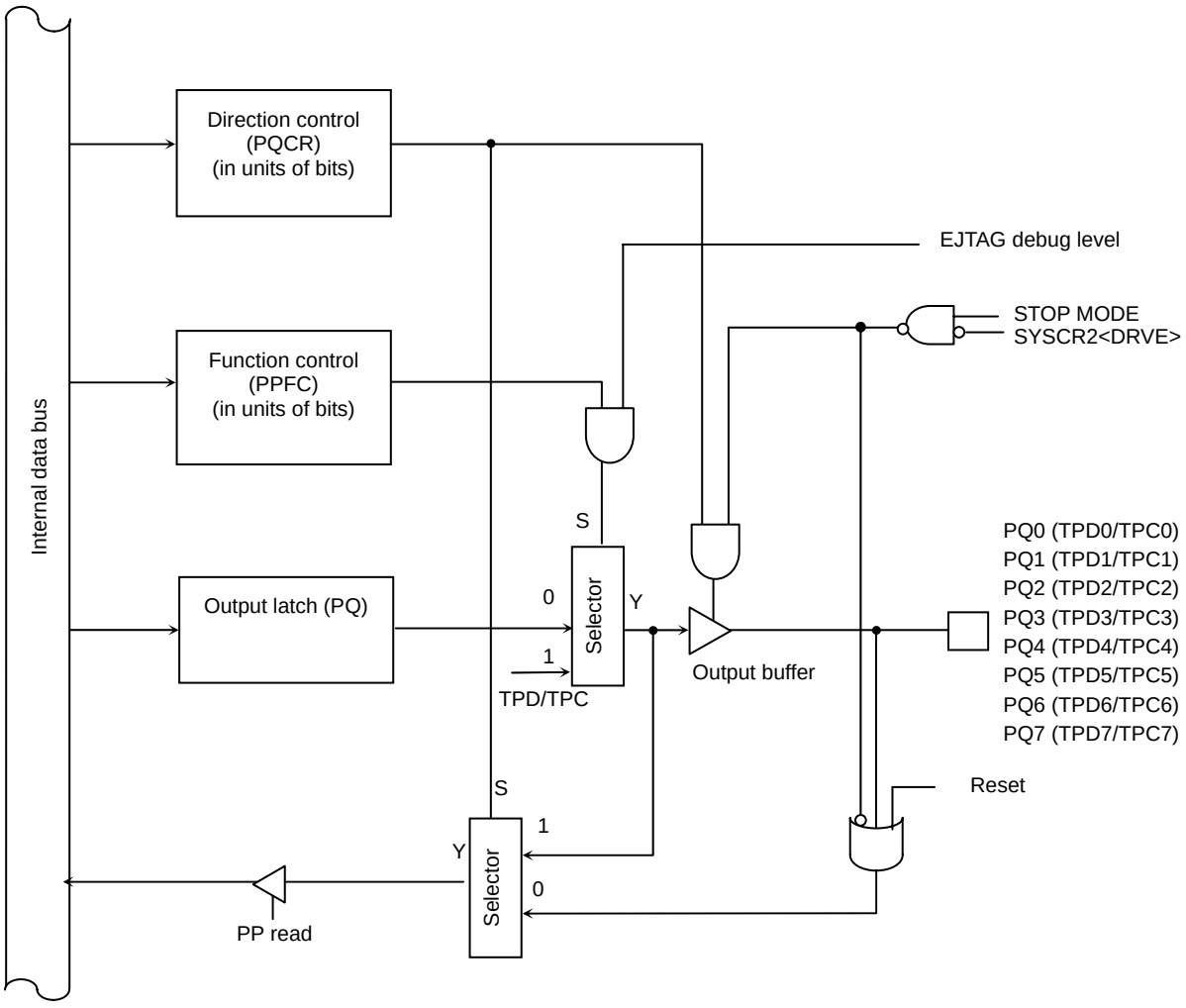


Fig. 7.25.1 Port Q (PQ0 through PQ7)

(Note) The above system diagram does not show the debug function.

Port Q register

		7	6	5	4	3	2	1	0
PQ	Bit Symbol	PQ7	PQ6	PQ5	PQ4	PQ3	PQ2	PQ1	PQ0
(0xFFFF_F0D1)	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port Q control register

		7	6	5	4	3	2	1	0
PQCR	Bit Symbol	PQ7C	PQ6C	PQ5C	PQ4C	PQ3C	PQ2C	PQ1C	PQ0C
(0xFFFF_F0D5)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Fig. 7.25.2 Port Q Registers

8. External Bus Interface

The TMP19A64 has a built-in external bus interface function to connect to external memory, I/Os, etc. This interface consists of an external bus interface circuit (EBIF), a chip selector (CS) and a wait controller.

The chip selector and wait controller designate mapping addresses in a 6-block address space and also control wait states and data bus widths (8- or 16-bit) in these and other external address spaces.

The external bus interface circuit (EBIF) controls the timing of external buses based on the chip selector and wait controller settings. The EBIF also controls the dynamic bus sizing and the bus arbitration with the external bus master.

- External bus mode
  Selectable address, data separator bus mode and multiplex mode
- Wait function
  This function can be enabled for each block.
  - A wait of up to 7 clocks can be automatically inserted.
  - A wait can be inserted via the $\overline{\text{WAIT}}$ / $\overline{\text{RDY}}$ pin.
- Data bus width
  Either an 8- or 16-bit width can be set for each block.
- Recovery cycle (read/write)
  If an external bus cycle is in progress, a dummy cycle of up to 2 clocks can be inserted and this dummy cycle can be specified for each block.
- Recovery cycle (chip selector)
  When an external bus is selected, a dummy cycle of up to 1 clock can be inserted and this dummy cycle can be specified for each block.
- Bus arbitration function

8.1 Address and Data Pins

(1) Address and data pin settings

The TMP19A64 can be set to either separate bus or multiplexed bus mode. Setting the BUSMD pin to the "L" level at a reset activates the separate bus mode, and setting the pin to the "H" level activates the multiplexed bus mode. Port pins 0, 1, 2, 5 and 6, which are to be connected to external devices (memory), are used as address buses, data buses and address/data buses. Table 8.1.1 shows these.

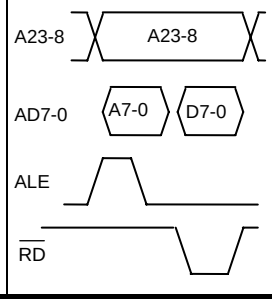
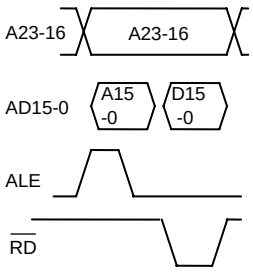
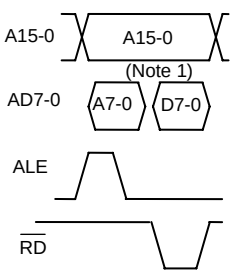
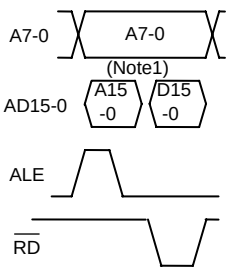
Table 8.1.1 Bus Mode, Address and Data Pins

	Separate BUSMD="L"	**Multiplex BUSMD="H"**
Port 0 (P00 to P07)	D0-D7	AD0-AD7
Port 1 (P10 to P17)	D8-D15	AD8-AD15/A8-A15
Port 2 (P20 to P27)	A16-A23	A0-A7/A16-A23
Port 5 (P50 to P57)	A0-A7	General-purpose port
Port 6 (P60 to P67)	A8-A15	General-purpose port
Port 37 (P37)	General-purpose port	ALE

Each port is put into input mode after a reset. To access an external device, set the address and data bus functions by using the port control register (PnCR) and the port function register (PnFC).

In the multiplex mode, the four types of functions can be selected, as shown in Table 8.1.2, by setting the port registers (PnCR and PnFC).

Table 8.1.2 Address and Data Pins in the Multiplex Mode

		①	②	③	④
Number of address buses		max.24 (-16 MB)	max.24 (-16 MB)	max.16 (-64 KB)	max.8 (-256 B)
Number of data buses		8	16	8	16
Number of address/data multiplexed buses		8	16	0	0
Port function	Port 0	AD0 to AD7	AD0 to AD7	AD0 to AD7	AD0 to AD7
	Port 1	A8 to A15	AD8 to AD15	A8 to A15	AD8 to AD15
	Port 2	A16 to A23	A16 to A23	A0 to A7	A0 to A7
Timing Diagram					

(Note 1) Even in cases of ③ and ④, address outputs are available as the data bus pins are also used for address buses.

(Note 2) Ports 0 to 2 are put into input modes after a reset, and they do not serve as address or data bus pins.

(Note 3) Any of ① to ④ can be selected by setting the P1CR, P1FC, P2CR and P2FC registers.

(2) Address HOLD when an internal area is accessed

When an internal area is being accessed, the address bus maintains the address output of the previously accessed external area and doesn't change it. Also, the data bus is in a state of high impedance.

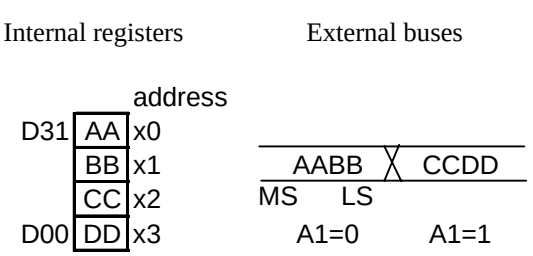
8.2 Data Format

Internal registers and external bus interfaces of the TMP19A64 are configured as described below.

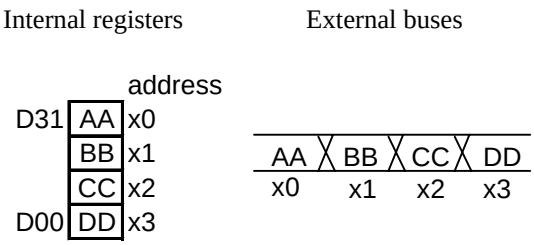
(1) Big-endian mode

① Word access

- 16-bit bus width

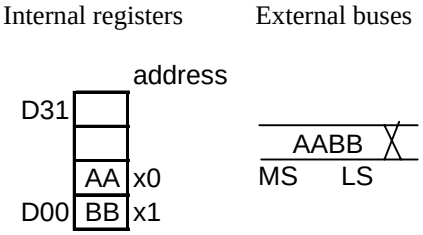


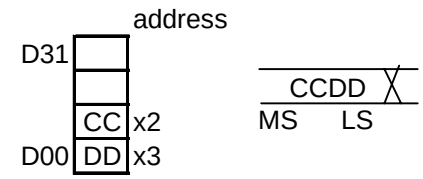
- 8-bit bus width



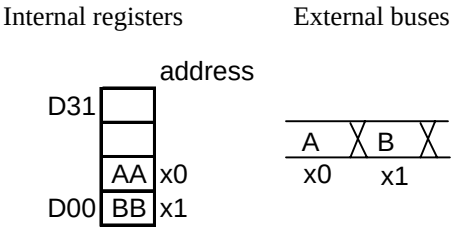
② Half word access

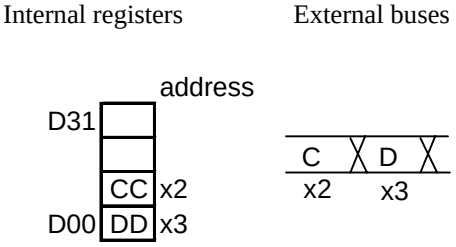
- 16-bit bus width





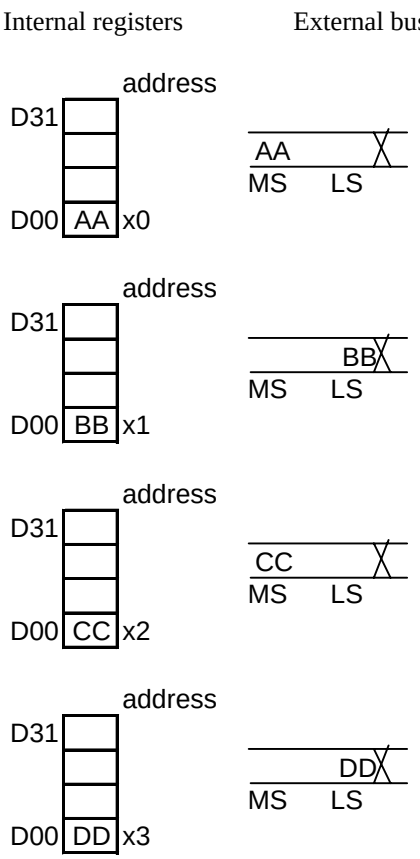
- 8-bit bus width



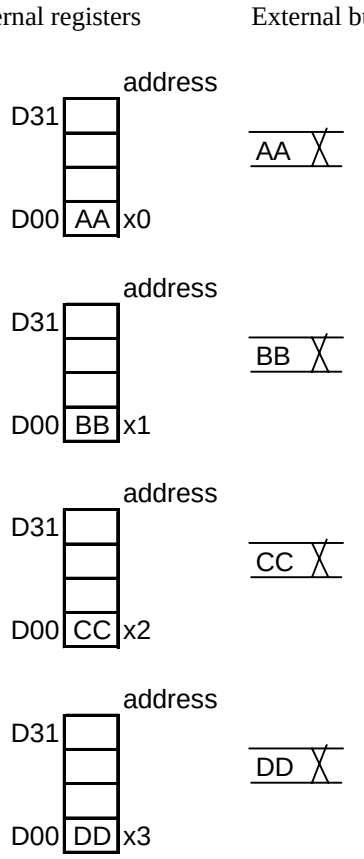


③ Byte access

- 16-bit bus width



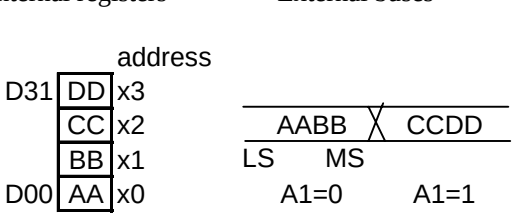
- 8-bit bus width



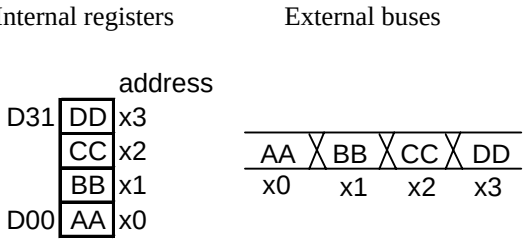
(2) Little-endian mode

① Word access

- 16-bit bus width

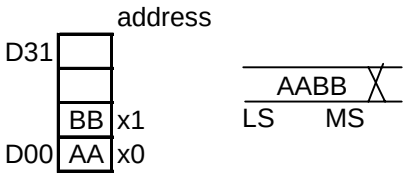


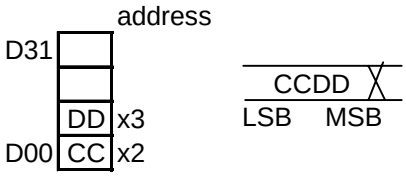
- 8-bit bus width



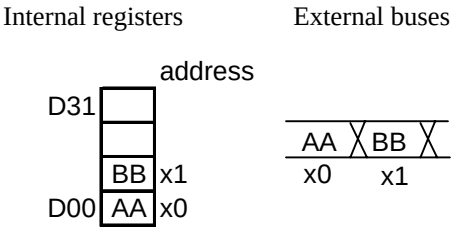
② Half word access

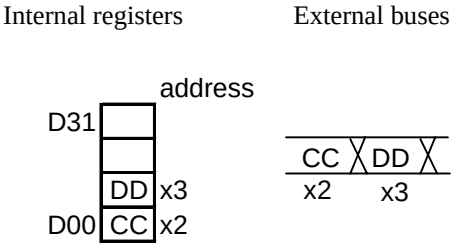
- 16-bit bus width





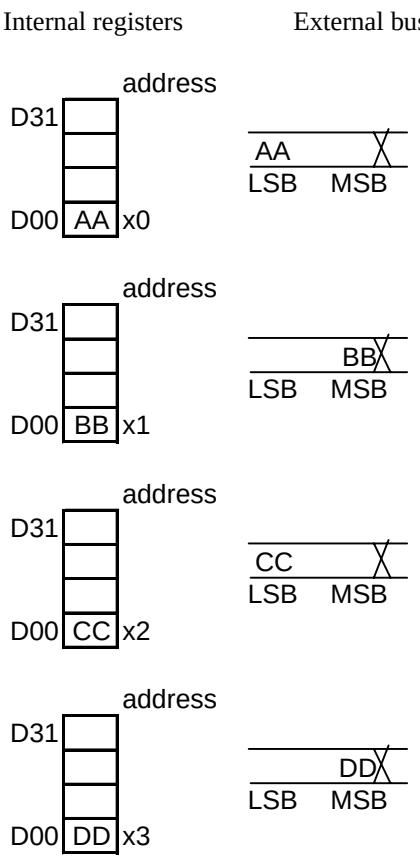
- 8-bit bus width



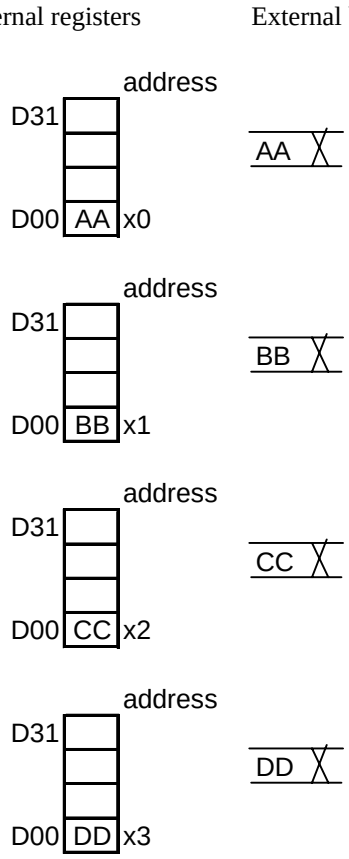


③ Byte access

- 16-bit bus width



- 8-bit bus width



8.3 External Bus Operations (Separate Bus Mode)

This section describes various bus timing values. The timing diagram shown below assumes that the address buses are A23 through A0 and that the data buses are D15 through D0.

(1) Basic bus operation

The external bus cycle of the TMP19A64 basically consists of three clock pulses and a wait can be inserted as mentioned later. The basic clock of an external bus cycle is the same as the internal system clock.

Fig. 8.3.1 shows read bus timing and Fig. 8.3.2 shows write bus timing. If internal areas are accessed, address buses remain unchanged as shown in these figures. Additionally, data buses are in a state of high impedance and control signals such as $\overline{RD}$ and $\overline{WR}$ do not become active.

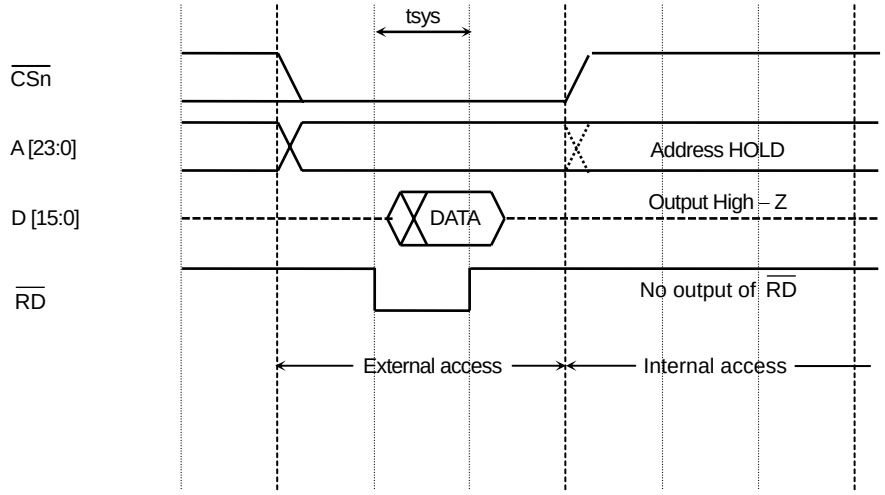


Fig. 8.3.1 Read Operation Timing Diagram

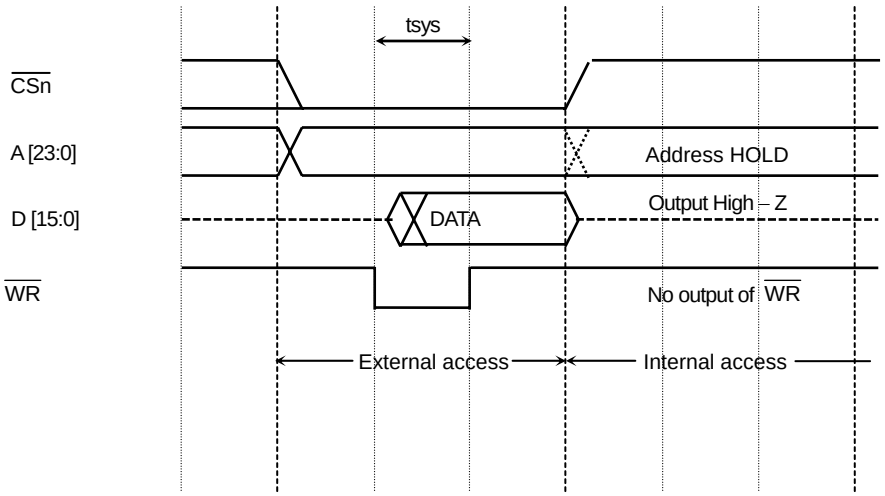


Fig. 8.3.2 Write Operation Timing Diagram

(2) Wait timing

A wait cycle can be inserted for each block by using the chip selector (CS) and wait controller.

The following three types of wait can be inserted:

① A wait of up to 7 clocks can be automatically inserted.

② A wait can be inserted via the $\overline{\text{WAIT}}$ pin (2+2N, 3+2N, 4+2N, 5+2N, 6+2N, 7+2N). Note: 2N is the number of external waits that can be inserted.

③ A wait can be inserted via the $\overline{\text{RDY}}$ pin (2+2N, 3+2N, 4+2N, 5+2N, 6+2N, 7+2N). Note: 2N is the number of external waits that can be inserted.

The setting of the number of waits to be automatically inserted and the setting of the external wait input can be made using the chip selector and wait controller registers, BmnCS<BnW>.

Fig. 8.3.3 through Fig. 8.3.10 show the timing diagrams in which waits have been inserted.

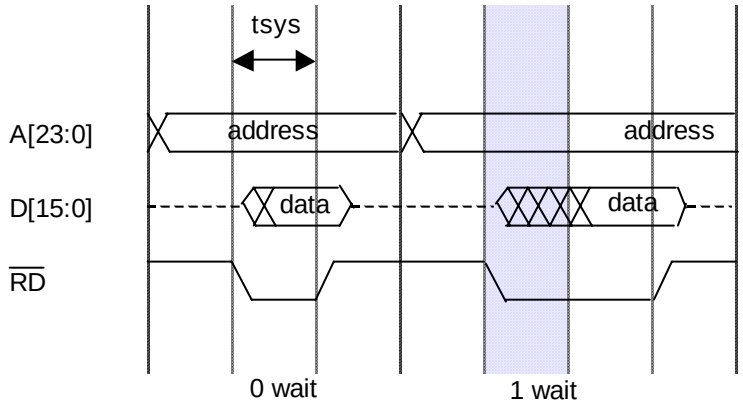


Fig. 8.3.3 Read Operation Timing Diagram (0 Wait and 1 Wait Automatically Inserted)

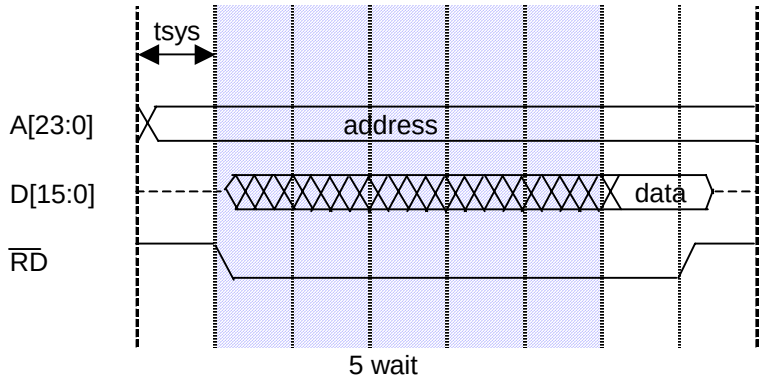


Fig. 8.3.4 Read Operation Timing Diagram (5 Waits Automatically Inserted)

Fig. 8.3.5 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the separate bus mode.

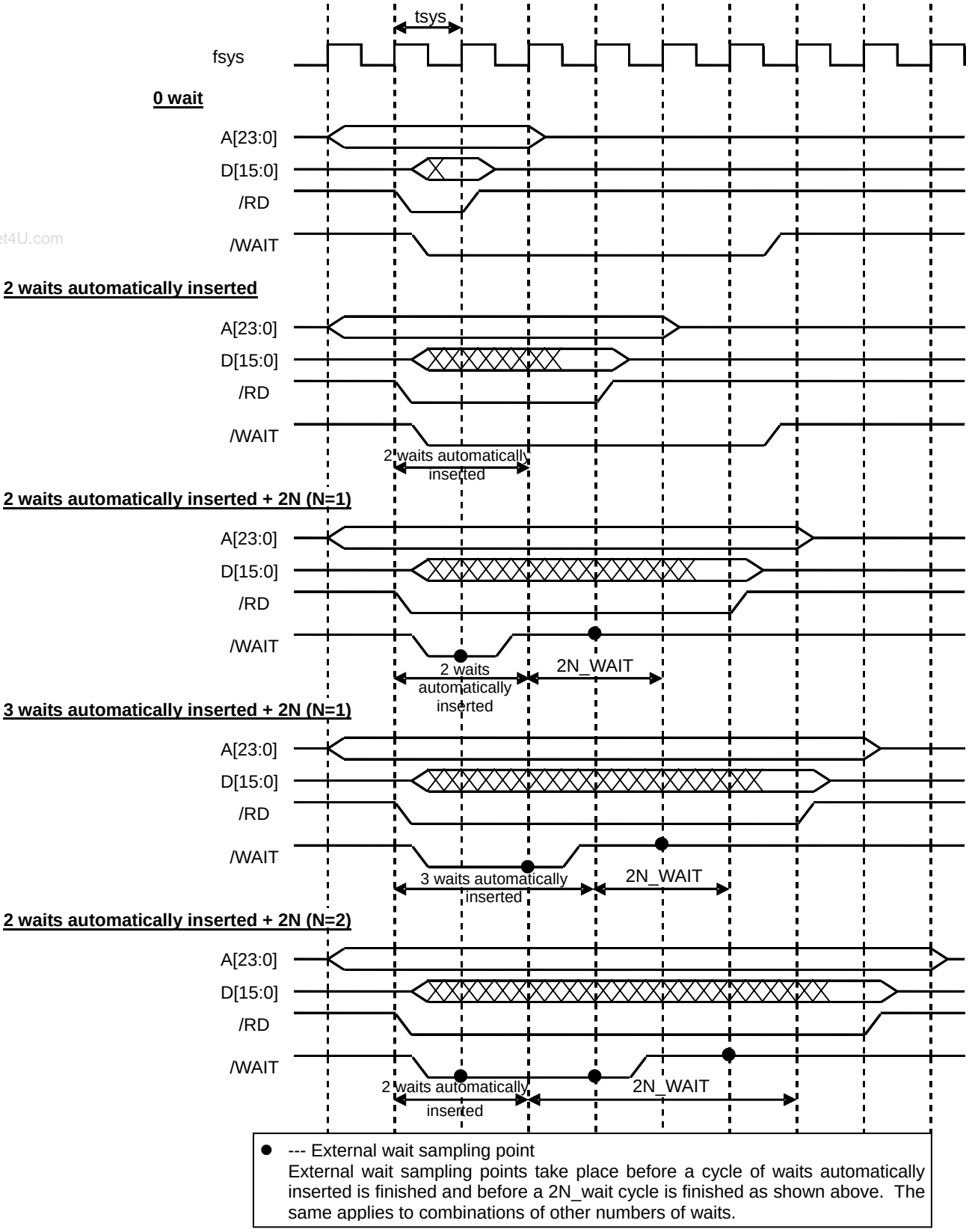


Fig. 8.3.5 Read Operation Timing Diagram

Fig. 8.3.6 shows the write operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the separate bus mode.

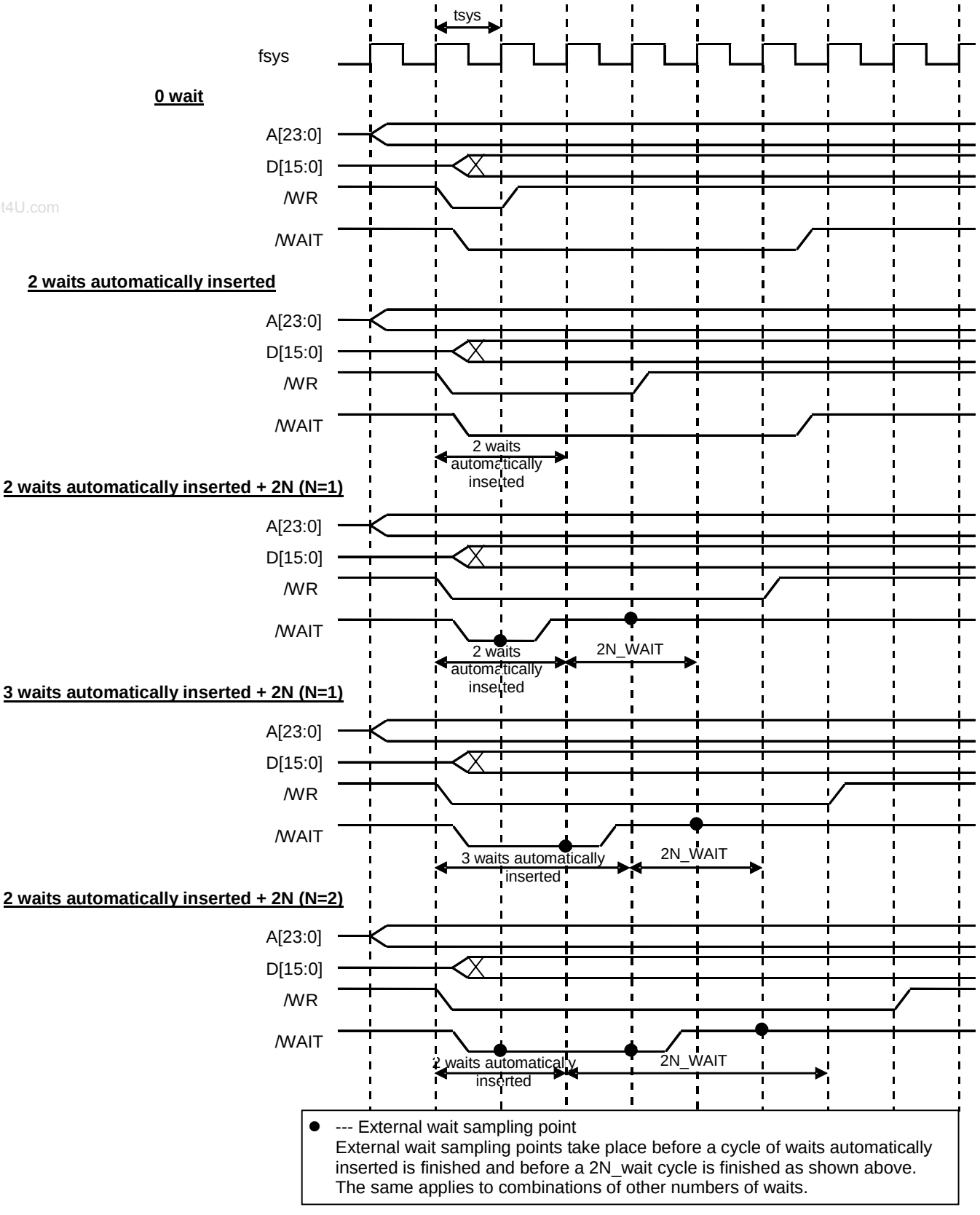


Fig. 8.3.6 Write Operation Timing Diagram

By setting the bit 3<P33F> of port 3 function register P3FC to "1," the $\overline{\text{WAIT}}$ input pin (P33) can also serve as the $\overline{\text{RDY}}$ input pin.

The $\overline{\text{RDY}}$ input is input to the external bus interface circuit as the logical reverse of the $\overline{\text{WAIT}}$ input. The number of waits is specified by the chip selector and wait controller register, BmnCS<BnW>.

Fig. 8.3.7 shows the $\overline{\text{RDY}}$ inputs and the number of waits.

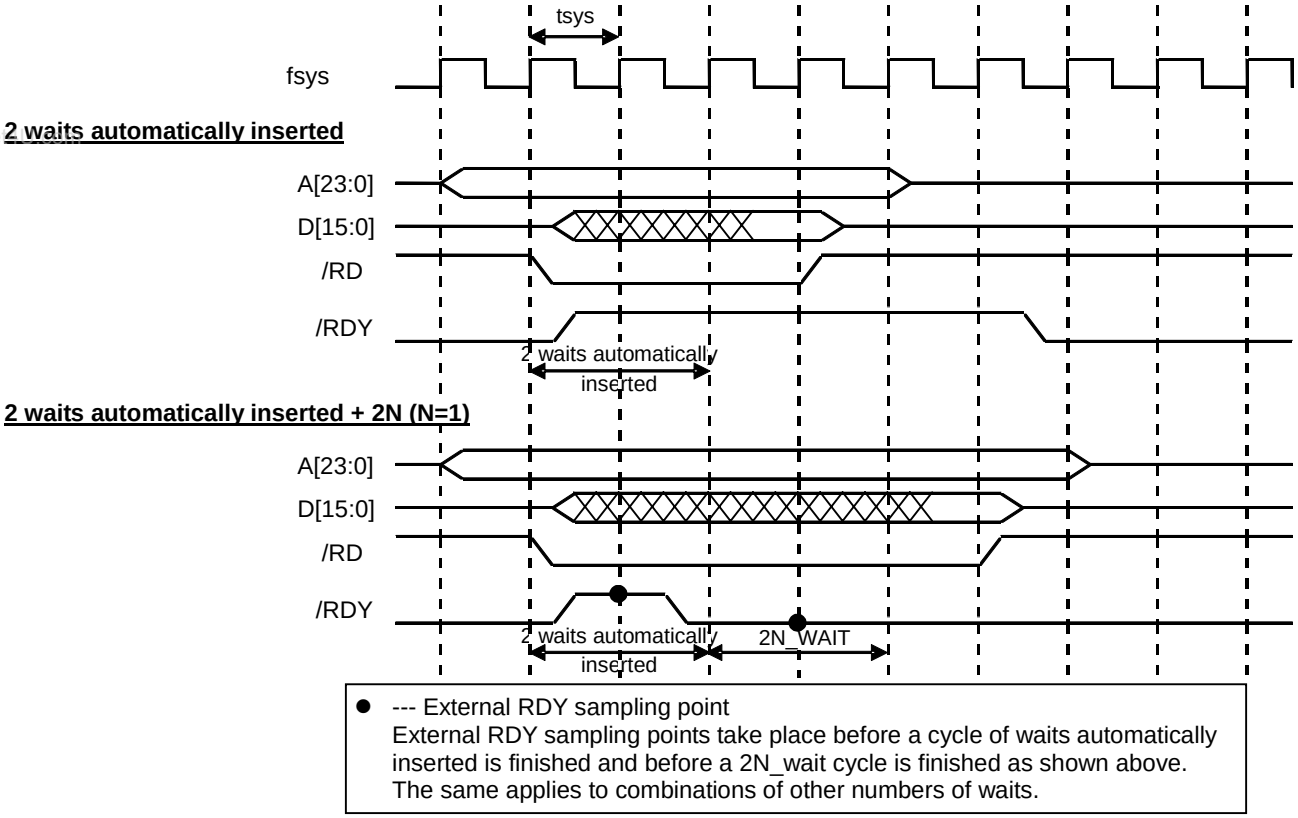


Fig. 8.3.7 $\overline{\text{RDY}}$ Input and Wait Operation Timing Diagram

(3) Time that it takes before ALE is asserted

When the external bus of the TMP19A64 is used as a multiplexed bus, the ALE width (assert time) can be specified by using the system control register SYSCR3 <ALESEL> in the CG. In the case of a separate bus mode, ALE is not output, but the time from when an address is established to the assertion of the $\overline{RD}$ or $\overline{WR}$ signal is different depending on the SYSCR3<ALESEL>.

During a reset, <ALESEL> = "1" is set and the $\overline{RD}$ or $\overline{WR}$ signal is asserted as a point of two system (internal) clocks after an address is established. If <ALESEL> is cleared to "0," the $\overline{RD}$ or $\overline{WR}$ signal is asserted at a point of one system (internal) clock after an address is established. This assert setting cannot be established for each block in an external area and the same setting is commonly used in an external address space.

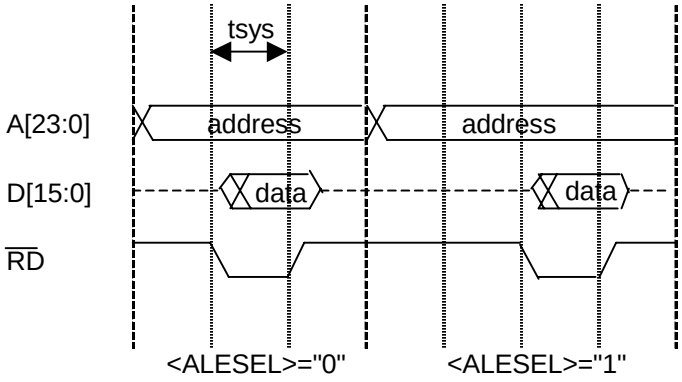


Fig. 8.3.13 SYSCR3<ALESEL> Set Value and External Bus Operation

(4) Recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip selector and wait controller registers, BmnCS<BnWCV> (write recovery cycle) and <BnRCV> (read recovery cycle). As for the number of dummy cycles, one or two system clocks (internal) can be specified for each block. Fig. 8.3.14 shows the timing of recovery time insertion.

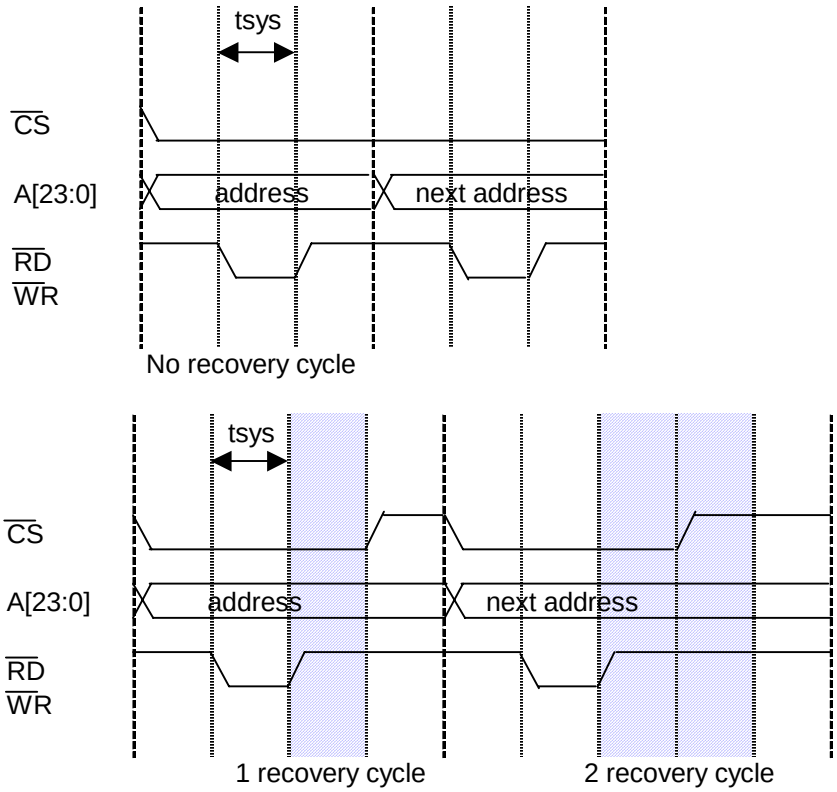
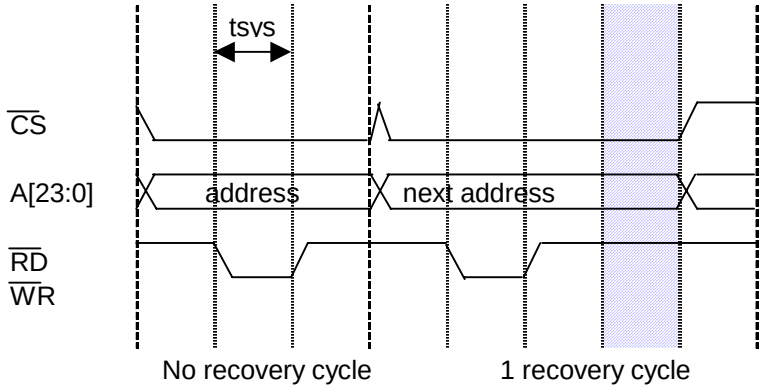


Fig. 8.3.14 Timing of Recovery Time Insertion

(5) Chip selector recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

The dummy cycle insertion setting can be made in the chip selector and wait controller registers, BmnCS<BnCSCV>. As for the number of dummy cycles, one system clock (internal) can be specified for each block. Fig. 8.3.15 shows the timing of recovery time insertion.



8.4 External Bus Operations (Multiplexed Bus Mode)

This section describes various bus timing values. The timing diagram shown below assumes that the address buses are A23 through A16 and that the address/data buses are AD15 through AD0.

(1) Basic bus operation

The external bus cycle of the TMP19A64 basically consists of three clock pulses and a wait can be inserted as mentioned later. The basic clock of an external bus cycle is the same as the internal system clock.

Fig. 8.4.1 shows read bus timing and Fig. 8.4.2 shows write bus timing. If internal areas are accessed, address buses remain unchanged and the ALE does not output latch pulse as shown in these figures. Additionally, address/data buses are in a state of high impedance and control signals such as $\overline{RD}$ and $\overline{WR}$ do not become active.

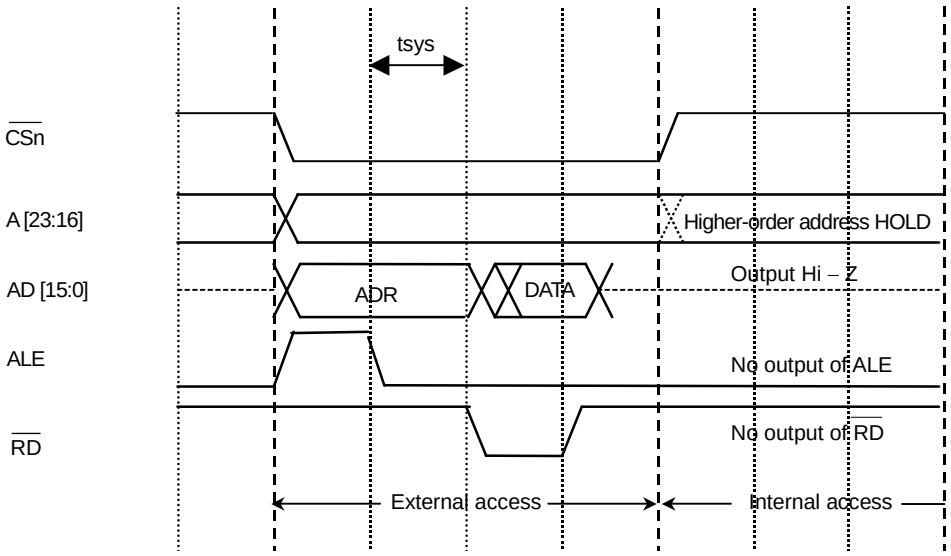


Fig. 8.4.1 Read Operation Timing Diagram

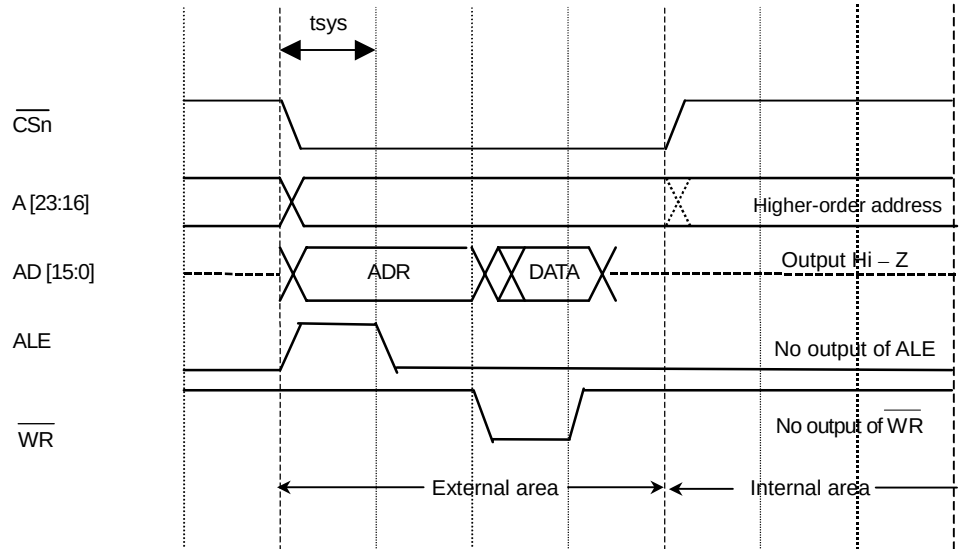


Fig. 8.4.2 Write Operation Timing Diagram

(2) Wait Timing

A wait cycle can be inserted for each block by using the chip selector (CS) and wait controller.

The following three types of wait can be inserted:

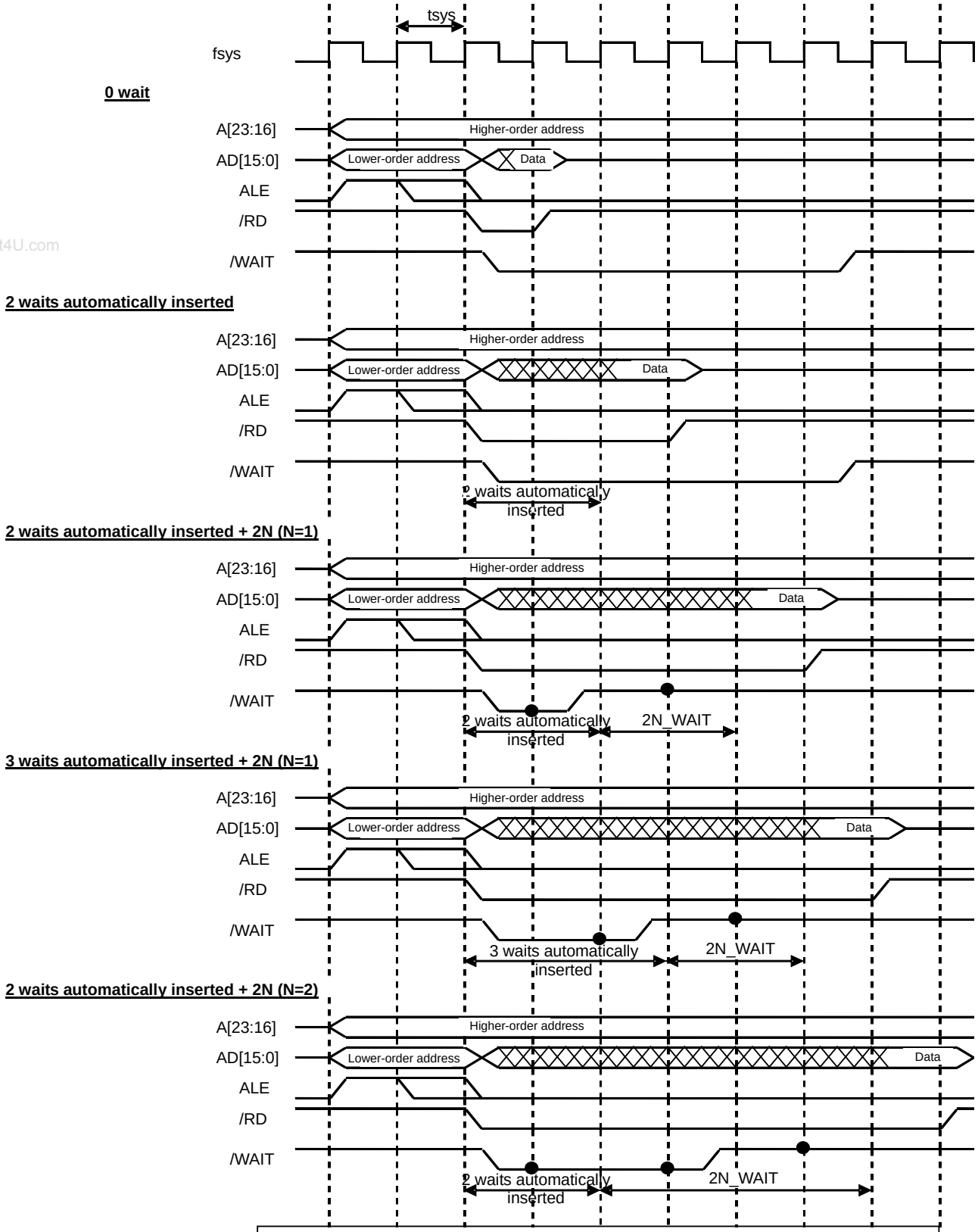
① A wait of up to 7 clocks can be automatically inserted.

② A wait can be inserted via the $\overline{\text{WAIT}}$ pin (2+2N, 3+2N, 4+2N, 5+2N, 6+2N, 7+2N).
Note: 2N is the number of external waits that can be inserted.

③ A wait can be inserted via the $\overline{\text{RDY}}$ pin (2+2N, 3+2N, 4+2N, 5+2N, 6+2N, 7+2N).
Note: 2N is the number of external waits that can be inserted.

The setting of the number of waits to be automatically inserted and the setting of the external wait input can be made using the chip selector and wait controller registers, BmnCS<BnW>.

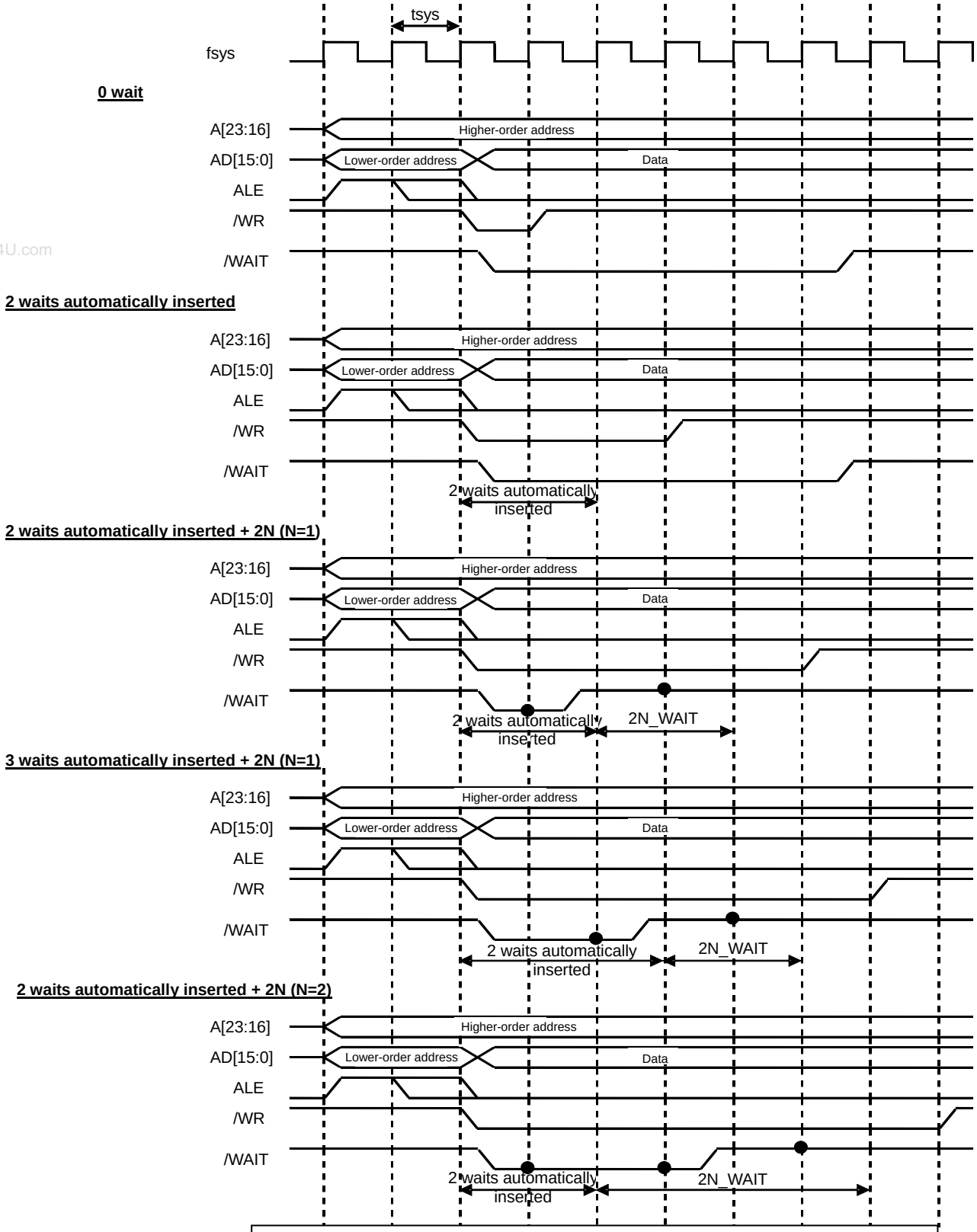
Fig. 8.4.3 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the multiplexed bus mode.



- --- External wait sampling point
  External wait sampling points take place before a cycle of waits automatically inserted is finished and before a 2N_wait cycle is finished as shown above. The same applies to combinations of other numbers of waits.

Fig. 8.4.3 Read Operation Timing Diagram

Fig. 8.4.4 shows the write operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the multiplexed bus mode.



- --- External wait sampling point
  External wait sampling points take place before a cycle of waits automatically inserted is finished and before a 2N_wait cycle is finished as shown above. The same applies to combinations of other numbers of waits.

Fig. 8.4.4 Write Operation Timing Diagram

(3) Time that it takes before ALE is asserted

Either 1 clock or 2 clocks can be selected as the time that it takes before ALE is asserted. The setting bit is located in the system clock control register. The default is 2 clocks. This assert setting cannot be established for each block in an external area and the same setting is commonly used in an external address space.

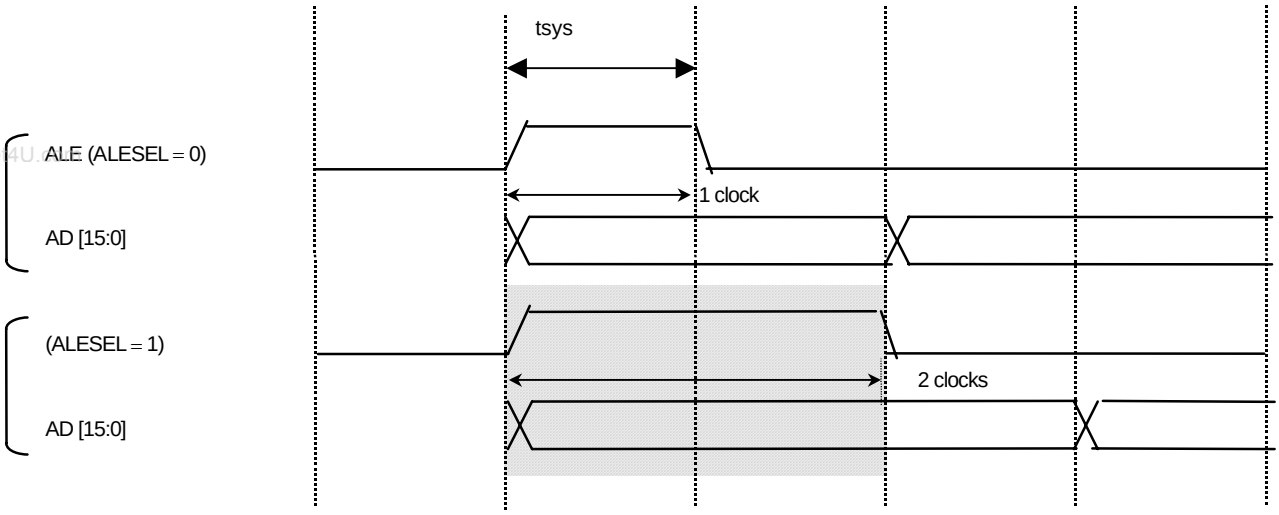


Fig. 8.4.12 Time That It Takes Before ALE Is Asserted

Fig. 8.4.13 shows the timing when the ALE is 1 clock or 2 clocks.

When the ALE is 1 clock or 2 clocks

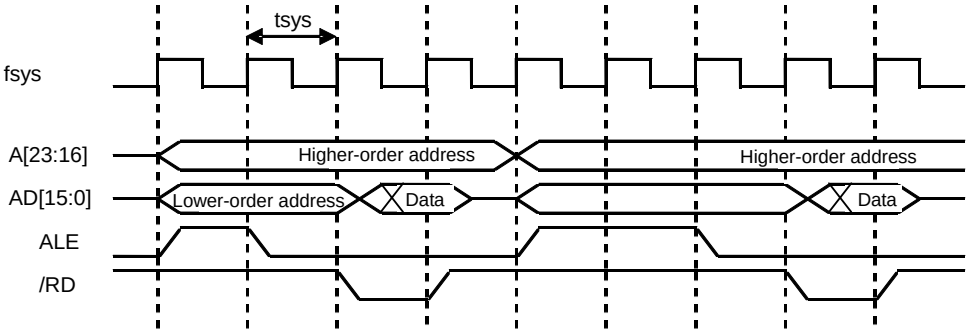


Fig. 8.4.13 Read Operation Timing Diagram (When the ALE is 1 Clock or 2 Clocks)

(4) Read and Write Recovery Time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip selector and wait controller registers, BmnCS<BnWCV> (write recovery cycle) and <BnRCV> (read recovery cycle). As for the number of dummy cycles, one or two system clocks (internal) can be specified for each block. Fig. 8.4.14 shows the timing of recovery time insertion.

When read/write recovery is inserted (ALE width:1fsys)

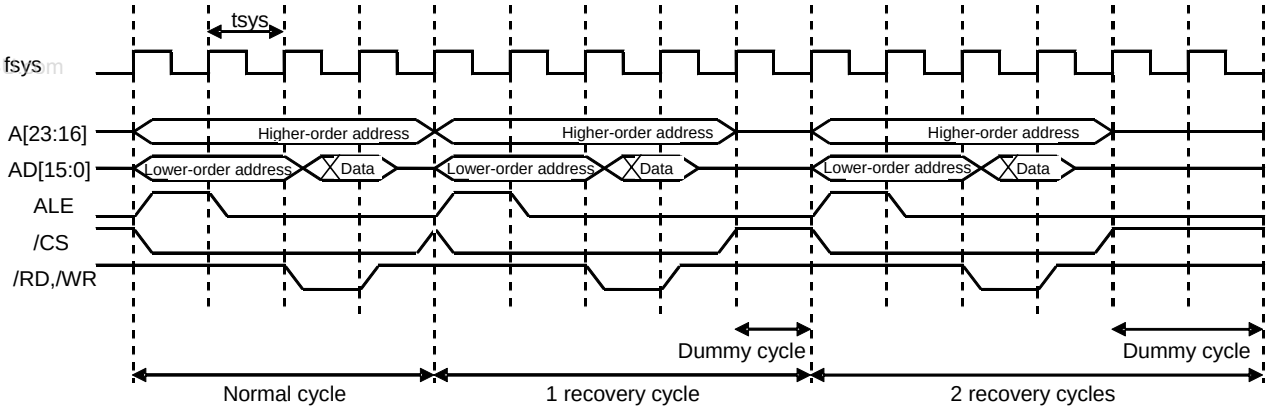


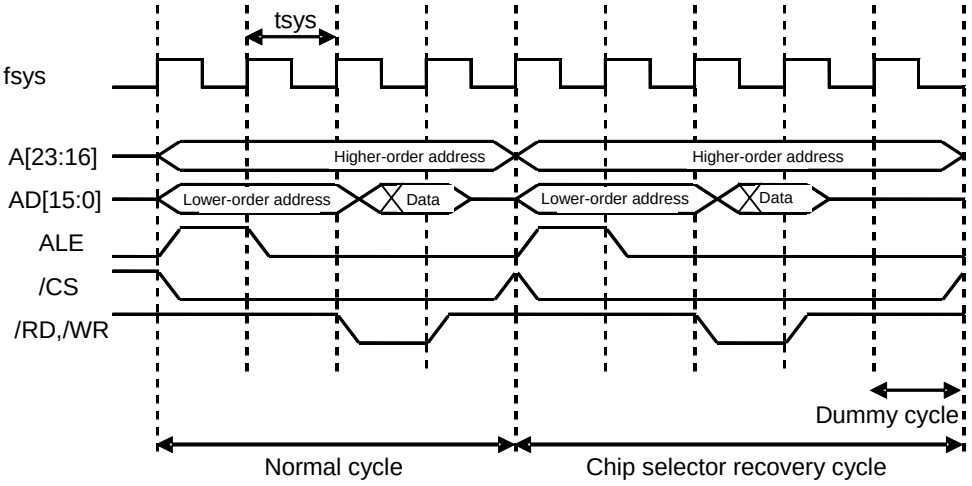
Fig. 8.4.14 Timing of Recovery Time Insertion

(5) Chip selector recovery time

If access to external areas occurs consecutively, a dummy cycle can be inserted for recovery time.

The dummy cycle insertion setting can be made in the chip selector and wait controller registers, BmnCS<BnCSCV>. As for the number of dummy cycles, one system clock (internal) can be specified for each block. Fig. 8.4.15 shows the timing of recovery time insertion.

When chip selector recovery is inserted (ALE width:1fsys)



8.5 Bus Arbitration

The TMP19A64 can be connected to an external bus master. The arbitration of bus control authority with the external bus master is executed by using the two signals, $\overline{\text{BUSRQ}}$ and $\overline{\text{BUSAK}}$. The external bus master can acquire control authority for TMP19A64 external buses only, and cannot acquire control authority for internal buses.

(1) Accessible range of external bus master

The external bus master can acquire control authority for TMP19A64 external buses only, and cannot acquire control authority for internal buses (G-BUS). Therefore, the external bus master cannot access the internal memories or the internal I/O. The arbitration of bus control authority for external buses is executed by the external bus interface circuit (EBIF), and this is independent of the CPU and the internal DMAC. Even when the external bus master holds the external bus control authority, the CPU and the internal DMAC can access the internal ROM, RAM and registers. On the other hand, if the CPU or the internal DMAC tries to access an external memory when the external bus master holds the external bus control authority, the CPU or the internal DMAC has to wait until the external bus master releases the bus. For this reason, if the $\overline{\text{BUSRQ}}$ remains active, the TMP19A64 can lock.

(2) Acquisition of bus control authority

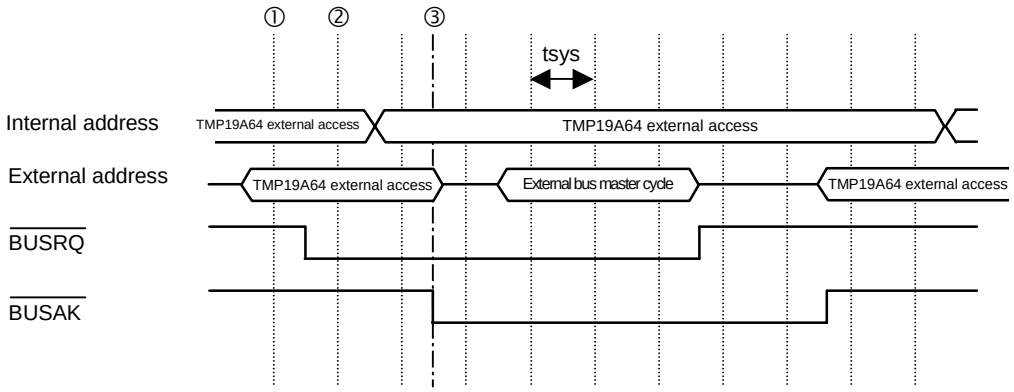
The external bus master requests the TMP19A64 for bus control authority by asserting the $\overline{\text{BUSRQ}}$ signal. The TMP19A64 samples the $\overline{\text{BUSRQ}}$ signal at the break of external bus cycles on the internal buses (G-BUS) and determines whether or not to give the bus control authority to the external bus master. When it gives the bus control authority to the external bus master, it asserts the $\overline{\text{BUSAK}}$ signal. At the same time, it makes address buses, data buses and bus control signals ($\overline{\text{RD}}$ and $\overline{\text{WR}}$) in a state of high impedance. (The internal pull-up is enabled for the R/$\overline{\text{W}}$, $\overline{\text{HWR}}$ and $\overline{\text{CSx}}$.)

Depending on the relationship between the size of data to be loaded or stored and the external memory bus width, two or more bus cycles can occur in response to a single data transfer (bus sizing). In this case, the end of the last bus cycle is the break of external bus cycles.

If access to external areas occurs consecutively on the TMP19A64, a dummy cycle can be inserted. Again, requests for buses are accepted at the break of external bus cycles on the internal buses (G-BUS). During a dummy cycle, the next external bus cycle is already started on the internal buses. Therefore, even if the $\overline{\text{BUSRQ}}$ signal is asserted during a dummy cycle, the bus is not released until the next external bus cycle is completed.

Keep asserting the $\overline{\text{BUSRQ}}$ signal until the bus control authority is released.

Fig. 8.5.1 shows the timing of acquiring bus control authority by the external bus master.



① $\overline{\text{BUSRQ}}$ is at the "H" level.

② The TMP19A64 recognizes that the $\overline{\text{BUSRQ}}$ is at the "L" level, and releases the bus at the end of the bus cycle.

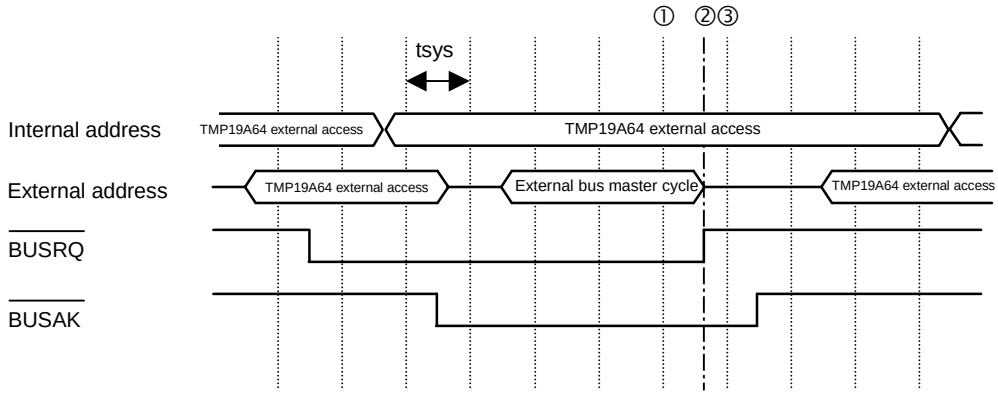
③ When the bus is completed, the TMP19A64 asserts $\overline{\text{BUSAK}}$. The external bus master recognizes that the $\overline{\text{BUSAK}}$ is at the "L" level, and acquires the bus control authority to start bus operations.

Fig. 8.5.1 Bus Control Authority Acquisition Timing

(3) Release of bus control authority

The external bus master releases the bus control authority when it becomes unnecessary.
If the external bus master no longer needs the bus control authority that it has held, it deasserts the $\overline{\text{BUSRQ}}$ signal and returns the bus control authority to the TMP19A64.

Fig. 8.5.2 shows the timing of releasing unnecessary bus control authority.



① The external bus master has the bus control authority.

② The external bus master deasserts the $\overline{\text{BUSRQ}}$, as it no longer requires the bus control authority.

③ The TMP19A64 recognizes that the $\overline{\text{BUSRQ}}$ is at the "H" level, and deasserts the $\overline{\text{BUSAK}}$.

Fig. 8.5.2 Timing of Releasing Bus Control Authority

9. The Chip Selector and Wait Controller

The TMP19A64 can be connected to external devices (I/O devices, ROM and SRAM).

6-block address spaces (CS0 through CS5) can be established in the TMP19A64 and three parameters can be specified for each 4-block address and other address spaces: data bus width, the number of waits and the number of dummy cycles.

$\overline{CS0}$ through $\overline{CS5}$ (also used as P40 through P45) are the output pins corresponding to spaces CS0 through CS5. These pins generate chip selector signals (for ROM and SRAM) to each space when the CPU designates an address in which spaces CS0 through CS5 are selected. For chip selector signals to be generated, however, the port 4 controller register (P4CR) and the port 4 function register (P4FC) must be set appropriately.

The specification of the spaces CS0 through CS5 is to be performed with a combination of base addresses (BAn, n=0 to 5) and mask addresses (MAn, n=0 to 5) using the base and mask address setting registers (BMA0 through BMA5).

Meanwhile, master enable, data bus width, the number of waits and the number of dummy cycles for each address space are specified in the chip selector and wait controller registers (B01CS, B23CS, B45CS and BEXCS).

A bus wait request pin ($\overline{WAIT}$) is provided as an input pin to control the status of these settings.

9.1 Specifying Address Spaces

Spaces CS0 through CS5 are specified using the base and mask address setting registers (BMA0 through BMA5).

In each bus cycle, a comparison is made to see if each address on the bus is located in the space CS0 through CS5. If the result of a comparison is a match, it is considered that the designated CS space has been accessed and chip selector signals are output from pins $\overline{CS0}$ through $\overline{CS5}$ and the operations specified by the chip selector and wait controller registers (B01CS, B23CS and B45CS) are executed. (Refer to "9.2 The Chip Selector and Wait Controller.")

9.1.1 Base and Mask Address Setting Registers

Figures 9.1.1 to 3 show base and mask address setting registers. For base addresses (BA0 through BA5), a start address in the space CS0 through CS5 is specified. In each bus cycle, the chip selector and wait controller compare values in their registers with addresses and those addresses with address bits masked by the mask address (MA0 through MA5) are not compared. The size of an address space is determined by the mask address setting.

(1) Base addresses

Base address BAn specifies the higher-order 16 bits (A31 through A16) of the start address. The lower-order 16 bits (A15 to A0) of the start address are always set to "0." Therefore, the start address begins with 0x0000_0000H and increases in 64 kilobyte units.

Fig. 9.1.4 shows the relationship between the start address and the BAn value.

(2) Mask addresses

Mask address (MAn) specifies which address bit value is to be compared. The address on the bus that corresponds to the bit for which "0" is written on the address mask MAn is to be included in address comparison to determine if the address is in the area of the CS0 to CS5 spaces. The bit for which "1" is written is not included in address comparison.

CS0 to CS5 spaces have different address bits that can be masked by MA0 to MA5.

CS0 space and CS1 space: A29 through A14

CS2 space and CS3 space: A30 through A15

CS4 space and CS5 space: A30 through A15

(Note) Address settings must be made using physical addresses.

Base and mask address setting registers BMA0 (0xFFFF_E400H)-BMA5 (0xFFFF_E414H)

BMA0 (0xFFFF_E400H)

	31	30	29	28	27	26	25	24
bit Symbol	BA0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							
	23	22	21	20	19	18	17	16
bit Symbol	BA0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	15	14	13	12	11	10	9	8
bit Symbol	MA0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	1	1
Function	Make sure that you write "0."							
	7	6	5	4	3	2	1	0
bit Symbol	MA0							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS0 space size setting 0: Address for comparison							

BMA1 (0xFFFF_E404H)

	31	30	29	28	27	26	25	24
bit Symbol	BA1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							
	23	22	21	20	19	18	17	16
bit Symbol	BA1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	15	14	13	12	11	10	9	8
bit Symbol	MA1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	1	1
Function	Make sure that you write "0."							
	7	6	5	4	3	2	1	0
bit Symbol	MA1							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS1 space size setting 0: Address for comparison							

(Note) Make sure that you write "0" for bits 10 through 15 for BMA0 and BMA1.
The size of both the CS0 and CS1 spaces can be a minimum of 16 KB to a maximum of 1 GB. The external address space of the TMP19A64 is 16 MB and so bits 10 through 15 must be set to "0" as addresses A24 through A29 are not masked.

Fig. 9.1.1 Base and Mask Address Setting Registers (BMA0, BMA1)

BMA2 (0xFFFF_E408H)

	31	30	29	28	27	26	25	24
bit Symbol	BA2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							
	23	22	21	20	19	18	17	16
bit Symbol	BA2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	15	14	13	12	11	10	9	8
bit Symbol	MA2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure that you write "0."							
	7	6	5	4	3	2	1	0
bit Symbol	MA2							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS2 space size setting 0: Address for comparison							

BMA3 (0xFFFF_E40CH)

	31	30	29	28	27	26	25	24
bit Symbol	BA3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							
	23	22	21	20	19	18	17	16
bit Symbol	BA3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	15	14	13	12	11	10	9	8
bit Symbol	MA3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure that you write "0."							
	7	6	5	4	3	2	1	0
bit Symbol	MA3							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS3 space size setting 0: Address for comparison							

(Note) Make sure that you write "0" for bits 9 through 15 for BMA2 and BMA3.
The size of both the CS2 and CS3 spaces can be a minimum of 32 KB to a maximum of 2 GB. The external address space of the TMP19A64 is 16 MB and so bits 9 through 15 must be set to "0" as addresses A24 through A30 are not masked.

Fig. 9.1.2 Base and Mask Address Setting Registers (BMA2, BMA3)

BMA4 (0xFFFF_E410H)

	31	30	29	28	27	26	25	24
bit Symbol	BA4							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							
	23	22	21	20	19	18	17	16
bit Symbol	BA4							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	15	14	13	12	11	10	9	8
bit Symbol	MA4							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure that you write "0."							
	7	6	5	4	3	2	1	0
bit Symbol	MA4							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS4 space size setting 0: Address for comparison							

BMA5 (0xFFFF_E414H)

	31	30	29	28	27	26	25	24
bit Symbol	BA5							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							
	23	22	21	20	19	18	17	16
bit Symbol	BA5							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	15	14	13	12	11	10	9	8
bit Symbol	MA5							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure that you write "0."							
	7	6	5	4	3	2	1	0
bit Symbol	MA5							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS5 space size setting 0: Address for comparison							

(Note) Make sure that you write "0" for bits 9 through 15 for BMA4 and BMA5.
The size of both the CS4 and CS5 spaces can be a minimum of 32 KB to a maximum of 2 GB. The external address space of the TMP19A64 is 16 MB and so bits 9 through 15 must be set to "0" as addresses A24 through A30 are not masked.

Fig. 9.1.3 Base and Mask Address Setting Registers (BMA4, BMA5)

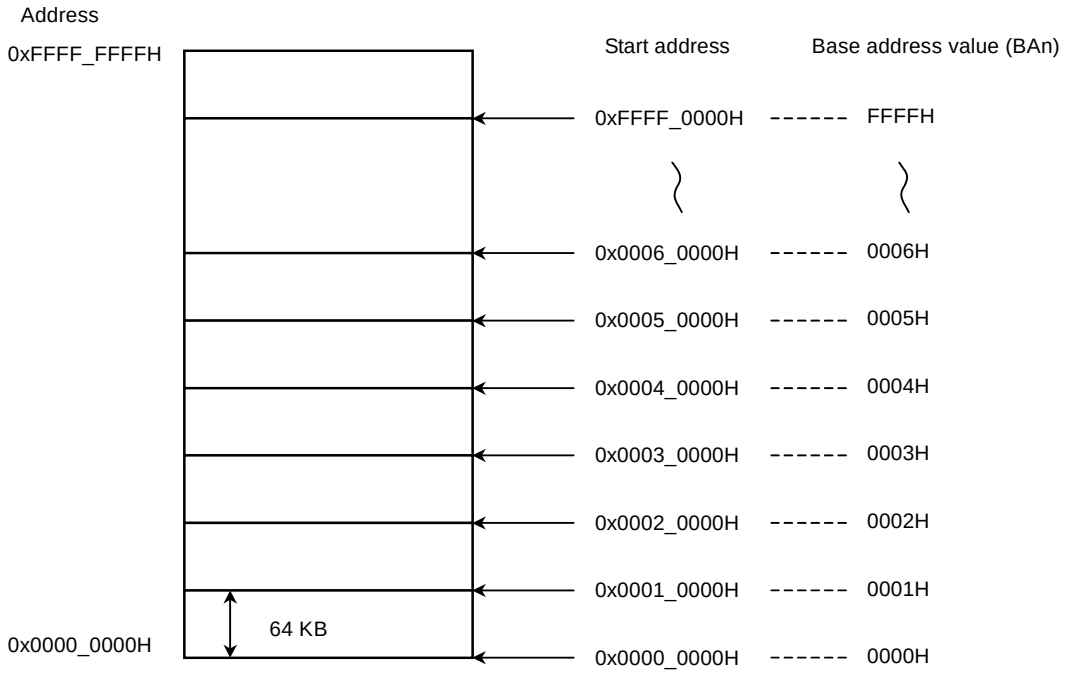
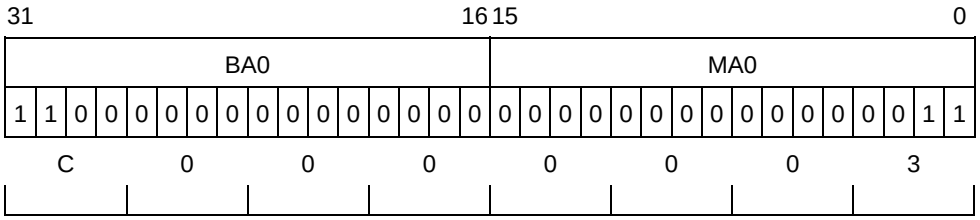


Fig. 9.1.4 Start and Base Address Register Values

9.1.2 How to Define Start Addresses and Address Spaces

- To specify a space of 64 KB starting at 0xC000_0000 in the CS0 space, the base and mask address registers must be programmed as shown below.

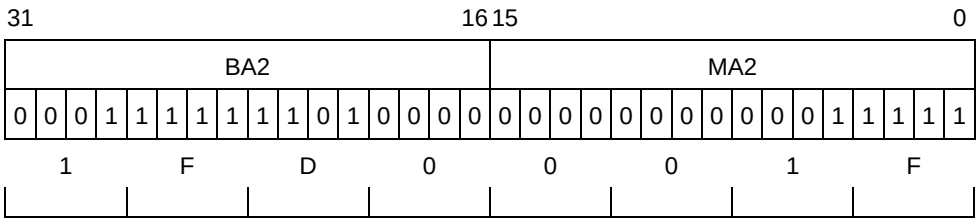


Values to be set in the base and mask address registers (BMA0)

In the base address (BA0), specify "0xC000" that corresponds to higher 16 bits of a start address, while in the mask address (MA0), specify whether a comparison of addresses in the space A29 through A16 is to be made or not. To ensure a comparison of A29 through A16, set bits 15 to 2 of the mask address (MA0) to "0." A comparison of addresses of A31 and A30 will definitely be made.

This setting allows A31 through A16 to be compared with the value specified as a start address. As A15 through A0 are masked, a space of 64 KB from 0xC000_0000 to 0xC000_FFFF is designated as a CS0 space and the $\overline{\text{CSO}}$ signal is asserted if there is a match with an address on the bus.

- To specify a space of 1 MB starting at 0x1FD0_0000 in the CS2 space, the base and mask address registers must be programmed as shown below.



Values to be set in the base and mask address registers (BMA2)

In the base address (BA2), specify "0x1FD0" that corresponds to higher 16 bits of a start address, while in the mask address (MA2), specify whether a comparison of addresses in the space A30 through A15 is to be made or not. To ensure a comparison of A30 through A20, set bits 15 to 5 of the mask address (MA2) to "0." A comparison of A31 will definitely be made.

This setting allows A31 through A20 to be compared with the value specified as a start address. As A19 through A0 are masked, a space of 1 MB from 0x1FD0_0000 to 0x1FDF_FFFF is designated as a CS2 space.

Note: The $\overline{\text{CSn}}$ signal is not asserted to the following address spaces in the TMP19A64:
0x1FC0_0000 to 0x1FCF_FFFF
0x4000_0000 to 0x400F_FFFF
0xFFFD_6000 to 0xFFFD_FFFF, 0xFFFF_6000 to 0xFFFF_DFFF

After a reset, the CS0, CS1, and CS3 through CS5 spaces are disabled, while the whole CS2 space (4 GB) is enabled as an address space.

Table 9.1.1 shows the relationship between CS space and space sizes. If two or more address spaces are specified simultaneously, a space or spaces with a smaller space number will be given priority in space selection.

Example: 0xC000_0000 as a start address of the CS0 space with a space size of 16 KB
0xC000_0000 as a start address of the CS1 space with a space size of 64 KB

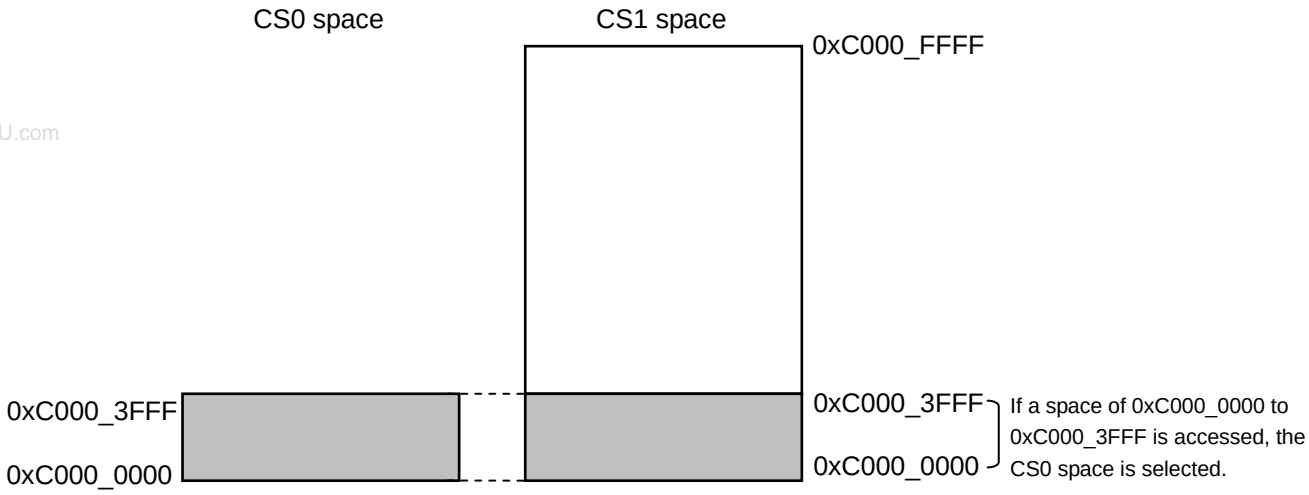


Table 9.1.1 CS Space and Space Sizes

Size (bytes) / CS space	16 K	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M	16 M
CS0	○	○	○	○	○	○	○	○	○	○	○
CS1	○	○	○	○	○	○	○	○	○	○	○
CS2		○	○	○	○	○	○	○	○	○	○
CS3		○	○	○	○	○	○	○	○	○	○
CS4		○	○	○	○	○	○	○	○	○	○
CS5		○	○	○	○	○	○	○	○	○	○

9.2 The Chip Selector and Wait Controller

Fig. 9.2.1 to Fig. 9.2.4 show the chip selector and wait controller registers. For each address space (spaces CS0 through CS5 and other address spaces), each chip selector and wait controller register (B01CS through B45CS, BEXCS) can be programmed to set master enable or disable, to select data bus width, to specify the number of waits and to insert dummy cycles.

If two or more address spaces are specified simultaneously, a space or spaces with a smaller space number will be given priority in space selection (order of priority: CS0>CS1>CS2>CS3>CS4>CS5>EXCS).

B01CS
(FFFFE480H)

	7	6	5	4	3	2	1	0
bit Symbol	B0OM			B0BUS	B0W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	15	14	13	12	11	10	9	8
bit Symbol		B0CSCV	B0WCV		B0E		B0RCV	
Read/Write	R	R/W	R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function		Specify the number of dummy cycles to be inserted. (CS0 recovery time) 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited		Enable or disable CS0. 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

	23	22	21	20	19	18	17	16
bit Symbol	B1OM			B1BUS	B1W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	31	30	29	28	27	26	25	24
bit Symbol		B1CSCV	B1WCV		B1E		B1RCV	
Read/Write	R	R/W	R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function		Specify the number of dummy cycles to be inserted. (CS1 recovery time) 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited		Enable or disable CS1. 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

Fig. 9.2.1 Chip Selector and Wait Controller Registers

B23CS (0xFFFF_E484H)

	7	6	5	4	3	2	1	0
bit Symbol	B2OM			B2BUS	B2W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	15	14	13	12	11	10	9	8
bit Symbol		B2CSCV	B2WCV		B2E	B2M	B2RCV	
Read/Write	R	R/W	R/W		R/W			
After reset	0	0	0	0	1	0	0	0
Function		Specify the number of dummy cycles to be inserted. (CS2 recovery time) 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited		Enable or disable CS2. 0: Disable 1: Enable	Select CS2 space. 0: 4G space 1: CS space	Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

	23	22	21	20	19	18	17	16
bit Symbol	B3OM			B3BUS	B3W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip select output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	31	30	29	28	27	26	25	24
bit Symbol		B3CSCV	B3WCV		B3E		B3RCV	
Read/Write	R	R/W	R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function		Specify the number of dummy cycles to be inserted. (CS3 recovery time) 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited		Enable or disable CS3. 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

Fig. 9.2.2 Chip Selector and Wait Controller Registers

B45CS (0xFFFF_E488H)

	7	6	5	4	3	2	1	0
bit Symbol	B4OM			B4BUS	B4W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	15	14	13	12	11	10	9	8
bit Symbol		B4CSCV	B4WCV		B4E		B4RCV	
Read/Write	R	R/W	R/W		R/W	R	R/W	
After reset	0	0	0	0	1	0	0	0
Function		Specify the number of dummy cycles to be inserted. (CS4 recovery time) 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited		Enable or disable CS4. 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

	23	22	21	20	19	18	17	16
bit Symbol	B5OM			B5BUS	B5W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip select output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	31	30	29	28	27	26	25	24
bit Symbol		B5CSCV	B5WCV		B5E		B5RCV	
Read/Write	R	R/W	R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function		Specify the number of dummy cycles to be inserted. (CS5 recovery time) 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited		Enable or disable CS5. 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

Fig. 9.2.3 Chip Selector and Wait Controller Registers

BEXCS (0xFFFF_E48CH)

	7	6	5	4	3	2	1	0
bit Symbol	BEXOM			BEXBUS	BEXW			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (automatic WAIT insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (external WAIT input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			

	15	14	13	12	11	10	9	8
bit Symbol		BECSCV	BEXWCV				BEXRCV	
Read/Write	R	R/W	R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function		Specify the number of dummy cycles to be inserted. 1: 1 cycle 0: None	Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited				Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Setting prohibited	

	23	22	21		20	19	18	
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function								

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R				R/W	R		
After reset	0	0	0	0	0	0	0	0
Function								

Fig. 9.2.4 Chip Selector and Wait Controller Registers

A reset of the TMP19A64 allows the port 4 controller register (P4CR) and the port 4 function register (P4FC) to be cleared to "0," and the CS signal output is disabled. To output the CS signals, set the corresponding bits to "1" at the P4FC and the P4CR in that order. The CS recovery time can be configured in any other areas than the CS setting areas, but CS signals will not be output.

10. DMA Controller (DMAC)

The TMP19A64 has a built-in 8-channel DMA Controller (DMAC).

10.1 Features

The DMAC of the TMP19A64 has the following features:

(1) DMA with 8 independent channels

(2) Two types of requests for bus control authority: With and without snoop requests

(3) Transfer requests: Internal requests (software initiated)/external requests (external interrupts, interrupt requests given by internal peripheral I/Os, and requests given by the $\overline{\text{DREQ}}$ pin)

Requests given by the $\overline{\text{DREQ}}$ pin (CH2, 3): Level mode (memory → memory)
Edge mode (memory → I/O, I/O to memory)

(4) Transfer mode: Dual address mode

(5) Transfer devices: Memory-to-memory, memory-to-I/O, I/O-to-memory

(6) Device size: 32-bit memory (8 or 16 bits can be specified using the CS/WAIT controller); I/O of 8, 16 or 32 bits

(7) Address changes: Increase, decrease, fixed, irregular increase, irregular decrease

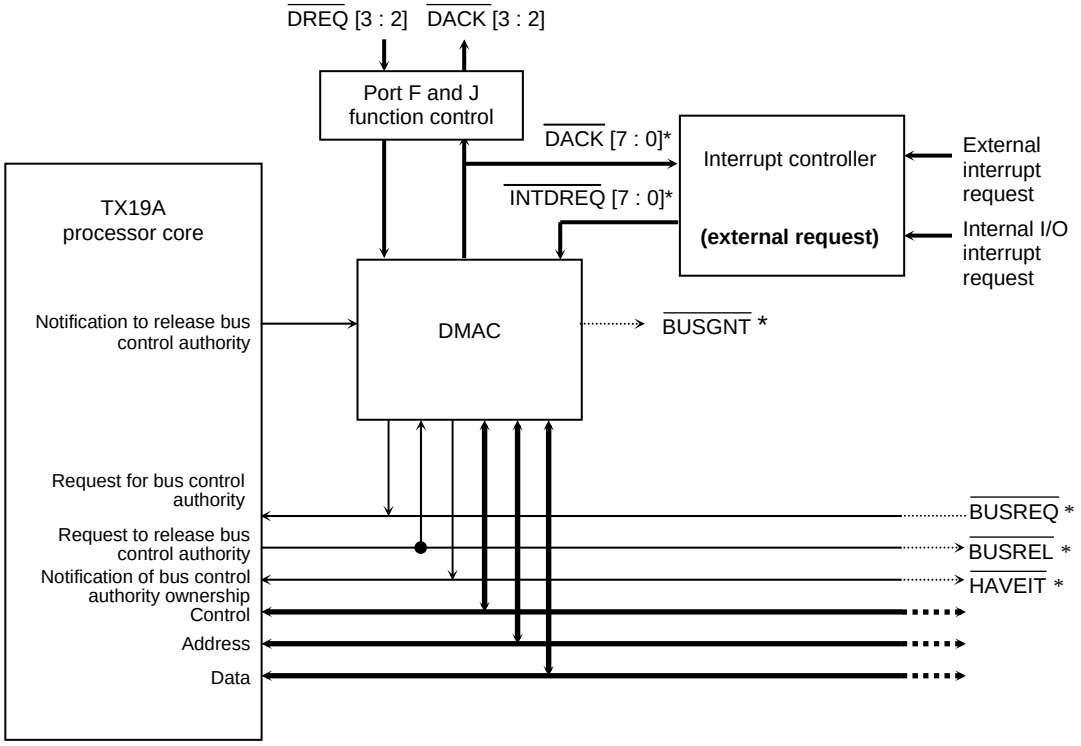
(8) Channel priority: Fixed (in ascending order of channel numbers)

(9) Endian switchover function

10.2 Configuration

10.2.1 Internal Connections of the TMP19A64

Fig. 10.2.1 shows the internal connections with the DMAC in the TMP19A64.



(Note) In Fig. 10.1, signals indicated by * are internal signals.

Fig. 10.2.1 DMAC Connections in the TMP19A64

The DMAC has eight DMA channels. Each of these channels handles the data transfer request signal ($\overline{\text{INTDREQn}}$) from the interrupt controller and the acknowledgment signal ($\overline{\text{DACKn}}$) generated in response to INTDREQn, where "n" is a channel number from 0 to 7. External pins ($\overline{\text{DREQ2}}$ and $\overline{\text{DREQ3}}$) are internally wired to allow them to function as pins of the port F and J. To use them as pins of the port F and J, they must be selected by setting the function control register PFFC and PJFC to an appropriate setting. If both ports are set to use the DMAC function, the port F is given priority in using the DMAC function.

Pins, $\overline{\text{DACK2}}$ and $\overline{\text{DACK3}}$, handle the data transfer request and acknowledge signal output supplied through external pins, $\overline{\text{DREQ2}}$ and $\overline{\text{DREQ3}}$. Channel 0 is given higher priority than channel 1, channel 1 higher priority than channel 2 and channel 2 higher priority than channel 3. Subsequent channels are given priority in the same manner.

The TX19A processor core has a snoop function. Using the snoop function, the TX19A processor core opens the core's data bus to the DMAC, thus allowing the DMAC to access the internal ROM and RAM linked to the core. The DMAC is capable of determining whether or not to use this snoop function. For further information on the snoop function, refer to 10.2.3 "Snoop Function."

Two types of bus control authority (SREQ and GREQ) are available to the DMAC and which type of control right to use depends on the use or nonuse of the snoop function. GREQ is a request for bus control authority if the DMAC does not use the snoop function, while SREQ is a request for bus control authority if the DMAC uses the snoop function. SREQ is given higher priority than GREQ.

10.2.2 DMAC Internal Blocks

Fig. 10.2.2 shows the internal blocks of the DMAC.

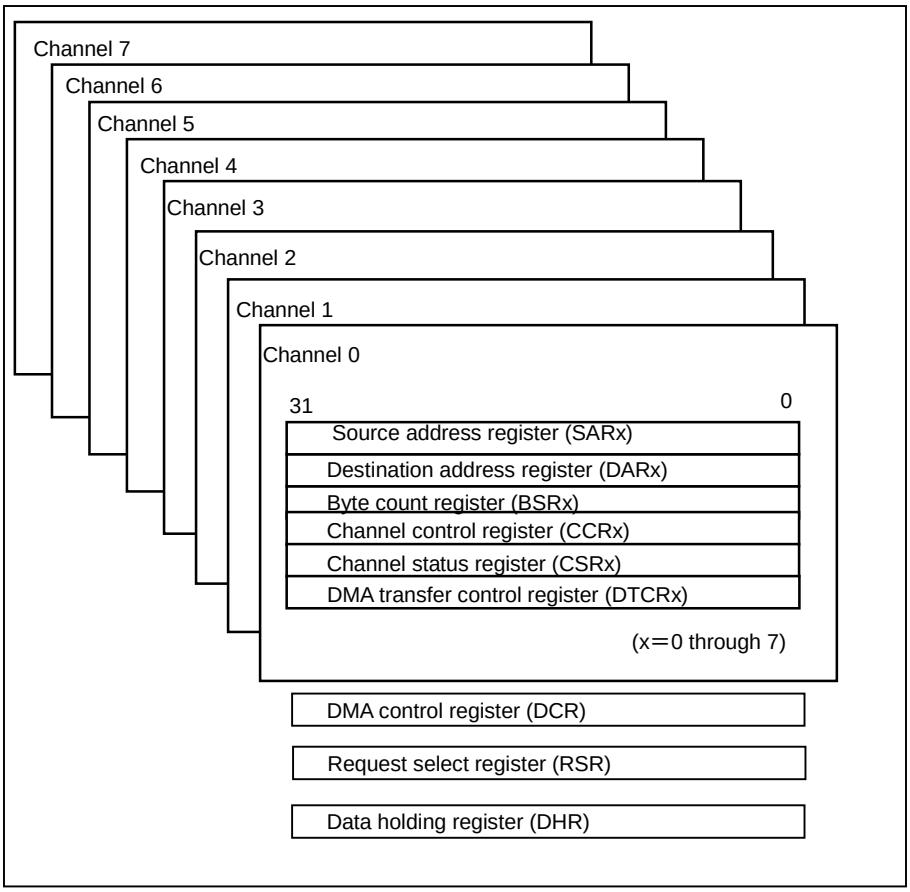


Fig. 10.2.2 DMAC Internal Blocks

10.2.3 Snoop Function

The TX19A processor core has a snoop function. If the snoop function is activated, the TX19A processor core opens the core's data bus to the DMAC and suspends its own operation until the DMAC withdraws a request for bus control authority. If the snoop function is enabled, the DMAC can access the internal RAM and ROM and therefore designate the RAM or ROM as a source or destination.

If the snoop function is not used, the DMAC cannot access the internal RAM or ROM. However, the G-Bus is opened to the DMAC. If the TX19A processor core attempts to access memory or the I/O by way of the G-Bus and if the DMAC does not accept a bus control release request, bus operations cannot be executed and, as a result, the pipeline stalls.

(Note) If the snoop function is not used, the TX19A processor core does not open the data bus to the DMAC. If the data bus is closed and the internal RAM or ROM is designated as a DMAC source or destination, an acknowledgment signal will not be returned in response to a DMAC transfer bus cycle and, as a result, the bus will lock.

10.3 Registers

The DMAC has fifty-one 32-bit registers. Table 10.3.1 shows the register map of the DMAC.

Table 10.3.1 DMAC Registers

Address	Register symbol	Register name
0xFFFF_E200	CCR0	Channel control register (ch. 0)
0xFFFF_E204	CSR0	Channel status register (ch. 0)
0xFFFF_E208	SAR0	Source address register (ch. 0)
0xFFFF_E20C	DAR0	Destination address register (ch. 0)
0xFFFF_E210	BCR0	Byte count register (ch. 0)
0xFFFF_E218	DTCR0	DMA transfer control register (ch. 0)
0xFFFF_E220	CCR1	Channel control register (ch. 1)
0xFFFF_E224	CSR1	Channel status register (ch. 1)
0xFFFF_E228	SAR1	Source address register (ch. 1)
0xFFFF_E22C	DAR1	Destination address register (ch. 1)
0xFFFF_E230	BCR1	Byte count register (ch. 1)
0xFFFF_E238	DTCR1	DMA transfer control register (ch. 1)
0xFFFF_E240	CCR2	Channel control register (ch. 2)
0xFFFF_E244	CSR2	Channel status register (ch. 2)
0xFFFF_E248	SAR2	Source address register (ch. 2)
0xFFFF_E24C	DAR2	Destination address register (ch. 2)
0xFFFF_E250	BCR2	Byte count register (ch. 2)
0xFFFF_E258	DTCR2	DMA transfer control register (ch. 2)
0xFFFF_E260	CCR3	Channel control register (ch. 3)
0xFFFF_E264	CSR3	Channel status register (ch. 3)
0xFFFF_E268	SAR3	Source address register (ch. 3)
0xFFFF_E26C	DAR3	Destination address register (ch. 3)
0xFFFF_E270	BCR3	Byte count register (ch. 3)
0xFFFF_E278	DTCR3	DMA transfer control register (ch. 3)
0xFFFF_E280	CCR4	Channel control register (ch. 4)
0xFFFF_E284	CSR4	Channel status register (ch. 4)
0xFFFF_E288	SAR4	Source address register (ch. 4)
0xFFFF_E28C	DAR4	Destination address register (ch. 4)
0xFFFF_E290	BCR4	Byte count register (ch. 4)
0xFFFF_E298	DTCR4	DMA transfer control register (ch. 4)
0xFFFF_E2A0	CCR5	Channel control register (ch. 5)
0xFFFF_E2A4	CSR5	Channel status register (ch. 5)
0xFFFF_E2A8	SAR5	Source address register (ch. 5)
0xFFFF_E2AC	DAR5	Destination address register (ch. 5)
0xFFFF_E2B0	BCR5	Byte count register (ch. 5)
0xFFFF_E2B8	DTCR5	DMA transfer control register (ch. 5)
0xFFFF_E2C0	CCR6	Channel control register (ch. 6)
0xFFFF_E2C4	CSR6	Channel status register (ch. 6)
0xFFFF_E2C8	SAR6	Source address register (ch. 6)
0xFFFF_E2CC	DAR6	Destination address register (ch. 6)
0xFFFF_E2D0	BCR6	Byte count register (ch. 6)
0xFFFF_E2D8	DTCR6	DMA transfer control register (ch. 6)

Table 10.3.1 DMAC Registers (2)

0xFFFF_E2E0	CCR7	Channel control register (ch. 7)
0xFFFF_E2E4	CSR7	Channel status register (ch. 7)
0xFFFF_E2E8	SAR7	Source address register (ch. 7)
0xFFFF_E2EC	DAR7	Destination address register (ch. 7)
0xFFFF_E2F0	BCR7	Byte count register (ch. 7)
0xFFFF_E2F8	DTCR7	DMA transfer control register (ch. 7)
0xFFFF_E300	DCR	DMA control register (DMAC)
0xFFFF_E304	RSR	Request select register (DMAC)
0xFFFF_E30C	DHR	Data holding register (DMAC)

10.3.1 DMA Control Register (DCR)

DCR (0xFFFF_E300H)

	7	6	5	4	3	2	1	0
bit Symbol	Rst7	Rst6	Rst5	Rst4	Rst3	Rst2	Rst1	Rst0
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	See detailed description.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function								
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function								
	31	30	29	28	27	26	25	24
bit Symbol	Rstall							
Read/Write	W	R						
After reset	0	0						
Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31	Rstall	Reset all	Performs a software reset of the DMAC. If the Rstall bit is set to 1, the values of all the internal registers of the DMAC are reset to their initial values. All transfer requests are canceled and all eight channels go into an idle state. 0: Don't care 1: Initializes the DMAC
7	Rst7	Reset 7	Performs a software reset of the DMAC channel 7. If the Rst7 bit is set to 1, internal registers of the DMAC channel 7 and a corresponding bit of the channel 7 of the RSR register are reset to their initial values. The transfer request of the channel 7 is canceled and the channel 7 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 7
6	Rst6	Reset 6	Performs a software reset of the DMAC channel 6. If the Rst6 bit is set to 1, internal registers of the DMAC channel 6 and a corresponding bit of the channel 6 of the RSR register are reset to their initial values. The transfer request of the channel 6 is canceled and the channel 6 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 6
5	Rst5	Reset 5	Performs a software reset of the DMAC channel 5. If the Rst5 bit is set to 1, internal registers of the DMAC channel 5 and a corresponding bit of the channel 5 of the RSR register are reset to their initial values. The transfer request of the channel 5 is canceled and the channel 5 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 5

Fig. 10.3.1 DMA Control Register (DCR) (1 of 2)

Bit	Mnemonic	Field name	Description
4	Rst4	Reset 4	Performs a software reset of the DMAC channel 4. If the Rst4 bit is set to 1, internal registers of the DMAC channel 4 and a corresponding bit of the channel 4 of the RSR register are reset to their initial values. The transfer request of the channel 4 is canceled and the channel 4 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 4
3	Rst3	Reset 3	Performs a software reset of the DMAC channel 3. If the Rst3 bit is set to 1, internal registers of the DMAC channel 3 and a corresponding bit of the channel 3 of the RSR register are reset to their initial values. The transfer request of the channel 3 is canceled and the channel 3 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 3
2	Rst2	Reset 2	Performs a software reset of the DMAC channel 2. If the Rst2 bit is set to 1, internal registers of the DMAC channel 2 and a corresponding bit of the channel 2 of the RSR register are reset to their initial values. The transfer request of the channel 2 is canceled and the channel 2 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 2
1	Rst1	Reset 1	Performs a software reset of the DMAC channel 1. If the Rst1 bit is set to 1, internal registers of the DMAC channel 1 and a corresponding bit of the channel 1 of the RSR register are reset to their initial values. The transfer request of the channel 1 is canceled and the channel 1 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 1
0	Rst0	Reset 0	Performs a software reset of the DMAC channel 0. If the Rst0 bit is set to 1, internal registers of the DMAC channel 0 and a corresponding bit of the channel 0 of the RSR register are reset to their initial values. The transfer request of the channel 0 is canceled and the channel 0 goes into an idle state. 0: Don't care 1: Initializes the DMAC channel 0

Fig. 10.3.1 DMA Control Register (DCR) (2 of 2)

(Note 1) If a write to the DCR register occurs during a software reset right after the last round of DMA transfer is completed, the interrupt to stop DMA transfer is not canceled although the channel register is initialized.

(Note 2) An attempt to execute a write (software reset) to the DCR register by DMA transfer must be strictly avoided.

10.3.2 Channel Control Registers (CCRn) (n=0 through 7)

CCRn		7	6	5	4	3	2	1	0
(0xFFFF_E200H)	bit Symbol	SAC	DIO	DAC		TrSiz		DPS	
(0xFFFF_E220H)	Read/Write	R/W	R/W	R/W		R/W		R/W	
(0xFFFF_E240H)	After reset	0	0	0	0	0	0	0	0
(0xFFFF_E260H)	Function	See detailed description.							
		15	14	13	12	11	10	9	8
(0xFFFF_E280H)	bit Symbol		ExR	PosE	Lev	SReq	RelEn	SIO	SAC
(0xFFFF_E2A0H)	Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
(0xFFFF_E2C0H)	After reset	0	0	0	0	0	0	0	0
(0xFFFF_E2E0H)	Function	Always set this bit to "0."	See detailed description.						
		23	22	21	20	19	18	17	16
	bit Symbol	NIEn	AbIEn					Big	
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	W
	After reset	1	1	1	0	0	0	1	0
	Function	See detailed description.		Always set this bit to "0."				See detailed description.	Always set this bit to "0."
		31	30	29	28	27	26	25	24
	bit Symbol	Str							
	Read/Write	W							W
	After reset	0	0	0	0	0	0	0	0
	Function	See detailed description.							Always set this bit to "0."

Bit	Mnemonic	Field name	Description
31	Str	Channel start	Start (initial value: 0) Starts channel operation. If this bit is set to 1, the channel goes into a standby mode and starts to transfer data in response to a transfer request. Only a write of 1 is valid to the Str bit and a write of 0 is ignored. A read always returns a 0. 1: Starts channel operation
24	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
23	NIEn	Normal completion interrupt enable	Normal Completion Interrupt Enable (initial value: 1) 1: Normal completion interrupt enable 0: Normal completion interrupt disable
22	AbIEn	Abnormal completion interrupt enable	Abnormal Completion Interrupt Enable (initial value: 1) 1: Abnormal completion interrupt enable 0: Abnormal completion interrupt disable
21	—	(Reserved)	This is a reserved bit. Although its initial value is "1," always set this bit to "0."
20	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
19	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
18	—	(Reserved)	This is a reserved bit. Always set this bit to "0."

Fig. 10.3.2 Channel Control Register (CCRn) (1 of 3)

Bit	Mnemonic	Field name	Description
17	Big	Big-endian	Big Endian (initial value: 1) 1: A channel operates by big-endian 0: A channel operates by little-endian
16	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
15	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
14	ExR	External request mode	External Request Mode (initial value: 0) Selects a transfer request mode. 1: External transfer request (interrupt request or external $\overline{\text{DREQ}}$n request) 0: Internal transfer request (software initiated)
13	PosE	Positive edge	Positive Edge (initial value: 0) The effective level of the transfer request signal $\overline{\text{INTDREQ}}$n or $\overline{\text{DREQ}}$n is specified. This function is valid only if the transfer request is an external transfer request (if the ExR bit is 1). If it is an internal transfer request (if the ExR bit is 0), the PosE value is ignored. Because the $\overline{\text{INTDREQ}}$n and $\overline{\text{DREQ}}$n signals are active at "L" level, make sure that this PosE bit is set to "0." 1: Setting prohibited 0: The falling edge of the $\overline{\text{INTDREQ}}$n or $\overline{\text{DREQ}}$n signal or the "L" level is effective. The $\overline{\text{DACK}}$n is active at "L" level.
12	Lev	Level mode	Level Mode (initial value: 0) Specifies which is used to recognize the external transfer request, signal level or signal change. This setting is valid only if a transfer request is the external transfer request (if the ExR bit is 1). If the internal transfer request is specified as a transfer request (if the ExR bit is 0), the value of the Lev bit is ignored. Because the $\overline{\text{INTDREQ}}$n signal is active at "L" level, make sure that you set the Lev bit to "1." The state of active $\overline{\text{DREQ}}$n is determined by the Lev bit setting. 1: Level mode The level of the $\overline{\text{DREQ}}$n signal is recognized as a data transfer request. (The "L" level is recognized if the PosE bit is 0. 0: Edge mode A change in the $\overline{\text{DREQ}}$n signal is recognized as a data transfer request. (A falling edge is recognized if the PosE bit is 0.)
11	SReq	Snoop request	Snoop Request (initial value: 0) The use of the snoop function is specified by asserting the bus control request mode. If the snoop function is used, the snoop function of the TX19A processor core is enabled and the DMAC can use the data bus of the TX19A processor core. If the snoop function is not used, the snoop function of the TX19A processor core does not work. 1: Use snoop function (SREQ) 0: Do not use snoop function (GREQ)
10	RelEn	Bus control release request enable	Release Request Enable (initial value: 0) Acknowledgment of the bus control release request made by the TX19A processor core is specified. This function is valid only if GREQ is generated. If SREQ is generated, the TX19A processor core cannot make a bus control release request and, therefore, this function cannot be used. 1: The bus control release request is acknowledged if the DMAC has control of the bus. If the TX19A processor core issues a bus control release request, the DMAC relinquishes control of the bus to the TX19A processor core during a pause in bus operation. 0: The bus control release request is not acknowledged.
9	SIO	Source I/O	Source Type: I/O (initial value: 0) Specifies the source device. 1: I/O device 0: Memory

Fig. 10.3.2 Channel Control Register (CCRn) (2/3)

Bit	Mnemonic	Field name	Description
8 : 7	SAC	Source address count	Source Address Count (initial value: 00) Source Address Count (initial value: 00) Specifies the manner of change in a source address. 1x: Address fixed 01: Address decrease 00: Address increase
6	DIO	Destination I/O	Destination Type: I/O (initial value: 0) Specifies a destination device. 1: I/O device 0: Memory
5 : 4	DAC	Destination address count	Destination Address Count (initial value: 00) Specifies the manner of change in a destination address. 1x: Address fixed 01: Address decrease 00: Address increase
3 : 2	TrSiz	Transfer unit	Transfer Size (initial value: 00) Specifies the amount of data to be transferred in response to one transfer request. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)
1 : 0	DPS	Device port size	Device Port Size (initial value: 00) Specifies the bus width of an I/O device designated as a source or destination device. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4 bytes)

Fig. 10.3.2 Channel Control Register (CCRn) (3/3)

(Note 1) The CCRn register setting must be completed before the DMAC is put into a standby mode.

(Note 2) When accessing the internal I/O or transferring data by DMA in response to the DREQ pin request, make sure that you set the transfer unit <TrSiz> and the device port size <DPS> to the same size.

(Note 3) In executing memory-to-memory data transfer, a value set in DPS becomes invalid.

10.3.3 Request Select Register (RSR)

RSR (0xFFFF_E304H)

	7	6	5	4	3	2	1	0
bit Symbol					ReqS3	ReqS2		
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Always set this bit to "0."				See detailed description.		Always set this bit to "0."	
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function								
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function								
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function								

Bit	Mnemonic	Field name	Description
3	ReqS3	Request select (ch.3)	Request Select (initial value: 0) Selects a source of the external transfer request for the DMA channel 3. 1: Request made by $\overline{\text{DREQ3}}$ 0: Request made by the interrupt controller (INTC)
2	ReqS2	Request select (ch.2)	Request Select (initial value: 0) Selects a source of the external transfer request for the DMA channel 2. 1: Request made by $\overline{\text{DREQ2}}$ 0: Request made by the interrupt controller (INTC)

Fig. 10.3.3 DMA Control Register (RSR)

(Note) Make sure that you write "0" to bits 0, 1 and 4 through 7 of the RSR register.

10.3.4 Channel Status Registers (CSRn) (n=0 through 7)

CSRn		7	6	5	4	3	2	1	0
	bit Symbol								
(0xFFFF_E204H)	Read/Write	R					R/W	R/W	R/W
(0xFFFF_E224H)	After reset	0					0	0	0
(0xFFFF_E244H)	Function						Always set this bit to "0."		
(0xFFFF_E264H)		15	14	13	12	11	10	9	8
(0xFFFF_E284H)	bit Symbol								
(0xFFFF_E2A4H)	Read/Write	R							
(0xFFFF_E2C4H)	After reset	0							
(0xFFFF_E2E4H)	Function								
		23	22	21	20	19	18	17	16
	bit Symbol	NC	AbC		BES	BED	Conf		
	Read/Write	R/W	R/W	R/W	R	R	R	R	
	After reset	0	0	0	0	0	0	0	
	Function	See detailed description.		Always set this bit to "0."	See detailed description.				
		31	30	29	28	27	26	25	24
	bit Symbol	Act							
	Read/Write	R	R						
	After reset	0	0						
	Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31	Act	Channel active	Channel Active (initial value: 0) Indicates whether the channel is in a standby mode: 1: In a standby mode 0: Not in a standby mode
23	NC	Normal completion	Normal Completion (initial value: 0) Indicates normal completion of channel operation. If an interrupt at normal completion is permitted by the CCR register, the DMAC requests an interrupt when the NC bit becomes 1. This setting can be cleared by writing 0 to the NC bit. If a request for an interrupt at normal completion was previously issued, the request is canceled if the NC bit becomes 0. If an attempt is made to set the Str bit to 1 when the NC bit is 1, an error occurs. To start the next transfer, the NC bit must be cleared to 0. A write of 1 will be ignored. 1: Channel operation has been completed normally. 0: Channel operation has not been completed normally

Fig. 10.3.4 Channel Status Registers (CSRn) (1/2)

Bit	Mnemonic	Field name	Description
22	AbC	Abnormal completion	Abnormal Completion (initial value: 0) Indicates abnormal completion of channel operation. If an interrupt at abnormal completion is permitted by the CCR register, the DMAC requests an interrupt when the AbC bit becomes 1. This setting can be cleared by writing 0 to the AbC bit. If a request for an interrupt at abnormal completion was previously issued, the request is canceled if the AbC bit becomes 0. Additionally, if the AbC bit is cleared to 0, each of the BES, BED and Conf bits are cleared to 0. If an attempt is made to set the Str bit to 1 when the AbC bit is 1, an error occurs. To start the next transfer, the AbC bit must be cleared to 0. A write of 1 will be ignored. 1: Channel operation has been completed abnormally. 0: Channel operation has not been completed abnormally.
21	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
20	BES	Source bus error	Source Bus Error (initial value: 0) 1: A bus error has occurred when the source was accessed. 0: A bus error has not occurred when the source was accessed.
19	BED	Destination bus error	Destination Bus Error (initial value: 0) 1: A bus error has occurred when the destination was accessed. 0: A bus error has not occurred when the destination was accessed.
18	Conf	Configuration error	Configuration Error (initial value: 0) 1: A configuration error has occurred. 0: A configuration error has not occurred.
2 : 0	—	(Reserved)	These three bits are reserved bits. Always set them to "0."

Fig. 10.3.4 Channel Status Registers (CSRn) (2/2)

10.3.5 Source Address Registers (SARn) (n=0 through 7)

		7	6	5	4	3	2	1	0
SARn	bit Symbol	SAddr7	SAddr6	SAddr5	SAddr4	SAddr3	SAddr2	SAddr1	SAddr0
(0xFFFF_E208H)	Read/Write	R/W							
(0xFFFF_E228H)	After reset	0							
(0xFFFF_E248H)	Function	See detailed description.							
(0xFFFF_E268H)		15	14	13	12	11	10	9	8
(0xFFFF_E288H)	bit Symbol	SAddr15	SAddr14	SAddr13	SAddr12	SAddr11	SAddr10	SAddr9	SAddr8
(0xFFFF_E2A8H)	Read/Write	R/W							
(0xFFFF_E2C8H)	After reset	0							
(0xFFFF_E2E8H)	Function	See detailed description.							
		23	22	21	20	19	18	17	16
	bit Symbol	SAddr23	SAddr22	SAddr21	SAddr20	SAddr19	SAddr18	SAddr17	SAddr16
	Read/Write	R/W							
	After reset	0							
	Function	See detailed description.							
		31	30	29	28	27	26	25	24
	bit Symbol	SAddr31	SAddr30	SAddr29	SAddr28	SAddr27	SAddr26	SAddr25	SAddr24
	Read/Write	R/W							
	After reset	0							
	Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31 : 0	SAddr	Source address	Source Address (initial value: 0) Specifies the address of the source from which data is transferred using a physical address. This address changes according to the SAC and TrSiz settings of CCRn and the SACM setting of DTCRn.

Fig. 10.3.5 Source Address Register (SARn)

10.3.6 Destination Address Register (DARn) (n=0 through 7)

		7	6	5	4	3	2	1	0
DARn	bit Symbol	DAddr7	DAddr6	DAddr5	DAddr4	DAddr3	DAddr2	DAddr1	DAddr0
(0xFFFF_E20CH)	Read/Write	R/W							
(0xFFFF_E22CH)	After reset	0							
(0xFFFF_E24CH)	Function	See detailed description.							
(0xFFFF_E26CH)		15	14	13	12	11	10	9	8
(0xFFFF_E28CH)	bit Symbol	DAddr15	DAddr14	DAddr13	DAddr12	DAddr11	DAddr10	DAddr9	DAddr8
(0xFFFF_E2ACH)	Read/Write	R/W							
(0xFFFF_E2CCH)	After reset	0							
(0xFFFF_E2ECH)	Function	See detailed description.							
		23	22	21	20	19	18	17	16
	bit Symbol	DAddr23	DAddr22	DAddr21	DAddr20	DAddr19	DAddr18	DAddr17	DAddr16
	Read/Write	R/W							
	After reset	0							
	Function	See detailed description.							
		31	30	29	28	27	26	25	24
	bit Symbol	DAddr31	DAddr30	DAddr29	DAddr28	DAddr27	DAddr26	DAddr25	DAddr24
	Read/Write	R/W							
	After reset	0							
	Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31 : 0	DAddr	Destination address	Destination Address (initial value: 0) Specifies the address of the destination to which data is transferred using a physical address. This address changes according to the DAC and TrSiz settings of CCRn and the DACM setting of DTCRn.

Fig. 10.3.6 Destination Address Register (DARn)

10.3.7 Byte Count Registers (BCRn) (n=0 through 7)

		7	6	5	4	3	2	1	0
BCRn	bit Symbol	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
(0xFFFF_E210H)	Read/Write	R/W							
(0xFFFF_E230H)	After reset	0							
(0xFFFF_E250H)	Function	See detailed description.							
(0xFFFF_E270H)		15	14	13	12	11	10	9	8
(0xFFFF_E290H)	bit Symbol	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
(0xFFFF_E2B0H)	Read/Write	R/W							
(0xFFFF_E2D0H)	After reset	0							
(0xFFFF_E2F0H)	Function	See detailed description.							
		23	22	21	20	19	18	17	16
	bit Symbol	BC23	BC22	BC21	BC20	BC19	BC18	BC17	BC16
	Read/Write	R/W							
	After reset	0							
	Function	See detailed description.							
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function								

Bit	Mnemonic	Field name	Description
23 : 0	BC	Byte count	Byte Count (initial value: 0) Specifies the number of bytes of data to be transferred. The address decreases by the number of pieces of data transferred (a value specified by TrSiz of CCRn).

Fig. 10.3.7 Byte Count Register (BCRn)

10.3.8 DMA Transfer Control Register (DTCRn) (n=0 through 7)

		7	6	5	4	3	2	1	0
DTCRn	bit Symbol			DACM			SACM		
(0xFFFF_E218H)	Read/Write	R		R/W			R/W		
(0xFFFF_E238H)	After reset	0		0	0	0	0	0	0
(0xFFFF_E258H)	Function			See detailed description.			See detailed description.		
(0xFFFF_E278H)		15	14	13	12	11	10	9	8
(0xFFFF_E298H)	bit Symbol								
(0xFFFF_E2B8H)	Read/Write	R							
(0xFFFF_E2D8H)	After reset	0							
(0xFFFF_E2F8H)	Function								
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function								
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0							
	Function								

Bit	Mnemonic	Field name	Description
5 : 3	DACM	Destination address count mode	Destination Address Count Mode Specifies the count mode of the destination address. 000: Counting begins from bit 0 001: Counting begins from bit 4 010: Counting begins from bit 8 011: Counting begins from bit 12 100: Counting begins from bit 16 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
2 : 0	SACM	Source address count mode	Source Address Count Mode Specifies the count mode of the source address. 000: Counting begins from bit 0 001: Counting begins from bit 4 010: Counting begins from bit 8 011: Counting begins from bit 12 100: Counting begins from bit 16 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Fig. 10.3.8 DMA Transfer Control Register (DTCRn)

10.3.9 Data Holding Register (DHR)

DHR (0xFFFF_E30CH)

	7	6	5	4	3	2	1	0
bit Symbol	DOT7	DOT6	DOT5	DOT4	DOT3	DOT2	DOT1	DOT0
Read/Write	R/W							
After reset	0							
Function	See detailed description.							
	15	14	13	12	11	10	9	8
bit Symbol	DOT15	DOT14	DOT13	DOT12	DOT11	DOT10	DOT9	DOT8
Read/Write	R/W							
After reset	0							
Function	See detailed description.							
	23	22	21	20	19	18	17	16
bit Symbol	DOT23	DOT22	DOT21	DOT20	DOT19	DOT18	DOT17	DOT16
Read/Write	R/W							
After reset	0							
Function	See detailed description.							
	31	30	29	28	27	26	25	24
bit Symbol	DOT31	DOT30	DOT29	DOT28	DOT27	DOT26	DOT25	DOT24
Read/Write	R/W							
After reset	0							
Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31 : 0	DOT	Data on transfer	Data on Transfer (initial value: 0) Data that is read from the source in a dual-address data transfer mode.

Fig. 10.3.9 Data Holding Register (DHR)

10.4 Functions

10.4.1 Overview

The DMAC is a 32-bit DMA controller capable of transferring data in a system using the TX19A processor core at high speeds without routing data via the core.

(1) Source and destination

The DMAC handles data transfers from memory to memory and between memory and an I/O device. A device from which data is transferred is called a source device and a device to which data is transferred is called a destination device. Both memory and I/O devices can be designated as a source or destination device. The DMAC supports data transfers from memory to I/O devices, from I/O devices to memory, and from memory to memory, but not between I/O devices.

The differences between memory and I/O devices are in the way they are accessed. When accessing an I/O device, the DMAC asserts a $\overline{\text{DACKn}}$ signal. Because there is only one line per channel that carries a $\overline{\text{DACKn}}$ signal, the number of I/O devices accessible during data transfer is limited to one. Therefore, data cannot be transferred between I/O devices.

An interrupt factor can be attached to a transfer request to be sent to the DMAC. If an interrupt factor is generated, the interrupt controller (INTC) issues a request to the DMAC (the TX19A processor core is not notified of the interrupt request. For details, see description on Interrupts.). The request issued by the INTC is cleared by the $\overline{\text{DACKn}}$ signal. Therefore, if an I/O device is designated as a device to which data is to be transferred, a request made to the DMAC is cleared after completion of the data transfer (transfer of the amount of data specified by TrSiz). On the other hand, during memory-to-memory transfers, the $\overline{\text{DACKn}}$ signal is asserted only when the number of bytes transferred (value set in the BCRn register) becomes "0." Therefore, one transfer request allows data to be transferred successively without a pause.

For example, if data is transferred between a internal I/O and the internal (external) memory of the TMP19A64, a request made by the internal I/O to the DMAC is cleared after completion of each data transfer and the transfer operation is always put in a standby mode for the next transfer request if the number of bytes transferred (value set in the BCRn register) does not become "0." Therefore, the DMA transfer operation continues until the value of the BCRn register becomes "0."

(2) Bus control arbitration (bus arbitration)

In response to a transfer request made inside the DMAC, the DMAC requests the TX19A processor core to arbitrate bus control authority. When a response signal is returned from the core, the DMAC acquires bus control authority and executes a data transfer bus cycle.

In acquiring bus control for the DMAC, use or nonuse of the data bus of the TX19A processor core can be specified; specifically either snoop mode or non-snoop mode can be specified for each channel by using bit 11 (SReq) of the CCRn register.

There are cases in which the TX19A processor core requests the release of bus control authority. Whether or not to respond to this request can be specified for each channel by using the bit 10 (RelEn) of the CCRn register. However, this function can only be used in non-snoop mode (GREQ). In snoop mode (SREQ), the TX19A processor core cannot request the release of bus control and, therefore, this function cannot be used.

When there are no more transfer requests, the DMAC releases control of the bus.

(Note 1) When the DMAC is acquiring bus control authority, $\overline{\text{NMI}}$ is put on hold.

(Note 2) Do not bring the TX19A to a halt when the DMAC is in operation.

(Note 3) To put the TX19A into IDLE (doze) mode when the snoop function is being used, you must first stop the DMAC.

(3) Transfer request modes

Two transfer request modes are used for the DMAC: an internal transfer request mode and an external transfer request mode.

In the internal transfer request mode, a transfer request is generated inside the DMAC. Setting a start bit (Str bit of the channel control register CCRn) in the internal register of the DMAC to "1" generates a transfer request, and the DMAC starts to transfer data.

In the external transfer request mode, after a start bit is set to "1," a transfer request is generated when a transfer request signal $\overline{\text{INTDREQ}}$n output by the INTC is input, or when a transfer request signal $\overline{\text{DREQ}}$n output by an external device is input. For the DMAC, two modes are provided: the level mode in which a transfer request is generated when the "L" level of the $\overline{\text{INTDREQ}}$n signal is detected and a mode in which a transfer request is generated when the falling edge or "L" level of the $\overline{\text{DREQ}}$n signal is detected.

(4) Address mode

For the DMAC of the TMP19A64, only one address mode is provided: a dual address mode. A single address mode is not available.

In the dual address mode, data can be transferred from memory to memory and between memory and an I/O device. Source and destination device addresses are output by the DMAC. To access an I/O device, the DMAC asserts the $\overline{\text{DACK}}$n signal. In the dual address mode, two bus operations, a read and a write, are executed. Data that is read from a source device for transfer is first put into the data holding register (DHR) inside the DMAC and then written to a destination device.

(5) Channel operation

The DMAC has eight channels (channels 0 through 7). A channel is activated and put into a standby mode by setting a start (Str) bit in the channel control register (CCRn) to "1."

If a transfer request is generated when a channel is in a standby mode, the DMAC acquires bus control authority and transfers data. If there is no transfer request, the DMAC releases bus control authority and goes into a standby mode. If data transfer has been completed, a channel is put in an idle state. Data transfer is completed either normally or abnormally (e.g. occurrence of errors). An interrupt signal can be generated upon completion of data transfer.

Fig. 10.4.1 shows the state transitions of channel operation.

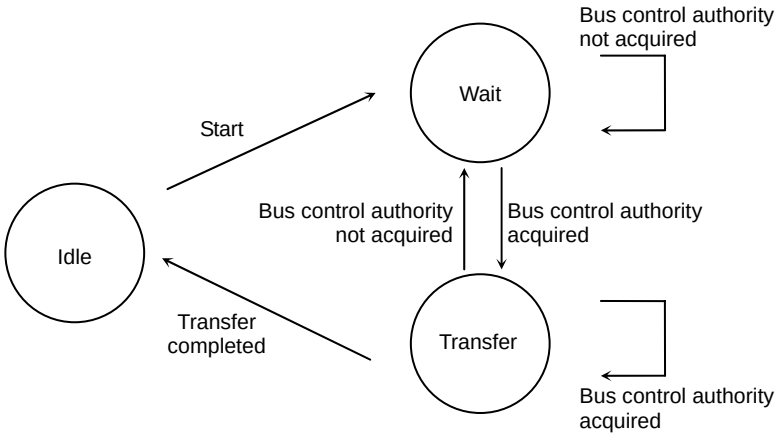


Fig. 10.4.1 Channel Operation State Transition

(6) Combinations of transfer modes

The DMAC can transfer data by combining each transfer mode as follows:

Transfer request	Edge/level	Address mode	Transfer devices
Internal	—	Dual	Memory → memory
External	"L" level ($\overline{\text{INTDREQn}}$)		Memory → memory
			Memory → I/O
			I/O → memory
External	"L" level ($\overline{\text{DREQn}}$)		Memory → memory
	Falling edge ($\overline{\text{DREQn}}$)		Memory → I/O
			I/O → memory

(7) Address changes

Address changes are broadly classified into three types: increases, decreases and fixed. The type of address change can be specified for each source and destination address by using SAC and DAC in the CCRn register. For a memory device, an increase, decrease or fixed can be specified. For an I/O device, however, only "fixed" can be specified. If an I/O device is selected as a source or destination device, SAC or DAC in the CCRn register must be set to "fixed."

If address increase or decrease is selected, the bit position for counting can be specified using SACM or DACM in the DTCRn register. To specify the bit position for counting a source address, SACM must be used, while DACM must be used to specify the bit position for a destination address. Any of the bits 0, 4, 8, 12 and 16 can be specified as the bit position for address counting. If 0 is selected, an address normally increases or decreases. By selecting bits 4, 8, 12 or 16, it is possible to increase or decrease an address irregularly.

Examples of address changes are shown below.

Example 1: Monotonic increase for a source device and irregular increase for a destination device

SAC: Address increase
DAC: Address increase
TrSiz: Transfer unit 32 bits
Source address: 0xA000_1000
Destination address: 0xB000_0000
SACM: 000 → counting to begin from bit 0 of the address counter
DACM: 001 → counting to begin from bit 4 of the address counter

	Source	Destination
1st	0xA000_1000	0xB000_0000
2nd	0xA000_1001	0xB000_0010
3rd	0xA000_1002	0xB000_0020
4th	0xA000_1003	0xB000_0030
…		…

Example 2: Irregular decrease for a source device and monotonic decrease for a destination device

SAC: Address decrease
DAC: Address decrease
TrSiz: Transfer unit 16 bits
Source address: Initial value 0xA000_1000
Destination address: 0xB000_0000
SACM: 010 → counting to begin from bit 8 of the address counter
DACM: 000 → counting to begin from bit 0 of the address counter

	Source	Destination
1st	0xA000_1000	0xB000_0000
2nd	0x9FFF_FF00	0xAFFF_FFFE
3rd	0x9FFF_FE00	0xAFFF_FFFC
4th	0x9FFF_FD00	0xAFFF_FFFA
	…	…

10.4.2 Transfer Request

For the DMAC to transfer data, a transfer request must be issued to the DMAC. There are two types of transfer request: an internal transfer request and an external transfer request. Either of these transfer requests can be selected and specified for each channel.

Whichever is selected, the DMAC acquires bus control authority and starts to transfer data if the transfer request is generated after the start of channel operation.

- Internal transfer request

  If the Str bit of CCR is set to "1" when the ExR bit of CCRn is "0," a transfer request is generated immediately. This transfer request is called an internal transfer request.

  The internal transfer request is valid until the channel operation is completed. Therefore, data can be transferred continuously if either of two events shown below does not occur:

  * A transition to a channel of higher priority

  * A shift of bus control authority to another bus master of higher priority

  In the case of the internal transfer request, data can only be transferred from memory to memory.

- External transfer request

  If the ExR bit of CCRn is "1," setting the Str bit of CCR to "1" allows a channel to go into a standby mode. The INTC or an external device then generates the $\overline{\text{INTDREQn}}$ or $\overline{\text{DREQn}}$ signal for this channel to notify the DMAC of a transfer request, and a transfer request is generated. This transfer request is called an external transfer request. In the case of the external transfer request, data can be transferred from memory to memory and between memory and an I/O device.

  The TMP19A64 recognizes the transfer request signal by detecting the "L" level of the $\overline{\text{INTDREQn}}$ signal or by detecting the falling edge or "L" level of the $\overline{\text{DREQn}}$ signal.

  The unit of data to be transferred in response to one transfer request is specified in the TrSiz field of CCRn, and 32, 16 or 8 bits can be selected.

  Transfer requests using $\overline{\text{INTDREQn}}$ and $\overline{\text{DREQn}}$ are described in detail on the next page.

① A transfer request made by the interrupt controller (INTC)

A transfer request made by the interrupt controller is cleared using the $\overline{\text{DACKn}}$ signal. This $\overline{\text{DACKn}}$ signal is asserted only if a bus cycle for an I/O device or the number of bytes (value set in the BCRn register) transferred from memory to memory becomes "0." Therefore, if data is transferred between memory and an I/O device, the amount of data specified by TrSiz is transferred only once because $\overline{\text{INTDREQn}}$ is cleared upon completion of one data transfer from one transfer request. On the other hand, if data is transferred from memory to memory, it can be transferred successively in response to a transfer request because $\overline{\text{INTDREQn}}$ is not cleared until the number of bytes transferred (value set in the BCRn register) becomes "0."

Note that if the DMAC acknowledges an interrupt set in $\overline{\text{INTDREQn}}$ and if this interrupt is cleared by the INTC before DMA transfer begins, there is a possibility that DMA transfer might be executed once after the interrupt is cleared, depending on the timing.

② A transfer request made by an external device

External pins ($\overline{\text{DREQ2}}$ and $\overline{\text{DREQ3}}$) are internally wired to allow them to function as pins of the port F and port J. These pins can be selected by setting the function control registers PFFC and PJFC to an appropriate setting. If both ports are set to use the DMAC function, the port F is given priority in using the DMAC function.

In the edge mode, the $\overline{\text{DREQn}}$ signal must be deasserted and then asserted for each transfer request to create an effective edge. In the level mode, however, successive transfer requests can be recognized by maintaining an effective level. In memory-to-memory transfer, only the "L" level mode can be used. In I/O-to-memory transfer, only the falling edge mode can be used.

– Level mode

In the level mode, the DMAC detects the "L" level of the $\overline{\text{DREQn}}$ signal upon the rising of the internal system clock. If it detects the "L" level of the $\overline{\text{DREQn}}$ signal when a channel is in a standby mode, it goes into transfer mode and starts to transfer data. To use the $\overline{\text{DREQn}}$ signal at an active level, the PosE bit (bit 13) of the CCRn register must be set to "0." The $\overline{\text{DACKn}}$ signal is active at the "L" level, as in the case of the $\overline{\text{DREQn}}$ signal.

If an external circuit asserts the $\overline{\text{DREQn}}$ signal, the $\overline{\text{DREQn}}$ signal must be maintained at the "L" level until the $\overline{\text{DACKn}}$ signal is asserted. If the $\overline{\text{DREQn}}$ signal is deasserted before the $\overline{\text{DACKn}}$ signal is asserted, a transfer request may not be recognized.

If the $\overline{\text{DREQn}}$ signal is not at the "L" level, the DMAC judges that there is no transfer request, and starts a transfer operation for other channels or releases bus control authority and goes into a standby mode.

The unit of a transfer request is specified in the TrSiz field (<bit3:2>) of the CCRn register.

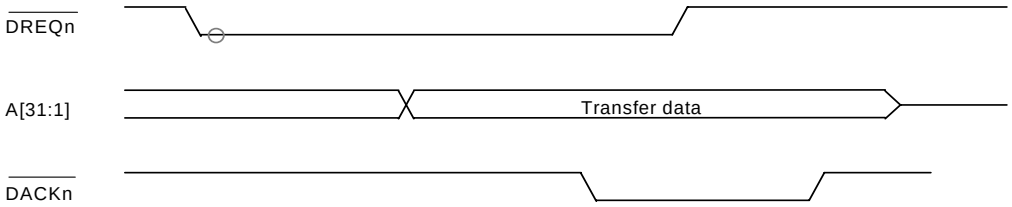


Fig. 10.4.2.1 Transfer Request Timing (Level Mode)

– Edge mode

In the edge mode, the DMAC detects the falling edge of the $\overline{\text{DREQn}}$ signal. If it detects the falling edge of the $\overline{\text{DREQn}}$ signal upon the rising of the internal system clock (the case in which the "L" level is detected upon the rising of the system clock although it was not detected upon the rising of the previous system clock) when a channel is in a standby mode, it judges that there is a transfer request, goes into transfer mode, and starts a transfer operation. To detect the falling edge of the $\overline{\text{DREQn}}$ signal, the PosE bit (bit 13) of the CCRn register must be set to "0," and the Lev bit (bit 12) must also be set to "0." The $\overline{\text{DACKn}}$ signal is active at the "L" level.

If the falling edge of the $\overline{\text{DREQn}}$ signal is detected after the $\overline{\text{DACKn}}$ signal is asserted, the next data is transferred without a pause.

If there is no falling edge of the $\overline{\text{DREQn}}$ signal after the $\overline{\text{DACKn}}$ signal is asserted, the DMAC judges that there is no transfer request, and starts a transfer operation for other channels or goes into a standby mode after releasing bus control authority.

The unit of a transfer request is specified in the TrSiz field (<bit3:2>) of the CCRn register.

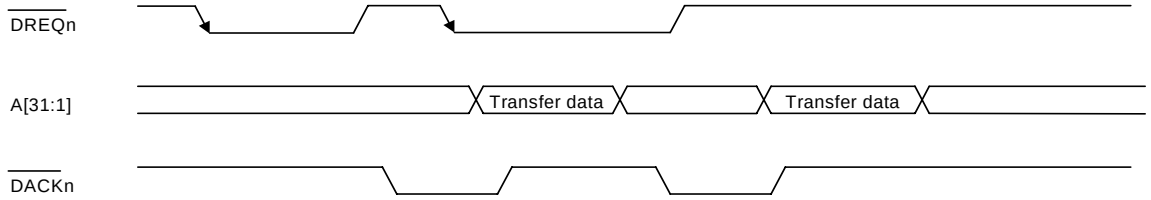


Fig. 10.4.2.2 Transfer Request Timing (Edge Mode)

10.4.3 Address Mode

In the address mode, whether the DMAC executes data transfers by outputting addresses to both source and destination devices or it does by outputting addresses to either a source device or a destination device is specified. The former is called the dual address mode, and the latter is called the single address mode. For TMP19A64, only the dual address mode is available.

In the dual address mode, The DMAC first performs a read of the source device by storing the data output by the source device in one of its registers (DHR). It then executes a write on the destination device by writing the stored data to the device, thereby completing the data transfer.

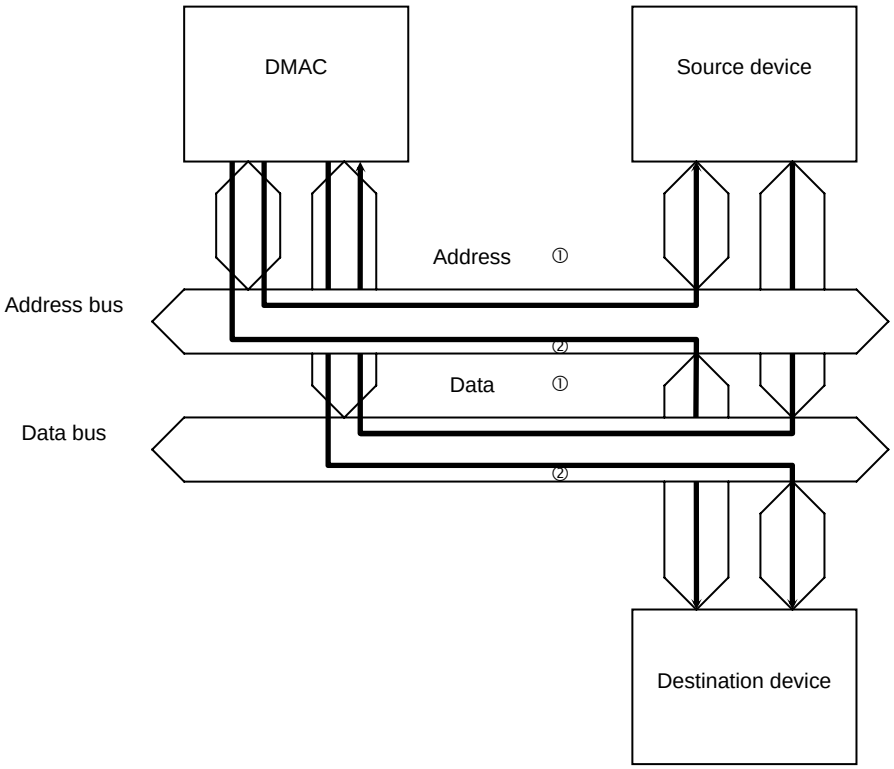


Fig. 10.4.3.1 Basic Concept of Data Transfer in the Dual Address Mode

The unit of data to be transferred by the DMAC is the amount of data (32, 16 or 8 bits) specified in the TrSiz field of the CCRn. One unit of data is transferred each time a transfer request is acknowledged.

In the dual address mode, the unit of data is read from the source device, put into the DHR and written to the destination device.

Access to memory takes place when the specified unit of data is transferred. If access to external memory takes place, 16-bit access takes place twice if the unit of data is set to 32 bits and if the bus width set in the CS wait controller is 16 bits. Likewise, if the unit of data is set to 32 bits and if the bus width set in the CS wait controller is 8 bits, 8-bit access takes place four times.

If data is to be transferred from memory to an I/O device or from an I/O device to memory, the unit of data to be transferred must be specified and, at the same time, the bus width of an I/O device (device port size) must be specified in the DPS field of the CCRn (32, 16 or 8 bits).

If the unit of data to be transferred is equal to a device port size, a read or write is executed once for an I/O device.

If a device port size is smaller than the unit of data to be transferred, the DMAC performs a read or write for an I/O device more than once. For example, if the unit of data to be transferred is 32 bits and if data is transferred from an I/O device whose device port size is 8 bits to memory, 8 bits of data are read from an I/O device four consecutive times and stored in the DHR. This 32-bit data is then written to memory all at once (twice if the data is written to external memory and if the bus width is 16 bits).

An address change occurs by the amount defined as the unit of data to be transferred. The BCRn value also changes by the same amount. A device port size must not be larger than the unit of data to be transferred. The relationships between units of data to be transferred and device port sizes are summarized in Table 10.4.3.2.

Table 10.4.3.2 Units of Data to Be Transferred and Device Port Sizes (Dual Address Mode)

TrSiz	DPS	Bus operations performed on I/O device
0x (32 bits)	0x (32 bits)	Once
0x (32 bits)	10 (16 bits)	Twice
0x (32 bits)	11 (8 bits)	4 times
10 (16 bits)	0x (32 bits)	Setting prohibited
10 (16 bits)	10 (16 bits)	Once
10 (16 bits)	11 (8 bits)	Twice
11 (8 bits)	0x (32 bits)	Setting prohibited
11 (8 bits)	10 (16 bits)	Setting prohibited
11 (8 bits)	11 (8 bits)	Once

10.4.4 Channel Operation

A channel is activated if the Str bit of the CCRn of a channel is set to "1." If a channel is activated, an activation check is conducted and if no error is detected, the channel is put into a standby mode.

If a transfer request is generated when a channel is in a standby mode, the DMAC acquires bus control authority and starts to transfer data.

Channel operation is completed either normally or abnormally (forced termination or occurrence of an error). Either normal completion or abnormal completion is indicated to the CSRn.

Start of channel operation

A channel is activated if the Str bit of the CCRn is set to "1."

When a channel is activated, a configuration error check is conducted and if no error is detected, the channel is put into a standby mode. If an error is detected, the channel is deactivated and this state of completion is considered to be abnormal completion. When a channel goes into a standby mode, the Act bit of the CSRn of that channel becomes "1."

If a channel is programmed to start operation in response to an internal transfer request, a transfer request is generated immediately and the DMAC acquires bus control authority and starts to transfer data. If a channel is programmed to start operation in response to an external transfer request, the DMAC acquires bus control authority after $\overline{\text{INTDREQn}}$ or $\overline{\text{DREQn}}$ is asserted, and starts to transfer data.

Completion of channel operation

A channel completes operation either normally or abnormally and either one of these states is indicated to the CSRn.

If an attempt is made to set the Str bit of the CCRn register to "1" when the NC or AbC bit of the CSRn register is "1," channel operation does not start and the completion of operation is considered to be abnormal completion.

Normal completion

Channel operation is considered to have been completed normally in the case shown below. For channel operation to be considered to have been completed normally, the transfer of a unit of data (value specified in the TrSiz field of CCRn) must be completed successfully.

- When the contents of BCRn become 0 and data transfer is completed

Abnormal completion

Cases of abnormal completion of DMAC operation are as follows:

- Completion due to a configuration error

  A configuration error occurs if there is a mistake in the DMA transfer setting. Because a configuration error occurs before data transfer begins, values specified in SARn, DARn and BCRn remain the same as when they were initially specified. If channel operation is completed abnormally due to a configuration error, the AbC bit of the CSRn is set to "1," along with the Conf bit. Causes of a configuration error are as follows:

  – Both SIO and DIO were set to "1."
  – The Str bit of CCRn was set to "1" when the NC bit or AbC bit of CSRn was "1."
  – A value that is not an integer multiple of the unit of data was set for BCRn.
  – A value that is not an integer multiple of the unit of data was set for SARn or DARn.
  – A prohibited combination of a device port size and a unit of data to be transferred was set.
  – The Str bit of CCRn was set to "1" when the BCRn value was "0."

- Completion due to a bus error

  If the DMAC operation has been completed abnormally due to a bus error, the AbC bit of CSRn is set to "1" and the BES or BED bit of CSRn is set to "1."

  – A bus error was detected during data transfer.

(Note) If the DMAC operation has been completed abnormally due to a bus error, BCR, SAR and DAR values cannot be guaranteed. If a bus error persists, refer to 21. "List of Functional Registers" which appear later in this document.

10.4.5 Order of Priority of Channels

Concerning the eight channels of the DMAC, the smaller the channel number assigned to each channel, the higher the priority. If a transfer request is generated to channels 0 and 1 simultaneously, a transfer request for channel 0 is processed with higher priority and the transfer operation is performed accordingly. When the transfer request for channel 0 is cleared, the transfer operation for channel 1 is performed if the transfer request still exists (An internal transfer request is retained if it is not cleared. The interrupt controller retains an external transfer request if the active state for an interrupt request assigned to DMA requests in the interrupt controller is set to edge mode. However, the interrupt controller does not retain an external transfer request if the active state is set to level mode. If the active state for an interrupt request assigned to DMA requests in the interrupt controller is set to level mode, it is necessary to continue asserting the interrupt request signal).

If a transfer request is generated when data is being transferred through channel 1, a channel transition occurs at channel 0, that is, data transfer through channel 1 is temporarily suspended and data transfer through channel 0 is started. When the transfer request for channel 0 is cleared, data transfer through channel 1 resumes.

Channel transitions occur upon the completion of data transfers (when the writing of all data in the DHR has been completed).

Interrupts

Upon completion of a channel operation, the DMAC can generate interrupt requests (INTDMAn: DMA transfer completion interrupt) to the TX19A processor core with two types of interrupts available: a normal completion interrupt and an abnormal completion interrupt.

- Normal completion interrupt

  If a channel operation is completed normally, the NC bit of CSRn is set to "1." If a normal completion interrupt is authorized for the NIEn bit of the CCRn, the DMAC requests the TX19A processor core to authorize an interrupt.

- Abnormal completion interrupt

  If a channel operation is completed abnormally, the AbC bit of CSRn is set to "1." If an abnormal completion interrupt is authorized for the AbIEn bit of the CCRn, the DMAC requests the TX19A processor core to authorize an interrupt.

10.5 Timing Diagrams

DMAC operations are synchronous to the rising edges of the internal system clock.

10.5.1 Dual Address Mode

- Memory-to-memory transfer

  Fig. 10.5.1.1 shows an example of the timing with which 16-bit data is transferred from one external memory (16-bit width) to another (16-bit width). Data is actually transferred successively until BCRn becomes "0."

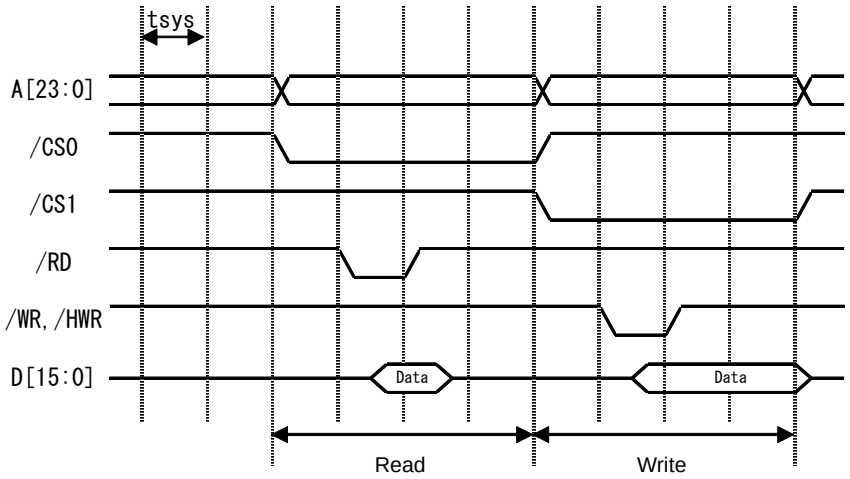


Fig. 10.5.1.1 Dual Address Mode (Memory-to-Memory)

- Memory-to-I/O device transfer

  Fig. 10.5.1.2 shows an example of the timing with which data is transferred from memory to an I/O device if the unit of data to be transferred is set to 16 bits and if the device port size is set to 8 bits.

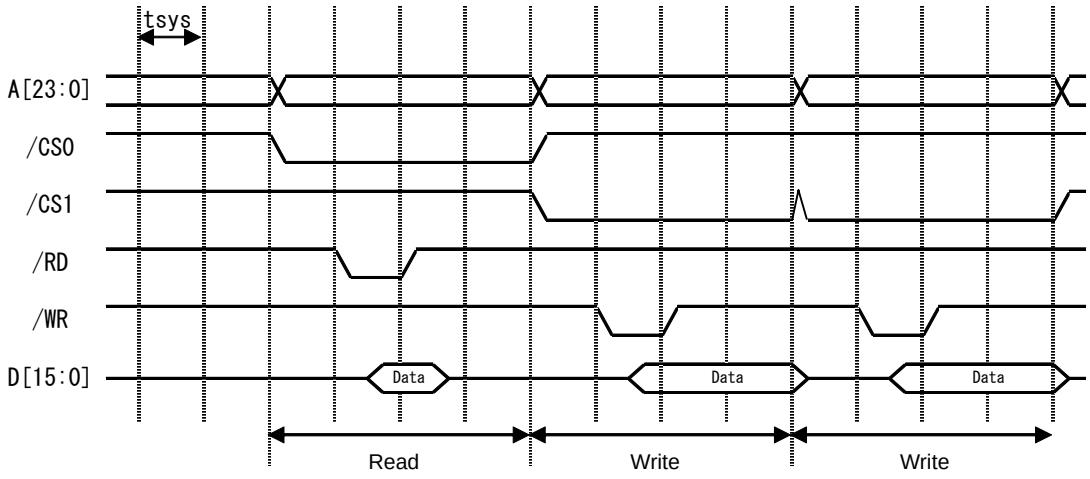


Fig. 10.5.1.2 Dual Address Mode (Memory-to-I/O Device)

- I/O device-to-memory transfer

  Fig. 10.5.1.3 shows an example of the timing with which data is transferred from an I/O device to memory if the unit of data to be transferred is set to 16 bits and if the device port size is set to 8 bits.

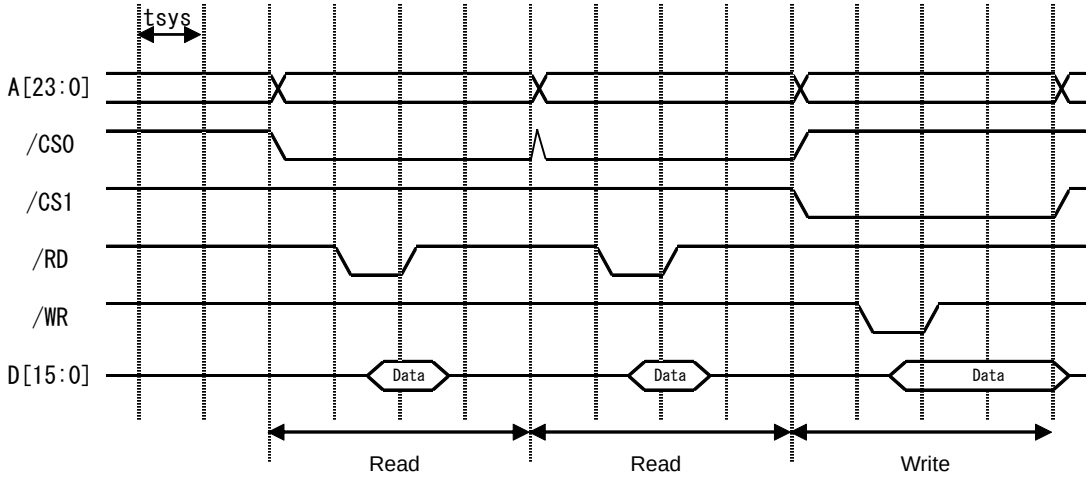


Fig. 10.5.1.3 Dual Address Mode (I/O Device-to-Memory)

10.5.2 DREQn-Initiated Transfer Mode

- Data transfer from internal RAM to external memory (multiplexed bus, 5-wait insertion, level mode)

  Fig. 10.5.2.1 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

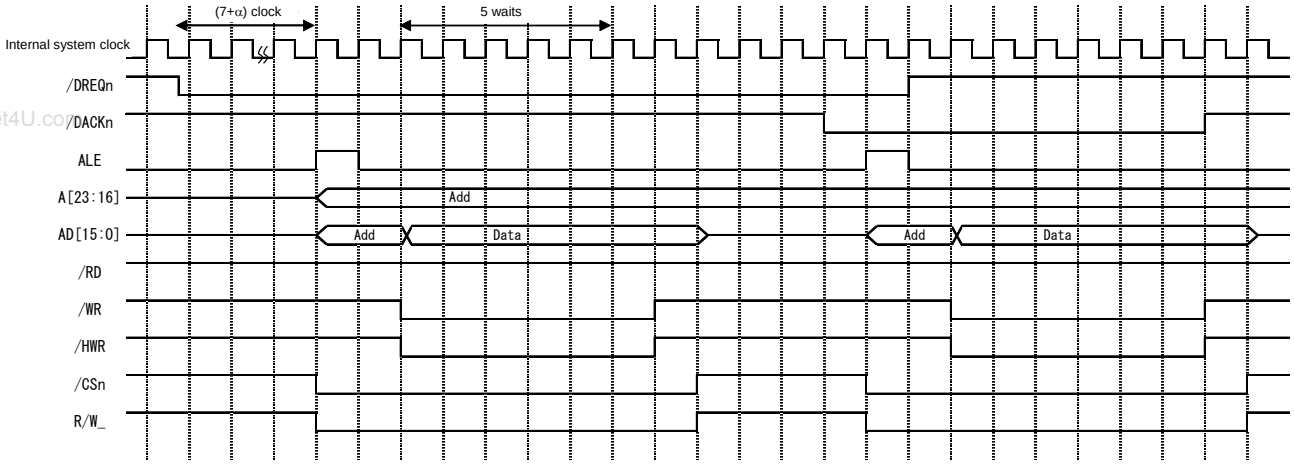


Fig. 10.5.2.1 Level Mode (from Internal RAM to External Memory)

- Data transfer from external memory to internal RAM (multiplexed bus, 5-wait insertion, level mode)

  Fig. 10.5.2.2 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bit width) to internal RAM.

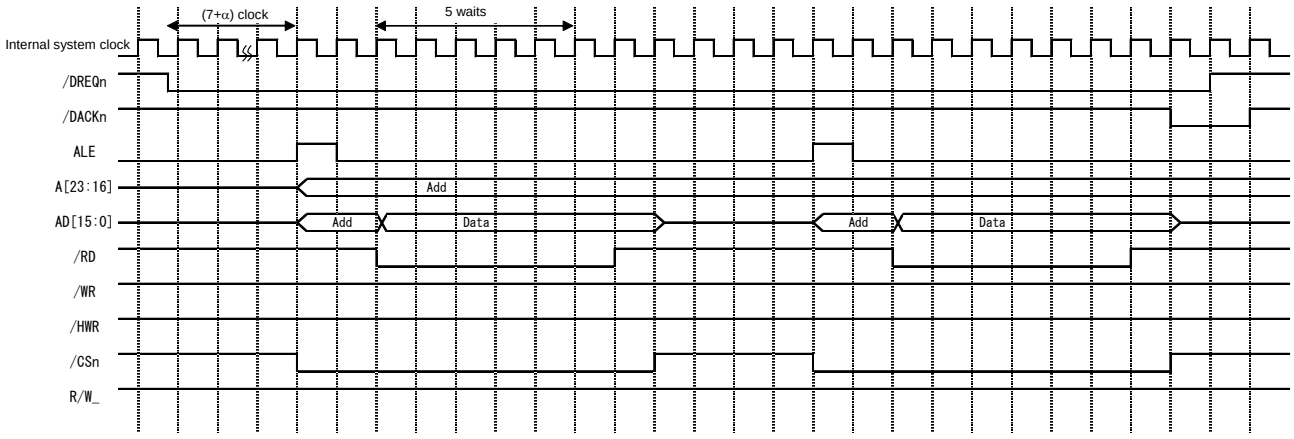


Fig. 10.5.2.2 Level Mode (from External Memory to Internal RAM)

- Data transfer from internal RAM to external memory (separate bus, 5-wait insertion, level mode)

  Fig. 10.5.2.3 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

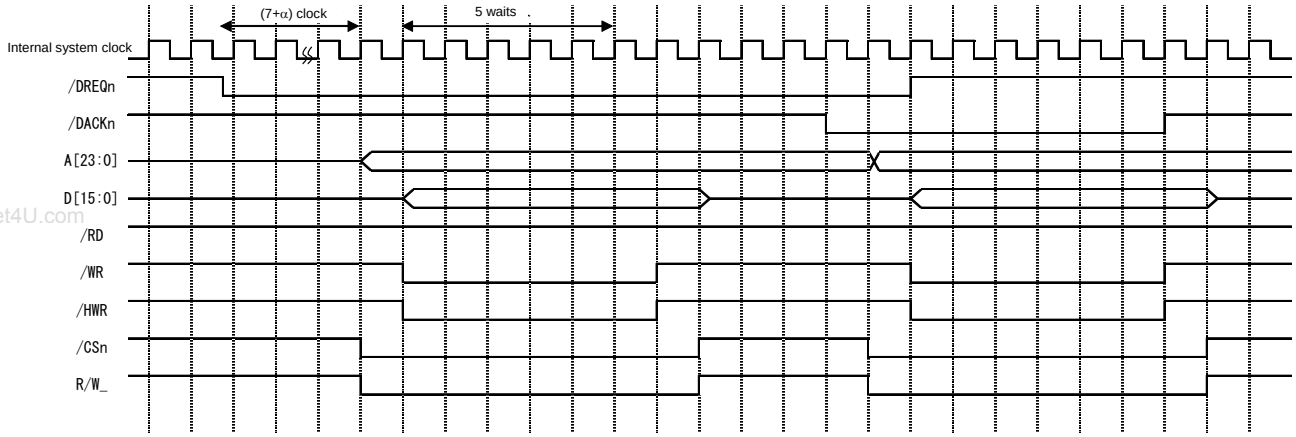


Fig. 10.5.2.3 Level Mode (Internal RAM to External Memory)

- Data transfer from external memory to internal RAM (separate bus, 5-wait insertion, level mode)

  Fig. 10.5.2.4 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bid width) to internal RAM.

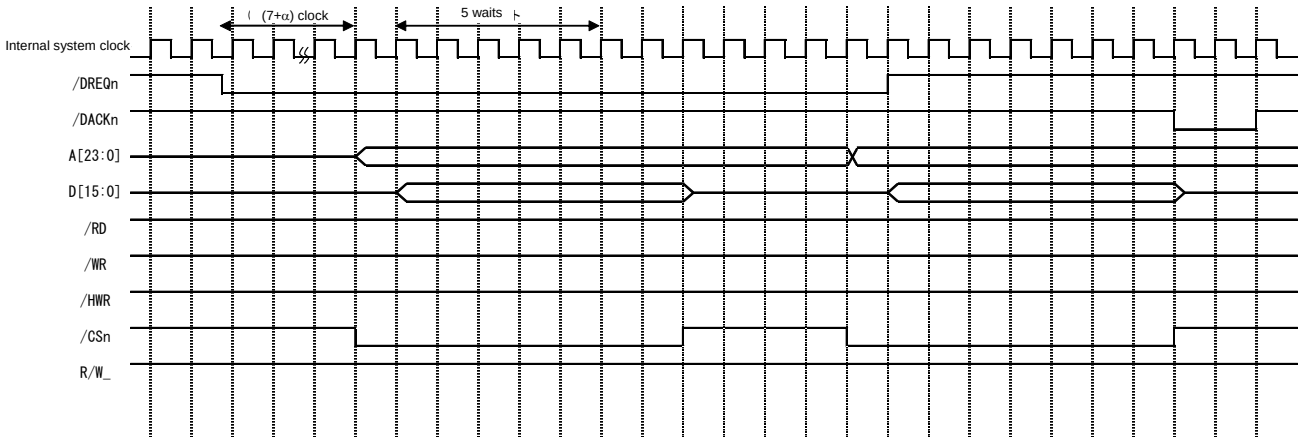


Fig. 10.5.2.4 Level Mode (from External Memory to Internal RAM)

- Data transfer from internal RAM to external memory (multiplexed bus, 5-wait insertion, edge mode)

  Fig. 10.5.2.5 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

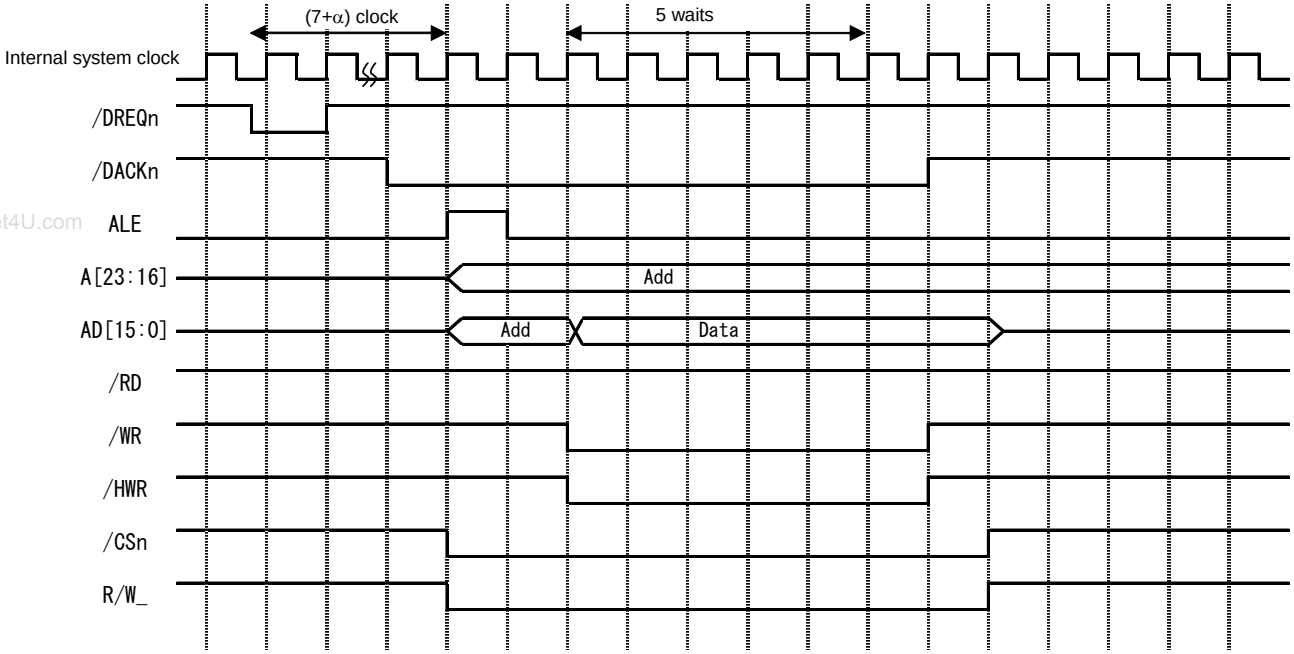


Fig. 10.5.2.5 Edge Mode (from Internal RAM to External Memory)

- Data transfer from external memory to internal RAM (multiplexed bus, 5-wait insertion, edge mode)

  Fig. 10.5.2.6 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

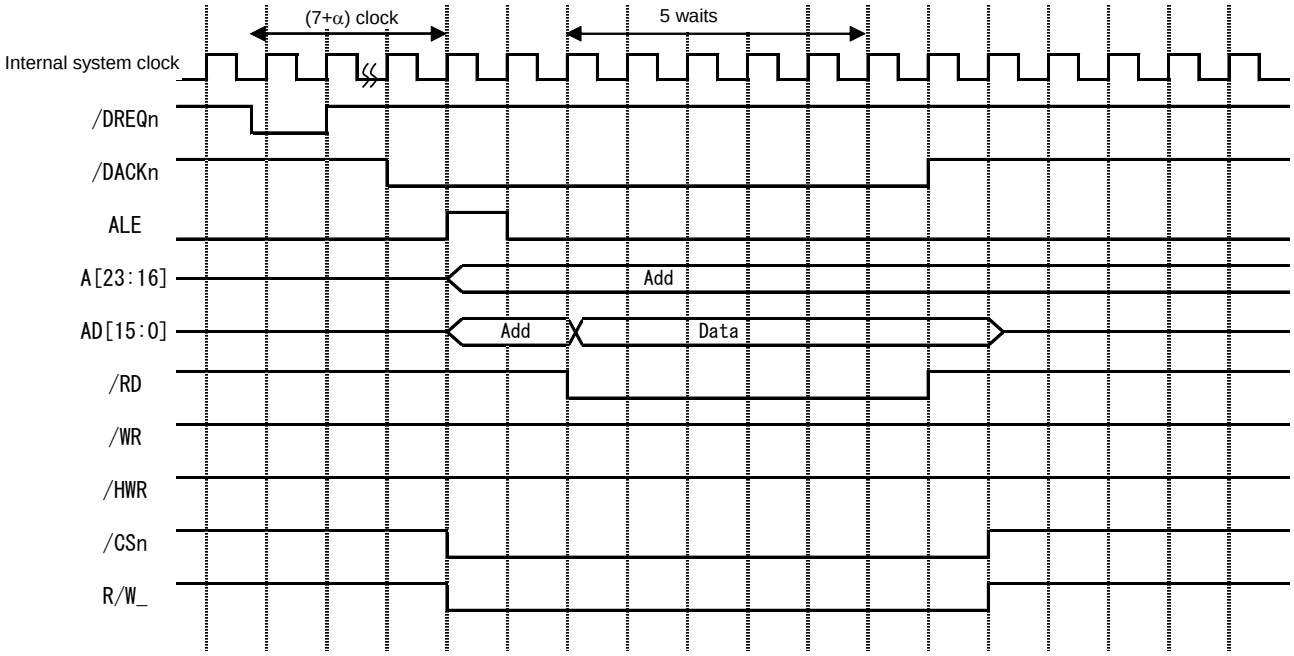


Fig. 10.5.2.6 Edge Mode (from External Memory to Internal RAM)

- Data transfer from internal RAM to external memory (separate bus, 5-wait insertion, edge mode)

  Fig. 10.5.2.7 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

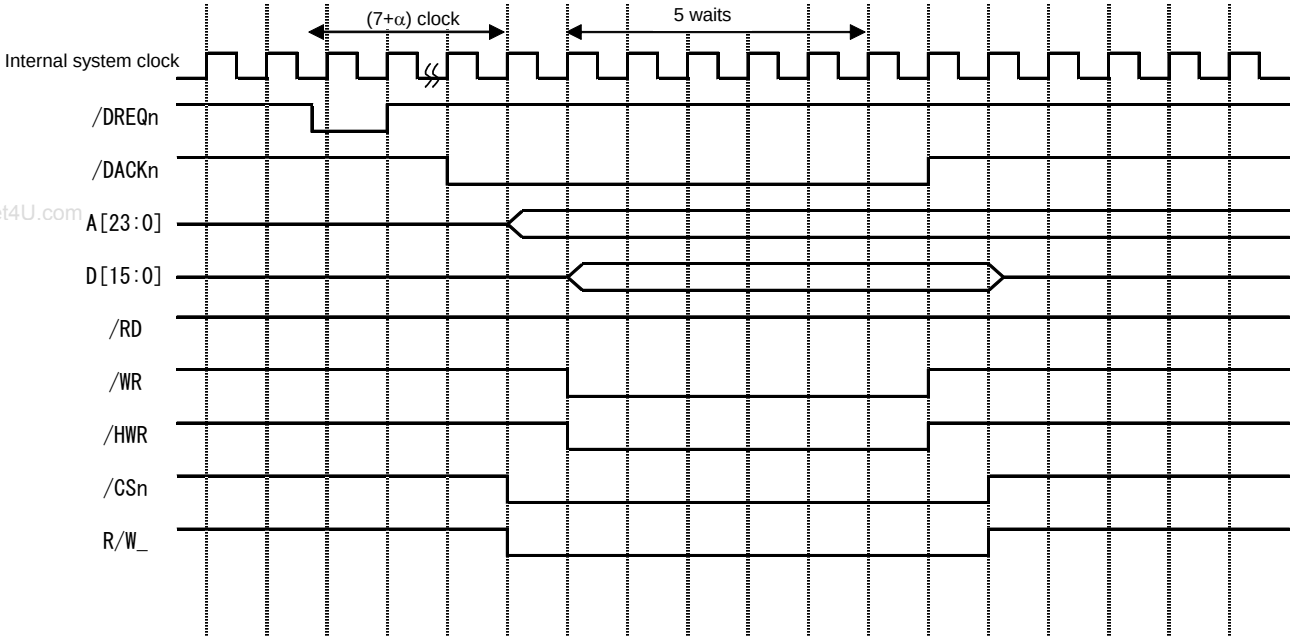


Fig. 10.5.2.7 Edge Mode (from Internal RAM to External Memory)

- Data transfer from external memory to internal RAM (separate bus, 5-wait insertion, edge mode)

  Fig. 10.5.2.8 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

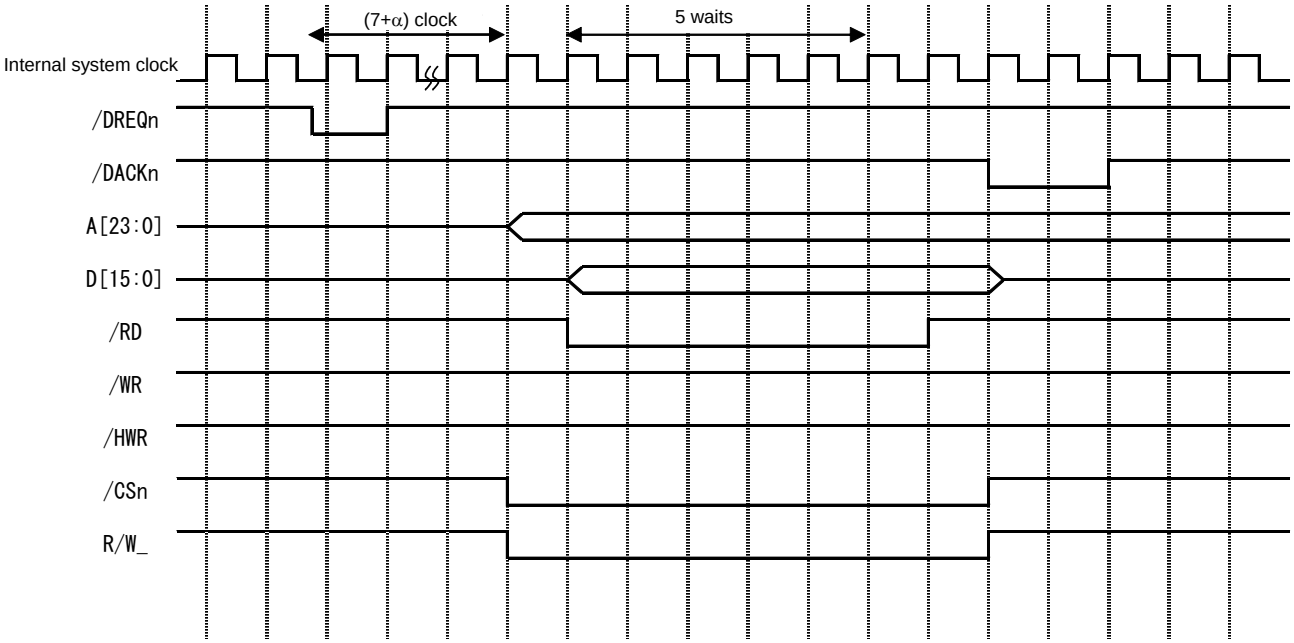


Fig. 10.5.2.8 Edge Mode (from External Memory to Internal RAM)

10.6 Case of Data Transfer

The settings described below relate to a case in which serial data received (SCnBUF) is transferred to the internal RAM by DMA transfer.

DMA (ch.0) is used to transfer data. The DMA0 is activated by a receive interrupt generated by SIO1.

<DMA setting>

- Channel used: 0
- Source address: SC1BUF
- Destination: (Physical address) 0xFFFF_9800
- Number of bytes transferred: 256 bytes

<Serial channel setting>

- Data length 8 bits: UART
- Serial channel: ch 1
- Transfer rate: 9600 bps

<SIO ch.1 setting>

```
IMC4      ←    0xxxxx_xx70    /* assigned to DMC0 activation factor * /
INTCLR    ←    0x40           /* IVR [9:4], INTRX1 interrupt factor * /
SC1MOD0   ←    0x29           /* UART mode, 8-bit length, baud rate generator * /
SC1CR     ←    0x00
BR1CR     ←    0x1F           /* @fc=54MHz, Transfer rate setting */
```

<DMA0 setting>

```
DCR       ←    0x8000_0000    /* DMA reset * /
IMCE      ←    0xxxxx_xx40    /* Disable interrupt setting */
INTCLR    ←    0xE0           /* IVR [8:0] value * /
IMCE      ←    0xxxxx_xx44    /* level = 4 (any given value) */
DTCR0     ←    0x0000_0000    /* DACM = 000 */
                              /* SACM = 000 */
SAR0      ←    0xFFFF_F208    /* physical address of SC1BUF */
DAR0      ←    0xFFFF_9800    /* physical address of destination to which data is transferred */
BCR0      ←    0x0000_00FF    /* 256 (number of bytes transferred) /
CCR0      ←    0x80c0_5B0f    /* DMA ch.0 setting */
```

11. 16-bit Timer/Event Counters (TMRBs)

Each of the eleven channels (TMRB0 through TMRBA) has a multi-functional, 16-bit timer/event counter. TMRBs operate in the following four operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square-wave output (PPG) mode
- Two-phase pulse input counter mode (quad-speed and TMRBA)

The use of the capture function allows TMRBs to operate in three other modes:

- Frequency measurement mode
- Pulse width measurement mode
- Time difference measurement mode

Each channel consists of a 16-bit up-counter, two 16-bit timer registers (one of which is double-buffered), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit.

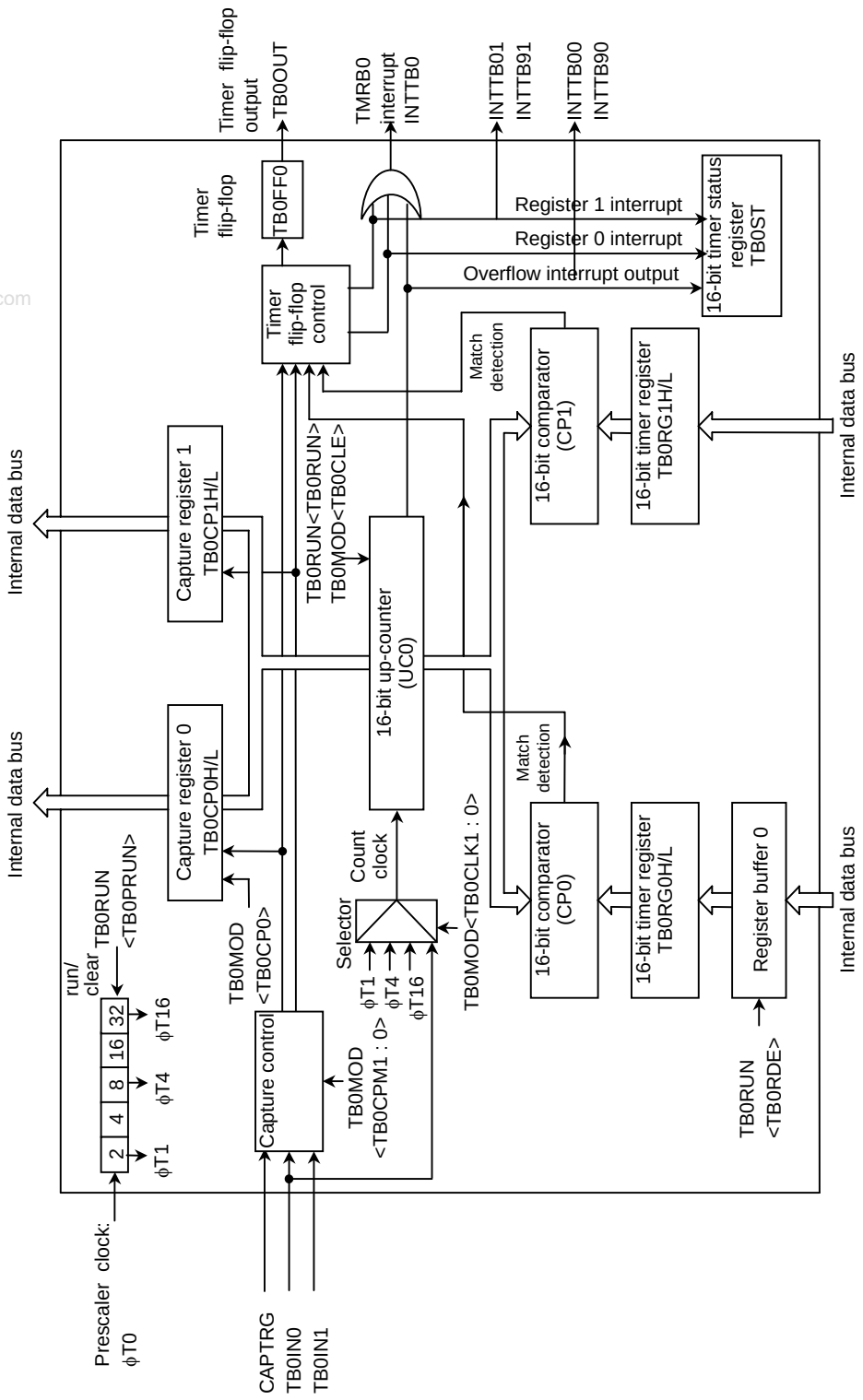
Each channel (TMRB0 through TMRBA) functions independently and while the channels operate in the same way, there are differences in their specifications as shown in Table 11.1 and the two-phase pulse count function. Therefore, the operational descriptions here are for TMRB0 only and for the two-phase pulse count function TMRBA only.

Table 11.1 Differences in the Specifications of TMRB Modules

Specification \ Channel		TMRB0	TMRB1	TMRB2	TMRB3	TMRB4	TMRB5
External pins	External clock/ capture trigger input pins	TB0IN0 (shared with PA0) TB0IN1 (shared with PA1)	TB1IN0 (shared with PA3) TB1IN1 (shared with PA4)	–	–	–	–
	Timer flip-flop output pin	TB0OUT (shared with PA2)	TB1OUT (shared with PA5)	TB2OUT (shared with PA6)	TB3OUT (shared with PA7)	TB4OUT (shared with PB0)	TB5OUT (shared with PB1)
Internal signals	Timer for capture triggers	TB9OUT	TB9OUT	TB9OUT	TB9OUT	TB9OUT	TB3OUT
Register names	Timer RUN register	TB0RUN	TB1RUN	TB2RUN	TB3RUN	TB4RUN	TB5RUN
	Timer control register	TB0CR	TB1CR	TB2CR	TB3CR	TB4CR	TB5CR
	Timer mode register	TB0MOD	TB1MOD	TB2MOD	TB3MOD	TB4MOD	TB5MOD
	Timer flip-flop control register	TB0FFCR	TB1FFCR	TB2FFCR	TB3FFCR	TB4FFCR	TB5FFCR
	Timer status register	TB0ST	TB1ST	TB2ST	TB3ST	TB4ST	TB5ST
	Timer UC preset register	TB0UCL TB0UCH	TB1UCL TB1UCH	TB2UCL TB2UCH	TB3UCL TB3UCH	TB4UCL TB4UCH	TB5UCL TB5UCH
	Timer register	TB0RG0L TB0RG0H TB0RG1L TB0RG1H	TB1RG0L TB1RG0H TB1RG1L TB1RG1H	TB2RG0L TB2RG0H TB2RG1L TB2RG1H	TB3RG0L TB3RG0H TB3RG1L TB3RG1H	TB0RG0L TB4RG0H TB4RG1L TB4RG1H	TB5RG0L TB5RG0H TB5RG1L TB5RG1H
	Capture register	TB0CP0L TB0CP0H TB0CP1L TB0CP1H	TB1CP0L TB1CP0H TB1CP1L TB1CP1H	TB2CP0L TB2CP0H TB2CP1L TB2CP1H	TB3CP0L TB3CP0H TB3CP1L TB3CP1H	TB4CP0L TB4CP0H TB4CP1L TB4CP1H	TB5CP0L TB5CP0H TB5CP1L TB5CP1H

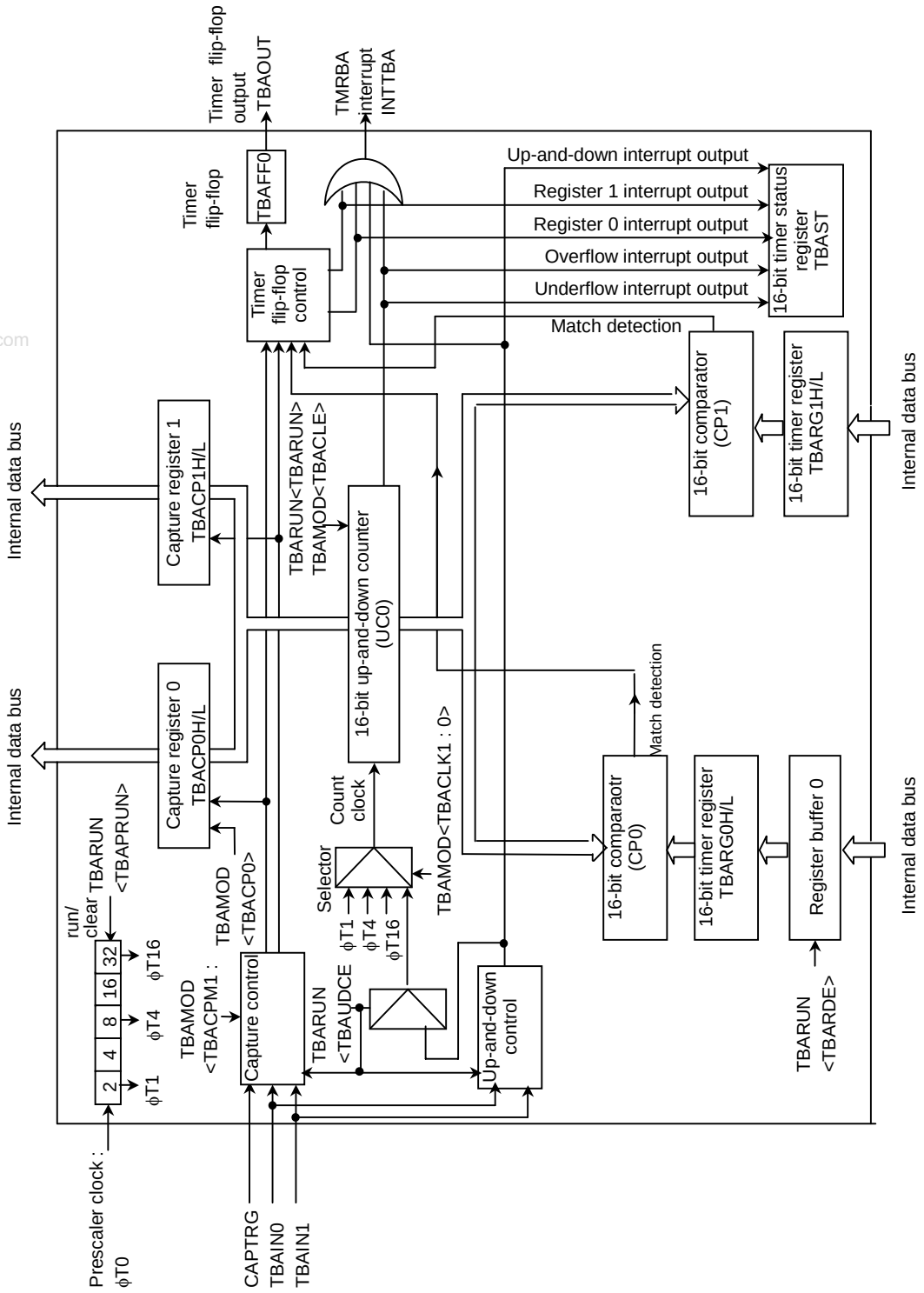
Specification \ Channel		TMRB6	TMRB7	TMRB8	TMRB9	TMRBA
External pins	External clock/ capture trigger input pins	–	–	–	–	TBAIN0 (shared with PB6) TBAIN1 (shared with PB7)
	Timer flip-flop output pin	TB6OUT (shared with PB2)	TB7OUT (shared with PB3)	TB8OUT (shared with PB4)	TB9OUT (shared with PB5)	–
Internal signals	Timer for capture triggers	TB3OUT	TB3OUT	TB3OUT	TB3OUT	TB3OUT
Register names	Timer RUN register	TB6RUN	TB7RUN	TB8RUN	TB9RUN	TBARUN
	Timer control register	TB6CR	TB7CR	TB8CR	TB9CR	TBACR
	Timer mode register	TB6MOD	TB7MOD	TB8MOD	TB9MOD	TBAMOD
	Timer flip-flop control register	TB6FFCR	TB7FFCR	TB8FFCR	TB9FFCR	TBAFFCR
	Timer status register	TB6ST	TB7ST	TB8ST	TB9ST	TBAST
	Timer UC preset register	TB6UCL TB6UCH	TB7UCL TB7UCH	TB8UCL TB8UCH	TB9UCL TB9UCH	TBAUCL TBAUCH
	Timer register	TB6RG0L TB6RG0H TB6RG1L TB6RG1H	TB7RG0L TB7RG0H TB7RG1L TB7RG1H	TB8RG0L TB8RG0H TB8RG1L TB8RG1H	TB9RG0L TB9RG0H TB9RG1L TB9RG1H	TBARG0L TBARG0H TBARG1L TBARG1H
	Capture register	TB6CP0L TB6CP0H TB6CP1L TB6CP1H	TB7CP0L TB7CP0H TB7CP1L TB7CP1H	TB8CP0L TB8CP0H TB8CP1L TB8CP1H	TB9CP0L TB9CP0H TB9CP1L TB9CP1H	TBACP0L TBACP0H TBACP1L TBACP1H

11.1 Block Diagram of Each Channel



(Note) TMRB2 through TMRB9 have no external clock and capture trigger input functions.

Fig. 11.1.1 TMRB0 Block Diagram (Same for Channels 1 through 9)



(Note) There is no TBAOUT external output.

Fig. 11.1.2 TMRBA Block Diagram

11.2 Description of Operations for Each Circuit

11.2.1 Prescaler

There is a 5-bit prescaler for acquiring the TMRB0 source clock. The prescaler input clock ϕT0 is fperiph/2, fperiph/4, fperiph/8 or fperiph/16 selected by SYSCR0<PRCK1:0> in the CG. The peripheral clock, fperiph, is either fgear, a clock selected by SYSCR1<FPSEL> in the CG, or fc, which is a clock before it is divided by the clock gear.

The operation or the stoppage of a prescaler is set with TB0RUN<TB0PRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 11.2.1 shows prescaler output clock resolutions.

Table 11.2.1 Prescaler Output Clock Resolutions

@fc = 54MHz

Release peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1 : 0>	Prescaler output clock resolutions		
			ϕT1	ϕT4	ϕT16
0 (fgear)	000 (fc)	00(fperiph/16)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^3(0.15 μs)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	fc/2^2(0.07 μs)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)
	100 (fc/2)	00(fperiph/16)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)	fc/2^{10}(18.96 μs)
		01(fperiph/8)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		10(fperiph/4)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
		11(fperiph/2)	fc/2^3(0.15 μs)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)
	110 (fc/4)	00(fperiph/16)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)	fc/2^{11}(37.93 μs)
		01(fperiph/8)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)	fc/2^{10}(18.96 μs)
		10(fperiph/4)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		11(fperiph/2)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
	111 (fc/8)	00(fperiph/16)	fc/2^8(4.74 μs)	fc/2^{10}(18.96 μs)	fc/2^{12}(75.85 μs)
		01(fperiph/8)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)	fc/2^{11}(37.93 μs)
		10(fperiph/4)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)	fc/2^{10}(18.96 μs)
		11(fperiph/2)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
1 (fc)	000 (fc)	00(fperiph/16)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^3(0.15 μs)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	fc/2^2(0.07 μs)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)
	100 (fc/2)	00(fperiph/16)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^3(0.15 μs)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	—	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)
	110 (fc/4)	00(fperiph/16)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	—	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	—	fc/2^4(0.30 μs)	fc/2^6(1.19 μs)
	111 (fc/8)	00(fperiph/16)	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	—	fc/2^6(1.19 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	—	fc/2^5(0.59 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	—	—	fc/2^6(1.19 μs)

(Note 1) The prescaler output clock ϕTn must be selected so that ϕTn<fsys/2 is satisfied (so that ϕTn is slower than fsys/2).

(Note 2) Do not change the clock gear while the timer is operating.

(Note 3) "—" denotes a setting prohibited.

11.2.2 Up-counter (UC0) and Up-counter Capture Registers (TB0UCL, TB0UCH)

This is the 16-bit binary counter that counts up in response to the input clock specified by TB0MOD<TB0CLK1:0>.

UC0 input clock can be selected from either three types - ϕT0, ϕT2 and ϕT8 - of prescaler output clock or the external clock of the TB0IN0 pin. For UC0, start, stop and clear are specified by TB0RUN<TB0RUN> and if UC0 matches the TB0RG1H/L timer register, it is cleared to "0" if the setting is "clear enable." Clear enable/disable is specified by TB0MOD<TB0CLE>.

If the setting is "clear disable," the counter operates as a free-running counter.

The current count value of the UC0 can be captured by reading the TB0UCL and TB0UCH registers.

Note Make sure that reading is performed in the order of low-order bits followed by high-order bits.

If UC0 overflow occurs, the INTTB01 overflow interrupt is generated.

TMRBA have the two-phase pulse input count function. The two-phase pulse count mode is activated by TBARUN<TBAUDCE>. This counter serves as the up-and-down counter, and is initialized to 0x7FFF. If a counter overflow occurs, the initial value 0x0000 is reloaded. If a counter underflow occurs, the initial value 0xFFFF is reloaded. When the two-phase pulse count mode is not active, the counter counts up only.

11.2.3 Timer Registers (TB0RG0H/L, TB0RG1H/L)

These are 16-bit registers for specifying counter values and two registers are built into each channel. If a value set on this timer register matches that on a UC0 up-counter, the match detection signal of the comparator becomes active.

To write data to the TB0RG0H/L and TB0RG1H/L timer registers, either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits can be used.

TB0RG0 of this timer register is paired with register buffer 0 - a double-buffered configuration. TB0RG0 uses TB0RUN<TB0RDE> to control the enabling/disabling of double buffering so that if <TB0RDE> = "0," double buffering is disabled and if <TB0RDE> = "1," it is enabled. If double buffering is enabled, data is transferred from register buffer 0 to the TB0RG0 timer register when there is a match between UC0 and TB0RG1.

The values of TB0RG0 and TB0RG1 become undefined after a reset so to use a 16-bit timer, it is necessary to write data to them beforehand. A reset initializes TB0RUN <TB0RDE> to "0" and sets double buffering to "disable." To use double buffering, write data to the timer register, set <TB0RDE> to "1" and then write the following data to the register buffers.

TB0RG0 and the register buffers are assigned to the same address: 0xFFFF_F18A/0xFFFF_F18B. If <TB0RDE> = "0," the same value is written to TB0RG0 and each register buffer; if <TB0RDE> = "1," the value is only written to each register buffer. To write an initial value to the timer register, therefore, the register buffers must be set to "disable."

11.2.4 Capture Registers (TB0CP0H/L, TB0CP1H/L)

To read data from the capture register, use 1-byte data transfer instruction twice and **make sure that reading is performed in the order of low-order bits followed by high-order bits**.

(Don't use 2-byte transfer instruction for data reading.)

11.2.5 Capture

This is a circuit that controls the timing of latching values from the UC0 up-counter into the TB0CP0 and TB0CP1 capture registers. The timing with which to latch data is specified by TB0MOD <TB0CPM1:0>.

Software can also be used to import values from the UC0 up-counter into the capture register; specifically, UC0 values are taken into the TB0CP0 capture register each time "0" is written to TB0MOD<TB0CP0>. To use this capability, the prescaler must be running (TB0RUN<TB0PRUN> = "1").

In the two-phase pulse count mode (TMRBA), the counter value is captured by using software.

(Note 1) Although a read of low-order 8 bits in the capture register suspends the capture operation, it is resumed by successively reading high-order 8 bits.

(Note 2) If the timer stops after a read of low-order 8 bits, the capture operation remains suspended even after the timer restarts. Please ensure that the timer is not stopped after a read of low-order 8 bits.

11.2.6 Comparators (CP0, CP1)

These are 16-bit comparators for detecting a match by comparing set values of the UC0 up-counter with set values of the TB0RG0 and TB0RG1 timer registers. If a match is detected, INTTB0 is generated.

11.2.7 Timer Flip-flop (TB0FF0)

The timer flip-flop (TB0FF0) is reversed by a match signal from the comparator and a latch signal to the capture registers. It can be enabled or disabled to reverse by setting the TB0FFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>.

The value of TB0FF0 becomes undefined after a reset. The flip-flop can be reversed by writing "00" to TB0FFCR<TB0FF0C1:0>. It can be set to "1" by writing "01," and can be cleared to "0" by writing "10."

The value of TB0FF0 can be output to the timer output pin, TB0OUT (shared with PA2). To enable timer output, the port A related registers PACR and PAFC must be programmed beforehand.

11.3 Register Description

TMRBn RUN register (n=0 through 9)

TBnRUN (0xFFFF_F1x0)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRDE				I2TBn	TBnPRUN		TBnRUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset						0	0	0
Function	Double Buffering 0: Disable 1: Enable	Write "0."	Write "0."	Write "0."	IDLE 0: Stop 1: Operate	Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as "0."		

<TBnRUN>: Controls the TMRBn count operation.

<TBnPRUN>: Controls the TMRBn prescaler operation.

<I2TBn>: Controls the operation in the IDLE mode.

<TBnRDE>: Controls enabling/disabling of double buffering.

TMRBA RUN register

TBARUN (0xFFFF_F1E0)

	7	6	5	4	3	2	1	0
bit Symbol	TBARDE		UDACK	TBAUDCE	I2TBA	TBAPRUN		TBARUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffering 0: Disable 1: Enable	Write "0."	Sampling clock 0: fs 1: ϕT0/4	Enable/ disable two-phase counter 0: Disable 1: Enable	IDLE 0: Stop 1: Operate	Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as "0."		

<TBARUN>: Controls the TMRBA count operation.

<TBAPRUN>: Controls the TMRBA prescaler operation.

<I2TBA>: Controls the operation in the IDLE mode.

<TBAUDCE>: Controls enabling/disabling of the two-phase pulse input count operation.

Enable: The counter counts up and counts down.

Disable: This is the normal timer mode and the counter counts up only.

<UDACK>: Selects the two-phase pulse input sampling clock.

<TBARDE>: Controls enabling/disabling of double buffering.

TMRBn control register (n=0 through A)

TBnCR (0xFFFF_F1x1)

	7	6	5	4	3	2	1	0
bit Symbol	TBnEN							
Read/Write	R/W	R/W	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	TMRBn operation 0: Disable 1: Enable	Write "0."	This can be read as "0."	This can be read as "0."	This can be read as "0."	This can be read as "0."	This can be read as "0."	This can be read as "0."

<TBnEN>: Specifies the TMRB operation. When the operation is disabled, no clock is supplied to the other registers in the TMRB module. This can reduce power dissipation. (This disables reading from and writing to the other registers.) To use the TMRB, enable the TMRB operation (set to "1") before programming each register in the TMRB module. If the TMRB operation is executed and then disabled, settings will be maintained in each register.

TMRBn mode register (n=0 through A)

TBnMOD (0xFFFF_F1x2)

	7	6	5	4	3	2	1	0
bit Symbol			TBnCP0	TBnCPM1	TBnCPM0	TBnCLE	TBnCLK1	TBnCLK0
Read/Write	R		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	This can be read as "00."		Capture control by software 0: Capture by software 1: Don't care	Capture timing 00: Disable 01: TBnIN0 ↑ TBnIN1 ↑ 10: TBnIN0 ↑ TBnIN0 ↓ 11: CAPTRG ↑ CAPTRG ↓		Up-counter control 0: Clear/disable 1: Clear/enable	Selects source clock 00: TB0IN0 pin input 01: φT1 10: φT4 11: φT16	

<TBnCLK1:0>: Selects the TMRBn timer count clock.

<TBnCLE>: Clears and controls the TMRBn up-counter.

"0": Disables clearing of the up-counter.

"1": Clears up-counter if there is a match with timer register 1 (TBnRG1).

<TBnCPM1:0>: Specifies TMRBn capture timing.

"00": Capture disable

"01": Takes count values into capture register 0 (TBnCP0) upon the rising of TBnIN0 pin input. Takes count values into capture register 1 (TBnCP1) upon the rising of TBnIN1 pin input.

"10": Takes count values into capture register 0 (TBnCP0) upon the rising of TBnIN0 pin input. Takes count values into capture register 1 (TBnCP1) upon the falling of TBnIN0 pin input.

"11": Takes count value into capture register 0 (TBnCP0) upon the rising of the timer output for capture trigger (CAPTRG) and into capture register 1 (TBnCP1) upon the falling of CAPTRG (TB9OUT serves as CAPTRG for TMRB0 through TMRB4, and TB3OUT serves for TMRB5 through TMRBA.)

<TBnCP0>: Captures count values by software and takes them into capture register 0 (TBnCP0).

(Note) The value read from bit 5 of TBnMOD is "1."

TMRBn flip-flop control register (n=0 through A)

TBnFFCR (0xFFFF_F1x3)

	7	6	5	4	3	2	1	0
bit Symbol			TBnC1T1	TBnC0T1	TBnE1T1	TBnE0T1	TBnFF0C1	TBnFF0C0
Read/Write	R		R/W				W	
After reset							1	1
Function	This is always read as "11."		TBnFF0 reverse trigger 0: Disable trigger 1: Enable trigger				TBnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * This is always as "11."	
			When the up-counter value is taken into TBnCP1	When the up-counter value is taken into TBnCP0	When the up-counter matches TBnRG1	When the up-counter matches TBnRG0		

<TBnFF0C1:0>: Controls the timer flip-flop.

"00": Reverses the value of TBnFF0 (reverse by using software).

"01": Sets TBnFF0 to "1."

"10": Clears TBnFF0 to "0."

"11": Don't care

(Note) Always read as "11."

<TBnE1:0>: Reverses the timer flip-flop when the up-counter matches the timer register 0,1 (TBnRG0,1).

<TBnC1:0>: Reverses the timer flip-flop when the up-counter value is taken into the capture register 0,1 (TBnCP0,1).

TMRBn status register (1)

TMRBn status register (n=0 through 9)

TBnST (0xFFFF_F1x4)

	7	6	5	4	3	2	1	0
bit Symbol						INTTBOFn	INTTBn1	INTTBn0
Read/Write	R					R		
After reset	0					0	0	0
Function	This can be read as "0."					0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

<INTTBn0>: Interrupt generated if there is a match with timer register 0 (TBnRG0)
<INTTBn1>: Interrupt generated if there is a match with timer register 1 (TBnRG1)
<INTTBOFn>: Interrupt generated if an up-counter overflow occurs

(Note) If any interrupt is generated, the flag that corresponds to the interrupt is set to TBnST and the generation of interrupt is notified to INTC. The flag is cleared by reading the TBnST register.

TMRBA status register (2)

① When TBARUN <TBAUDCE> = 0: Normal timer mode

TBAST (0xFFFF_F1E4)

	7	6	5	4	3	2	1	0
bit Symbol						INTTBOFA	INTTBA1	INTTBA0
Read/Write	R					R		
After reset	0					0	0	0
Function	This can be read as "0."					0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

<INTTBA0>: Interrupt generated if there is a match with timer register 0 (TBARG0)
<INTTBA1>: Interrupt generated if there is a match with timer register 1 (TBARG1)
<INTTBOFA>: Interrupt generated if an up-counter overflow occurs

② When TBARUN <TBAUDCE> = 1: Two-phase pulse input count mode

TBAST (0xFFFF_F1E4)

	7	6	5	4	3	2	1	0
bit Symbol				INTTBUDA	INTTBUDFA	INTTBOUFA		
Read/Write	R			R			R	
After reset	0			0	0	0	0	
Function	This can be read as "0."			Up-and-down count 0: Not generated 1: Generated	Underflow 0: Not generated 1: Generated	Overflow 0: Not generated 1: Generated	This can be read as "0."	

<INTTBOVFA>: Interrupt generated if an up-and-down counter overflow occurs
<INTTBUDFA>: Interrupt generated if an up-and-down counter underflow occurs
<INTTBUDA>: Interrupt generated if an up- or down-count occurs

(Note) If any interrupt is generated, the flag that corresponds to the interrupt is set to TBAST and the generation of interrupt is notified to INTC. The flag is cleared by reading the TBAST register.

TBnRG0H/L and TBnRG1H/L timer registers

TBnRG0H/L timer registers (n=0 through A)

TBnRG0L (0xFFFF_F1x8)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRG0L7	TBnRG0L6	TBnRG0L5	TBnRG0L4	TBnRG0L3	TBnRG0L2	TBnRG0L1	TBnRG0L0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of low-order 8 bits							

TBnRG0H (0xFFFF_F1x9)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRG0H7	TBnRG0H6	TBnRG0H5	TBnRG0H4	TBnRG0H3	TBnRG0H2	TBnRG0H1	TBnRG0H0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of low-order 8 bits							

(Note) To write data to the timer registers, use either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.

TBnRG1H/L timer registers (n=0 through A)

TBnRG1L (0xFFFF_F1xA)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRG1L7	TBnRG1L6	TBnRG1L5	TBnRG1L4	TBnRG1L3	TBnRG1L2	TBnRG1L1	TBnRG1L0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of low-order 8 bits							

TBnRG1H (0xFFFF_F1xB)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRG1H7	TBnRG1H6	TBnRG1H5	TBnRG1H4	TBnRG1H3	TBnRG1H2	TBnRG1H1	TBnRG1H0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of high-order 8 bits							

(Note) To write data to the timer registers, use either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.

TBnCP0H/L and TBnCP1H/L capture registers

TBnCP0H/L capture registers (n=0 through A)

TBnCP0L (0xFFFF_F1xC)

	7	6	5	4	3	2	1	0
bit Symbol	TBnCP0L7	TBnCP0L6	TBnCP0L5	TBnCP0L4	TBnCP0L3	TBnCP0L2	TBnCP0L1	TBnCP0L0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of low-order 8 bits							

TBnCP0H (0xFFFF_F1xD)

	7	6	5	4	3	2	1	0
bit Symbol	TBnCP0H7	TBnCP0H6	TBnCP0H5	TBnCP0H4	TBnCP0H3	TBnCP0H2	TBnCP0H1	TBnCP0H0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of high-order 8 bits							

(Note) To read data from the capture registers, use a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.
Don't use a 2-byte data transfer instruction.

TBnCP1H/L capture registers (n=0 through A)

TBnCP1L (0xFFFF_F1xE)

	7	6	5	4	3	2	1	0
bit Symbol	TBnCP1L7	TBnCP1L6	TBnCP1L5	TBnCP1L4	TBnCP1L3	TBnCP1L2	TBnCP1L1	TBnCP1L0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of low-order 8 bits							

TBnCP1H (0xFFFF_F1xF)

	7	6	5	4	3	2	1	0
bit Symbol	TBnCP1H7	TBnCP1H6	TBnCP1H5	TBnCP1H4	TBnCP1H3	TBnCP1H2	TBnCP1H1	TBnCP1H0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of high-order 8 bits							

(Note) To read data from the capture registers, use a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.
Don't use a 2-byte data transfer instruction.

11.4 Description of Operations for Each Mode

11.4.1 16-bit Interval Timer Mode

<< Generating interrupts at periodic cycles >>

To generate the INTTB0 interrupt, specify a time interval in the TB0RG1 timer register.

		7	6	5	4	3	2	1	0	
TB0CR		1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	0	0	0	0	–	0	X	0	Stops TMRB0.
IMC5	←	X	1	1	0	X	1	0	0	Enables INTTB0, and sets it to level 4.
		X	–	–	0	X	–	–	–	(Setting of INTTB0 only is shown here.
		X	–	–	0	X	–	–	–	This is a 32-bit register and requires settings of other interrupts as well.)
		X	–	–	0	X	–	–	–	
TB0FFCR	←	X	X	0	0	0	0	–	–	Disables the trigger.
TB0MOD	←	X	X	1	0	0	1	*	*	Designates the prescaler output clock as the input clock,
TB0RG1L	←	*	*	*	*	*	*	*	*	and specifies the time interval.
TB0RG1H		*	*	*	*	*	*	*	*	(16 bits)
TB0RUN	←	0	0	0	0	–	1	X	1	Starts TMRB0.

X; Don't care –; no change

11.4.2 16-bit Event Counter Mode

<<By using an input clock as an external clock (TB0IN0 pin input), it is possible to make it the event counter>>

The up-counter counts up on the rising edge of TB0IN0 pin input. By capturing a value using software and reading the captured value, it is possible to read the count value.

		7	6	5	4	3	2	1	0	
TB0CR	←	1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	0	0	0	0	–	0	X	0	Stops TMRB0.
PACR	←	–	–	–	–	–	–	–	0	Sets P20 to the input mode.
PAFC	←	–	–	–	–	–	–	–	1	
IMC5	←	X	1	1	0	X	1	0	0	Enables INTTB0, and sets it to level 4.
		X	–	–	0	X	–	–	–	(Setting of INTTB0 only is shown here. This is a 32-bit register and requires settings of other interrupts as well.)
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	
TB0FFCR	←	X	X	0	0	0	0	–	–	Disables the trigger.
TB0MOD	←	X	X	1	0	0	1	0	0	Designates the TB0IN0 pin input as the input clock.
TB0RUN	←	0	0	0	0	–	1	X	1	Starts TMRB0.
TB0MOD	←	X	X	0	0	0	1	0	0	Captures a value using software.
TB0CP0L	←	*	*	*	*	*	*	*	*	Reads the count value of low-order 8 bits.
TB0CP0H	←	*	*	*	*	*	*	*	*	Reads the counter value of high-order 8 bits.

X; Don't care –; no change

To be used as the event counter, put the prescaler in a "RUN" state (TB0RUN<TB0PRUN> = "1").

11.4.3 16-bit PPG (Programmable Square Wave) Output Mode

Square waves with any frequency and any duty (programmable square waves) can be output. The output pulse can be either low-active or high-active.

Programmable square waves can be output from the TB0OUT pin by triggering the timer flip-flop (TB0FF) to reverse when the set value of the up-counter matches the set values of the timer registers (TB0RG0H/L and TB0RG1H/L). Note that the set values of TB0RG0H/L and TB0RG1H/L must satisfy the following requirement:

(Set value of TB0RG0H/L) < (Set value of TB0RG1H/L)

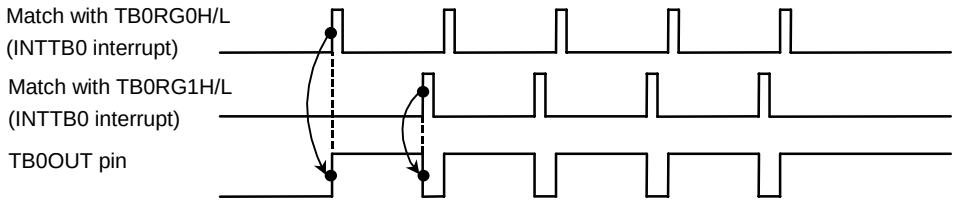


Fig. 11.4.3.1 Example of Output of Programmable Square Wave (PPG)

In this mode, by enabling the double buffering of TB0RG0H/L, the value of register buffer 0 is shifted into TB0RG0H/L when the set value of the up-counter matches the set value of TB0RG1H/L. This facilitates handling of small duties.

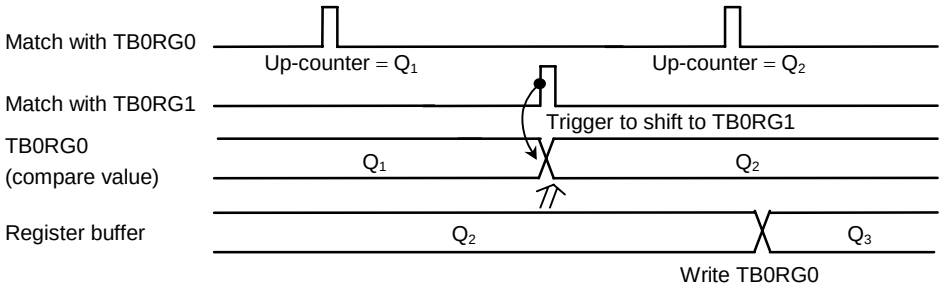


Fig. 11.4.3.2 Register Buffer Operation

The block diagram of the 16-bit PPG (programmable square wave) output mode is shown below.

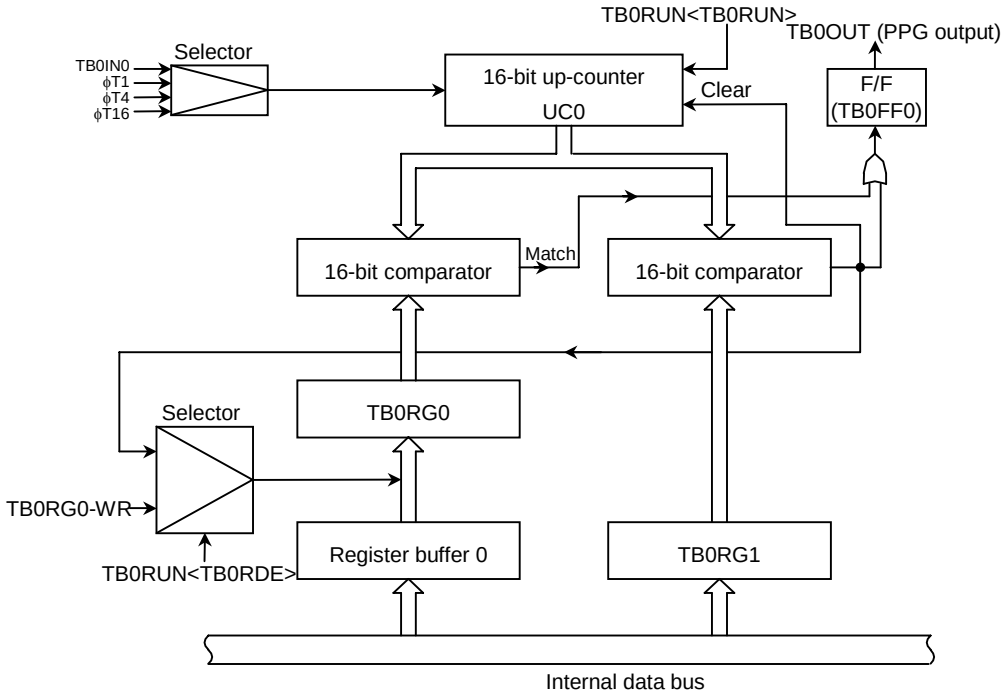


Fig. 11.4.3.3 Block Diagram of 16-bit PPG Mode

<< Example of setting of each register in the 16-bit PPG output mode >>

		7	6	5	4	3	2	1	0	
TB0CR	←	1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	0	0	0	0	–	0	X	0	Disables the TB0RG0 double buffering and stops TMRB0.
TB0RG0L	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
TB0RG0H	←	*	*	*	*	*	*	*	*	
TB0RG1L	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
TB0RG1H	←	*	*	*	*	*	*	*	*	
TB0RUN	←	1	0	0	0	–	0	X	0	Enables the TB0RG0 double buffering. (Changes the duty/cycle when the INTTB0 interrupt is generated)
TB0FFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TB0FF0 to reverse when a match with TB0RG0 or TB0RG1 is detected, and sets the initial value of TB0FF0 to "0."
TB0MOD	←	X	X	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function. (** = 01, 10, 11)
PACR	←	–	–	–	–	–	1	–	–	Assigns PA2 to TB0OUT.
PAFC	←	–	–	–	–	–	1	–	–	
TB0RUN	←	1	0	0	0	–	1	X	1	Starts TMRB0.

X; Don't care –; no change

11.4.4 Applications using the Capture Function

The capture function can be used to develop many applications, including those described below:

① One-shot pulse output triggered by an external pulse

② Frequency measurement

③ Pulse width measurement

④ Time difference measurement

① One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter (UC0) is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TB0IN0 pin. A trigger is generated at the rising of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TB0CP0H/L).

The INTC must be programmed so that an interrupt INT5 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TB0RG0H/L) to the sum of the TB0CP0H/L value (c) and the delay time (d), (c + d), and set the timer registers (TB0RG1H/L) to the sum of the TB0RG0H/L values and the pulse width (p) of one-shot pulse, (c + d + p).

In addition, the timer flip-flop control registers (TB0FFCR<TB0E1T1, TB0E0T1>) must be set to "11." This enables triggering the timer flip-flop (TB0FF0) to reverse when UC0 matches TB0RG0H/L and TB6RG1H/L. This trigger is disabled by the INTTB0 interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in Fig. 11.4.4.1.

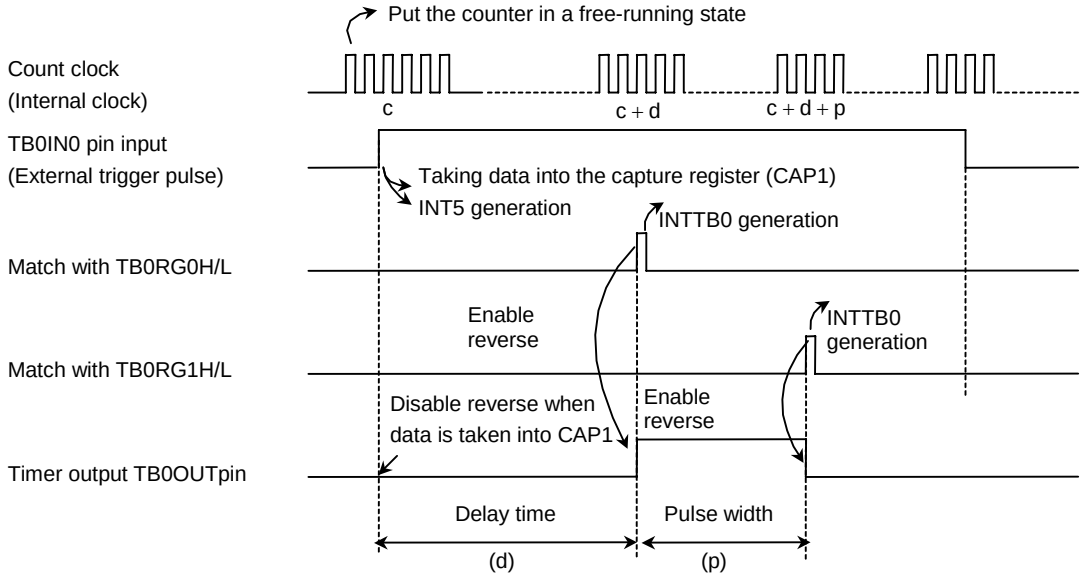


Fig. 11.4.4.1 One-shot Pulse Output (With Delay)

Programming example: Output a 2-ms one-shot pulse triggered by an external pulse from the TB0IN0 pin with a 3-ms delay

* Clock condition

System clock : High speed (fc)
High-speed clock gear : 1X (fc)
Prescaler clock : fperiph/4 (fperiph fsys)

Main programming

		7	6	5	4	3	2	1	0	
TB0CR	←	1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0MOD	←	X	X	1	0	1	0	0	1	Puts to a free-running state. Uses ϕT1 for counting.
										Takes data into TB0CP0 at the rising of TB0IN0 input
TB0FFCR	←	X	X	0	0	0	0	1	0	
										Clears TB0FF0 to zero
										Disables TB0FF0 to reverse
PACR	←	–	–	–	–	–	1	–	–	Assigns PA2 pin to TB0OUT
PAFC	←	–	–	–	–	–	1	–	–	
IMC1	←	X	–	–	0	X	–	–	–	
		X	1	1	0	X	1	0	0	Enables INT5
		X	–	–	0	X	–	–	–	These are 32-bit registers and must be all processed.
		X	–	–	0	X	–	–	–	
IMC5	←	X	1	1	0	X	0	0	0	
		X	–	–	0	X	–	–	–	Disables INTTB0
		X	–	–	0	X	–	–	–	These are 32-bit registers and must be all processed.
		X	–	–	0	X	–	–	–	
TB0RUN	←	–	0	0	0	–	1	X	1	Starts the TMRB0 module.

INT0 programming

TB0RG0L	←	*	*	*	*	*	*	*	*	TB0CP0 + 3ms/ϕT1
TB0RG0H	←	*	*	*	*	*	*	*	*	
TB0RG1L	←	*	*	*	*	*	*	*	*	TB0RG0 + 2ms/ϕT1
TB0RG1H	←	*	*	*	*	*	*	*	*	
TB0FFCR	←	X	X	–	–	1	1	–	–	
										Enables TB0FF0 to reverse when there is a match with TB0RG0, 1
IMC5	←	X	1	1	0	X	1	0	0	
		X	–	–	0	X	–	–	–	Enables INTTB0
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	

INTTB0 programming

TB0FFCR	←	X	X	–	–	0	0	–	–	
										Disables TB0FF0 to reverse when there is a match with TB0RG0, 1
IMC5	←	X	1	1	0	X	0	0	0	
		X	–	–	0	X	–	–	–	Disables INTTB0
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	

X; Don't care —;no change

If a delay is not required, TB0FF0 is reversed when data is taken into TB0CP0H/L, and TB0RG1L/H is set to the sum of the TB0CP0H/L value (c) and the one-shot pulse width (p), (c + p), by generating the INT5 interrupt. TB0FF0 is enabled to reverse when UC0 matches with TB0RG1L/H, and is disabled by generating the INTTB0 interrupt.

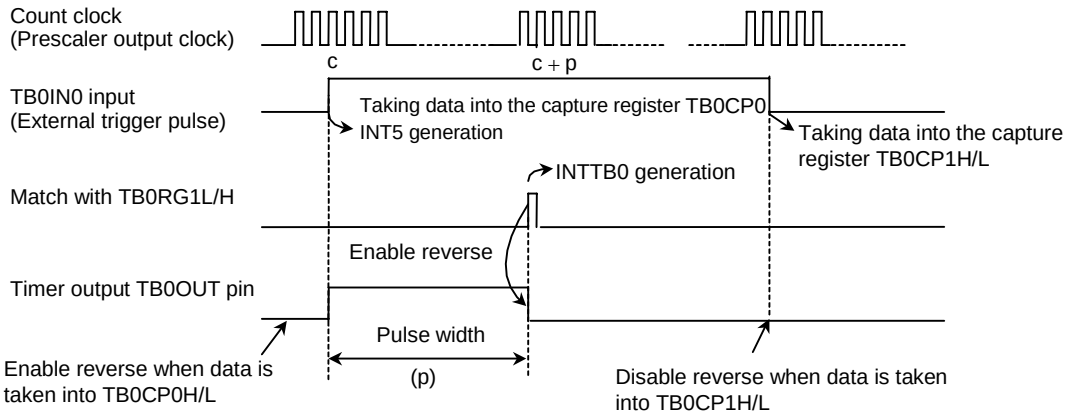


Fig. 11.4.4.2 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

② Frequency measurement

By using the capture function, the frequency of an external clock can be measured.

To measure frequency, another 16-bit timer (TMRB3) is used in combination with the 16-bit event counter mode (TMRB3 reverses TB3FFCR to specify the measurement time).

The TB0IN0 pin input is selected as the TMRB0 count clock to perform the count operation using an external input clock. TB0MOD<TB0CPM1:0> is set to "11." This setting allows a count value of the 16-bit UC0 up-counter to be taken into the capture register (TB0CP0) upon the rising of a timer flip-flop (TB3FFCR) of the 16-bit timer (TMRB3), and an UC0 counter value to be taken into the capture register (TB0CP1H/L) upon the falling of TB3FF of the 16-bit timer (TMRB3).

A frequency is then obtained from the difference between TB0CP0H/L and TB0CP1H/L based on the measurement, by generating the INTTB3 16-bit timer interrupt.

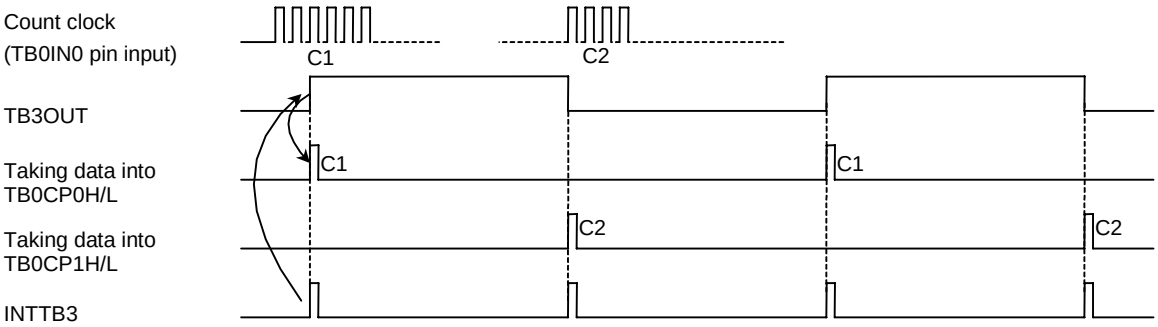


Fig. 11.4.4.3 Frequency Measurement

For example, if the set width of TB3FF level "1" of the 16-bit timer is 0.5 s and if the difference between TB0CP0H/L and TB0CP1H/L is 100, the frequency is 100 / 0.5 s = 200 Hz.

③ Pulse width measurement

By using the capture function, the "H" level width of an external pulse can be measured. Specifically, an external pulse is input through the TB0IN0 pin and the up-counter (UC0) is made to count up by putting it in a free-running state using the prescaler output clock. A trigger is generated at each rising and falling edge of the external pulse by using the capture function and the value of the up-counter is taken into the capture registers (TB0CP0H/L, TB0CP1H/L). The INTC must be programmed so that INT5 is generated at the falling edge of an external pulse input through the TB0IN0 pin.

The "H" level pulse width can be calculated by multiplying the difference between TB0CP0H/L and TB0CP1H/L by the clock cycle of an internal clock.

For example, if the difference between TB0CP0H/L and TB0CP1H/L is 100 and the cycle of the prescaler output clock is 0.5 μs, the "H" level pulse width is 100 × 0.5 μs = 50 μs.

Caution must be exercised when measuring pulse widths exceeding the UC0 maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.

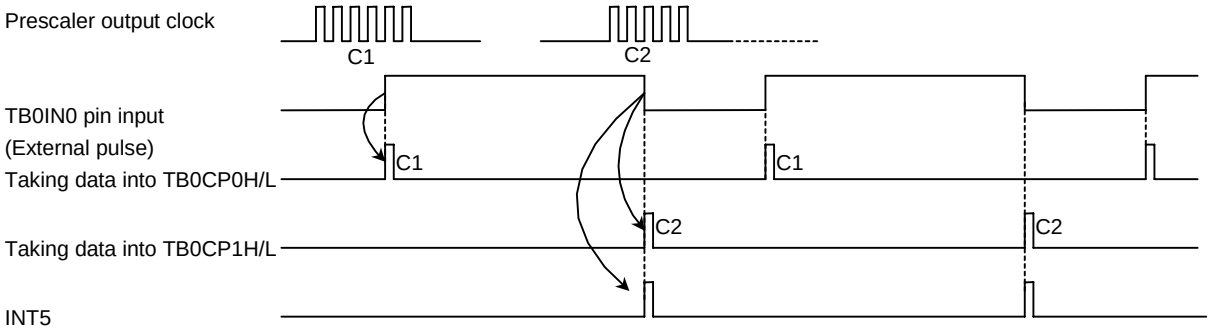


Fig. 11.4.4.4 Pulse Width Measurement

The "L" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INT5 interrupt processing as shown in Fig. 11.4.4.5 and this difference is multiplied by the cycle of the prescaler output clock to obtain the "L" level width.

④ Time Difference Measurement

By using the capture function, the time difference between two events can be measured. Specifically, the up-counter (UC0) is made to count up by putting it in a free-running state using the prescaler output clock. The value of UC0 is taken into the capture register (TB0CP0H/L) at the rising edge of the TB0IN0 pin input pulse. The INTC must be programmed to generate INT5 interrupt at this time.

The value of UC0 is taken into the capture register TB0CP1H/L at the rising edge of the TB0IN1 pin input pulse. The INTC must be programmed to generate INT6 interrupt at this time.

The time difference can be calculated by multiplying the difference between TB0CP1H/L and TB0CP0H/L by the clock cycle of an internal clock.

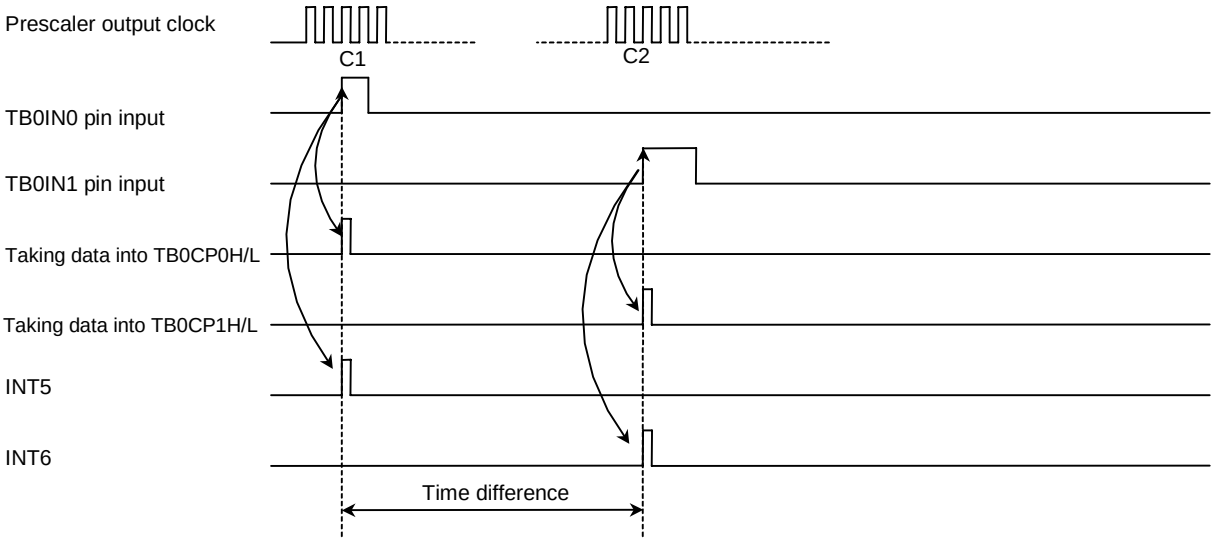


Fig. 11.4.4.5 Time Difference Measurement

11.4.5 Two-phase Pulse Input Count Mode (TMRBA)

In this mode, the counter is incremented or decremented by one depending on the state transition of the two-phase clock that is input through TBAIN0 and TBAIN1 and has phase difference. An interrupt is output when a counter overflow or underflow occurs in the up-and-down counter mode, and when the counting operation is executed.

This is the multiplication-by-4 mode in which the counter counts up/down at each count.

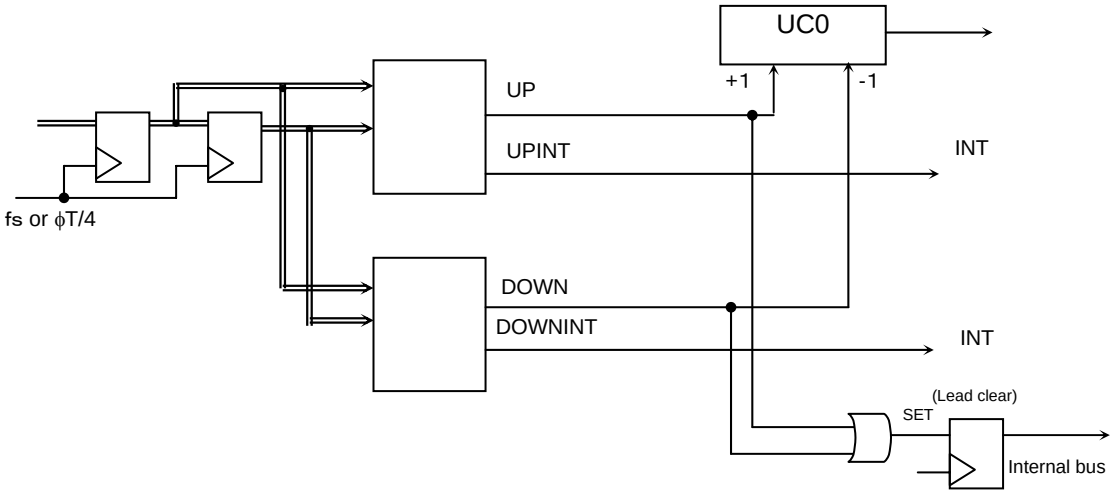
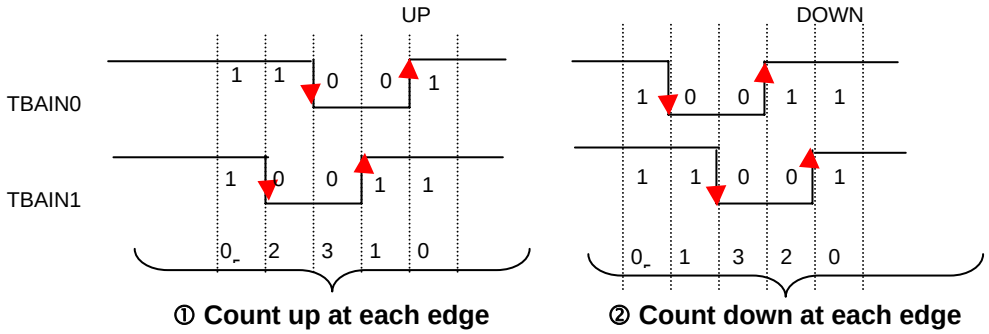


Fig. 11.4.5.1 Count Circuit of Two-phase Counter

11.4.6 Multiplication-by-4 Mode



	Pin state					
Count condition	UP			DOWN		
TBAIN0,TBAIN1	0	→	2	0	→	1
	2		3	1		3
	3		1	3		2
	1		0	2		0

TMRBA RUN register (TBARUN)

TBARUN (0xFFFF_F1E0)

	7	6	5	4	3	2	1	0
bit Symbol	TBARDE		UDACK	TBAUDCE	I2TBA	TBAPRUN		TBARUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0: Disable 1: Enable	Write "0."	Select sampling clock 0:fs 1: ϕT0/4	Enable/ disable two-phase counter 0: Disable 1: Enable	IDLE 0: Stop 1: Operate	Timer Run/Stop Control 0: Stop & Clear 1: Run (Count Up)		

Fig. 11.4.6.1 Two-phase Pulse Input Count Mode Setting Register

For the sampling clock, the fifth bit <UDACK> of the TBARUN register is set to "1."

<< Recovery from the SLEEP mode >>

The two-phase counter counts up or down depending on the SLEEP release input state.

① Operation mode

Register setting determines whether the external input signals from the TBAIN0 and TBAIN1 input pins are input to the normal 16-bit timer (capture input) or the up-and-down counter.

- In the up-and-down counter mode, capture is executed by the software only. Capture at the external clock timing does not work.
- In the up-and-down counter mode, the comparator is disabled and it does not execute comparison with timer registers.
- The input clock sampling is executed by fs (32 KHz) or the high-speed clock (system clock). The maximum input frequency is 4 kHz for fs and ϕT0/4 [Hz] for the high-speed clock.

<< How to program the up-and-down counter >>

Set the TBAMOD register <TBACLK0, TBACLK1> to "00" (prescaler OFF). Then, program the fourth bit <TBAUDCE> of the TBARUN register to determine whether to operate the counter as the up-and-down counter or as the conventional up-counter for external clock input.

TBAUDCE (Enable the up-and-down counter) = "0": Normal 16-bit timer operation
= "1": Up-and-down counter operation

② Interrupt

In the NORMAL or SLOW mode

The INTTBA interrupt is generated by counting up or down. Reading the status register TBAST during interrupt handling allows simultaneous check for occurrences of an overflow and an underflow. If TBAST<INTTBOUFA> is "1," it indicates that an overflow has occurred. If <INTTBUDFA> is "1," it indicates that an underflow has occurred. This register is cleared after it is read. The counter becomes 0x0000 when an overflow occurs, and it becomes 0xFFFF when an underflow occurs. After that, the counter continues the counting operation.

TBAST (0xFFFF_F1E4)

	7	6	5	4	3	2	1	0
bit Symbol				INTTBUDA	INTTBUDFA	INTTBOUFA		
Read/Write	R			R			R	
After reset	0			0	0	0	0	
Function	This can be read as "0."			Up-and-down count 0: Not occurred 1: Occurred	Underflow 0: Not occurred 1: Occurred	Overflow 0: Not occurred 1: Occurred	This can be read as "0."	

Fig. 11.4.6.2 TMRBA Status Register

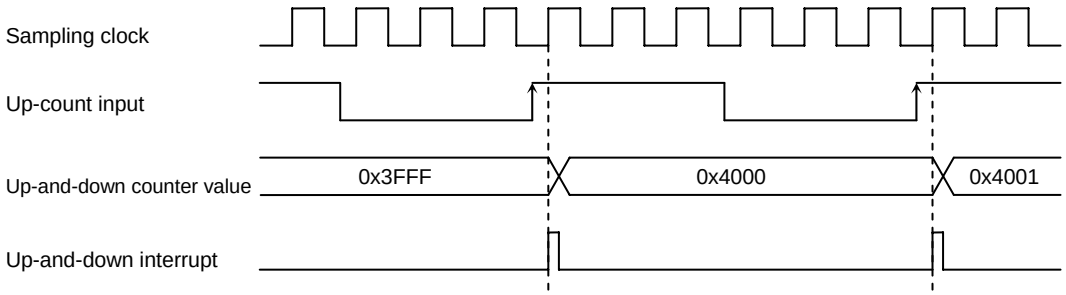
Note: The status is cleared after the register is read.

In the SLEEP mode

The INTTBA interrupt is enabled using the interrupt controller (INTC). The INTTBA interrupt is generated by the count-up or count-down input, and the system recovers from the SLEEP mode. Reading the status register TBAST during interrupt handling allows simultaneous check for occurrences of an overflow and an underflow

③ Up-and-down counter

When the two-phase input count mode is selected (TBARUN<TBAUDCE> = "1"), the up-counter becomes the up-and-down counter and it is initialized to 0x7FFF. If a counter overflow occurs, the counter returns to 0x0000. If a counter underflow occurs, the counter returns to 0xFFFF. After that, the counter continues the counting operation. Therefore, the state can be checked by reading the counter value and the status flag TBAST after an interrupt is generated.



(Note 1) The up (down) count input must be set to the "H" level for the states before and after an input.

(Note 2) Reading of counter value must be executed during INTTBA interrupt handling

12. 32-bit Input Capture (TMRC)

TMRC consists of one channel with a 32-bit time base timer (TBT), four channels (TCCAP0 through TCCAP3) each with a 32-bit input capture register, and ten channels (TCCMP0 through TCCMP9) each with a 32-bit compare register.

Fig. 12-1 shows the TMRC block diagram.

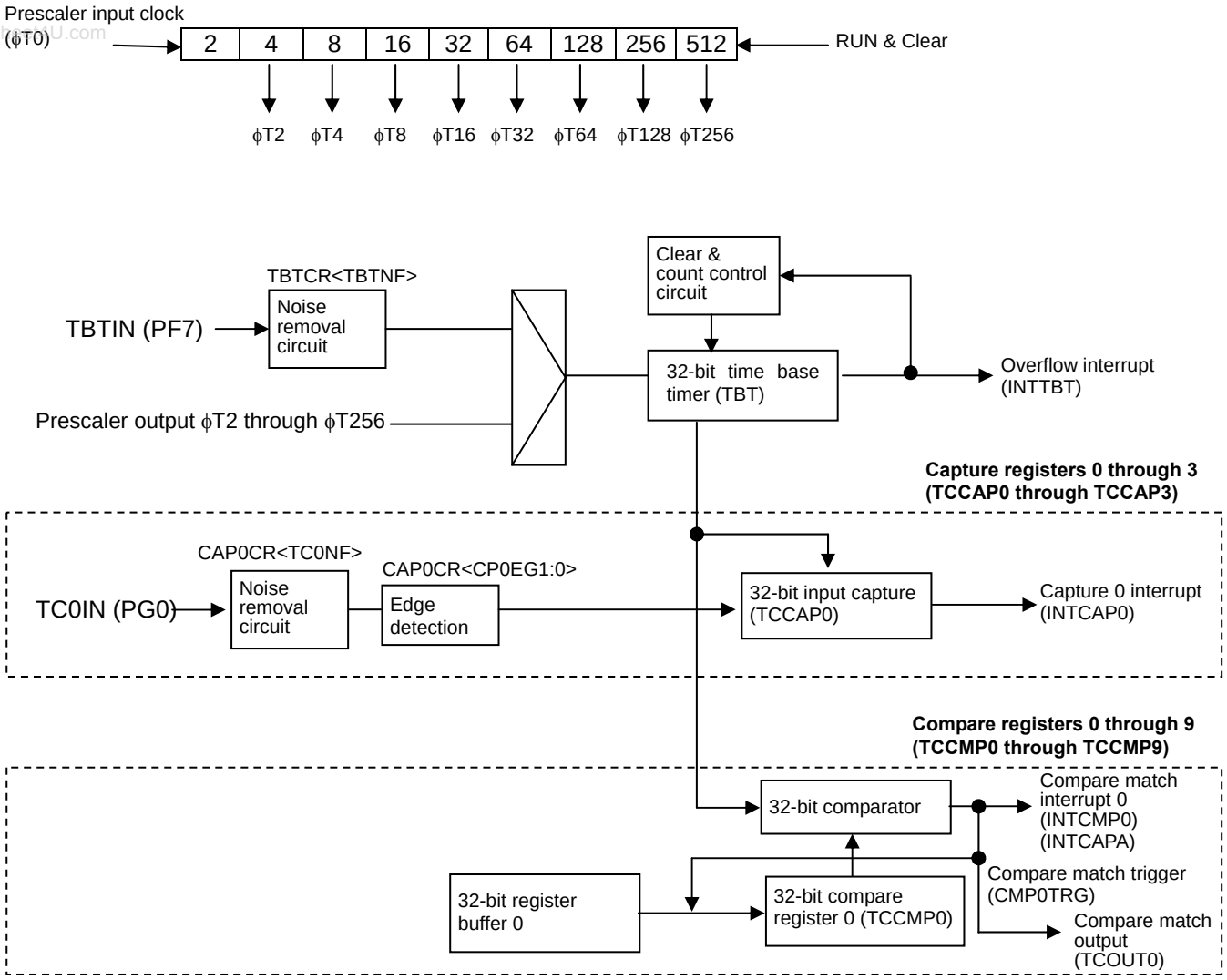


Fig. 12.1 Timer C Block Diagram

12.1 Description for Operations of Each Circuit

12.2.1 Prescaler

The prescaler is provided to acquire the TMRC source clock. The prescaler input clock ϕT0 is fperiph/2, fperiph/4, fperiph/8 or fperiph/16 selected by SYSCR0<PRCK1:0> in the CG. ϕT2 through ϕT256 generated by dividing ϕT0 are available as TMRC prescaler input clocks and can be selected with TBTCR<TBTCLK3:0>.

Fperiph is either "fgear" which is a clock selected by SYSCR1<FPSEL> in the CG, or "fc" which is a clock before it is divided by the clock gear.

The operation or stoppage of the prescaler is set with TBTRUN<TBTPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 12-1 shows the prescaler output clock resolutions.

Table 12.1 Prescaler Output Clock Resolutions
(if any of high-speed clock gears 8/8, 4/8, 2/8 and 1/8 is selected)

@fc = 54MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolution			
			ϕT2	ϕT4	ϕT8	ϕT16
0(fgear)	000(fc)	00(fperiph/16)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	fc/2^3(0.15 μs)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)
	100(fc/2)	00(fperiph/16)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)	fc/2^{10}(18.96 μs)
		01(fperiph/8)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		10(fperiph/4)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
		11(fperiph/2)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)
	110(fc/4)	00(fperiph/16)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		01(fperiph/8)	fc/2^7(2.z/37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)	fc/2^{10}(18.96 μs)
		10(fperiph/4)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		11(fperiph/2)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
	111(fc/8)	00(fperiph/16)	fc/2^9(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		01(fperiph/8)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		10(fperiph/4)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)	fc/2^{10}(18.96 μs)
		11(fperiph/2)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
1(fc)	000(fc)	00(fperiph/16)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	fc/2^3(0.15 μs)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)
	100(fc/2)	00(fperiph/16)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	fc/2^3(0.15 μs)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)
	110(fc/4)	00(fperiph/16)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	—	fc/2^4(0.30 μs)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)
	111(fc/8)	00(fperiph/16)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)	fc/2^9(9.48 μs)
		01(fperiph/8)	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)	fc/2^8(4.74 μs)
		10(fperiph/4)	—	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)	fc/2^7(2.37 μs)
		11(fperiph/2)	—	—	fc/2^5(0.59 μs)	fc/2^6(1.19 μs)

@fc = 54MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolution			
			ϕT32	ϕT64	ϕT128	ϕT256
0(fgear)	000(fc)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	100(fc/2)	00(fperiph/16)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)
		01(fperiph/8)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		10(fperiph/4)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		11(fperiph/2)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
	110(fc/4)	00(fperiph/16)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)	fc/2^{15}(606.8 μs)
		01(fperiph/8)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)
		10(fperiph/4)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		11(fperiph/2)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
	111(fc/8)	00(fperiph/16)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)	fc/2^{15}(606.8 μs)	fc/2^{16}(1213.6 μs)
		01(fperiph/8)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)	fc/2^{15}(606.8 μs)
		10(fperiph/4)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)
		11(fperiph/2)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
1(fc)	000(fc)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	100(fc/2)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	110(fc/4)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	111(fc/8)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)

(Note 1) The prescaler output clock ϕTn must be selected so that ϕTn<fsys/2 is satisfied (so that ΦTn is slower than fsys/2).

(Note 2) Do not change the clock gear while the timer is operating.

(Note 3) "—" denotes "setting prohibited."

12.2.2 Noise Removal Circuit

The noise removal circuit removes noises from an external clock source input (TBTIN) and a capture trigger input (TcnIN) of the time base timer (TBT). It can also output input signals without removing noises from them.

12.2.3 32-bit Time Base Timer (TBT)

This is a 32-bit binary counter that counts up upon the rising of an input clock specified by the TBT control register TBTCR of the time base timer.

Based on the TBTCR<TBTCLK3:0> setting, an input clock is selected from external clocks supplied through the TBTIN pin and eight prescaler output clocks ϕT2, ϕT4, ϕT8, ϕT16, ϕT32, ϕT64, ϕT128, and ϕT256.

"Count," "stop" or "clear" of the up-counter can be selected with TBTRUN<TBTRUN>. When a reset is performed, the up-counter is in a cleared state and the timer is in an idle state. As counting starts, the up-counter operates in a free-running condition. As it reaches an overflow state, the overflow interrupt INTTBT is generated; subsequently, the count value is cleared to 0 and the up-counter restarts a count-up operation. INTTBT is controlled by the TCGST and TCGIM that are grouped in the same way as INTCAP0 through INTCAP3 are (see the explanation about the 32-bit capture register).

This counter can perform a read capture operation. When it is performing a read capture operation, it is possible to read a counter value by accessing the TBT read capture register (TBTRDCAP) in units of 32 bits.

However, a counter value cannot be read (captured) if the register is accessed in units of 8 or 16 bits.

12.2.4 Edge Detection Circuit

By performing sampling, this circuit detects the input edge of an external capture input (TcnIN). It can be set to "rising edge," "falling edge," "both edges" or "not capture" by provisioning the capture control register CAPnCR<CPnEG1:0>. Fig. 12.2.4.1 shows capture inputs, outputs (capture factor outputs) produced by the edge detection circuit, and specific detection circuit settings.

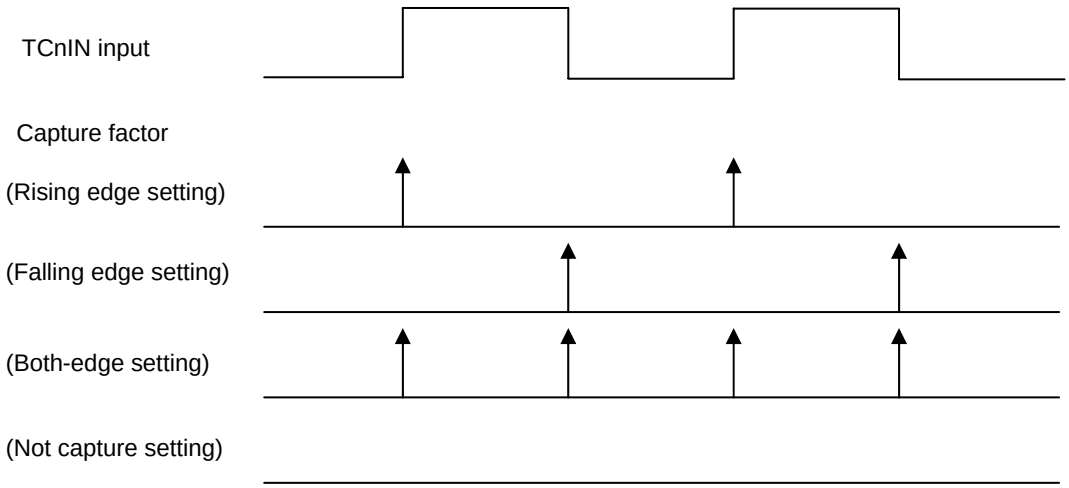


Fig. 12.2.4.1 Capture Inputs and Capture Factor Outputs (Outputs Produced by the Edge Detection Circuit)

12.2.5 32-bit Capture Register

This is a 32-bit register for capturing count values of the time base timer by using capture factors as triggers. If a capture operation is performed, the capture interrupt INTCAPn is generated. Four interrupt requests INTCAP0 through INTCAP3 are grouped into one set of interrupt requests which are then notified to the interrupt controller. Which one of interrupt requests INTCAP0 through INTCAP3 must be processed can be identified by reading the status register TCGST during interrupt processing. Additionally, it is possible to mask unnecessary interrupts by setting the interrupt mask register TCGIM to an appropriate bit setting. While a read of the capture register is ongoing, count values cannot be captured even if there are triggers.

12.2.6 32-bit Compare Register

This is a 32-bit register for specifying a compare value. TMRC has ten built-in compare registers, TCCMP0 through TCCMP9. If values set in these compare registers match the value of the time base timer TBT, the match detection signal of a comparator becomes active. "Compare enable" or "compare disable" can be specified with the compare control register CMPCTL<CMPEN1:0>.

To set TCCMPn to a specific value, data must be transferred to TCCMPn in the order of lower to higher bits by using a byte data transfer instruction. If a byte data transfer instruction is used, data is transferred four times to TCCMPn.

Each compare register has a double-buffer structure, that is, TCCMPn forms a pair with a register buffer "n." "Enable" or "disable" of the double buffers is controlled by the compare control register CMPCTL <CMPRDEn>. If <CMPRDEn> is set to "0," the double buffers are disabled. If <CMPRDEn> is set to "1," they are enabled.

If the double buffers are enabled, data transfer from the register buffer "n" to the compare register TCCMPn takes place when the value of TBT matches that of TCCMPn.

Because TCCMPn is indeterminate when a reset is performed, it is necessary to prepare and write data in advance. A reset initializes CMPCTL <CMPRDEn> to "0" and disables the double buffers. To use the double buffers, data must be written to the compare register, <CMPRDEn> must be set to "1," and then the following data must be written to the register buffer.

TCCMPn and the register buffer are assigned to the same address. If <CMPRDEn> is "0," the same value is written to TCCMPn and each register buffer. If <CMPRDEn> is "1," data is written to each register buffer only. Therefore, to write an initial value to the compare register, it is necessary to set the double buffers to "disable."

12.3 Register Description

TMRC Control Register

TCCR (FFFFF400H)

	7	6	5	4	3	2	1	0
bit Symbol	TCEN	I2TBT						
Read/Write	R/W		R					
After reset	0	0	0	0	0	0	0	0
Function	TMRC operation 0: Disable 1: Enable	IDLE 0: Stop 1: Run						

<I2TBT>: Controls the operation in IDLE mode

<TCEN>: Specifies enabling/disabling of the TMRC operation. If set to "disable," a clock is not supplied to other registers of the TMRC module and, therefore, a reduction in power consumption is possible (a read of or a write to other registers cannot be executed). To use TMRC, the TMRC operation must be set to "enable" ("1") before making individual register settings of TMRC modules. If TMRC is operated and then set to "disable," individual register settings are retained.

(Note) Values read from bits 0 through 5 of TCCR are all "0."

TBTRUN Register

TBTRUN (FFFFF401H)

	7	6	5	4	3	2	1	0
bit Symbol						TBTCAP	TBTPRUN	TBTRUN
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function					Ensure this is set to "0."	TBT counter software capture 0: Don't Care 1: Software capture	Timer Run/Stop Control 0: Stop & clear 1: Count	

<TBTRUN>: Controls the TBT count operation

<TBTPRUN>: Controls the TBT prescaler operation

<TBTCAP>: If this is set to "1," the count value of the time base timer (TBT) is taken into the capture register TBTCAPn.

(Note) Values read from bits 4 through 7 of TBTRUN are all "0."

Fig. 12.3.1 TMRC-related Registers

TBT Control Register

TBTCR (FFFFF402H)

	7	6	5	4	3	2	1	0
bit Symbol	TBTNF				TBTCLK3	TBTCLK2	TBTCLK1	TBTCLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	TBTIN Input noise removal 0: Disable 1: Enable	Ensure this is set to "0."			TBT source clock 0000: ϕT2 0001: ϕT4 0010: ϕT8 0011: ϕT16 0100: ϕT32 0101: ϕT64 0110: ϕT128 0111: ϕT256 1111: TBTIN pin input			

<TBTCLK3:0>: This is an input clock for TBT. Clocks from "0000" to "0111" are available as prescaler output clocks. A clock "1111" is input through the TBTIN pin.

<TBTNF>: Controls the noise removal for the TBTIN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph=fc=54MHz) is accepted as a source clock for TBT, at whichever level the TBTIN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph=fc=54MHz) is regarded as noise and removed, at whichever level the TBTIN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

TBT Capture Register (TBTCAP)

TBTCAP0 (FFFFF404H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP07	CAP06	CAP05	CAP04	CAP03	CAP02	CAP01	CAP00
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 7 through 0)							

TBTCAP1 (FFFFF405H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10	CAP09	CAP08
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 15 through 8)							

TBTCAP2 (FFFFF406H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP23	CAP22	CAP21	CAP20	CAP19	CAP18	CAP17	CAP16
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 23 through 16)							

TBTCAP3 (FFFFF407H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP31	CAP30	CAP29	CAP28	CAP27	CAP26	CAP25	CAP24
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 31 through 24)							

Fig. 12.3.2 TMRC-related Registers

TBT Capture Register (TBTRDCAP)

TBTRDCAP0 (FFFFF408H)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP07	RDCAP06	RDCAP05	RDCAP04	RDCAP03	RDCAP02	RDCAP01	RDCAP00
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 7 through 0)							

TBTRDCAP1 (FFFFF409H)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP17	RDCAP16	RDCAP15	RDCAP14	RDCAP13	RDCAP12	RDCAP11	RDCAP10
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 15 through 8)							

TBTRDCAP2 (FFFFF40AH)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP27	RDCAP26	RDCAP25	RDCAP24	RDCAP23	RDCAP22	RDCAP21	RDCAP20
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 23 through 16)							

TBTRDCAP3 (FFFFF40BH)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP37	RDCAP36	RDCAP35	RDCAP34	RDCAP33	RDCAP32	RDCAP31	RDCAP30
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bits 31 through 24)							

Fig. 12.3.3 TMRC-related Registers

TMRC Capture 0 Control Register

CAP0CR (FFFFF410H)

	7	6	5	4	3	2	1	0
bit Symbol	TC0NF						CP0EG1	CP0EG0
Read/Write	R/W	R					R/W	
After reset	0	0	0	0	0	0	0	0
Function	TC0IN Input noise removal 0: Disable 1: Enable						Select effective edge of TC0IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP0EG1:0>: Selects the effective edge of an input to the trigger input pin TC0IN of the capture 0 register (TCCAP0). If this is set to "00," the capture operation is disabled.

<TC0NF>: Controls the noise removal for the TC0IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph=fc=54MHz) is accepted as a trigger input for TCCAP0, at whichever level the TC0IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph=fc=54MHz) is regarded as noise and removed, at whichever level the TC0IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) Values read from bits 2 through 6 of CAPOCR are all "0."

TMRC Capture 0 Register (TCCAP0)

TCCAP0LL (FFFFF414H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP007	CAP006	CAP005	CAP004	CAP003	CAP002	CAP001	CAP000
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data (bits 7 through 0)							

TCCAP0LH (FFFFF415H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP017	CAP016	CAP015	CAP014	CAP013	CAP012	CAP011	CAP010
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data (bits 15 through 8)							

TCCAP0HL (FFFFF416H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP027	CAP026	CAP025	CAP024	CAP023	CAP022	CAP021	CAP020
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data (bits 23 through 16)							

TCCAP0HH (FFFFF417H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP037	CAP036	CAP035	CAP034	CAP033	CAP032	CAP031	CAP030
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data (bits 31 through 24)							

(Note) Data is not captured during a read of the capture register.

Fig. 12.3.4 TMRC-related Registers

TMRC Capture 1 Control Register

CAP1CR (FFFFF418H)

	7	6	5	4	3	2	1	0
bit Symbol	TC1NF						CP1EG1	CP1EG0
Read/Write	R/W	R					R/W	
After reset	0	0	0	0	0	0	0	0
Function	TC1IN Input noise removal 0: Disable 1: Enable						Select effective edge of TC1IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP1EG1:0>: Selects the effective edge of an input to the trigger input pin TC1IN of the capture 1 register (TCCAP1). If this is set to "00," the capture operation is disabled.

<TC1NF>: Controls the noise removal for the TC1NF pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph=fc=54MHz) is accepted as a trigger input for TCCAP1, at whichever level TC1IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph=fc=54MHz) is regarded as noise and removed, at whichever level the TC1IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) Values read from bits 2 through 6 of CAP1CR are all "0."

TMRC Capture 1 Register (TCCAP1)

TCCAP1LL (FFFFF41CH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP107	CAP106	CAP105	CAP104	CAP103	CAP102	CAP101	CAP100
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 1 data (bits 7 through 0)							

TCCAP1LH (FFFFF41DH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP117	CAP116	CAP115	CAP114	CAP113	CAP112	CAP111	CAP110
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 1 data (bits 15 through 8)							

TCCAP1HL (FFFFF41EH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP127	CAP126	CAP125	CAP124	CAP123	CAP122	CAP121	CAP120
Read/Write	R							
After reset								
Function	Capture 1 data (bits 23 through 16)							

TCCAP1HH (FFFFF41FH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP137	CAP136	CAP135	CAP134	CAP133	CAP132	CAP131	CAP130
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 1 data (bits 31 through 24)							

(Note) Data is not captured during a read of the capture register.

Fig. 12.3.5 TMRC-related Registers

TMRC Capture 2 Control Register

CAP2CR (FFFFF420H)

	7	6	5	4	3	2	1	0
bit Symbol	TC2NF						CP2EG1	CP2EG0
Read/Write	R/W	R					R/W	
After reset	0	0	0	0	0	0	0	0
Function	TC2IN Input noise removal 0: Disable 1: Enable						Select effective edge of TC2IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP2EG1:0>: Selects the effective edge of an input to the trigger input pin TC2IN of the capture 2 register (TCCAP2). If this is set to "00," the capture operation is disabled.

<TC2NF>: Controls the noise removal for the TC2IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph=fc=54MHz) is accepted as a trigger input for TCCAP2, at whichever level the TC2IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph=fc=54MHz) is regarded as noise and removed, at whichever level the TC2IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) Values read from bits 2 through 6 of CAP2CR are all "0."

TMRC Capture 2 Register (TCCAP2)

TCCAP2LL (FFFFF424H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP207	CAP206	CAP205	CAP204	CAP203	CAP202	CAP201	CAP200
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 2 data (bits 7 through 0)							

TCCAP2LH (FFFFF425H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP217	CAP216	CAP215	CAP214	CAP213	CAP212	CAP211	CAP210
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 2 data (bits 15 through 8)							

TCCAP2HL (FFFFF426H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP227	CAP226	CAP225	CAP224	CAP223	CAP222	CAP221	CAP220
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 2 data (bits 23 through 16)							

TCCAP2HH (FFFFF427H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP237	CAP236	CAP235	CAP234	CAP233	CAP232	CAP231	CAP230
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 2 data (bits 31 through 24)							

(Note) Data is not captured during a read of the capture register.

Fig. 12.3.6 TMRC-related Registers

TMRC Capture 3 Control Register

CAP3CR (FFFFF428H)

	7	6	5	4	3	2	1	0
bit Symbol	TC3NF						CP3EG1	CP3EG0
Read/Write	R/W	R					R/W	
After reset	0	0	0	0	0	0	0	0
Function	TC3IN Input noise removal 0: Disable 1: Enable						Select effective edge of TC3IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP3EG1:0>: Selects the effective edge of an input to the trigger input pin TC3IN of the capture 3 register (TCCAP3). If this is set to "00," the capture operation is disabled.

<TC3NF>: Controls the noise removal for the TC3IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph=fc=54MHz) is accepted as a trigger input for TCCAP3, at whichever level the TC3IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph=fc=54MHz) is regarded as noise and removed, at whichever level the TC3IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) Values read from bits 2 through 6 of CAP3CR are all "0."

TMRC Capture 3 Register (TCCAP3)

TCCAP3LL (FFFFF42CH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP307	CAP306	CAP305	CAP304	CAP303	CAP302	CAP301	CAP300
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 3 data (bits 7 through 0)							

TCCAP3LH (FFFFF42DH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP317	CAP316	CAP315	CAP314	CAP313	CAP312	CAP311	CAP310
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 3 data (bits 15 through 8)							

TCCAP3HL (FFFFF42EH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP327	CAP326	CAP325	CAP324	CAP323	CAP322	CAP321	CAP320
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 3 data (bits 23 through 16)							

TCCAP3HH (FFFFF42FH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP337	CAP336	CAP335	CAP334	CAP333	CAP332	CAP331	CAP330
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 3 data (bits 31 through 24)							

(Note) Data is not captured during a read of the capture register.

Fig. 12.3.7 TMRC-related Registers

TMRCG Interrupt Mask Register

TCGIM (FFFFF40CH)

	7	6	5	4	3	2	1	0
bit Symbol				TBTIM	TCIM3	TCIM2	TCIM1	TCIM0
Read/Write	R			R/W				
After reset	0	0	0	0	0	0	0	0
Function				Mask 1: INTTBT	Mask 1: INTCAP3	Mask 1: INTCAP2	Mask 1: INTCAP1	Mask 1: INTCAP0

(Note) Values read from bits 5, 6 and 7 of TCGIM are all "0."

TMRCG Status Register

TCGST (FFFFF40DH)

	7	6	5	4	3	2	1	0
bit Symbol				INTTBT	INTCAP3	INTCAP2	INTCAP1	INTCAP0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function				0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

(Note 1) A read of TCGST clears bits 0, 1, 2, 3 and 4.

(Note 2) Values read from bits 5, 6 and 7 of TCGST are all "0."

Fig. 12.3.8 TMRC-related Registers

TMRC Compare Control Register (CMPCTLn)

CMPCTL0 (FFFFF470H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN0	TCFFC01	TCFFC00			CMPRDE0	CMPEN0
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF0 reversal 0: Disable 1: Enable	TCFF0 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 0 0: Disable 1: Enable	Compare 0 enable 0: Disable 1: Enable

CMPCTL1 (FFFFF471H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN1	TCFFC11	TCFFC10			CMPRDE1	CMPEN1
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF1 reversal 0: Disable 1: Enable	TCFF1 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 1 0: Disable 1: Enable	Compare 1 enable 0: Disable 1: Enable

CMPCTL2 (FFFFF472H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN2	TCFFC21	TCFFC20			CMPRDE2	CMPEN2
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF2 reversal 0: Disable 1: Enable	TCFF2 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 2 0: Disable 1: Enable	Compare 2 enable 0: Disable 1: Enable

CMPCTL3 (FFFFF473H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN3	TCFFC31	TCFFC30			CMPRDE3	CMPEN3
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF3 reversal 0: Disable 1: Enable	TCFF3 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 3 0: Disable 1: Enable	Compare 3 enable 0: Disable 1: Enable

<CMPENn>: Controls enable/disable of the compare match detection.

<CMPRDEn>: Controls enable/disable of double buffers of the compare register.

<TCFFCn1:0>: Controls F/F of the compare match output.

<TCFFENn>: Controls enable/disable of F/F reversal of the compare match output.

(Note) Values read from bits 7, 3 and 2 of CMPCTLn are all "0."

Fig. 12.3.9 TMRC-related Registers

TMRC Compare Control Register (CMPCTLn)

CMPCTL4 (FFFFF474H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN4	TCFFC41	TCFFC40			CMPRDE4	CMPEN4
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF4 reversal 0: Disable 1: Enable	TCFF4 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 4 0: Disable 1: Enable	Compare 4 enable 0: Disable 1: Enable

CMPCTL5 (FFFFF475H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN5	TCFFC51	TCFFC50			CMPRDE5	CMPEN5
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF5 reversal 0: Disable 1: Enable	TCFF5 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 5 0: Disable 1: Enable	Compare 5 enable 0: Disable 1: Enable

CMPCTL6 (FFFFF476H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN6	TCFFC61	TCFFC60			CMPRDE6	CMPEN6
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF6 reversal 0: Disable 1: Enable	TCFF6 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 6 0: Disable 1: Enable	Compare 6 enable 0: Disable 1: Enable

CMPCTL7 (FFFFF477H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN7	TCFFC71	TCFFC70			CMPRDE7	CMPEN7
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF7 reversal 0: Disable 1: Enable	TCFF7 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 7 0: Disable 1: Enable	Compare 7 enable 0: Disable 1: Enable

<CMPENn>: Controls enable/disable of the compare match detection.

<CMPRDEn>: Controls enable/disable of double buffers of the compare register.

<TCFFCn1:0>: Controls F/F of the compare match output.

<TCFFENn>: Controls enable/disable of F/F reversal of the compare match output.

(Note) Values read from bits 7, 3 and 2 of CMPCTLn are all "0."

Fig. 12.3.10 TMRC-related Register

TMRC Compare Control Register (CMPCTLn)

CMPCTL8 (FFFFF478H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN8	TCFFC81	TCFFC80			CMPRDE8	CMPEN8
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF8 reversal 0: Disable 1: Enable	TCFF8 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 8 0: Disable 1: Enable	Compare 8 enable 0: Disable 1: Enable

CMPCTL9 (FFFFF479H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN9	TCFFC91	TCFFC90			CMPRDE9	CMPEN9
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF9 reversal 0: Disable 1: Enable	TCFF9 control 00: Reversal 01: Set 10: Clear 11: Don't care				Double buffers 9 0: Disable 1: Enable	Compare 9 enable 0: Disable 1: Enable

<CMPENn>: Controls enable/disable of the compare match detection.

<CMPRDEn>: Controls enable/disable of double buffers of the compare register.

<TCFFCn1:0>: Controls F/F of the compare match output.

<TCFFENn>: Controls enable/disable of F/F reversal of the compare match output.

(Note) Values read from bits 7, 3 and 2 of CMPCTLn are all "0."

Fig. 12.3.11 TMRC-related Registers

TMRC Compare Register 0 (TCCMP0)

TCCMP0LL (FFFFF440H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP007	CMP006	CMP005	CMP004	CMP003	CMP002	CMP001	CMP000
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bits 7 through 0)							

TCCMP0LH (FFFFF441H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP017	CMP016	CMP015	CMP014	CMP013	CMP012	CMP011	CMP010
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bits 15 through 8)							

TCCMP0HL (FFFFF442H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP027	CMP026	CMP025	CMP024	CMP023	CMP022	CMP021	CMP020
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bits 23 through 16)							

TCCMP0HH (FFFFF443H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP037	CMP036	CMP035	CMP034	CMP033	CMP032	CMP031	CMP030
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bits 31 through 24)							

TMRC Compare Register 1 (TCCMP1)

TCCMP1LL (FFFFF444H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP107	CMP106	CMP105	CMP104	CMP103	CMP102	CMP101	CMP100
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bits 7 through 0)							

TCCMP1LH (FFFFF445H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP117	CMP116	CMP115	CMP114	CMP113	CMP112	CMP111	CMP110
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bits 15 through 8)							

TCCMP1HL (FFFFF446H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP127	CMP126	CMP125	CMP124	CMP123	CMP122	CMP121	CMP120
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bits 23 through 16)							

TCCMP1HH (FFFFF447H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP137	CMP136	CMP135	CMP134	CMP133	CMP132	CMP131	CMP130
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bits 31 through 24)							

Fig. 12.3.12 TMRC-related Registers

TMRC Compare Register 2 (TCCMP2)

TCCMP2LL (FFFFF448H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP207	CMP206	CMP205	CMP204	CMP203	CMP202	CMP201	CMP200
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 2 data (bits 7 through 0)							

TCCMP2LH (FFFFF449H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP217	CMP216	CMP215	CMP214	CMP213	CMP212	CMP211	CMP210
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 2 data (bits 15 through 8)							

TCCMP2HL (FFFFF44AH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP227	CMP226	CMP225	CMP224	CMP223	CMP222	CMP221	CMP220
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 2 data (bits 23 through 16)							

TCCMP2HH (FFFFF44BH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP237	CMP236	CMP235	CMP234	CMP233	CMP232	CMP231	CMP230
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 2 data (bits 31 through 24)							

TMRC Compare Register 3 (TCCMP3)

TCCMP3LL (FFFFF44CH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP307	CMP306	CMP305	CMP304	CMP303	CMP302	CMP301	CMP300
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 3 data (bits 7 through 0)							

TCCMP3LH (FFFFF44DH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP317	CMP316	CMP315	CMP314	CMP313	CMP312	CMP311	CMP310
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 3 data (bits 15 through 8)							

TCCMP3HL (FFFFF44EH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP327	CMP326	CMP325	CMP324	CMP323	CMP322	CMP321	CMP320
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 3 data (bits 23 through 16)							

TCCMP3HH (FFFFF44FH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP337	CMP336	CMP335	CMP334	CMP333	CMP332	CMP331	CMP330
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 3 data (bits 31 through 24)							

Fig. 12.3.13 TMRC-related Registers

TMRC Compare Register 4 (TCCMP4)

TCCMP4LL (FFFFF450H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP407	CMP406	CMP405	CMP404	CMP403	CMP402	CMP401	CMP400
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 4 data (bits 7 through 0)							

TCCMP4LH (FFFFF451H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP417	CMP416	CMP415	CMP414	CMP413	CMP412	CMP411	CMP410
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 4 data (bits 15 through 8)							

TCCMP4HL (FFFFF452H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP427	CMP426	CMP425	CMP424	CMP423	CMP422	CMP421	CMP420
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 4 data (bits 23 through 16)							

TCCMP4HH (FFFFF453H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP437	CMP436	CMP435	CMP434	CMP433	CMP432	CMP431	CMP430
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 4 data (bits 31 through 24)							

TMRC Compare Register 5 (TCCMP5)

TCCMP5LL (FFFFF454H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP507	CMP506	CMP505	CMP504	CMP503	CMP502	CMP501	CMP500
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 5 data (bits 7 through 0)							

TCCMP5LH (FFFFF455H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP517	CMP516	CMP515	CMP514	CMP513	CMP512	CMP511	CMP510
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 5 data (bits 15 through 8)							

TCCMP5HL (FFFFF456H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP527	CMP526	CMP525	CMP524	CMP523	CMP522	CMP521	CMP520
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 5 data (bits 23 through 16)							

TCCMP5HH (FFFFF457H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP537	CMP536	CMP535	CMP534	CMP533	CMP532	CMP531	CMP530
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 5 data (bits 31 through 24)							

Fig. 12.3.14 TMRC-related Registers

TMRC Compare Register 6 (TCCMP6)

TCCMP6LL (FFFFF458H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP607	CMP606	CMP605	CMP604	CMP603	CMP602	CMP601	CMP600
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 6 data (bits 7 through 0)							

TCCMP6LH (FFFFF459H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP617	CMP616	CMP615	CMP614	CMP613	CMP612	CMP611	CMP610
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 6 data (bits 15 through 8)							

TCCMP6HL (FFFFF45AH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP627	CMP626	CMP625	CMP624	CMP623	CMP622	CMP621	CMP620
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 6 data (bits 23 through 16)							

TCCMP6HH (FFFFF45BH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP637	CMP636	CMP635	CMP634	CMP633	CMP632	CMP631	CMP630
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 6 data (bits 31 through 24)							

TMRC Compare Reg7 (TCCMP7)

TCCMP7LL (FFFFF45CH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP707	CMP706	CMP705	CMP704	CMP703	CMP702	CMP701	CMP700
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 7 data (bits 7 through 0)							

TCCMP7LH (FFFFF45DH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP717	CMP716	CMP715	CMP714	CMP713	CMP712	CMP711	CMP710
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 7 data (bits 15 through 8)							

TCCMP7HL (FFFFF45EH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP727	CMP726	CMP725	CMP724	CMP723	CMP722	CMP721	CMP720
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 7 data (bits 23 through 16)							

TCCMP7HH (FFFFF45FH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP737	CMP736	CMP735	CMP734	CMP733	CMP732	CMP731	CMP730
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 7 data (bits 31 through 24)							

Fig. 12.3.15 TMRC-related Registers

TMRC Compare Register 8 (TCCMP8)

TCCMP8LL (FFFFF460H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP807	CMP806	CMP805	CMP804	CMP803	CMP802	CMP801	CMP800
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 8 data (bits 7 through 0)							

TCCMP8LH (FFFFF461H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP817	CMP816	CMP815	CMP814	CMP813	CMP812	CMP811	CMP810
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 8 data (bits 15 through 8)							

TCCMP8HL (FFFFF462H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP827	CMP826	CMP825	CMP824	CMP823	CMP822	CMP821	CMP820
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 8 data (bits 23 through 16)							

TCCMP8HH (FFFFF463H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP837	CMP836	CMP835	CMP834	CMP833	CMP832	CMP831	CMP830
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 8 data (bits 31 through 24)							

TMRC Compare Register 9 (TCCMP9)

TCCMP9LL (FFFFF464H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP907	CMP906	CMP905	CMP904	CMP903	CMP902	CMP901	CMP900
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 9 data (bits 7 through 0)							

TCCMP9LH (FFFFF465H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP917	CMP916	CMP915	CMP914	CMP913	CMP912	CMP911	CMP910
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 9 data (bits 15 through 8)							

TCCMP9HL (FFFFF466H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP927	CMP926	CMP925	CMP924	CMP923	CMP922	CMP921	CMP920
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 9 data (bits 23 through 16)							

TCCMP9HH (FFFFF467H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP937	CMP936	CMP935	CMP934	CMP933	CMP932	CMP931	CMP930
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 9 data (bits 31 through 24)							

Fig. 12.3.16 TMRC-related Registers

13. Serial Channel (SIO)

13.1 Features

This device has seven serial I/O channels: SIO0 to SIO6. Each channel operates in either the UART mode (asynchronous communication) or the I/O interface mode (synchronous communication) which is selected by the user.

I/O interface mode — Mode 0: This is the mode to send and receive I/O data and associated synchronization signals (SCLK) to extend I/O.

Asynchronous (UART) mode:
- Mode 1: TX/RX Data Length: 7 bits
- Mode 2: TX/RX Data Length: 8 bits
- Mode 3: TX/RX Data Length: 9 bits

In the above modes 1 and 2, parity bits can be added. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). Figure shows the block diagram of SIO0.

Each channel consists of a prescaler, a serial clock generation circuit, a receive buffer and its control circuit, and a send buffer and its control circuit. Each channel functions independently.

As the SIOs 0 to 6 operate in the same way, Only SIO0 is described here.

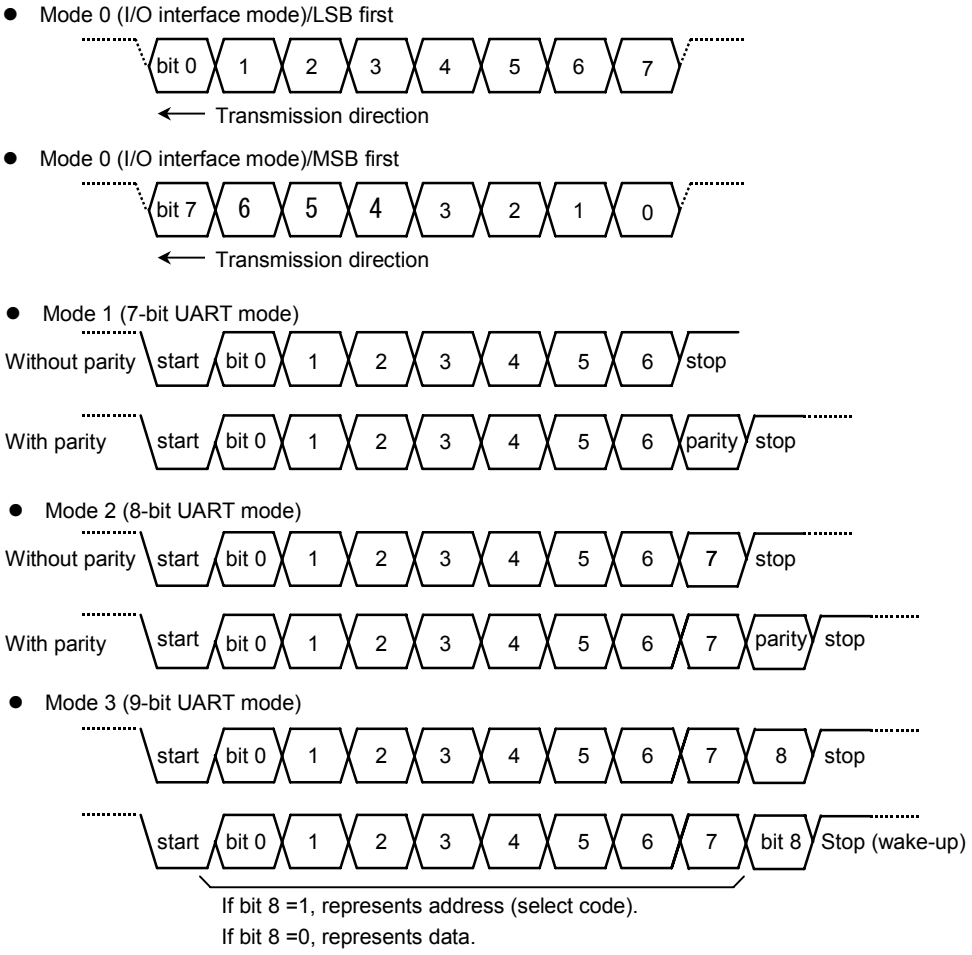


Fig. 13.1 Data Format

13.2 Block Diagram (Channel 0)

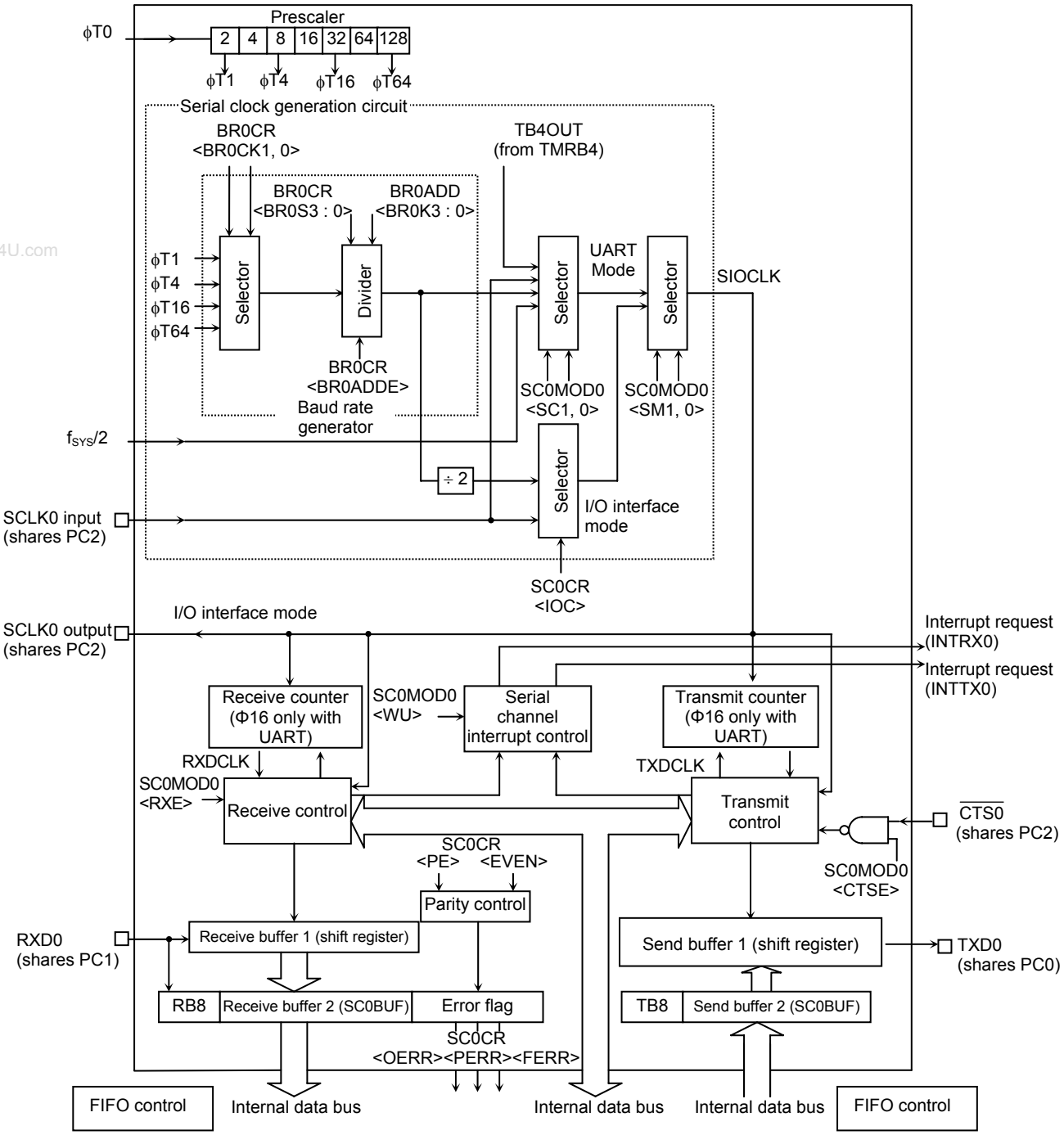


Fig. 13.2.1 SIO0 Block Diagram

13.3 Operation of Each Circuit (Channel 0)

13.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock ϕT0 to the prescaler is selected by SYSCR of CG <PRCK1:0> to provide the frequency of either fperiph/2, fperiph/4, fperiph/8, or fperiph/16. The clock frequency fperiph is either the clock "fgear," to be selected by SYSCR1<FPSEL> of CG, or the clock "fc" before it is divided by the clock gear.

The prescaler becomes active only when the baud rate generator is selected for generating the serial transfer clock. Table 13.3.1 lists the prescaler output clock resolution.

Table 13.3.1 Clock Resolution to the Baud Rate Generator

@fc = 54MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK1:0>	Prescaler output clock resolution			
			ϕT1	ϕT4	ϕT16	ϕT64
0 (fgear)	000(fc)	00(fperiph/16)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	fc/2^3(0.15μs)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)
		11(fperiph/2)	fc/2^2(0.07μs)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)
	100(fc/2)	00(fperiph/16)	fc/2^6(1.2μs)	Fc/2^8(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)
		01(fperiph/8)	fc/2^5(0.6μs)	Fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		10(fperiph/4)	fc/2^4(0.3μs)	Fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)
		11(fperiph/2)	fc/2^3(0.15μs)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)
	110(fc/4)	00(fperiph/16)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)	fc/2^{13}(152μs)
		01(fperiph/8)	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)
		10(fperiph/4)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		11(fperiph/2)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7 μs)	fc/2^{10}(19.0μs)
	111(fc/8)	00(fperiph/16)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)	fc/2^{14}(303μs)
		01(fperiph/8)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)	fc/2^{13}(152μs)
		10(fperiph/4)	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)
		11(fperiph/2)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
1 (fc)	000(fc)	00(fperiph/16)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	fc/2^3(0.15μs)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)
		11(fperiph/2)	fc/2^2(0.07μs)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)
	100(fc/2)	00(fperiph/16)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	fc/2^3(0.15μs)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)
		11(fperiph/2)	—	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)
	110(fc/4)	00(fperiph/16)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	—	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)
		11(fperiph/2)	—	fc/2^4(0.3μs)	fc/2^6(1.2μs)	fc/2^8(4.7μs)
	111(fc/8)	00(fperiph/16)	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	—	fc/2^6(1.2μs)	fc/2^8(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	—	fc/2^5(0.6μs)	fc/2^7(2.4μs)	fc/2^9(9.5μs)
		11(fperiph/2)	—	—	fc/2^6(1.2μs)	fc/2^8(4.7μs)

(Note 1) The prescaler output clock ϕTn must be selected so that the relationship "ϕTn < fsys/2" is satisfied (so that ϕTn is slower than fsys/2).

(Note 2) Do not change the clock gear while SIO is operating.

(Note 3) The horizontal lines in the above table indicate that the setting is prohibited.

The serial interface baud rate generator uses four different clocks, i.e., ϕT1, ϕT4, ϕT16 and ϕT64, supplied from the prescaler output clock.

13.3.2 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses either the ϕT1, ϕT4, ϕT16 or ϕT64 clock supplied from the 7-bit prescaler. This input clock selection is made by setting the baud rate setting register, BR0CR <BR0CK1:0>.

The baud rate generator contains built-in dividers for divide by 1, (N + m/16), and 16 where N is a number from 2 to 15 and m is a number from 0 to 15. The division is performed according to the settings of the baud rate control registers BR0CR <BR0ADDE> <BR0S3:0> and BR0ADD <BR0K3:0> to determine the resulting transfer rate.

- UART Mode:

  1) If BR0CR <BR0ADDE> = 0,

     The setting of BR0ADD <BR0K3:0> is ignored and the counter is divided by N where N is the value set to BR0CR <BR0S3:0>. (N = 1 to 16).

  2) If BR0CR <BR0ADDE> = 1,

     The N + (16 - K)/16 division function is enabled and the division is made by using the values N (set in BR0CR <BR0S3:0>) and K (set in BR0ADD<BR0K3:0>). (N = 2 to 15, K = 1 to 15)

     Note For the N values of 1 and 16, the above N+(16-K)/16 division function is inhibited. So, be sure to set BR0CR<BR0ADDE> to "0."

- I/O interface mode:

  The N + (16 - K)/16 division function cannot be used in the I/O interface mode. Be sure to divide by N, by setting BR0CR <BR0ADDE> to "0."

- Baud rate calculation to use the baud rate generator:

  1) UART mode

     $$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} /16$$

     The highest baud rate out of the baud rate generator is 843.75 kbps when ϕT1 is 13.5 MHz.

     The fsys/2 frequency, obtained by dividing the system clock by 2, can be used as the serial clock. In this case, the highest baud rate will be 1.68 Mbps when fsys is 54 MHz.

2) I/O interface mode

$$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} /2$$

The highest baud rate will be generated when ϕT1 is 13.5 MHz. If double buffering is used, the divide ratio can be set to "1" and the resulting output baud rate will be 6.75 Mbps. (If double buffering is not used, the highest baud rate will be 3.375 Mbps applying the divide ratio of "2.")

- Example baud rate setting:

1) Division by an integer (divide by N):

Selecting fc = 54 MHz for fperiph, setting ϕT0 to fperiph/16, using the baud rate generator input clock ϕT1, setting the divide ratio N (BR0CR<BR0S3:0>) = 4, and setting BR0CR<BR0ADDE> = "0," the resulting baud rate in the UART mode is calculated as follows:

* Clocking conditions
  - System clock : High-speed (fc)
  - High speed clock gear : x 1 (fc)
  - Prescaler clock : fperiph/16 (fperiph = fsys)

$$\text{Baud rate} = \frac{fc/32}{4} /16$$

$$= 54 \times 10^6 / 32 / 4 / 16 = 26367 \text{ (bps)}$$

(Note) The divide by (N + (16-K)/16) function is inhibited and thus BR0ADD <BR0K3:0> is ignored.

2) For divide by N + (16-K)/16 (only for UART mode):

Selecting fc = 54 MHz MHz for fperiph, setting ϕT0 to fperiph/16, using the baud rate generator input clock ϕT2, setting the divide ratio N (BR0CR<BR0S3:0>) = 4, setting K (BR0ADD<BR0K3:0>) = 14, and selecting BR0CR<BR0ADDE> = 1, the resulting baud rate is calculated as follows:

* Clocking conditions
  - System clock : High-speed (fc)
  - High-speed clock gear : x 1 (fc)
  - Prescaler clock : fperiph/16 (fperiph = fsys)

$$\text{Baud rate} = \frac{fc/32}{4 + \frac{(16-14)}{16}} /16$$

$$= 54 \times 10^6 / 32 / (4 + \frac{2}{16}) / 16 = 25568 \text{ (bps)}$$

Also, an external clock input may be used as the serial clock. The resulting baud rate calculation is shown below:

- Baud rate calculation for an external clock input:

  1) UART mode

     Baud Rate = external clock input / 16

     In this, the period of the external clock input must be equal to or greater than 4/fsys.

     If fsys = 54 MHz, the highest baud rate will be 54 / 4 / 16 = 844 (kbps).

  2) I/O interface mode

     Baud Rate = external clock input

     When double buffering is used, it is necessary to satisfy the following relationship:

     External clock input period > 12/fsys

     Therefore, when fsys = 54 MHz, the baud rate must be set to a rate lower than 54 / 12 = 4.5 (Mbps).

     When double buffering is not used, it is necessary to satisfy the following relationship:

     External clock input period > 16/fsys

     Therefore, when fsys = 54 MHz, the baud rate must be set to a rate lower than 54 / 16 = 3.375 (Mbps).

     Example baud rates for the UART mode are shown in Table 13.3.2.1 and Table 13.3.2.2.

Table 13.3.2.1 Selection of UART Baud Rate

(Use the baud rate generator with BR0CR <BR0ADDE> = 0)

Unit (kbps)

fc [MHz]	Input clock / Divide ratio N (Set to BR0CR <BR0S3:0>)	ϕT1 (fc/4)	ϕT4 (fc/16)	ϕT16 (fc/64)	ϕT64 (fc/256)
19.6608	1	307.200	76.800	19.200	4.800
↑	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	0	19.200	4.800	1.200	0.300
24.576	5	76.800	19.200	4.800	1.200
↑	A	38.400	9.600	2.400	0.600
29.4912	1	460.800	115.200	28.800	7.200
↑	2	230.400	57.600	14.400	3.600
↑	3	153.600	38.400	9.600	2.400
↑	4	115.200	28.800	7.200	1.800
↑	6	76.800	19.200	4.800	1.200
↑	C	38.400	9.600	2.400	0.600

(Note) This table shows the case where the system clock is set to fc, the clock gear is set to fc/1, and the prescaler clock is set to fperiph/2.

Table 13.3.2.2 Selection of UART Baud Rate

(The TMRB4 timer output (internal TB4OUT) is used with the timer input clock set to ϕT0.)

Unit (kbps)

fc / TB4RG0H/L	29.4912 MHz	24.576 MHz	24 MHz	19.6608 MHz	16 MHz	12.288 MHz
0001H	230.4	192	187.5	153.6	125	96
0002H	115.2	96	93.75	76.8	62.5	48
0003H	76.8	64	62.5	51.2	41.67	32
0004H	57.6	48	46.88	38.4	31.25	24
0005H	46.08	38.4	37.5	30.72	25	19.2
0006H	38.4	32	31.25	25.6	20.83	16
0008H	28.8	24	23.44	19.2	15.63	12
000AH	23.04	19.2	18.75	15.36	12.5	9.6
0010H	14.4	12	11.72	9.6	7.81	6
0014H	11.52	9.6	9.38	7.68	6.25	4.8

Baud rate calculation to use the TMRB4 timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by SYSCR0} < \text{PRCK1 : 0} >}{\text{TB4REG} \times \underline{2} \times 16}$$

(When input clock to the timer TMRB4 is ϕT0)

(Note 1) In the I/O interface mode, the TMRB4 timer output signal cannot be used internally as the transfer clock.

(Note 2) This table shows the case where the system clock is set to fc, the clock gear is set to fc/1, and the prescaler clock is set to fperiph/4.

13.3.3 Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

- I/O interface mode:

  In the SCLK output mode with the SC0CR <IOC> serial control register set to "0," the output of the previously mentioned baud rate generator is divided by 2 to generate the basic clock.

  In the SCLK input mode with SC0CR <IOC> set to "1," rising and falling edges are detected according to the SC0CR <SCLKS> setting to generate the basic clock.

- Asynchronous (UART) mode:

  According to the settings of the serial control mode register SC0MOD0 <SC1:0>, either the clock from the baud rate register, the system clock ($f_{SYS}/2$), the internal output signal of the TMRB4 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

13.3.4 Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by SIOCLK. Sixteen SIOCLK clock pulses are used in receiving a single data bit while the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

13.3.4 Receive Control Unit

- I/O interface mode:

  In the SCLK output mode with SC0CR <IOC> set to "0," the RXD0 pin is sampled on the rising edge of the shift clock output to the SCLK0 pin.

  In the SCLK input mode with SC0CR <IOC> set to "1," the serial receive data RXD0 pin is sampled on the rising or falling edge of SCLK input depending on the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

  The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

13.3.5 Receive Buffer

The receive buffer is of a dual structure to prevent overrun errors. The first receive buffer (a shift register) stores the received data bit-by-bit. When a complete set of bits have been stored, they are moved to the second receive buffer (SC0BUF). At the same time, the receive buffer full flag (SC0MOD2 "RBFLL") is set to "1" to indicate that valid data is stored in the second receive buffer. However, if the receive FIFO is set enabled, the receive data is moved to the receive FIFO and this flag is immediately cleared.

If the receive FIFO has been disabled (SCOFCNF <CNFG> = 0 and SC0MOD1<FDPX1:0>=01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SCNFCNF <CNFG> = 1 and SC0MOD1<FDPX1:0>=01/11), an interrupt will be generated according to the SC0RFC <RIL2:0> setting.

The CPU will read the data from either the second receive buffer (SC0BUF) or from the receive FIFO (the address is the same as that of the receive buffer). If the receive FIFO has not been enabled, the receive buffer full flag SC0MOD2<RBFLL> is cleared to "0" by the read operation. The next data received can be stored in the first receive buffer even if the CPU has not read the previous data from the second receive

buffer (SC0BUF) or the receive FIFO.

If SCLK is set to generate clock output in the I/O interface mode, the double buffer control bit SC0MOD2 <WBUF> can be programmed to enable or disable the operation of the second receive buffer (SCOBUF).

By disabling the second receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (SCOFCNF <CNFG> = 0 and <FDPX1:0> = 01), handshaking with the other side of communication can be enabled and the SCLK output stops each time one frame of data is transferred. In this setting, the CPU reads data from the first receive buffer. By the read operation of CPU, the SCLK output resumes.

If the second receive buffer (i.e., double buffering) is enabled but the receive FIFO is not enabled, the SCLK output is stopped when the first receive data is moved from the first receive buffer to the second receive buffer and the next data is stored in the first buffer filling both buffers with valid data. When the second receive buffer is read, the data of the first receive buffer is moved to the second receive buffer and the SCLK output is resumed upon generation of the receive interrupt INTRX. Therefore, no buffer overrun error will be caused in the I/O interface SCLK output mode regardless of the setting of the double buffer control bit SC0MOD2 <WBUF>.

If the second receive buffer (double buffering) is enabled and the receive FIFO is also enabled (SCNFCNF <CNFG> = 1 and <FDPX1:0> = 01/11), the SCLK output will be stopped when the receive FIFO is full (according to the setting of SCOFNCF <RFST>) and both the first and second receive buffers contain valid data. Also in this case, if SCOFCNF <RXTXCNT> has been set to "1," the receive control bit RXE will be automatically cleared upon suspension of the SCLK output. If it is set to "0," automatic clearing will not be performed.

(Note) In this mode, the SC0CR <OEER> flag is insignificant and the operation is undefined. Therefore, before switching from the SCLK output mode to another mode, the SC0CR register must be read to initialize this flag.

In other operating modes, the operation of the second receive buffer is always valid, thus improving the performance of continuous data transfer. If the receive FIFO is not enabled, an overrun error occurs when the data in the second receive buffer (SC0BUF) has not been read before the first receive buffer is full with the next receive data. If an overrun error occurs, data in the first receive buffer will be lost while data in the second receive buffer and the contents of SC0CR <RB8> remain intact. If the receive FIFO is enabled, the FIFO must be read before the FIFO is full and the second receive buffer is written by the next data through the first buffer. Otherwise, an overrun error will be generated and the receive FIFO overrun error flag will be set. Even in this case, the data already in the receive FIFO remains intact.

The parity bit to be added in the 8-bit UART mode as well as the most significant bit in the 9-bit UART mode will be stored in SC0CR <RB8>.

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1."

13.3.6 Receive FIFO Buffer

In addition to the double buffer function already described, data may be stored using the receive FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte receive buffer can be enabled. Also, in the UART mode or I/O interface mode, data may be stored up to a predefined fill level. When the receive FIFO buffer is to be used, be sure to enable the double buffer function.

If data with parity bit is to be received in the UART mode, parity check must be performed each time a data frame is received.

13.3.7 Receive FIFO Operation

① I/O interface mode with SCLK output:

The following example describes the case a 4-byte data stream is received in the half duplex mode:

SC0RFC<7:6>=01: Clears receive FIFO and sets the condition of interrupt generation.

SC0RFC<1:0>=00: Sets the interrupt to be generated at fill level 4.

SC0FCNF <1:0>=10111: Automatically inhibits continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

In this condition, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. After receiving 4 bytes, the RXE bit is automatically cleared and the receive operation is stopped (SCLK is stopped).

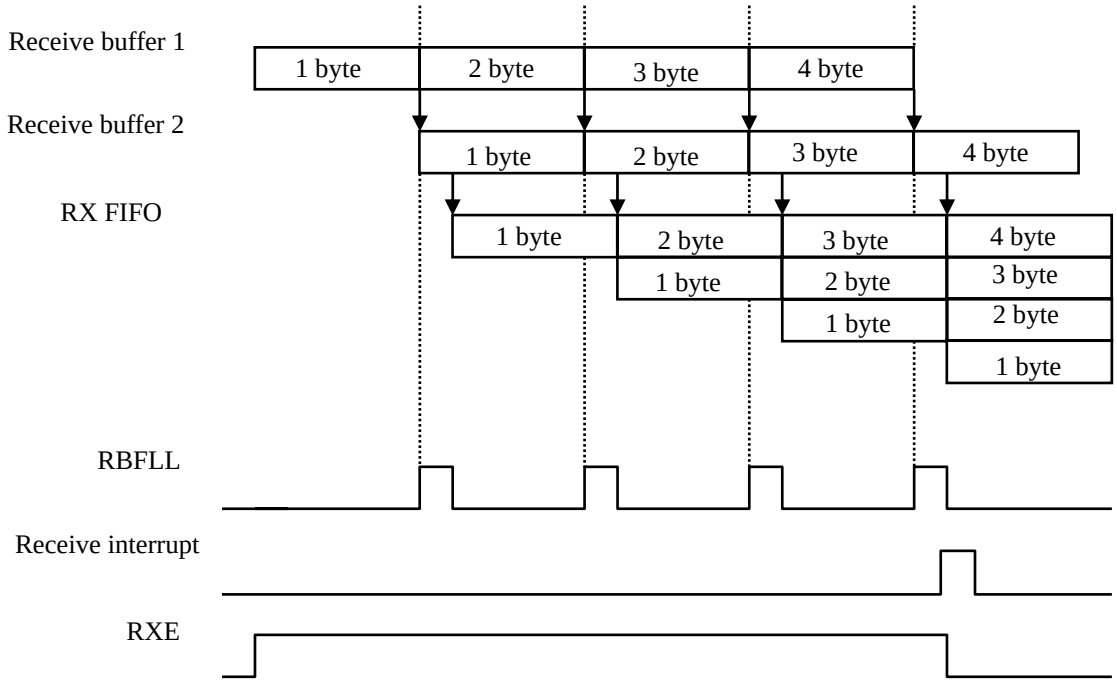


Fig. 13.3.7.1 Receive FIFO Operation

② I/O interface mode with SCLK input:

The following example describes the case a 10-byte data stream is received:

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

SC0RFC <1:0> = 00: Sets the interrupt to be generated at fill level 4.

SC0FCNF <1:0> = 10101: Automatically allows continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the maximum allowable number.

In this condition, 4-byte data reception can be initiated along with the input clock by setting the half duplex transmission mode and writing "1" to the RXE bit. When the 4-byte data reception is completed, the receive FIFO interrupt will be generated.

Note that preparation for the next data reception can be managed in this setting, i.e., the next 4-byte data can be received before data is fully read from the FIFO.

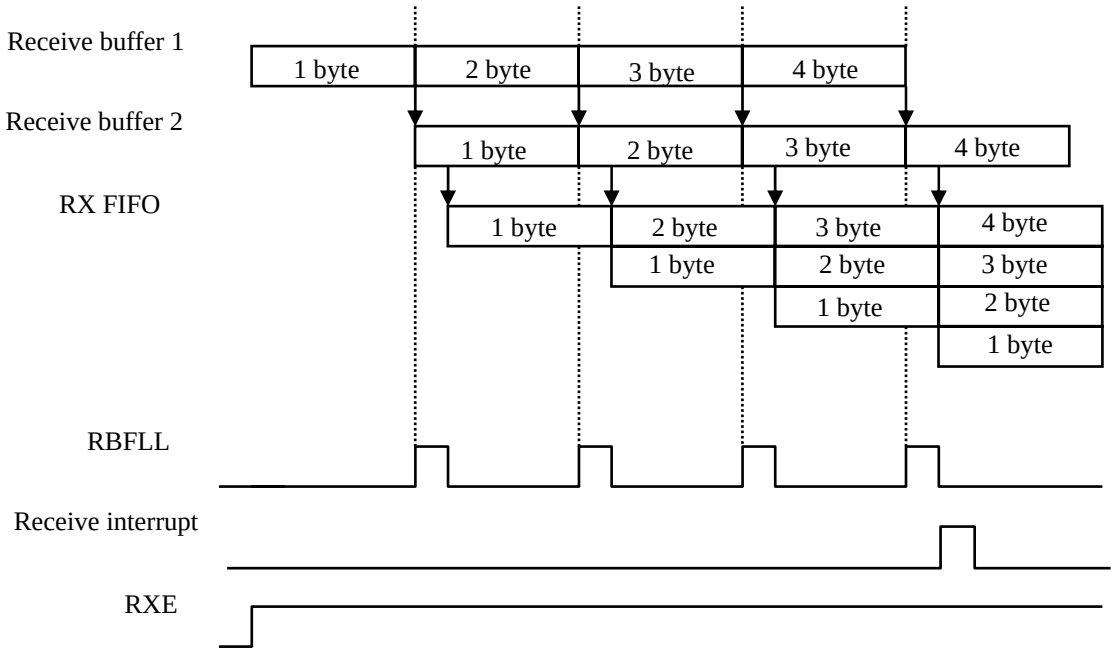


Fig. 13.3.7.2 Receive FIFO Operation

13.3.8 Transmit Counter

The transmit counter is a 4-bit binary counter used in the asynchronous communication (UART) mode. It is counted by SIOCLK as in the case of the receive counter and generates a transmit clock (TXDCLK) on every 16th clock pulse.

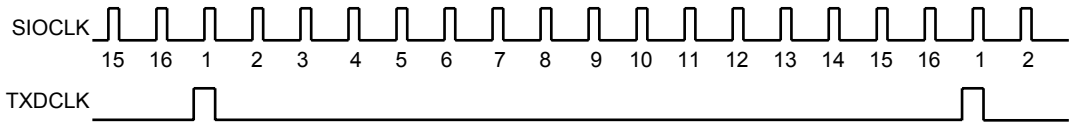


Fig. 13.3.8.1 Transmit Clock Generation

13.3.9 Transmit Control Unit

- I/O interface mode:

  In the SCLK output mode with SC0CR <IOC> set to "0," each bit of data in the send buffer is output to the TXD0 pin on the rising edge of the shift clock output from the SCLK0 pin.

  In the SCLK input mode with SC0CR <IOC> set to "1," each bit of data in the send buffer is output to the TXD0 pin on the rising or falling edge of the input SCLK signal according to the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

  When the CPU writes data to the send buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock (TXDSFT) is also generated.

- Handshake function

The $\overline{CTS}$ pin enables frame by frame data transmission so that overrun errors can be prevented. This function can be enabled or disabled by SC0MOD0 <CTSE>.

When the $\overline{CTS}$ pin is set to the "H" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{CTS}$ pin returns to the "L" level. However in this case, the INTTX0 interrupt is generated, the next transmit data is requested to the CPU, data is written to the send buffer, and it waits until it is ready to transmit data.

Although no $\overline{RTS}$ pin is provided, a handshake control function can be easily implemented by assigning a port for the $\overline{RTS}$ function. By setting the port to "H" level upon completion of data reception (in the receive interrupt routine), the transmit side can be requested to suspend data transmission.

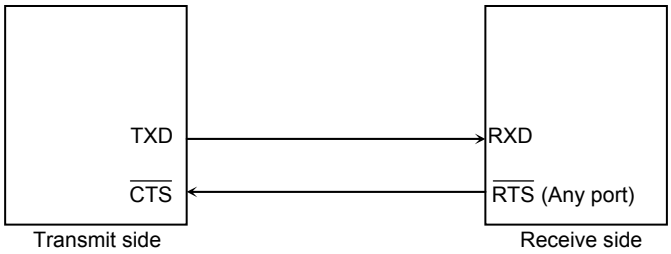
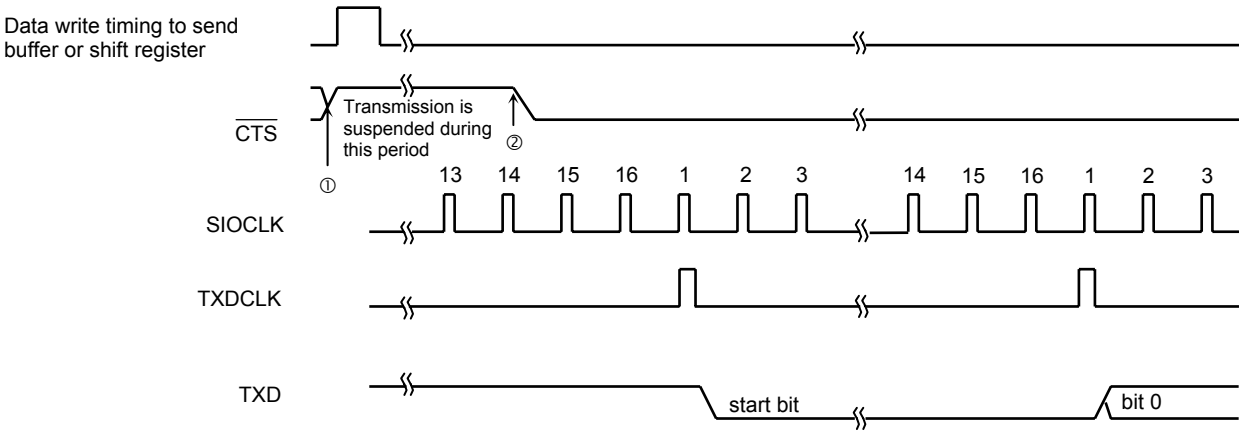


Fig. 13.3.9.1 Handshake Function



(Note)

① If the $\overline{CTS}$ signal is set to "H" during transmission, the next data transmission is suspended after the current transmission is completed.

② Data transmission starts on the first falling edge of the TXDCLK clock after $\overline{CTS}$ is set to "L."

Fig. 13.3.9.2 $\overline{CTS}$ (Clear to Send) Signal Timing

13.3.10 Transmit Buffer

The send buffer (SC0BUF) is in a dual structure. The double buffering function may be enabled or disabled by setting the double buffer control bit <WBUF> in serial mode control register 2 (SC0MOD2). If double buffering is enabled, data written to send buffer 2 (SCOBUF) is moved to send buffer 1 (shift register).

If the transmit FIFO has been disabled (SCOFCNF <CNFG> = 0 or 1 and <FDPX1:0> = 01), the INTTX interrupt is generated at the same time and the send buffer empty flag <TBEMP> of SC0MOD2 is set to "1." This flag indicates that send buffer 2 is now empty and that the next transmit data can be written. When the next data is written to send buffer 2, the <TBEMP> flag is cleared to "0."

If the transmit FIFO has been enabled (SCNFCNF <CNFG> = 1 and <FDPX1:0> = 10/11), any data in the transmit FIFO is moved to the send buffer 2 and <TBEMP> flag is immediately cleared to "0." The CPU writes data to send buffer 2 or to the transmit FIFO.

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in send buffer 2 before the next frame clock input, which occurs upon completion of data transmission from send buffer 1, an under-run error occurs and a serial control register (SC0CR) <PERR> parity/under-run flag is set.

If the transmit FIFO is enabled in the I/O interface SCLK input mode, when data transmission from send buffer 1 is completed, the send buffer 2 data is moved to send buffer 1 and any data in transmit FIFO is moved to send buffer 2 at the same time.

If the transmit FIFO is disabled in the I/O interface SCLK output mode, when data in send buffer 2 is moved to send buffer 1 and the data transmission is completed, the SCLK output stops. So, no under-run errors can be generated.

If the transmit FIFO is enabled in the I/O interface SCLK output mode, the SCLK output stops upon completion of data transmission from send buffer 1 if there is no valid data in the transmit FIFO.

Note) In the I/O interface SCLK output mode, the SC0CR <PEER> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the SCLK output mode to another mode, SC0CR must be read in advance to initialize the flag.

If double buffering is disabled, the CPU writes data only to send buffer 1 and the transmit interrupt INTTX is generated upon completion of data transmission.

If handshaking with the other side is necessary, set the double buffer control bit <WBUF> to "0" (disable) to disable send buffer 2; any setting for the transmit FIFO should not be performed.

13.3.11 Transmit FIFO Buffer

In addition to the double buffer function already described, data may be stored using the transmit FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte send buffer can be enabled. In the UART mode or I/O interface mode, up to 4 bytes of data may be stored.

If data is to be transmitted with a parity bit in the UART mode, parity check must be performed on the receive side each time a data frame is received.

13.3.12 Transmit FIFO Operation

① I/O interface mode with SCLK output (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

SC0TFC <7:6> = 01: Clears transmit FIFO and sets the condition of interrupt generation

SC0TFC <1:0> = 00: Sets the interrupt to be generated at fill level 0.

SC0FCNF <1:0> = 01011: Inhibits continued transmission after reaching the fill level.

In this condition, data transmission can be initiated by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the send buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.

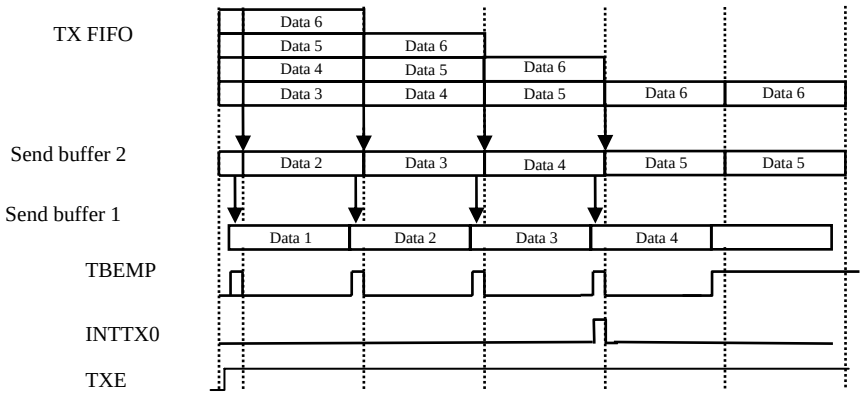


Fig. 13.3.12.1 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

SC0TFC <1:0> = 01: Clears the transmit FIFO and sets the condition of interrupt generation.

SC0TFC <7:2> = 000000: Sets the interrupt to be generated at fill level 0.

SC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the send buffer, the transmit FIFO interrupt is generated

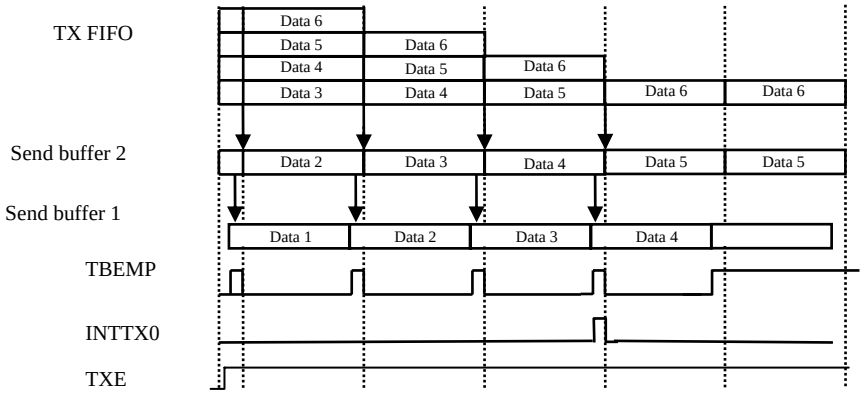


Fig. 13.3.12.2 Transmit FIFO Operation

13.3.13 Parity Control Circuit

If the parity addition bit <PE> of the serial control register SC0CR is set to "1," data is sent with the parity bit. Note that the parity bit may be used only in the 7- or 8-bit UART mode. The <EVEN> bit of SC0CR selects either even or odd parity.

Upon data transmission, the parity control circuit automatically generates the parity with the data written to the send buffer (SC0BUF). After data transmission is complete, the parity bit will be stored in SC0BUF bit 7 <TB7> in the 7-bit UART mode and in bit 7 <TB8> in the serial mode control register SC0MOD in the 8-bit UART mode. The <PE> and <EVEN> settings must be completed before data is written to the send buffer.

Upon data reception, the parity bit for the received data is automatically generated while the data is shifted to receive buffer 1 and moved to receive buffer 2 (SC0BUF). In the 7-bit UART mode, the parity generated is compared with the parity stored in SC0BUF <RB7>, while in the 8-bit UART mode, it is compared with the bit 7 <RB8> of the SC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the SC0CR register is set.

In the I/O interface mode, the SC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

13.3.14 Error Flag

Three error flags are provided to increase the reliability of received data.

1. Overrun error <OERR>: Bit 4 of the serial control register SC0CR

   In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame receive data before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied). This flag is set to "0" when it is read. In the I/O interface SCLK output mode, no overrun error is generated and therefore, this flag is inoperative and the operation is undefined.

2. Parity error/under-run error <PERR>: Bit 3 of the SC0CR register

   In the UART mode, this bit is set to "1" when a parity error is generated. A parity error is generated when the parity generated from the received data is different from the parity received. This flag is set to "0" when it is read.

   In the I/O interface mode, this bit indicates an under-run error. When the double buffer control bit <WBUF> of the serial mode control register SC0MOD2 is set to "1" in the SCLK input mode, if no data is set to the transmit double buffer before the next data transfer clock after completing the transmission from the transmit shift register, this error flag is set to "1" indicating an under-run error. If the transmit FIFO is enabled, any data content in the transmit FIFO will be moved to the buffer. When the transmit FIFO and the double buffer are both empty, an under-run error will be generated. Because no under-run errors can be generated in the SCLK output mode, this flag is inoperative and the operation is undefined. If send buffer 2 is disabled, the under-run flag <PERR> will not be set. This flag is set to "0" when it is read.

3. Framing error <FERR>: Bit 2 of the SC0CR register

In the UART mode, this bit is set to "1" when a framing error is generated. This flag is set to "0" when it is read. A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the <SBLEN> (stop bit length) setting of the serial mode control register 2, SC0MOD2, the stop bit status is determined by only 1 bit on the receive side.

Operation mode	Error flag	Function
UART	OERR	Overrun error flag
	PERR	Parity error flag
	FERR	Framing error flag
I/O interface (SCLK input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O interface (SCLK output)	OERR	Operation undefined
	PERR	Operation undefined
	FERR	Fixed to 0

13.3.15 Direction of Data Transfer

In the I/O interface mode, the direction of data transfer can be switched between "MSB first" and "LSB first" by the data transfer direction setting bit <DRCHG> of the SC0MOD2 serial mode control register 2. Don't switch the direction when data is being transferred.

13.3.16 Stop Bit Length

In the UART mode transmission, the stop bit length can be set to either 1 or 2 bits by bit 4 <SBLEN> of the SC0MOD2 register.

13.3.17 Status Flag

If the double buffer function is enabled (SC0MOD2 <WBUF> = "1"), the bit 6 flag <RBFLL> of the SC0MOD2 register indicates the condition of receive buffer full. When one frame of data has been received and transferred from buffer 1 to buffer 2, this bit is set to "1" to show that buffer 2 is full (data is stored in buffer 2). When the receive buffer is read by CPU/DMAC, it is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag.

When double buffering is enabled (SC0MOD2 <WBUF> = "1"), the bit 7 flag <TBEMP> of the SC0MOD2 register indicates that send buffer 2 is empty. When data is moved from send buffer 2 to send buffer 1 (shift register), this bit is set to "1" indicating that send buffer 2 is now empty. When data is set to the send buffer by CPU/DMAC, the bit is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag.

13.3.18 Configurations of Send/Receive Buffers

		<WBUF> = 0	<WBUF> = 1
UART	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O interface (SCLK input)	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O interface (SCLK output)	Transmit buffer	Single	Double
	Receive buffer	Single	Double

13.3.19 Signal Generation Timing

① UART Mode:

Receive Side

Mode	9-bit	8-bit with parity	8-bit, 7-bit, and 7-bit with parity
Interrupt generation timing	Around the center of the 1st stop bit	Around the center of the 1st stop bit	Around the center of the 1st stop bit
Framing error timing	Around the center of the stop bit	Around the center of the stop bit	Around the center of the stop bit
Parity error generation timing	—	Around the center of the last (parity) bit	Around the center of the last (parity) bit
Overrun error generation timing	Around the center of the stop bit	Around the center of the stop bit	Around the center of the stop bit

Transmit Side

Mode	9-bit	8-bit with parity	8-bit, 7-bit, and 7-bit with parity
Interrupt generation timing (<WBUF> = 0)	Just before the stop bit is sent	Just before the stop bit is sent	Just before the stop bit is sent
Interrupt generation timing (<WBUF> = 1)	Immediately after data is moved to send buffer 1 (just before start bit transmission)	Immediately after data is moved to send buffer 1 (just before start bit transmission)	Immediately after data is moved to send buffer 1 (just before start bit transmission)

② I/O interface mode:

Receive Side

Interrupt generation timing (WBUF = 0)	SCLK output mode	Immediately after the rising edge of the last SCLK
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (WBUF = 1)	SCLK output mode	Immediately after the rising edge of the last SCLK (just after data transfer to receive buffer 2) or just after receive buffer 2 is read
	SCLK input mode	Immediately after the rising edge or falling edge of the last SCLK depending on the rising or falling edge triggering mode, respectively (right after data is moved to receive buffer 2)
Overrun error generation timing	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)

Transmit Side

Interrupt generation timing (WBUF = 0)	SCLK output mode	Immediately after the rising edge of the last SCLK
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (WBUF = 1)	SCLK output mode	Immediately after the rising edge of the last SCLK or just after data is moved to send buffer 1
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for the rising or falling edge mode, respectively) or just after data is moved to send buffer 1
Under-run error generation timing	SCLK input mode	Immediately after the falling or rising edge of the next SCLK (for the rising or falling edge triggering mode, respectively)

Note 1) Do not modify any control register when data is being sent or received (in a state ready to send or receive).
Note 2) Do not stop the receive operation (by setting SC0MOD0 <RXE> = "0") when data is being received.
Note 3) Do not stop the transmit operation (by setting SC0MOD1 <TXE> = "0") when data is being transmitted.

13.4 Register Description (Only for Channel 0)

SC0MOD0 (0xFFFF_F262)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Send data Bit 8	Handshake function control 0: Disables CTS 1: Enables CTS	Receive control 0: Disables reception 1: Enables reception	Wake-up function 0: Disable 1: Enable	Serial transfer mode 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode		Serial transfer clock (for UART) 00: Timer TB4OUT 01: Baud rate generator 10: Internal $f_{SYS}/2$ clock 11: External clock (SCLK0 input)	

Note) In the I/O interface mode, the serial control register (SC0CR) is used for clock selection.

Wakeup function

	9-bit UART	Other mode
0	Interrupt when received	don't care
1	Interrupt at RB8=1	

Handshake function ($\overline{\text{CTS}}$ pin) enable

0	Disable (transmission is always allowed)
1	Enable

Note) With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2). Then set <RXE> to "1."

Fig. 13.4.1 Serial Mode Control Register 0 (for SIO0, SC0MOD0)

SC0MOD1 (0xFFFF_F265)

	7	6	5	4	3	2	1	0
bit Symbol	I2S0	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	IDLE 0: Stop 1: Start	Transfer mode setting 00: Transfer prohibited 01: Half duplex (RX) 10: Half duplex (TX) 11: Full duplex		Transmit control 0: Disable 1: Enable	Interval time of continuous transmission 000: None 100: 8SCLK 001: 1SCLK 101:16SCLK 010: 2SCLK 110: 32SCLK 011: 4SCLK 111: 64SCLK			Write "0."

Fig. 13.4.2 Serial Mode Control Register 1 (for SIO0, SC0MOD1)

<SINT2:0>: Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode. This parameter is invalid for the UART mode or when an external clock is used.

<TXE>: This bit enables transmission and is valid for all the transfer modes. If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<FDPX1:0>: Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<I2S0>: Specifies the Idle mode operation.

SC0MOD2 (0xFFFF_F266)

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R/W						W	W
After reset	1	0	0	0	0	0	0	0
Function	Send buffer empty flag 0: full 1: Empty	Receive buffer full flag 0: Empty 1: full	In transmission flag 0: Stop 1: Start	Stop bit 0: 1-bit 1: 2-bit	Setting transfer direction 0: LSB first 1: MSB first	W-buffer 0: Disable 1: Enable	Soft reset Overwrite "01" on "10" to reset	

<SWRST1:0>: Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the mode register parameters SC0MOD0 <RXE>, SC0MOD1<TXE>, SC0MOD2 <TBEMP>, <RBFLL>, and <TXRUN>, control register parameters SC0CR <OERR>, <PERR>, and <FERR>, and their internal circuits are initialized.

<WBUF>: This parameter enables or disables the send/receive buffers to send (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART. In all other modes, double buffering is enabled regardless of the <WBUF> setting.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, it is fixed to LSB first.

<TXRUN>: This is a status flag to show that data transmission is in progress.
When this bit is set to "1," it indicates that data transmission operation is in progress. If it is "0," the bit 7 <TBEMP> is set to "1" to indicate that the transmission has been fully completed and the same <TBEMP> is set to "0" to indicate that the send buffer contains some data waiting for the next transmission.

<RBFLL>: This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0."
If double buffering is disabled, this flag is insignificant.

<TBEMP>: This flag shows that the send double buffers are empty. When data in the send double buffers is moved to the send shift register and the double buffers are empty, this bit is set to "1." Writing data again to the double buffers sets this bit to "0."
If double buffering is disabled, this flag is insignificant.

<SBLEN>: This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

(Note) While data transmission is in progress, any software reset operation must be executed twice in succession.

Fig. 13.4.3 Serial Mode Control Register

SC0CR (0xFFFF_F261)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to "0" when read)			R/W	
After reset	0	0	0	0	0	0	0	0
Function	Receive data Bit 8	Parity 0: Odd 1: Even	Add parity 0: Disable 1: Enable	0: Normal operation 1: Error			0: SCLK0 ↑ 1: SCLK0 ↓	0: Baud rate generator 1: SCLK0 pin input
				Overrun	Parity/ under-run	Framing		

I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK0 input operation

0	Data send/receive at rising edges of SCLK0 ↑
1	Data send/receive at falling edges of SCLK0 ↓

Framing error flag
Parity error/under-run error flag
Overrun error flag
} Cleared to "0" when read

Add/check even parity

0	Odd parity
1	Even parity

(Note) Any error flag is cleared when read.

Fig. 13.4.4 Serial Control Register (for SIO0, SC0CR)

BR0CR (0xFFFF_F263)

	7	6	5	4	3	2	1	0
bit Symbol		BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Write "0."	N+(16-K)/16 divider function 0: Disable 1: Enable	00: φT1 01: φT4 10: φT16 11: φT64		Divide ratio "N"			

Select input clock to the baud rate generator

00	Internal clock φT1
01	Internal clock φT4
10	Internal clock φT16
11	Internal clock φT64

BR0ADD (0xFFFF_F264)

	7	6	5	4	3	2	1	0
bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function					Specify K for the "N + (16 - K)/16" division			

Setting divide ratio of the baud rate generator

	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0
BR0CR <BR0S3:0> / BR0ADD <BR0K3:0>	0000 (N = 16) ~ 0001 (N = 1)	0010 (N = 2) ~ 1111 (N = 15)	0001 (N = 1) (ONLY UART) ~ 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) ~ 1111 (K = 15)	Disable	$N+\frac{(16-K)}{16}$ Division	Divide by N

(Note 1) In the UART mode, the division ratio "1" of the baud rate generator can be specified only when the "N + (16 - K)/16" division function is not used. In the I/O interface mode, the division ratio "1" of the baud rate generator can be specified only when double buffering is used.

(Note 2) To use the "N + (16 - K)/16" division function, be sure to set BR0CR <BR0ADDE> to "1" after setting the K value (K = 1 to 15) to BR0ADD <BR0K3:0>. However, don't use the "N + (16 - K)/16" division function when BR0CR <BR0S3:0> is set to either "0000" or "0001" (N = 16 or 1).

(Note 3) The "N + (16 - K)/16" division function can only be used in the UART mode. In the I/O interface mode, the "N + (16 - K)/16" division function must be disabled (prohibited) by setting BR0CR <BR0ADDE> to "0."

Fig. 13.4.5 Baud Rate Generator Control (for SIO0, BR0CR, BR0ADD)

SC0BUF (0xFFFF_F260)

	7	6	5	4	3	2	1	0
bit Symbol	TB7/RB7	TB6/RB6	TB5/RB5	TB4/RB4	TB3/RB3	TB2/RB2	TB1/RB1	TB0/RB0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	TB7 to TB0: Send buffer + FIFO RB7 to RB0: Receive buffer + FIFO							

Note: HSCBUF works as a send buffer for WR operation and as a receive buffer for RD operation.

Fig. 13.4.6 SIO0 Send/Receive Buffer Register

SC0FCNF (0xFFFF_F26C)

	7	6	5	4	3	2	1	0
bit Symbol				RFST	TFIE	RFIE	RXTXCNT	CNFG
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	**Be sure to write "000."**			Bytes used in RX FIFO 0: Maximum 1: Same as Fill level of RX FIFO	TX interrupt for TX FIFO 0: Disable 1: Enable	RX interrupt for RX FIFO 0: Disable 1: Enable	Automatic disable of RXE/TXE 0: None 1: Auto Disable	FIFO Enable 0: Disable 1: Enable

<CNFG>: If enabled, the SCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows:

<FDPX1:0> = 01 (Half duplex RX) ---- 4-byte RX FIFO

<FDPX1:0> = 10 (Half duplex TX) ---- 4-byte TX FIFO

<FDPX1:0> = 11 (Full duplex) --------- 2-Byte RX FIFO + 2-Byte TX FIFO

<RXTXCNT>:0 The function to automatically disable RXE/TXE bits is disabled.

1: If enabled, the SCOMOD1 <FDPX1:0> is used to set as follows:

<FDPX1:0> = 01 (Half duplex RX) ------ When the RX FIFO is filled up to the specified number of valid bytes, RXE is automatically set to "0" to inhibit further reception.

<FDPX1:0> = 10 (Half duplex TX) ------ When the TX FIFO is empty, TXE is automatically set to "0" to inhibit further transmission.

<FDPX1:0> = 11 (Full duplex) ---------- When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<RFIE>: When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<TFIE>: When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFST>: When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected.

0: The maximum number of bytes of the FIFO configured 4 bytes when <FDPX1:0> = 01 (Half duplex RX) and 2 bytes for <FDPX1:0> = 11 (Full duplex)

1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL5:0>.

(Note 1) Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.

Fig. 13.4.7 FIFO Configuration Register

SC0RFC (0xFFFF_F268)

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS					RIL1	RIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0	0	0	0	0	0
Function	Clear RX FIFO 1: Clear Always reads "0."	Select interrupt generation condition					FIFO fill level to generate RX interrupts 00: 4 bytes (2 bytes if full duplex) 01: 1byte 10: 2byte 11: 3byte Note: RIL1 is ignored when FDPX1:0 = 11 (full duplex)	

0: An interrupt is generated when the specified fill level is reached.
1: An interrupt is generated when the specified fill level is reached or if the specified fill level has been exceeded at the time data is read.

Fig. 13.4.8 Receive FIFO Control Register

Transmit FIFO Configuration Register

SC0TFC (0xFFFF_F269)

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS					TIL1	TIL0
Read/Write	w	R/W	R				R/W	
After reset	0	0	0	0	0	0	0	0
Function	Clear TX FIFO 1: Clear Always reads "0."	Select interrupt generation condition					FIFO fill level to generate TX interrupts 00: Empty 01: 1byte 10: 2byte 11: 3byte Note: TIL1 is ignored when FDPX1:0 = 11 (full duplex).	

0: An interrupt is generated when the specified fill level is reached.
1: An interrupt is generated when the specified fill level is reached or if the level is lower than the specified fill level at the time new data is written.

Fig. 13.4.9 Transmit FIFO Configuration Register

SC0RST (0xFFFF_F26A)

	7	6	5	4	3	2	1	0
bit Symbol	ROR					RLVL2	RLVL1	RLVL0
Read/Write	R	R				R		
After reset	0	0	0	0	0	0	0	0
Function	RX FIFO Overrun 1: Generated					Status of RX FIFO fill level 000: Empty 001: 1Byte 010: 2Byte 011: 3Byte 100: 4Byte		

(Note) The <ROR> bit is cleared to "0" when receive data is read from the SC0BUF register.

Fig. 13.4.10 Receive FIFO Status Register

SC0TST (0xFFFF_F26B)

	7	6	5	4	3	2	1	0
bit Symbol	TUR					TLVL2	TLVL1	TLVL0
Read/Write	R	R				R		
After reset	1	0	0	0	0	0	0	0
Function	TX FIFO Under run 1: Generated					Status of TX FIFO fill level 000: Empty 001: 1Byte 010: 2Byte 011: 3Byte 100: 4Byte		

(Note) The <TUR> bit is cleared to "0" when transmit data is written to the SC0BUF register.

Fig. 13.4.11 Transmit FIFO Status Register

SC0EN (0xFFFF_F267)

	7	6	5	4	3	2	1	0
bit Symbol								SIOE
Read/Write	R							R/W
After reset	0	0	0	0	0	0	0	0
Function								SIO operation 0: Disable 1: Enable

<SIOE>: It specifies SIO operation. When SIO operation is disabled, the clock will not be supplied to the SIO module except for the register part and thus power dissipation can be reduced (other registers cannot be accessed for read/write operation). When SIO is to be used, be sure to enable SIO by setting "1" to this register before setting any other registers of the SIO module. If SIO is enabled once and then disabled, any register setting is maintained.

Fig. 13.4.12 SIO Enable Register

13.5 Operation in Each Mode

13.5.1 Mode 0 (I/O Interface Mode)

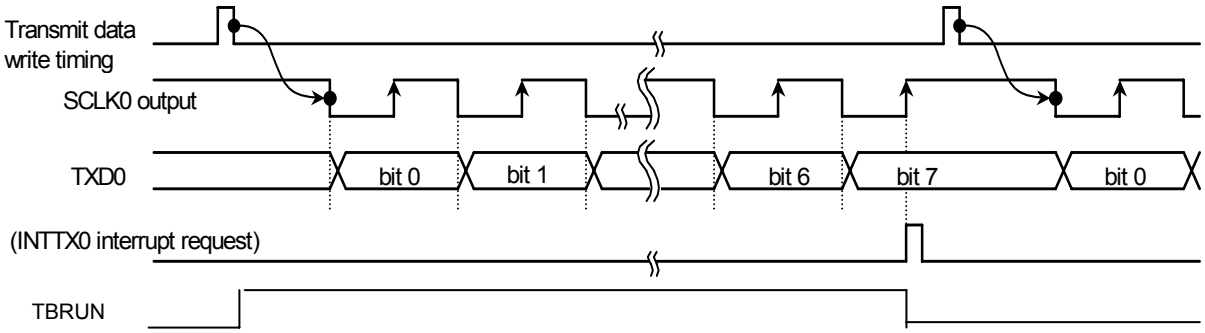
Mode 0 consists of two modes, i.e., the "SCLK output" mode to output synchronous clock and the "SCLK input" mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

① Sending data

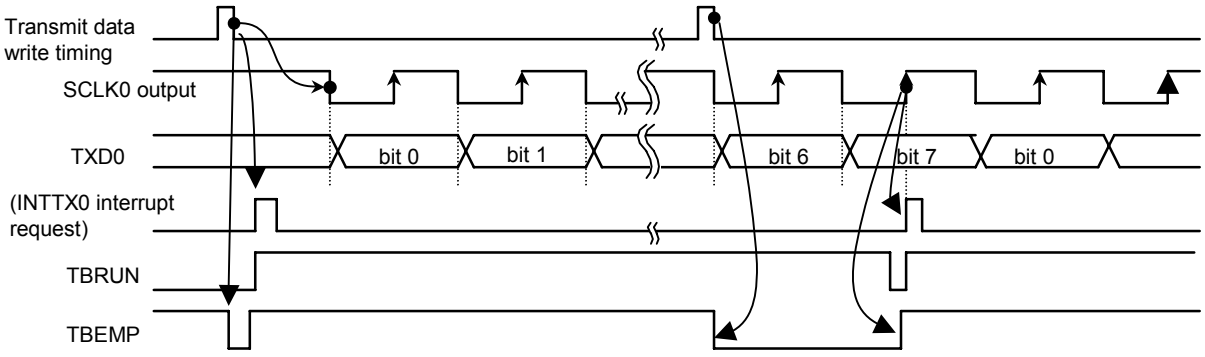
SCLK output mode

In the SCLK output mode, if SC0MOD2<WBUF> is set to "0" and the send double buffers are disabled, 8 bits of data are output from the TXD0 pin and the synchronous clock is output from the SCLK0 pin each time the CPU writes data to the send buffer. When all data is output, the INTTX0 interrupt is generated.

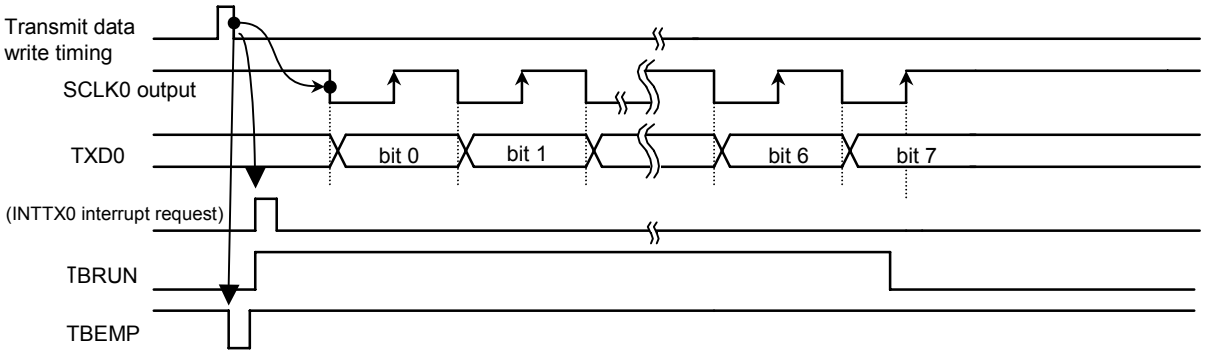
If SC0MOD2 <WBUF> is set to "1" and the send double buffers are enabled, data is moved from send buffer 2 to send buffer 1 when the CPU writes data to send buffer 2 while data transmission is halted or when data transmission from send buffer 1 (shift register) is completed. When data is moved from send buffer 2 to send buffer 1, the send buffer empty flag SC0MOD2 <TBEMP> is set to "1," and the INTTX0 interrupt is generated. If send buffer 2 has no data to be moved to send buffer 1, the INTTX0 interrupt is not generated and the SCLK0 output stops.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



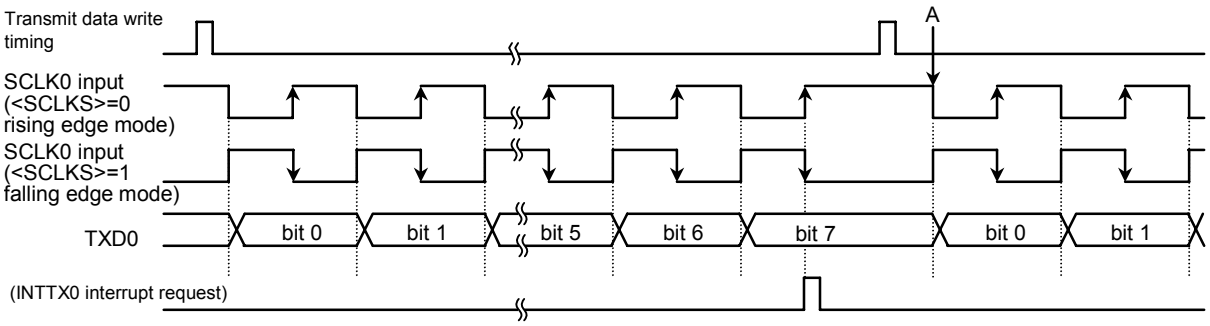
<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 13.5.1.11 Send Operation in the I/O Interface Mode (SCLK0 Output Mode)

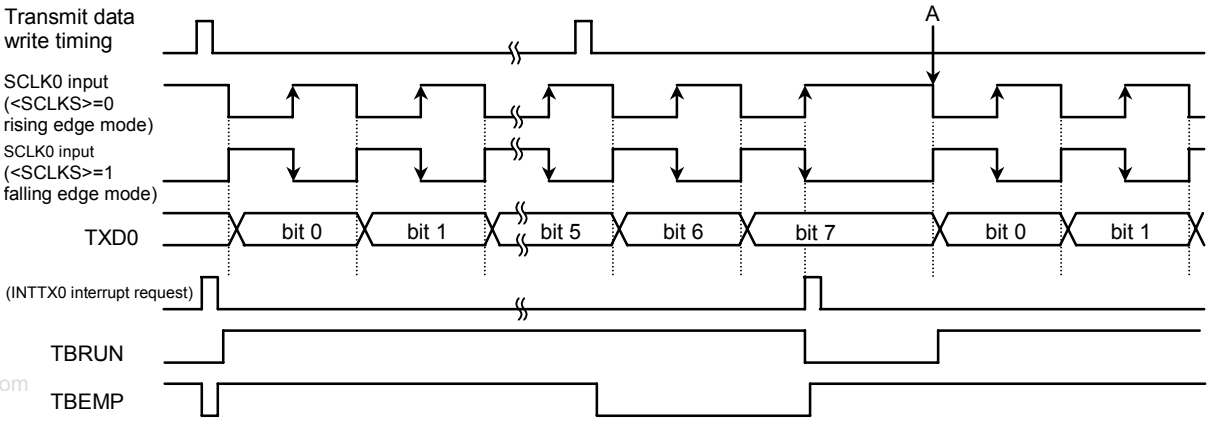
SCLK input mode

In the SCLK input mode, if SC0MOD2 <WBUF> is set to "0" and the send double buffers are disabled, 8-bit data that has been written in the send buffer is output from the TXD0 pin when the SCLK0 input becomes active. When all 8 bits are sent, the INTTX0 interrupt is generated. The next send data must be written before the timing point "A" as shown in Fig. 13.5.1.2.

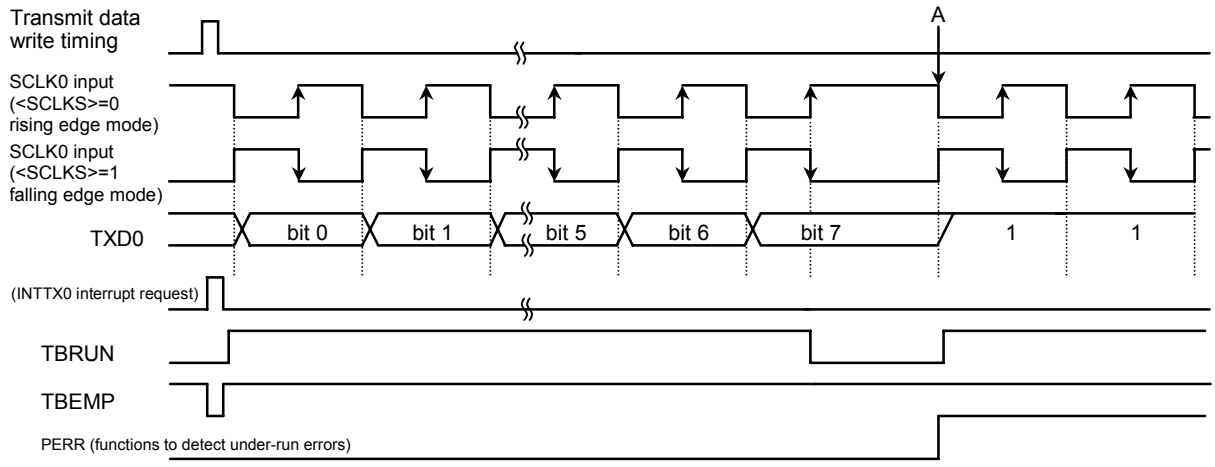
If SC0MOD2 <WBUF> is set to "1" and the send double buffers are enabled, data is moved from send buffer 2 to send buffer 1 when the CPU writes data to send buffer 2 before the SCLK0 becomes active or when data transmission from send buffer 1 (shift register) is completed. As data is moved from send buffer 2 to send buffer 1, the send buffer empty flag SC0MOD2 <TBEMP> is set to "1" and the INTTX0 interrupt is generated. If the SCLK0 input becomes active while no data is in send buffer 2, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is sent.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 13.5.1.2 Send Operation in the I/O Interface Mode (SCLK0 Input Mode)

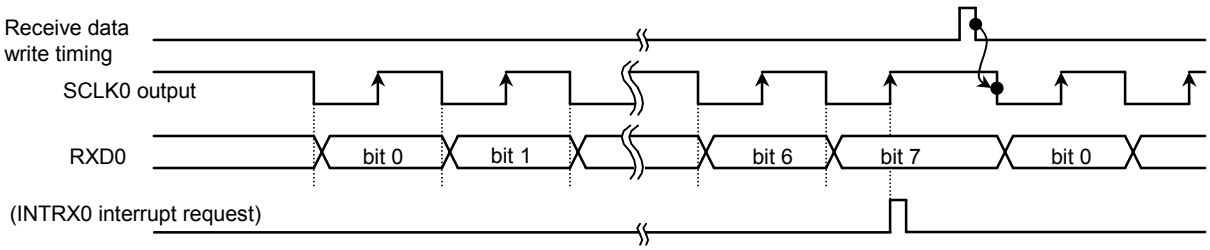
② Receiving data

SCLK output mode

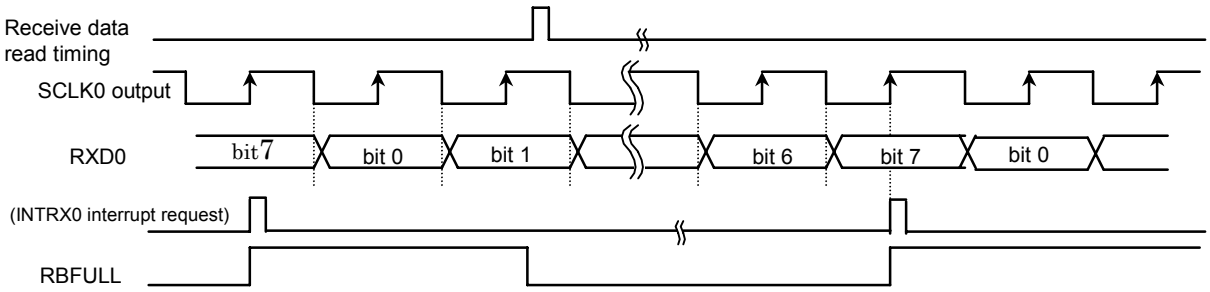
In the SCLK output mode, if SC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

The first SCLK output can be started by setting the receive enable bit SC0MOD0 <RXE> to "1." If the receive double buffering is enabled with SC0MOD2 <WBUF> set to "1," the first frame received is moved to receive buffer 2 and receive buffer 1 can receive the next frame successively. As data is moved from receive buffer 1 to receive buffer 2, the receive buffer full flag SC0MOD2 <RBFULL> is set to "1" and the INTRX0 interrupt is generated.

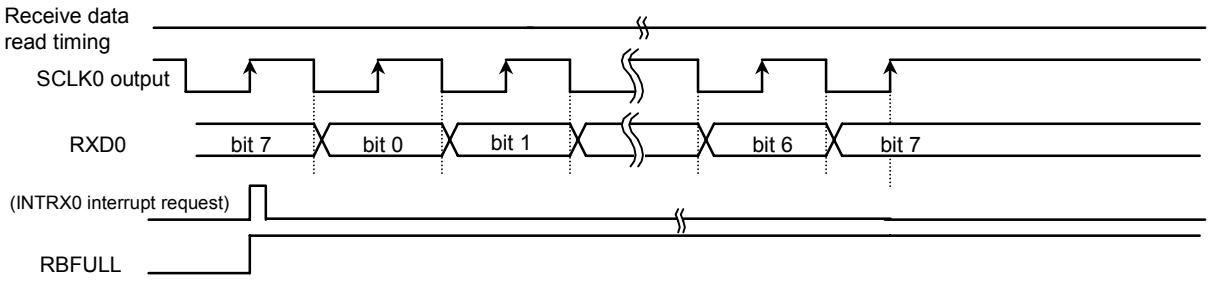
While data is in receive buffer 2, if CPU/DMAC cannot read data from receive buffer 2 in time before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 2 and thus the INTRX0 interrupt is generated and data reception resumes.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if data is read from buffer 2)



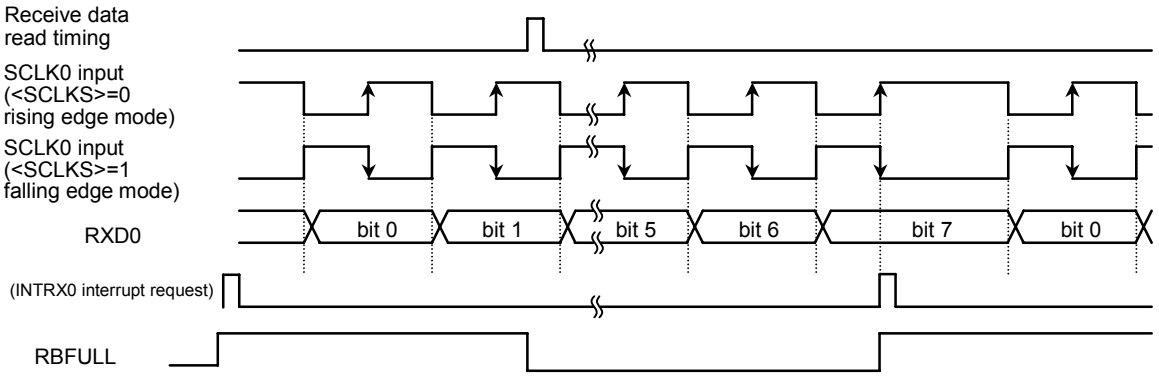
<WBUF> = "1" (if double buffering is enabled) (if data cannot be read from buffer 2)

Fig. 13.5.1.3 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

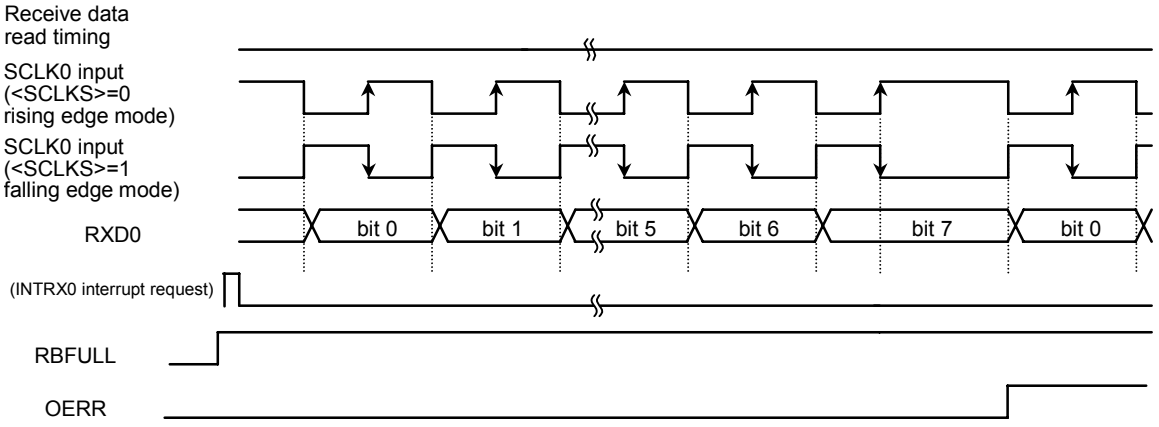
SCLK input mode

In the SCLK input mode, since receive double buffering is always enabled, the received frame can be moved to receive buffer 2 and receive buffer 1 can receive the next frame successively.

The INTRX receive interrupt is generated each time received data is moved to received buffer 2.



If data is read from buffer 2



If data cannot be read from buffer 2

Fig. 13.5.1.4 Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

(Note) To receive data, SC0MOD <RXE> must always be set to "1" (receive enable) regardless of the SCLK input or output mode.

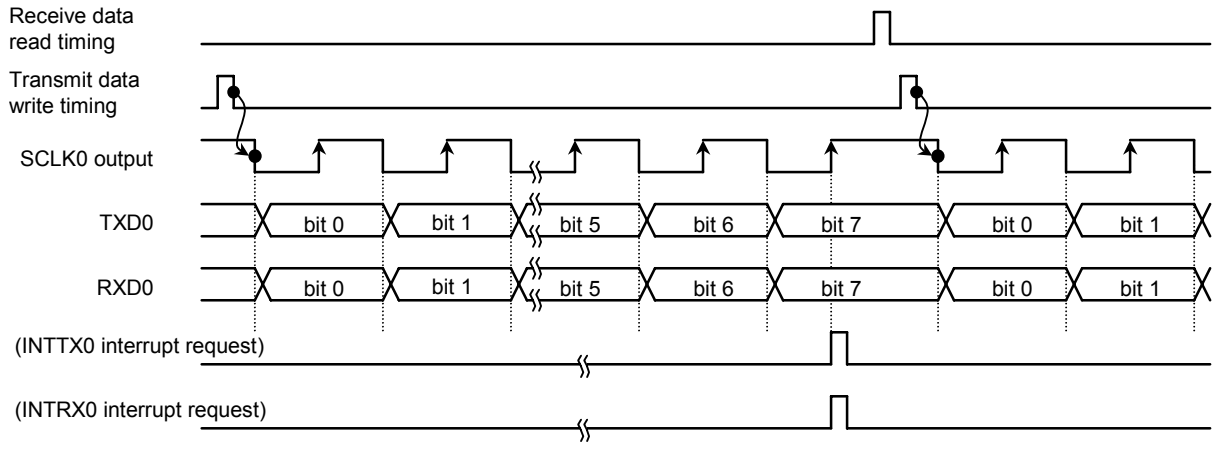
③ Send and receive (full-duplex)

The full-duplex mode is enabled by setting bit 6 <FDPX0> of the serial mode control register 1 (SC0MOD1) to "1."

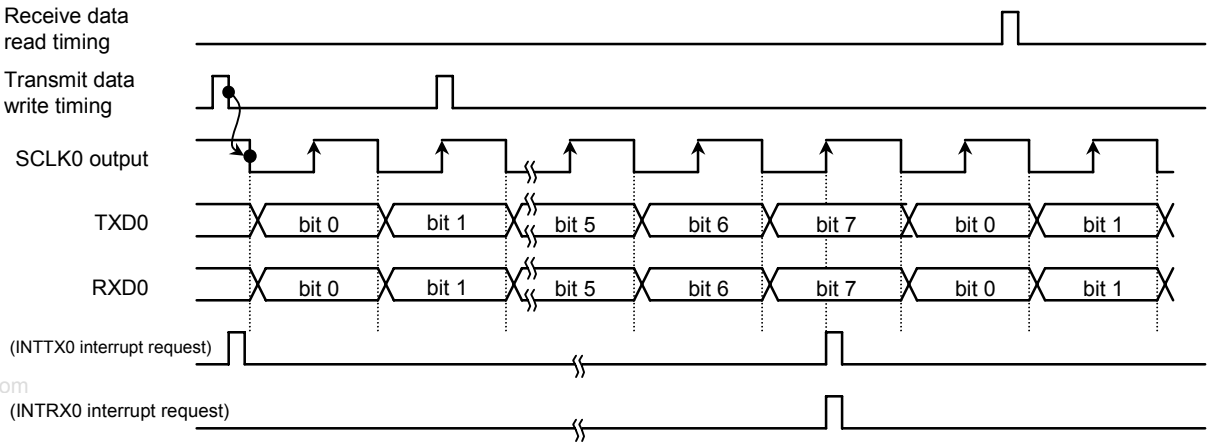
SCLK output mode

In the SCLK output mode, if SC0MOD2 <WBUF> is set to "0" and both the send and receive double buffers are disabled, SCLK is output when the CPU writes data to the send buffer. Subsequently, 8 bits of data are shifted into receive buffer 1 and the INTRX0 receive interrupt is generated. Concurrently, 8 bits of data written to the send buffer are output from the TXD0 pin, the INTTX0 send interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops. In this, the next round of data transmission and reception starts when the data is read from the receive buffer and the next send data is written to the send buffer by the CPU. The order of reading the receive buffer and writing to the send buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

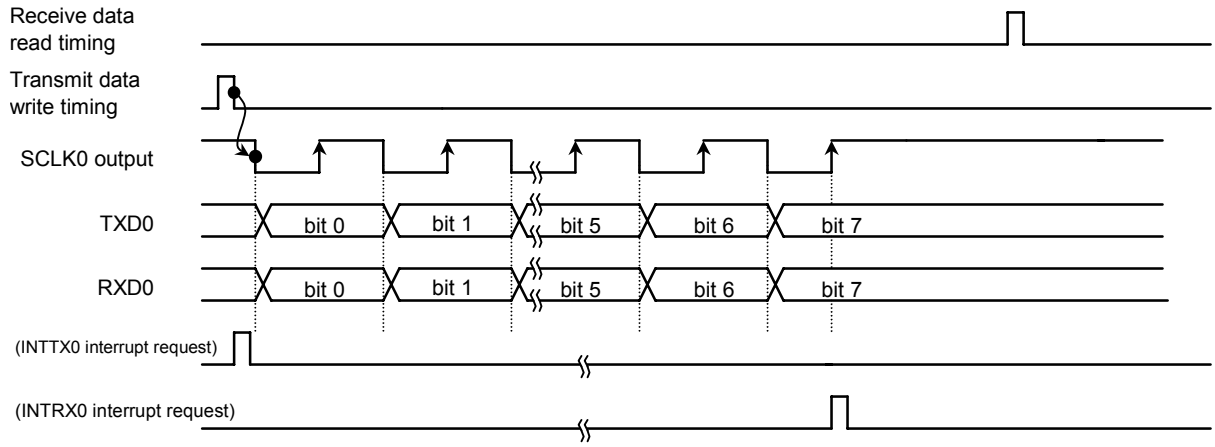
If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, SCLK is output when the CPU writes data to the send buffer. Subsequently, 8 bits of data are shifted into receive buffer 1, moved to receive buffer 2, and the INTRX0 interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is output from the TXD0 pin. When all data bits are sent out, the INTTX0 interrupt is generated and the next data is moved from the send buffer 2 to send buffer 1. If send buffer 2 has no data to be moved to send buffer 1 (SC0MOD2 <TBEMP> = 1) or when receive buffer 2 is full (SC0MOD2 <RBFULL> = 1), the SCLK clock is stopped. When both conditions are satisfied, i.e., receive data is read and send data is written, the SCLK output is resumed and the next round of data transmission is started.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled)



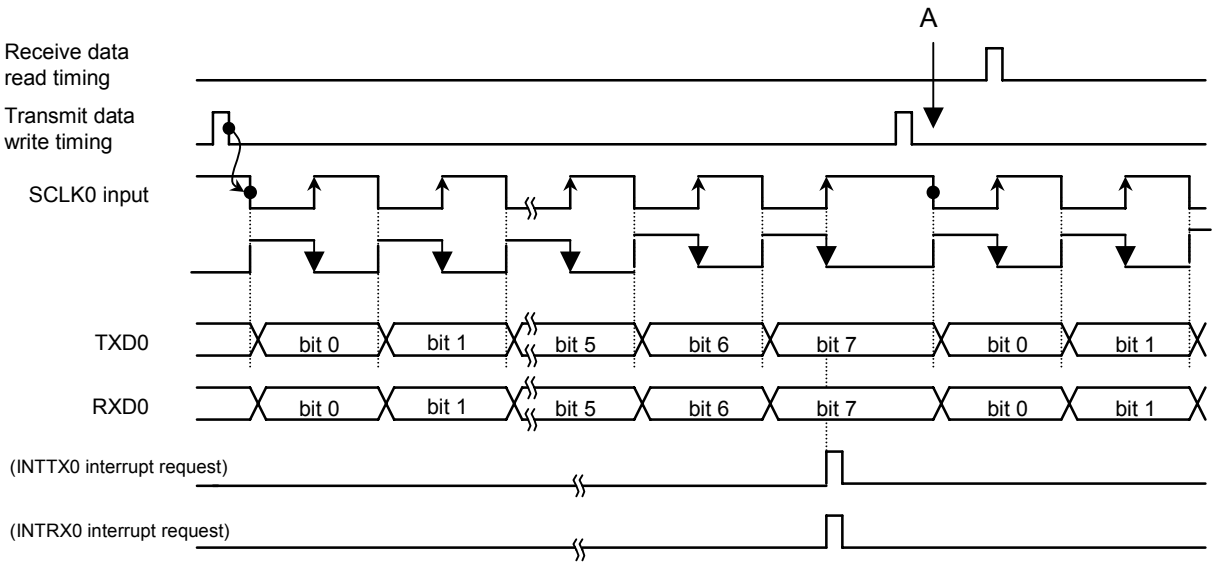
<WBUF> = "1" (if double buffering is enabled)

Fig. 13.5.1.5 Send/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

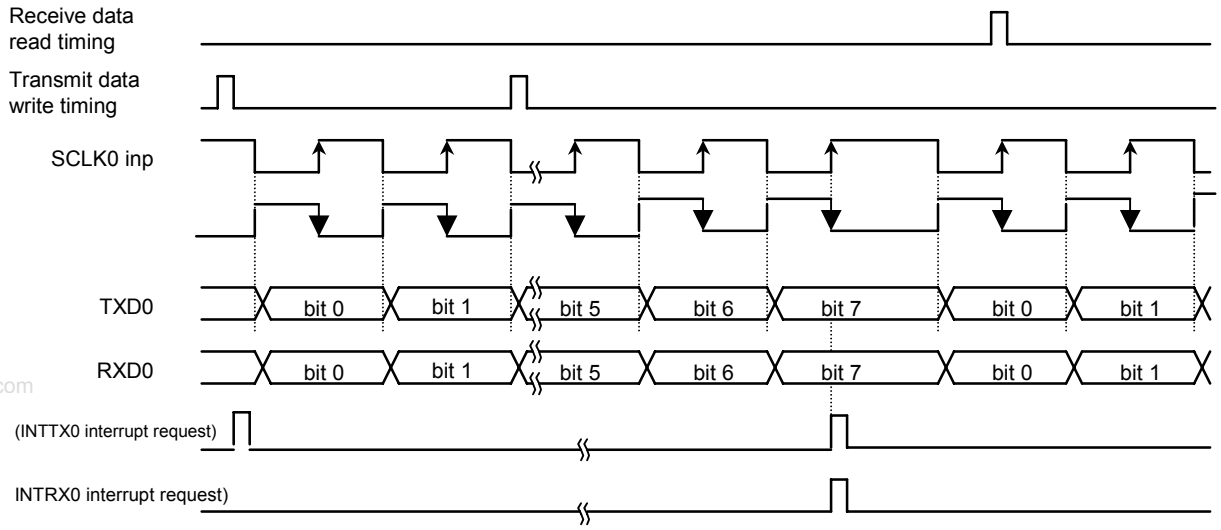
SCLK input mode

In the SCLK input mode with SC0MOD2 <WBUF> set to "0" and the send double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the send buffer is output from the TXD0 pin and 8 bits of data is shifted into the receive buffer when the SCLK0 input becomes active. The INTTX0 interrupt is generated upon completion of data transmission and the INTRX0 interrupt is generated at the instant the received data is moved from receive buffer 1 to receive buffer 2. Note that transmit data must be written into the send buffer before the SCLK input for the next frame (data must be written before the point A in Fig. 13.5.1.6). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

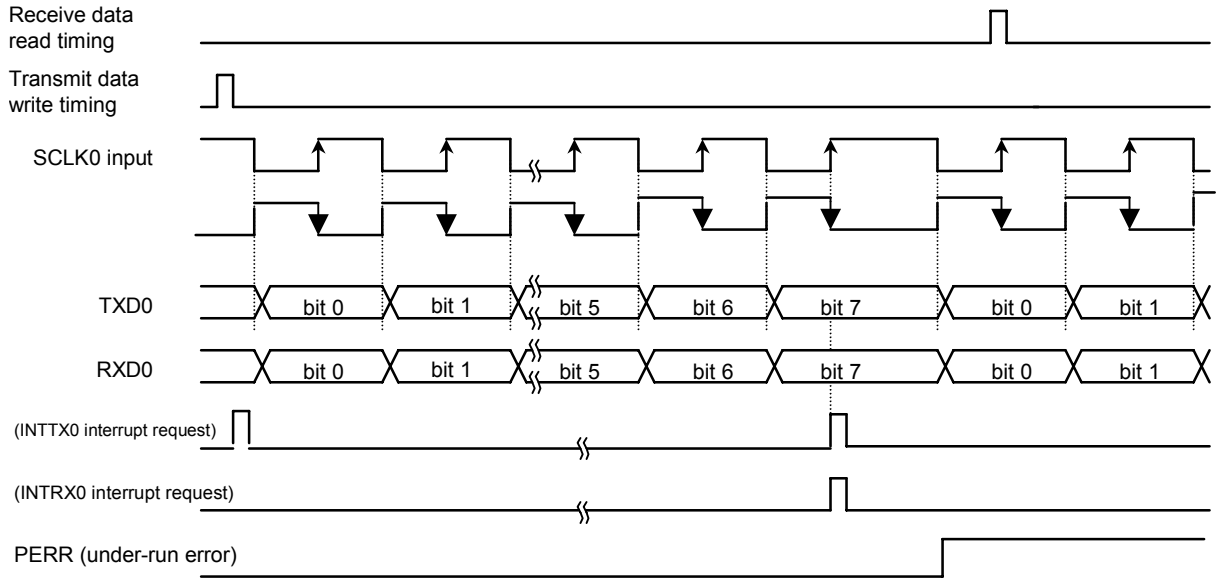
If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing send buffer 2 data is moved to send buffer 1 after completing data transmission from send buffer 1. At the same time, the 8 bits of data received is shifted to buffer 1, moved to receive buffer 2, and the INTRX0 interrupt is generated. Upon the SCLK input for the next frame, transmission from send buffer 1 (in which data has been moved from send buffer 2) is started while receive data is shifted into receive buffer 1 simultaneously. If data in receive buffer 2 has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to send buffer 2 when SCLK for the next frame is input, an under-run error occurs.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (no errors)



<WBUF> = "1" (if double buffering is enabled) (error generation)

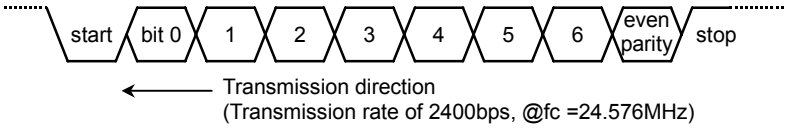
Fig. 13.5.1.6 Send/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

13.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SC0MOD <SM1, 0>) to "01."

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SC0CR <EVEN> bit. The length of the stop bit can be specified using SC0MOD2<SBLEN>.

Example: The control register settings for transmitting in the following data format are listed in the following table.



Transmission direction
(Transmission rate of 2400bps, @fc =24.576MHz)

* Clocking conditions

System clock	: High-speed (fc)
High-speed clock gear	: x 1 (fc)
Prescaler clock	: fperiph/4 (fperiph = fsys)

	7	6	5	4	3	2	1	0	
PCCR	←	_	_	_	_	_	_	_	1
PCFC	←	_	_	_	_	_	_	_	1
SC0MOD	←	X	0	_	X	0	1	0	1
SC0CR	←	X	1	1	X	X	X	0	0
BR0CR	←	0	0	1	0	1	0	1	0
IMC3	←	_	1	1	_	0	1	0	0
SC0BUF	←	*	*	*	*	*	*	*	*

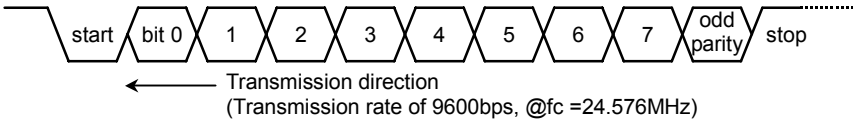
PCCR, PCFC: Designates PC0 as the TXD0 pin.
SC0MOD: Sets the 7-bit UART mode.
SC0CR: Adds even parity.
BR0CR: Sets the data rate to 2400 bps.
IMC3: Enables the INTTX0 interrupt and sets to level 4 by the <31:24> bits of the 32 bit register.
SC0BUF: Sets the data to be sent.

Note: X: don't care - : no change

13.5.3 Mode 2 (8-bit UART Mode)

The 8-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SC0CR <PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SC0CR <EVEN>.

Example: The control register settings for receiving data in the following format are as follows:



Transmission direction
(Transmission rate of 9600bps, @fc =24.576MHz)

* Clocking conditions

System clock	: High-speed (fc)
High-speed clock gear	: x 1 (fc)
Prescaler clock	: fperiph/4 (fperiph = fsy_s)

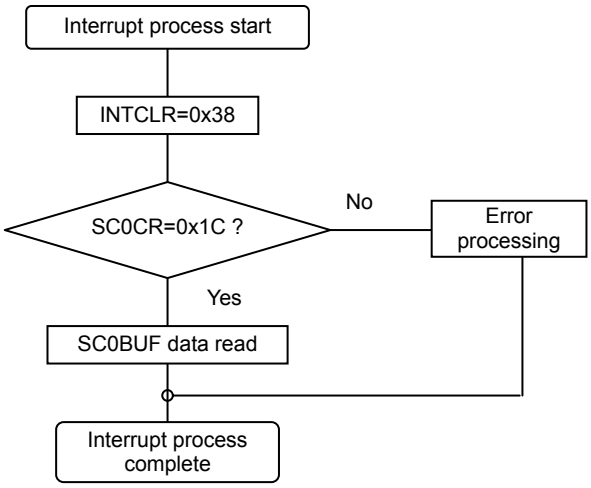
Main routine settings

		7	6	5	4	3	2	1	0	
PCCR	←	_	_	_	_	_	_	0	_	Designates PC1 as the RXD0 pin.
PCFC	←	_	_	_	_	_	_	1	_	
SC0MOD	←	_	0	0	X	1	0	0	1	Selects the 8-bit UART mode.
SC0CR	←	X	0	1	X	X	X	0	0	Sets odd parity.
BR0CR	←	0	0	0	1	0	1	0	1	Sets the data rate to 9600 bps.
IMC3	←	_	1	1	_	0	1	0	0	Enables the INTRX0 interrupt and sets to level 4 by the <23:16> bits of the 32 bit register.
SC0MOD	←	_	_	1	X	_	_	_	_	Enables reception of data.

An example interrupt routine process

INTCLR 0 0 0 1 1 1 0 0 0	Clears the interrupt request. 0x0000_0038
Reg. ← SC0CR AND 0x1C	Performs error check
if reg. is not "0" then error processing	
Set SC0BUF to Reg.	Reads received data.
Interrupt processing is completed	

Note: X: don't care - : no change



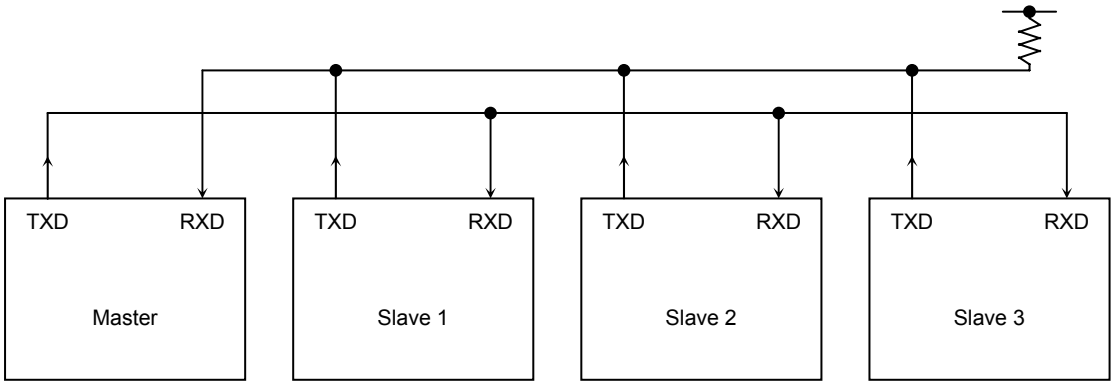
13.5.4 Mode 3 (9-bit UART)

The 9-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "11." In this mode, parity bits must be disabled (SC0CR <PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SC0MOD0) for transmit data and it is stored in bit 7 <RB8> of the serial control register SC0CR upon receiving data. When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SC0BUF. The stop bit length can be specified using SC0MOD2 <SBLEN>.

Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1."



(Note) The TXD pin of the slave controller must be set to the open drain output mode using the ODE register.

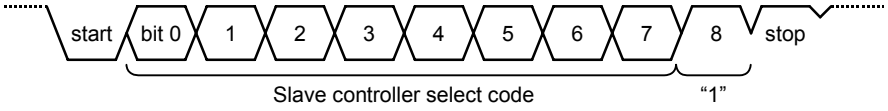
Fig. 13.5.4.1 Serial Links to Use Wake-up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

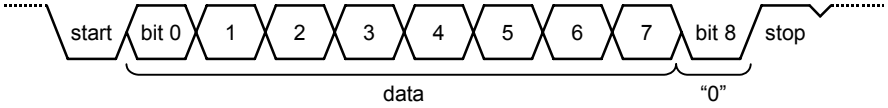
② Set SC0MOD <WU> to "1" for the slave controllers to make them ready to receive data.

③ The master controller is to send a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1."



④ Every slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0."

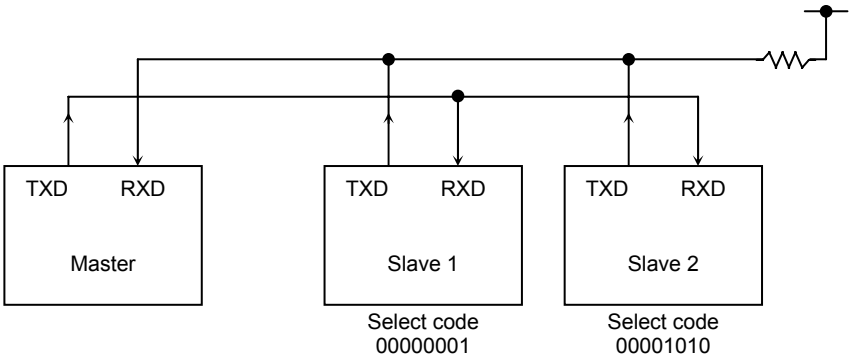
⑤ The master controller transmits data to the designated slave controller (the controller of which SC0MOD <WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0."



⑥ The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRX0) is generated.

Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

Example setting: Using the internal clock fsys/2 as the transfer clock, two slave controllers are serially linked as follows:



① Master controller setting

Main routine

PCCR	← _ _ _ _ _ _ 0 1	Designates PC0/PC1 as the TXD0/RXD0 pins, respectively.
PCFC	← _ _ _ _ _ _ 1 1	
IMC3	← _ 1 1 _ 0 1 0 1	Enables the INTRX0 interrupt and sets to level 5 by the <23:16> bits of the 32 bit register.
	← _ 1 1 _ 0 1 0 0	Enables the INTTX0 interrupt and sets to level 4 by the <31:24> bits of the 32 bit register.
SC0MOD0	← 1 0 1 0 1 1 1 0	Sets the 9-bit UART mode and fsys/2 transfer clock.
SC0BUF	← 0 0 0 0 0 0 0 1	Sets the select code of Slave 1.

Interrupt routine (INTTX0)

INTCLR	0 0 0 1 1 1 1 0 0	Clears the interrupt request. (0x0000_003C)
SC0MOD0	← 0 _ _ _ _ _ _ _	Sets TB8 to "0."
SC0BUF	← * * * * * * * *	Sets the data to be sent.

Interrupt processing is completed.

② Slave controller setting

Main routine

PCCR	← _ _ _ _ _ _ 0 1	Designates PC0 as TXD (open drain output) and PC1 as RXD.
PCFC	← _ _ _ _ _ _ 1 1	
PCODE	← _ _ _ _ _ _ _ 1	
IMC3	← _ 1 1 _ 0 1 1 0	Enables INTTX0 and INTRX0.
	← _ 1 1 _ 0 1 0 1	
SC0MOD0	← 0 0 1 1 1 1 1 0	Sets the 9-bit UART mode and f_{SYS}/2 transfer clock and sets <WU> to "1."

Interrupt routine (INTRX0)

INTCLR	0 0 0 1 1 1 0 0 0	Clears the interrupt request.
Reg.	← SC0BUF	
if Reg. = select code, Then		
SC0MOD0	← _ _ _ 0 _ _ _ _	Clears <WU> to "0."

14. Serial Bus Interface (SBI)

The TMP19A64 contains a Serial Bus Interface (SBI) channel, which has the following two operating modes:

- I^2C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I^2C bus mode, the SBI is connected to external devices via PF0 (SDA) and PF1 (SCL). In the clock-synchronous 8-bit SIO mode, the SBI is connected to external devices via PF2 (SCK), PF0 (SO) and PF1 (SI).

The following table shows the programming required to put the SBI in each operating mode.

	PFODE <PFODE1:0>	PFCR <PF2C, PF1C, PF0C>	PFFC <PF2F, PF1F, PF0F>
I2C bus mode	11	X11	011
Clock-synchronous 8-bit SIO mode	XX	101 (clock output) 001 (clock input)	111

X: Don't care

14.1 Configuration

The configuration is shown in Fig. 14.1.

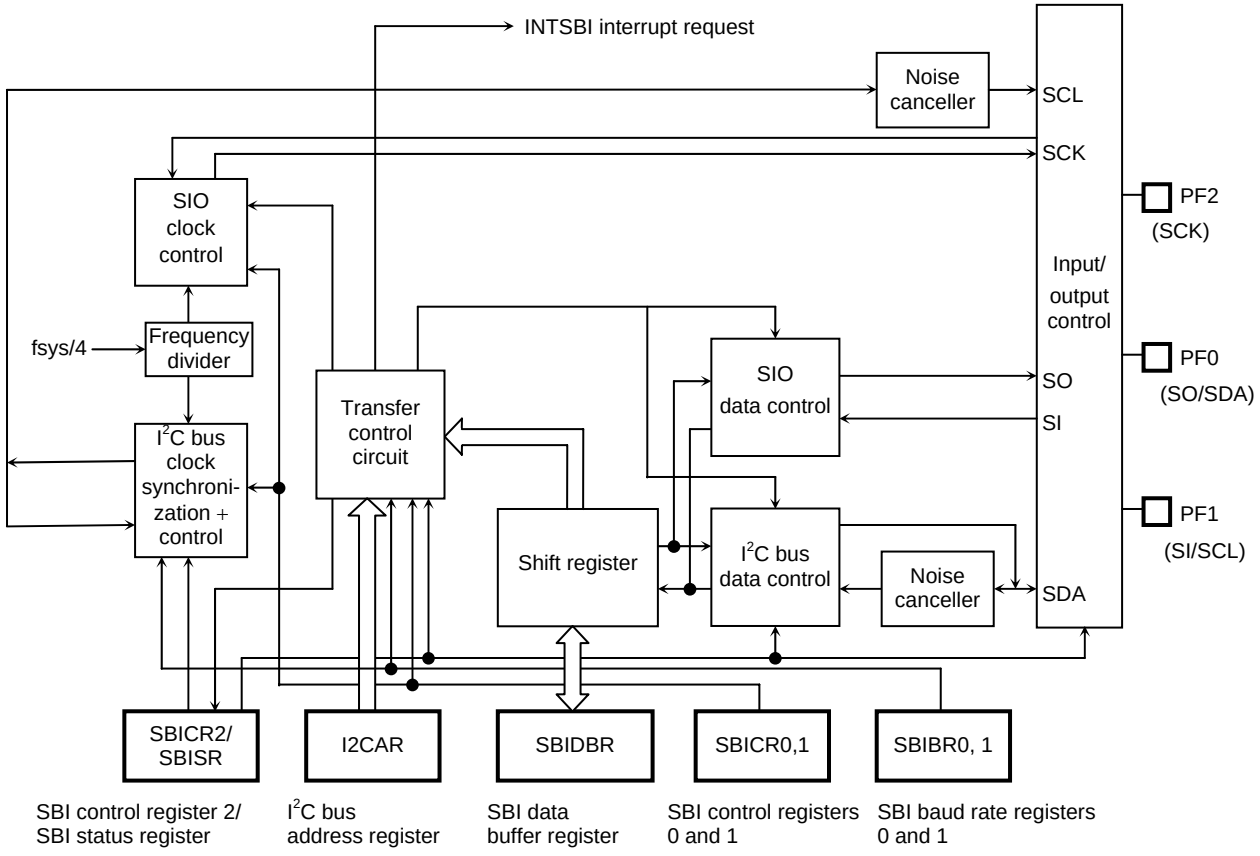


Fig. 14.1 SBI Block Diagram

14.2 Control

The following registers control the serial bus interface and provide its status information for monitoring.

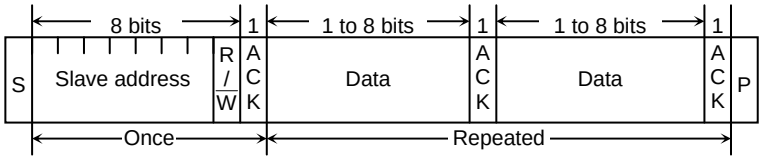
- Serial bus interface control register 0 (SBICR0)
- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register (SBISR)
- Serial bus interface baud rate register 0 (SBIBR0)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to "14.5 Control in the I^2C Bus Mode" and "14.7 Control in the Clock-synchronous 8-bit SIO Mode."

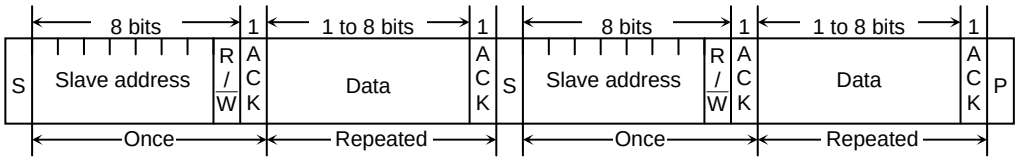
14.3 I^2C Bus Mode Data Formats

Fig. 14.3 shows the data formats used in the I^2C bus mode.

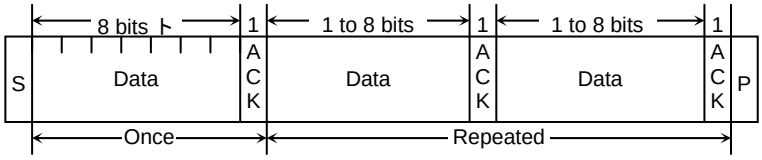
(a) Addressing format



(b) Addressing format (with repeated start condition)



(c) Free data format (master-transmitter to slave-receiver)



Note: S: Start condition
R / W : Direction bit
ACK: Acknowledge bit
P: Stop condition

Fig. 14.3 I^2C Bus Mode Data Formats

14.4 Control Registers in the I²C Bus Mode

The following registers control the serial bus interface (SBI) in the I²C bus mode and provide its status information for monitoring.

Serial bus interface control register 0

SBICR0 (0xFFFF_F257)

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0	0	0	0	0	0	0
Function	SBI operation 0: Disable 1: Enable							

<SBIEN>: To use the SBI, enable the SBI operation ("1") before setting each register in the SBI module.

(Note) Bits 0 to 6 of SBICRO are read as "0."

Fig. 14.4.1 I²C Bus Mode Register

Serial bus interface control register 1

SBICR1 (0xFFFF_F250)

	7	6	5	4	3	2	1	0
bit Symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	R/W			R/W	R	R/W		R/W
After reset	0	0	0	0	1	0	0	1
Function	Select the number of bits per transfer (Note 1)			Acknow-ledgment clock 0: Not generate 1: Generate		Select internal SCL output clock frequency (Note 2) and reset monitor		

On writing <SCK2:0>: Select internal SCL output clock frequency

000	n=5	265 kHz
001	n=6	201 kHz
010	n=7	136 kHz
011	n=8	83 kHz
100	n=9	46 kHz
101	n=10	25 kHz
110	n=11	13 kHz
111		reserved

System clock : fsys (=54 MHz)

Clock gear : fc/1

$$\text{Frequency} = \frac{\text{fsys}/2}{2^n + 70} \text{ [Hz]}$$

On reading <SWRMON>: Software reset status monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

Select the number of bits per transfer

<BC2:0>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

(Note 1) Clear <BC2:0> to "000" before switching the operation mode to the clock-synchronous 8-bit SIO mode.

(Note 2) For details on the SCL line clock frequency, refer to "14.5.3 Serial Clock."

(Note 3) After a reset, the <SCK0/SWRMON> bit is read as "1." However, if the SIO mode is selected at the SBICR2 register, the initial value of the <SCK0> bit is "0."

Fig. 14.4.2 I^2C Bus Mode Register

Serial bus interface control register 2

SBICR2 (0xFFFF_F253)

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W		W	
After reset	0	0	0	1	0	0	0	0
Function	Select master/slave 0: Slave 1: Master	Select transmit/ receive 0: Receive 1: Transmit	Start/stop condition generation 0: Stop condition generated 1: Start condition generated	Clear INTSBI interrupt request 0: – 1: Clear interrupt request	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Software reset generation Write "10" followed by "01" to generate a reset.	

Select serial bus interface operating mode (Note 2)

00	Port mode (Serial bus interface output disabled)
01	Clock-synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

(Note 1) Reading this register causes it to function as the SBISR register.

(Note 2) Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "H" level before switching the operating mode from the port mode to the I^2C bus or clock-synchronous 8-bit SIO mode.

Fig. 14.4.3 I^2C Bus Mode Register

Table 14.4.4 Base Clock Resolution

@fsys = 54 MHz

Clock gear value <GEAR2:0>	Base clock resolution
000 (fc)	fsys/2^2 (0.07 μs)
100 (fc/2)	fsys/2^3 (0.14 μs)
110 (fc/4)	fsys/2^4 (0.28 μs)
111 (fc/8)	fsys/2^5 (0.58 μs)

Serial bus interface status register

SBISR (0xFFFF_F253)

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/slave selection monitor 0: Slave 1: Master	Transmit/ receive selection monitor 0: Receive 1: Transmit	I^2C bus state monitor 0: Free 1: Busy	INTSBI interrupt request monitor 0: Interrupt request generated 1: Interrupt request cleared	Arbitration lost detection 0: – 1: Detected	Slave address match detection 0: – 1: Detected	General call detection 0: – 1: Detected	Last received bit monitor 0: "0" 1: "1"

Last received bit monitor

0	The last bit received was "0."
1	The last bit received was "1."

Addressed as slave

0	–
1	Addressed as slave or general call detected

Arbitration lost

0	–
1	Arbitration was lost to another master

(Note) Writing to this register causes it to function as SBICR2.

Fig. 14.4.5 I^2C Bus Mode Register

Serial bus interface baud rate register0

SBIBR0 (0xFFFF_F254)

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI						
Read/Write	R	R/W	R					R/W
After reset	1	0	1	1	1	1	1	0
Function		IDLE 0: Stop 1: Operate						Make sure that you write "0."

Operation in the IDLE mode

0	Stop
1	Operate

Serial bus interface data buffer register

SBIDBR (0xFFFF_F251)

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	0							

(Note) Transmit data must be written to this register, with bit 7 being the most-significant bit (MSB).

I^2C bus address register

I2CAR (0xFFFF_F252)

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set the slave address when the SBI acts as a slave device.							Specify address recognition mode

Specify address recognition mode

0	Recognizes the slave address.
1	Does not recognize slave address.

Fig. 14.4.6 I^2C Bus Mode Register

14.5 Control in the I²C Bus Mode

14.5.1 Setting the Acknowledgement Mode

Setting SBICR1<ACK> to "1" selects the acknowledge mode. When operating as a master, the SBI adds one clock for acknowledgment signals. As a transmitter, the SBI releases the SDA pin during this clock cycle to receive acknowledgment signals from the receiver. As a receiver, the SBI pulls the SDA pin to the "L" level during this clock cycle and generates acknowledgment signals.

Setting <ACK> to "0" selects the non-acknowledgment mode. When operating as a master, the SBI does not generate clock for acknowledgement signals.

14.5.2 Setting the Number of Bits per Transfer

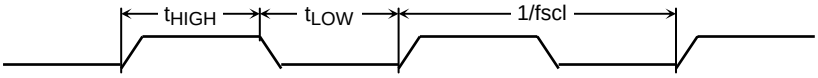
SBICR1 <BC2:0> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC2:0> is set to "000," causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC2:0> keeps a previously programmed value.

14.5.3 Serial Clock

① Clock source

SB SBICR1 <SCK2:0> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$t_{LOW} = 2^{n-1}/(fsys/2) + 58/(fsys/2)$

$t_{HIGH} = 2^{n-1}/(fsys/2) + 12/(fsys/2)$

$fscl = 1/(t_{Low} + t_{HIGH})$

$$= \frac{fsys/2}{2^n + 70}$$

SBI0CR1 <SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Fig. 14.5.3.1 Clock Source

The highest speeds in the standard and high-speed modes are specified to 100 KHz and 400 KHz respectively in the communications standards. Note that the internal SCL clock frequency is determined by the fsys used and the calculation formula shown above.

② Clock Synchronization

The I^2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "L" level overrides other masters producing the "H" level on their clock lines. This must be detected and responded by the masters producing the "H" level.

Clock synchronization assures correct data transfer on a bus that has two or more masters.

For example, the clock synchronization procedure for a bus with two masters is shown below.

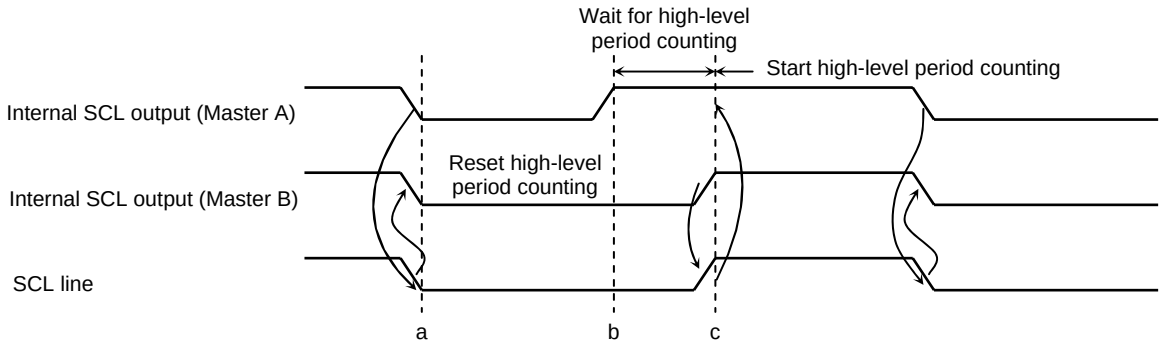


Fig. 14.5.3.2 Example of Clock Synchronization

At point a, Master A pulls its internal SCL output to the "L" level, bringing the SCL bus line to the "L" level. Master B detects this transition, resets its "H" level period counter, and pulls its internal SCL output level to the "L" level.

Master A completes counting of its "L" level period at point b, and brings its internal SCL output to the "H" level. However, Master B still keeps the SCL bus line at the "L" level, and Master A stops counting of its "H" level period counting. After Master A detects that Master B brings its internal SCL output to the "H" level and brings the SCL bus line to the "H" level at point c, it starts counting of its "H" level period.

This way, the clock on the bus is determined by the master with the shortest "H" level period and the master with the longest "L" level period among those connected to the bus.

14.5.4 Slave Addressing and Address Recognition Mode

When the SBI is configured to operate as a slave device, the slave address <SA6:0> and <ALS> must be set at I2CAR. Setting <ALS> to "0" selects the address recognition mode

14.5.5 Configuring the SBI as a Master or a Slave

Setting SBICR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when the stop condition has been detected on the bus or when arbitration has been lost.

14.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBICR2 <TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

In the slave mode, the SBI receives the direction bit ($R/\overline{W}$) from the master device on the following occasions:

- when data is transmitted in the addressing format
- when the received slave address matches the value specified at I2CCR
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros

If the value of the direction bit ($R/\overline{W}$) is "1," <TRX> is set to "1" by the hardware. If the bit is "0," <TRX> is set to "0."

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0," <TRX> changes to "1." If the SBI does not receive acknowledgement, <TRX> retains the previous value

<TRX> is cleared to "0" by the hardware when the stop condition has been detected on the bus or when arbitration has been lost.

14.5.7 Generating Start and Stop Conditions

When SBISR<BB> is "0," writing "1" to SBICR2 <MST, TRX, BB, PIN> causes the SBI to generate the start condition on the bus and output 8-bit data. <ACK> must be set to "1" in advance.

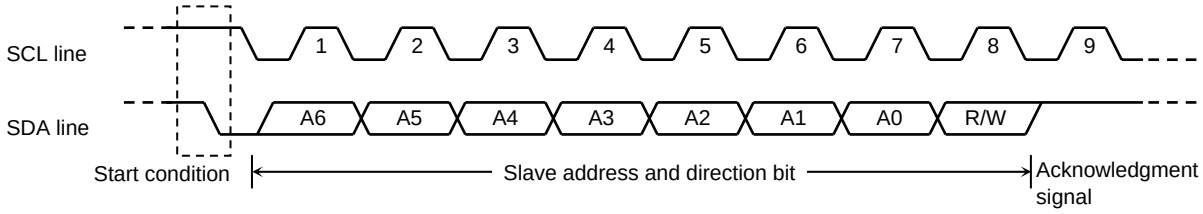


Fig. 14.5.7.1 Generating the Start Condition and a Slave Address

When <BB> is "1," writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

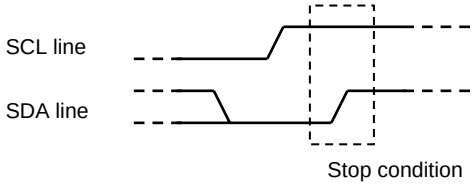


Fig. 14.5.7.2 Generating the Stop Condition

SBISR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and set to "0" when the stop condition is detected (the bus is free).

14.5.8 Interrupt Service Request and Release

When a serial bus interface interrupt request (INTSBI) is generated, SBICR2 <PIN> is cleared to "0." While <PIN> is "0," the SBI pulls the SCL line to the "L" level.

After transmission or reception of one data word, <PIN> is cleared to "0." It is set to "1" when data is written to or read from SBIDBR. It takes a period of t_{LOW} for the SCL line to be released after <PIN> is set to "1."

In the address recognition mode (<ALS> = "0"), <PIN> is cleared to "0" when the received slave address matches the value specified at I2CAR or when a general-call address is received; i.e., the eight bits following the start condition are all zeros. When the program writes "1" to SBICR2<PIN>, it is set to "1." However, writing "0" does clear this bit to "0."

14.5.9 Serial Bus Interface Operating Modes

SBICR2 <SBIM1:0> selects an operating mode of the serial bus interface. <SBIM1:0> must be set to "10" to configure the SBI for the I^2C bus mode. Make sure that the bus is free before switching the operating mode to the port mode.

14.5.10 Lost-arbitration Detection Monitor

The I^2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines. The I^2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below. Up until point a, Master A and Master B output the same data. At point a, Master A outputs the "L" level and Master B outputs the "H" level. Then Master A pulls the SDA bus line to the "L" level because the line has the wired-AND connection. When the SCL line goes high at point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid. In other words, Master B loses arbitration. Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

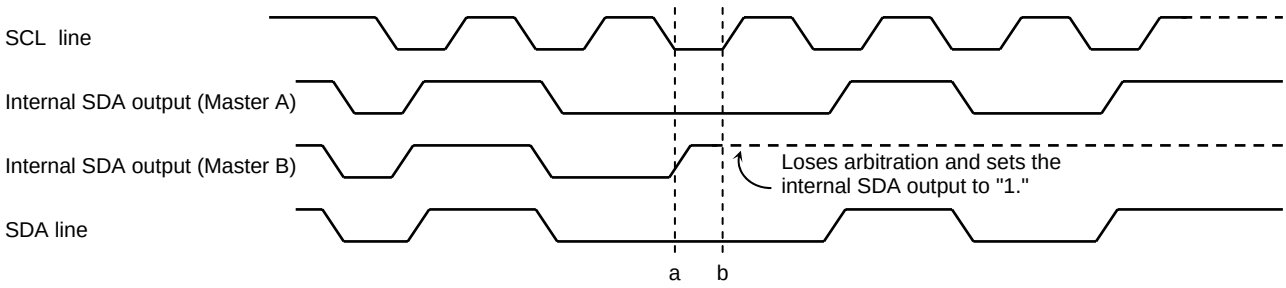


Fig. 14.5.10.1 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, the master loses arbitration and sets SBI0SR <AL> to "1."

When <AL> is set to "1," SBISR <MST, TRX> are cleared to "0," causing the SBI to operate as a slave receiver. <AL> is cleared to "0" when data is written to or read from SBIDBR or data is written to SBICR2.

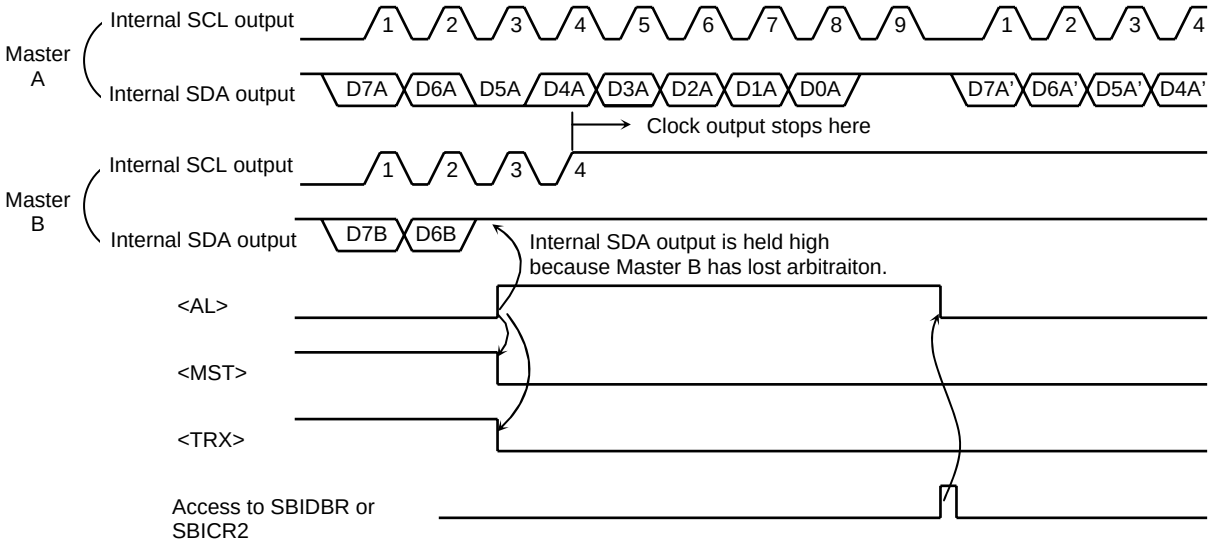


Fig. 14.5.10.2 Example of Master B Losing Arbitration (D7A = D7B, D6A = D6B)

14.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (I2CCR <ALS> = "0"), SBISR <AAS> is set to "1" on receiving the general-call address or the slave address that matches the value specified at I2CCR. When <ALS> is "1," <AAS> is set to "1" when the first data word has been received. <AAS> is cleared to "0" when data is written to or read from SBIDBR.

14.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBISR <AD0> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros. <AD0> is cleared to "0" when the start or stop condition is detected on the bus.

14.5.13 Last Received Bit Monitor

SBISR <LRB> is set to the SDA line value that was read at the rising of the SCL line. In the acknowledgment mode, reading SBISR <LRB> immediately after generation of the INTSBI interrupt request causes ACK signal to be read.

14.5.14 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBICR2 <SWRST1:0> generates a reset signal that initializes the serial bus interface circuit. After a reset, all control registers and status flags are initialized to their reset values. When the serial bus interface is initialized, <SWRST> is automatically cleared to "0."

(Note) After a software reset, the operating mode is also reset from the I^2C mode to the synchronous communication mode.

14.5.15 Serial Bus Interface Data Buffer Register (SBIDBR)

Reading or writing SBIDBR initiates reading received data or writing transmitted data. When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

14.5.16 I^2C Bus Address Register (I2CAR)

When the SBI is configured as a slave device, the I2CAR<SA6:0> bit is used to specify a slave address. If I2C0AR <ALS> is set to "0," the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format. If <ALS> is set to "1," the SBI does not recognize a slave address and receives data in the free data format.

14.5.17 IDLE Setting Register (SBIBR0)

The SBIBR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode. This register must be programmed before executing an instruction to switch to the standby mode.

14.6 Data Transfer Procedure in the I^2C Bus Mode

14.6.1 Device Initialization

First, program SBICR1<ACK, SCK2:0> by writing "0" to bits 7 to 5 and bit 3 in SBICR1.

Next, program I2CAR by specifying a slave address at <SA6:0> and an address recognition mode at <ALS>. (<ALS> must be set to "0" when using the addressing format.)

Next, program SBICR2 to initially configure the SBI in the slave receiver mode by writing "0" to <MST, TRX, BB> , "1" to <PIN> , "10" to <SBIM1:0> and "0" to bits 1 and 0.


```
                  7 6 5 4 3 2 1 0
SBICR1    ← 0 0 0 X 0 X X X     Specifies ACK and SCL clock.
I2CAR     ← X X X X X X X X     Specifies a slave address and an address recognition mode.
SBICR2    ← 0 0 0 1 1 0 0 0     Configures the SBI as a slave receiver.
(Note) X: Don't care
```

14.6.2 Generating the Start Condition and a Slave Address

① Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBICR1 <ACK> to select the acknowledgment mode. Write to SBIDBR a slave address and a direction bit to be transmitted.

When <BB> = "0," writing "1111" to SBICR2 <MST, TRX, BB, PIN> generates the start condition on the bus. Following the start condition, the SBI generates nine clocks from the SCL pin. The SBI outputs the slave address and the direction bit specified at SBIDBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0." In the master mode, the SBI holds the SCL line at the "L" level while <PIN> is "0." <TRX> changes its value according to the transmitted direction bit at generation of the INTSBI interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Settings in main routine

```
                  7 6 5 4 3 2 1 0
┌→ Reg.       ← SBISR
│  Reg.       ← Reg. e 0x20
│  if Reg.    ≠ 0x00              Ensures that the bus is free.
└─ Then
   SBICR1     ← X X X 1 0 X X X   Selects the acknowledgement mode.
   SBIDR1     ← X X X X X X X X   Specifies the desired slave address and direction.
   SBICR2     ← 1 1 1 1 1 0 0 0   Generates the start condition.
```

Example of INTSBI interrupt routine

```
INTCLR ← 0x50                     Clears the interrupt request.
Processing
End of interrupt
```

② Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line. If the received address matches its slave address specified at I2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "L" level during the ninth clock and outputs an acknowledgment signal.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0." In the slave mode, the SBI holds the SCL line at the "L" level while <PIN> is "0."

(Note) The user can only use a DMA transfer:
- **when there is only one master and only one slave and**
- **continuous transmission or reception is possible.**

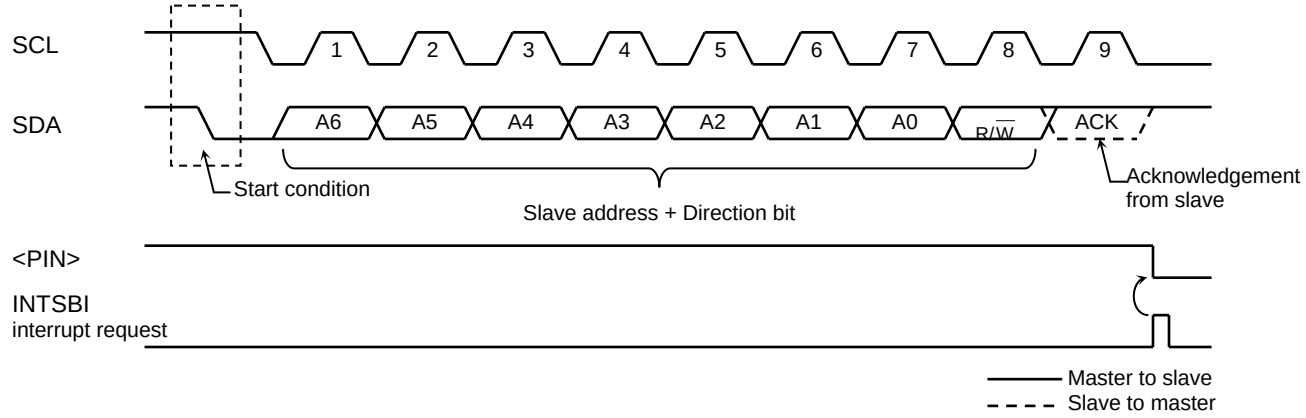


Fig. 14.6.2.1 Generation of the Start Condition and a Slave Address

14.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBI interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

① Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1," that means the receiver requires no further data. The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0," that means the receiver requires further data. If the next data to be transmitted has eight bits, the data is written into SBIDBR. If the data has different length, <BC2:0> and <ACK> are programmed and the transmit data is written into SBIDBR. Writing the data makes <PIN> to"1," causing the SCL pin to generate a serial clock for transfer of a next data word, and the SDA pin to transfer the data word. After the transfer is completed, the INTSBI interrupt request is generated, <PIN> is set to "0," and the SCL pin is pulled to the "L" level. To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBI interrupt

if MST = 0
Then go to the slave-mode processing
if TRX = 0
Then go to the receiver-mode processing
if LRB = 0
Then go to processing for generating the stop condition
SBICR1 ← X X X X 0 X X X Specifies the number of bits to be transmitted and specify whether ACK is required.
SBIDBR ← X X X X X X X X Writes the transmit data.
End of interrupt processing
(Note) X: Don't care

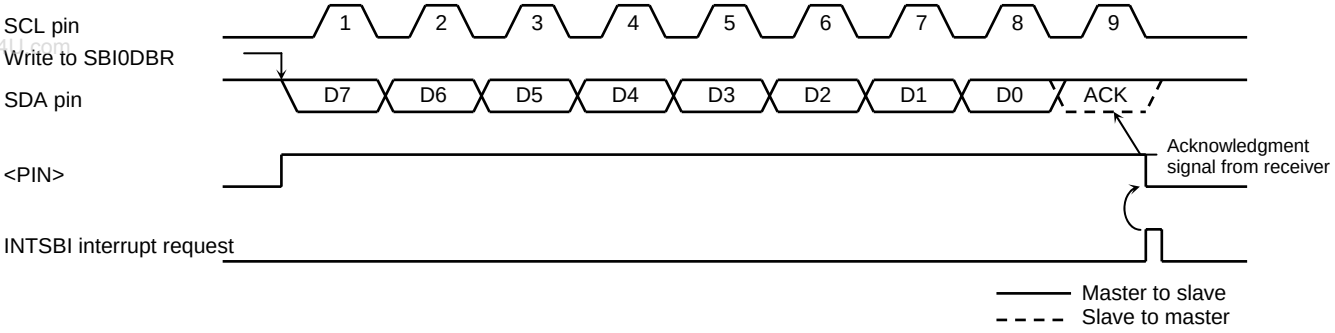


Fig. 14.6.3.1 <BC2:0> = "000" and <ACK> = "1" (Transmitter Mode)

Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBIDBR. If the data has different length, <BC2:0> and <ACK> are programmed and the received data is read from SBIDBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.) On reading the data, <PIN> is set to "1," and the serial clock is output to the SCL pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the "L" level, "0" is output to the SDA pin.

After that, the INTSBI interrupt request is generated, and <PIN> is cleared to "0," pulling the SCL pin to the "L" level. Each time the received data is read from SBIDBR, one-word transfer clock and an acknowledgement signal are output.

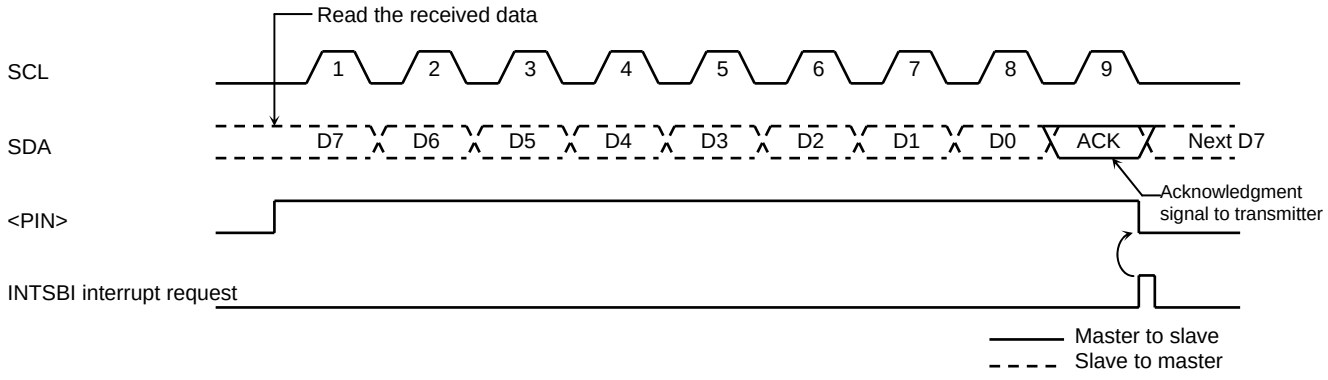


Fig. 14.6.3.2 <BC2:0> = "000" and <ACK> = "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be set to "0" immediately before reading the second to last data word. This disables generation of an acknowledgment clock for the last data word. When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC2:0> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer. At this time, the master receiver holds the SDA bus line at the "H" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

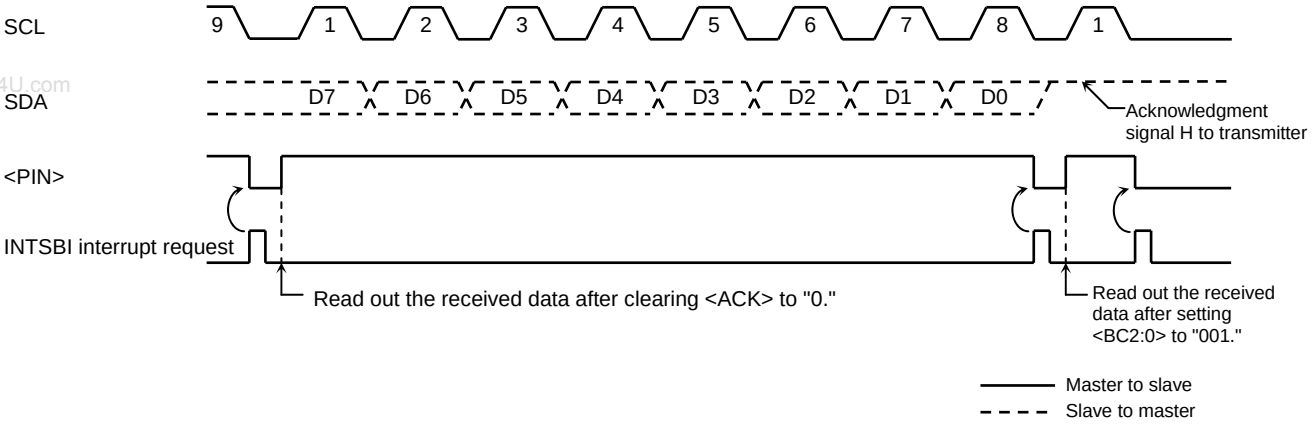


Fig. 14.6.3.3 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data words

INTSBI interrupt (after data transmission)

```
                  7 6 5 4 3 2 1 0
SBICR1    ← X X X X 0 X X X      Sets the number of bits of data to be received and specify
                                 whether ACK is required.
Reg.      ← SBI0CBR              Reads dummy data.
End of interrupt
```

INTSBI interrupt (first to (N-2)th data reception)

```
                  7 6 5 4 3 2 1 0
Reg.      ← SBIDBR               Reads the first to (N-2)th data words.
End of interrupt
```

INTSBI interrupt ((N-1)th data reception)

```
                  7 6 5 4 3 2 1 0
SBI0CR1   ← X X X 0 0 X X X      Disables generation of acknowledgement clock.
Reg.      ← SBIDBR               Reads the (N-1)th data word.
End of interrupt
```

INTSBI interrupt (Nth data reception)

```
                  7 6 5 4 3 2 1 0
SBI0CR1   ← 0 0 1 0 0 X X X      Generates a clock for 1-bit transfer.
Reg.      ← SBIDBR               Reads the Nth data word.
End of interrupt
```

INTSBI interrupt (after completing data reception)

```
Processing to generate the stop condition   Terminates the data transmission.
End of interrupt
```

(Note) X: Don't care

② Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBI interrupt request on four occasions: 1) when the SBI has received any slave address from the master, 2) when the SBI has received a general-call address, 3) when the received slave address matches its own address, and 4) when a data transfer has been completed in response to a general-call. Also, if the SBI loses arbitration in the master mode, it switches to the slave mode. Upon the completion of data word transfer in which arbitration is lost, the INTSBI interrupt request is generated, <PIN> is cleared to "0," and the SCL pin is pulled to the "L" level. When data is written to or read from SBIDBR or when <PIN> is set to "1," the SCL pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of lost arbitration is carried out.

SBISR <AL>, <TRX>, <AAS> and <AD0> are tested to determine the processing required. Table 14.6.3.4 shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode

INTSBI interrupt

```
if TRX = 0
Then go to other processing
if AL = 1
Then go to other processing
if AAS = 0
Then go to other processing
SBICR1   ← X X X 1 0 X X X    Sets the number of bits to be transmitted.
SBIDBR   ← X X X X 0 X X X    Sets the transmit data.
```

(Note) X: Don't care

Table 14.6.3.4 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration was lost while the slave address was being transmitted, and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC2:0> and write the transmit data into SBI0DBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1," that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0," that means the receiver requires further data. Set the number of bits in the data word to <BC2:0> and write the transmit data to the SBIDBR.
0	1	1	1/0	Arbitration was lost while a slave address was being transmitted, and the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration was lost while a slave address or a data word was being transmitted, and the transfer terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC2:0> and read the received data from SBIDBR.

14.6.4 Generating the Stop Condition

When SBISR <BB> is "1," writing "1" to SBICR2 <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. Do not alter the contents of <MST, TRX, BB, PIN> until the stop condition appears on the bus.

If another device is holding down the SCL bus line, the SBI waits until the SCL line is released. After that, the SDA pin goes high, causing the stop condition to be generated.

		7	6	5	4	3	2	1	0
SBICR2	←	1	1	0	1	1	0	0	0

Generates the stop condition.

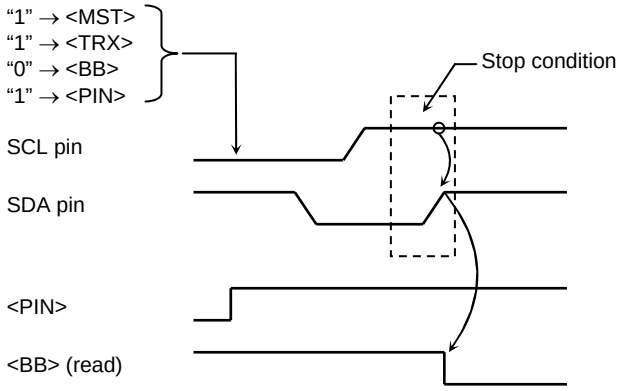


Fig. 14.6.4.1 Generating the Stop Condition

14.6.5 Repeated Start Procedure

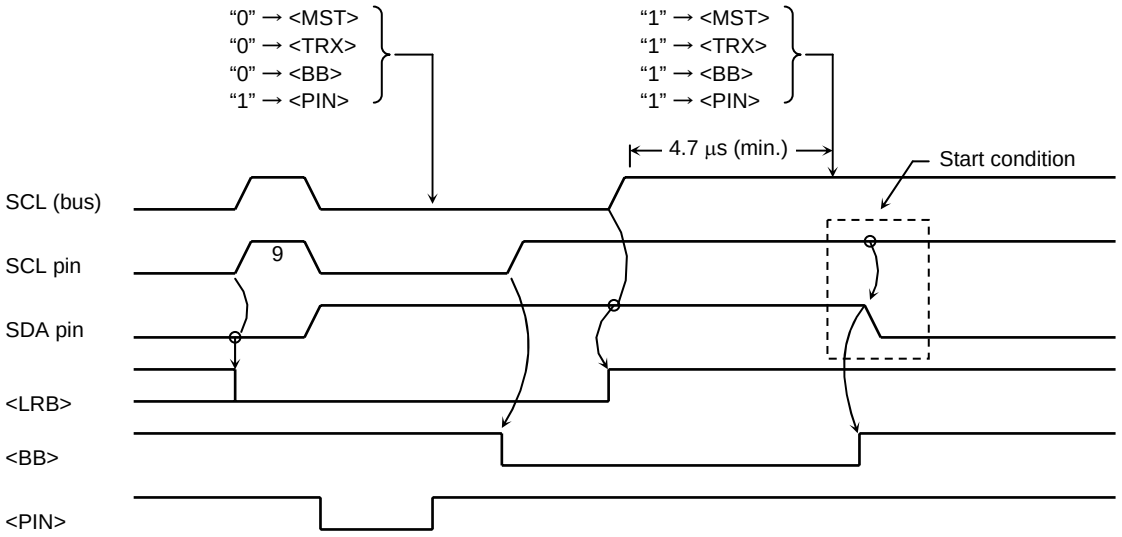
Repeated start is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a repeated start in the master mode is described below.

First, set SBICR2 <MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDA pin is held at the "H" level and the SCL pin is released. Because no stop condition is generated on the bus, other devices think that the bus is busy. Then, test SBISR <BB> and wait until it becomes "0" to ensure that the SCL pin is released. Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCL bus line to the "L" level. Once the bus is determined to be free this way, use the steps described above in (2) to generate the start condition.

To satisfy the setup time of repeated start, at least 4.7-μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

	7 6 5 4 3 2 1 0	
SBICR2	← 0 0 0 1 1 0 0 0	Releases the bus.
if SBISR<BB> ≠ 0		Checks that the SCL pin is released.
Then		
if SBISR<LRB> ≠ 1		Checks that no other device is pulling the SCL pin to the "L" level.
Then		
4.7 μs Wait		
SBICR1	← X X X 1 0 X X X	Selects the acknowledgment mode.
SBIDBR	← X X X X X X X X	Sets the desired slave address and direction.
SBICR2	← 1 1 1 1 1 0 0 0	Generates the start condition.

(Note) X: Don't care



(Note) Do not write <MST> to "0" when it is "0." (Repeated start cannot be done.)

Fig. 14.6.5.1 Timing Chart of Generating a Repeated Start

14.7 Control in the Clock-synchronous 8-bit SIO Mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

Serial bus interface control register 0

SBICR0 (0xFFFF_F257)

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0	0	0	0	0	0	0
Function	SBI operation 0: Disable 1: Enable							

<SBIEN>: To use the SBI, enable the SBI operation ("1") before setting each register of SBI module.

(Note) Bits 0 to 6 of SBICRO are read as "0."

Serial bus interface control register 1

SBICR1 (0xFFFF_F250)

	7	6	5	4	3	2	1	0
bit Symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	R/W				R	R/W		R/W
After reset	0	0	0	0	1	0	0	1
Function	Start transfer 0: Stop 1: Start	Abort transfer 0: Continue 1: Abort	Select transfer mode 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode			Select serial clock frequency		

On writing <SCK2:0>: Select serial clock frequency

000	n = 4	1.69 MHz
001	n = 5	844 kHz
010	n = 6	422 kHz
011	n = 7	211 kHz
100	n = 8	105 kHz
101	n = 9	53 kHz
110	n =10	26 kHz
111	—	External clock

System clock : fsys (=54 MHz)
Clock gear : fc/1
Frequency = $\frac{fsys/2}{2^n}$ [Hz]

(Note) Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.

(Note) After a reset, the <SCK0> bit is read as "1." If the SIO mode is selected at the SBICR2 register, the initial value of the <SCK0> bit becomes "0."

Serial bus interface data buffer register

SBIDBR (0xFFFF_F251)

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	0							

Fig. 14.7.1.1 SIO Mode Registers

Serial bus interface control register 2

SBICR2 (0xFFFF_F253)

	7	6	5	4	3	2	1	0
bit Symbol					SBIM1	SBIM0		
Read/Write	R				W		R	
After reset	1	1	1	1	0	0	1	1
Function					Select serial bus interface operating mode 00: Port mode 01: Clock-synchronous 8-bit SIO mode 10: I^2C bus mode 11: (Reserved)			

Serial bus interface register

SBISR (0xFFFF_F253)

	7	6	5	4	3	2	1	0
bit Symbol					SIOF	SEF		
Read/Write	R				R		R	
After reset	1	1	1	1	0	0	1	1
Function					Serial transfer status monitor 0: Terminated 1: In progress	Shift operation status monitor 0: Terminated 1: In progress		

Serial bus interface baud rate register 0

SBIBR0 (0xFFFF_F254)

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI						
Read/Write	R	R/W	R					R/W
After reset	1	0	1	1	1	1	1	0
Function		IDLE 0: Stop 1: Operate						Make sure that you write "0."

Fig. 14.7.1.2 SIO Mode Registers

14.7.1 Serial Clock

① Clock source

Internal or external clocks can be selected by programming SBICR1 <SCK2:0>.

Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCK pin. At the beginning of a transfer, the SCK pin output becomes the "H" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

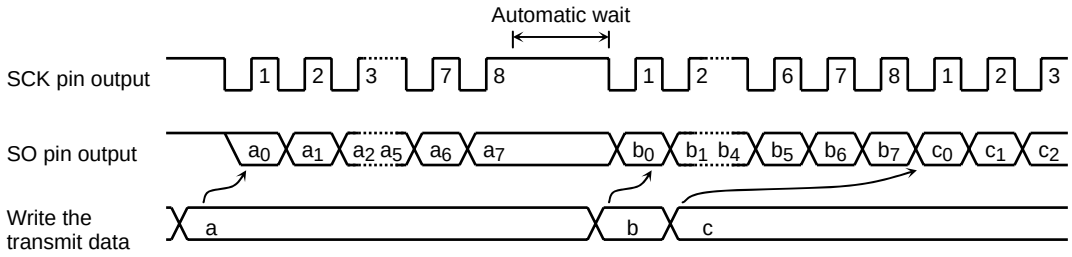


Fig. 14.7.1.3 Automatic Wait

External clock (<SCK2:0> = "111")

The SBI uses an external clock supplied from the outside to the SCK pin as a serial clock. For proper shift operations, the serial clock at the "H" and "L" levels must have the pulse widths as shown below.

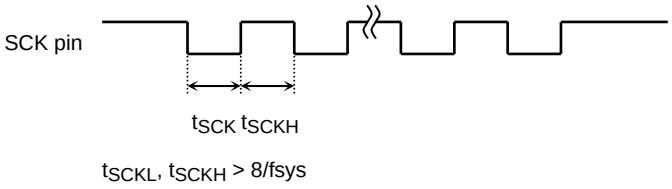


Fig. 14.7.1.4 Maximum Transfer Frequency of External Clock Input

② Shift Edge

Leading-edge shift is used in transmission. Trailing-edge shift is used in reception.

Leading-edge shift

Data is shifted at the leading edge of the serial clock (or the falling edge of the SCK pin input/output).

Trailing-edge shift

Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCK pin input/output).

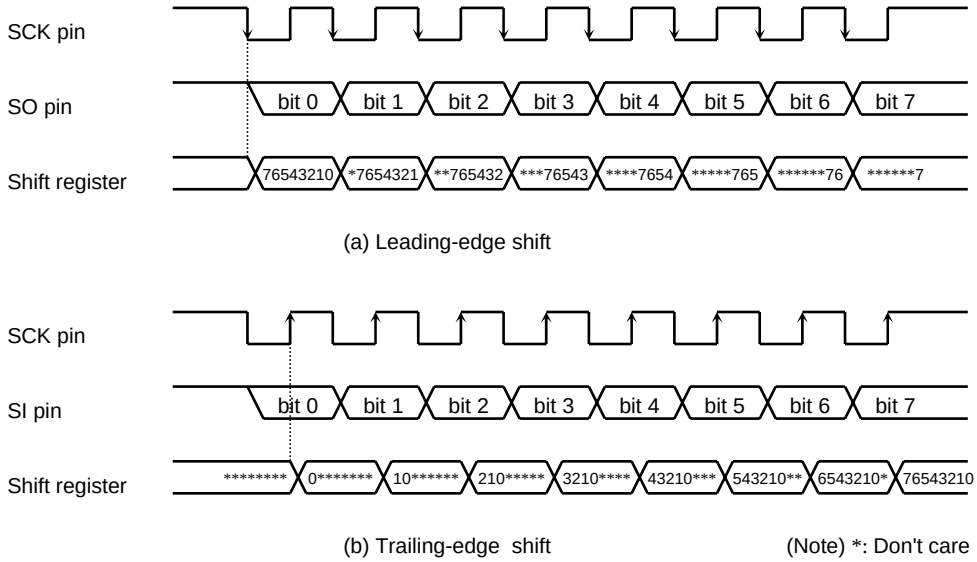


Fig. 14.7.1.5 Shift Edge

14.7.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBICR1 <SIOM1:0>.

① 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBIDBR.

After writing the transmit data, writing "1" to SBICR1 <SIOS> starts the transmission. The transmit data is moved from SBIDBR to a shift register and output to the SO pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the transmit data is transferred to the shift register, SBIDBR becomes empty, and the INTSBI (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBIDBR is loaded with the next transmit data.

In the external clock mode, SBIDBR must be loaded with data before the next data shift operation is started. Therefore, the data transfer rate varies depending on the maximum latency between when the interrupt request is generated and when SBIDBR is loaded with data in the interrupt service program.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting SBISR <SIOF> to "1" to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBI0SR <SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1," the transmission is aborted immediately and <SIOF> is cleared to "0."

In the external clock mode, <SIOS> must be set to "0" before the next transmit data shift operation is started. Otherwise, operation will stop after dummy data is transmitted.

```
                  7 6 5 4 3 2 1 0
SBICR1    ← 0 1 0 0 0 X X X      Selects the transmit mode.

SBIDBR    ← X X X X X X X X      Writes the transmit data.
SBICR1    ← 1 0 0 0 0 X X X      Starts transmission.
```

INTSBI interrupt

```
SBIDBR    ← X X X X X X X X      Writes the transmit data.
```

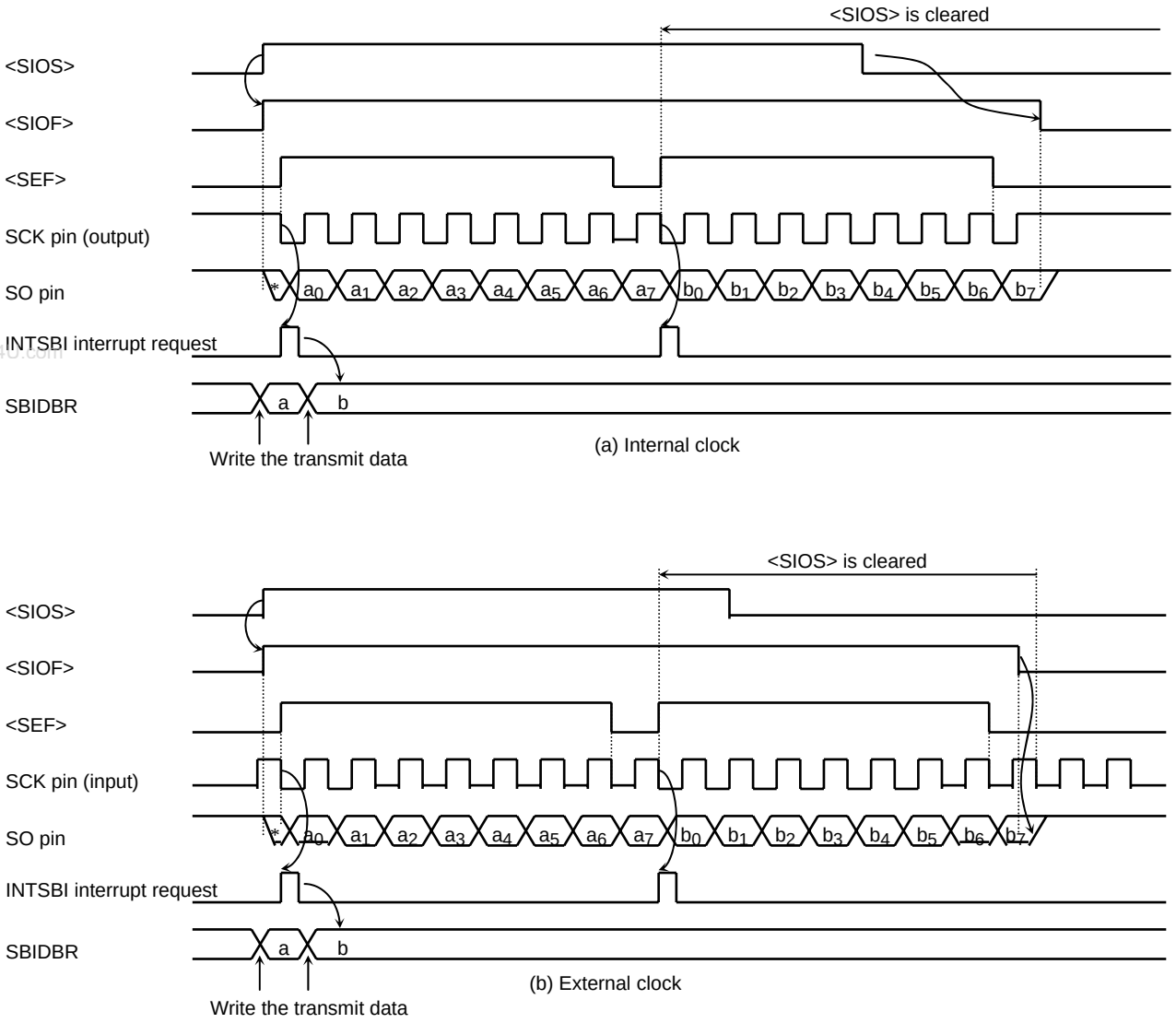


Fig. 14.7.2.1 Transmit Mode

Example: Example of programming (MIPS16) to terminate transmission by <SIO> (external clock)

```
                ADDIU   r3,  r0,  0x04
STEST1  :       LB      r2, (SBISR)          ;  If SBISR<SEF> = 1 then loop
                AND     r2,  r3
                BNEZ    r2,  STEST1
                ADDIU   r3,  r0,  0x20
STEST2  :       LB      r2, (Px)             ;  If SCK = 0 then loop
                AND     r2,  r3
                BEQZ    r2,  STEST2
                ADDIU   r3,  r0,  0y00000111
                STB     r3, (SBICR1)         ;  <SIOS> ← 0
```

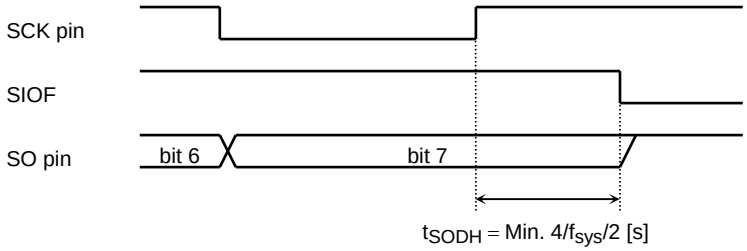


Fig. 14.7.2.2 Transmit Data Retention Time at the End of Transmission

② 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBICR1 <SIOS> enables reception. Data is taken into the shift register from the SI pin, with the least-significant bit (LSB) first, in synchronization with the serial clock. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIDBR and the INTSBI (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBIDBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data.

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBIDBR. The program checks SBISR <SIOF> to determine whether reception has come to an end. <SIOF> is cleared to "0" at the end of reception. After confirming the completion of the reception, last received data is read. If <SIOINH> is set to "1," the reception is aborted immediately and <SIOF> is cleared to "0." (The received data becomes invalid, and there is no need to read it out.)

(Note) The contents of SBIDBR will not be retained after the transfer mode is changed. The ongoing reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

```
                  7 6 5 4 3 2 1 0
SBICR1    ← 0 1 1 1 0 X X X      Selects the receive mode.

SBICR1    ← 1 0 1 1 0 0 0 0      Starts reception.
```

INTSBI interrupt

```
Reg.      ← SBIDBR               Reads the received data.
```

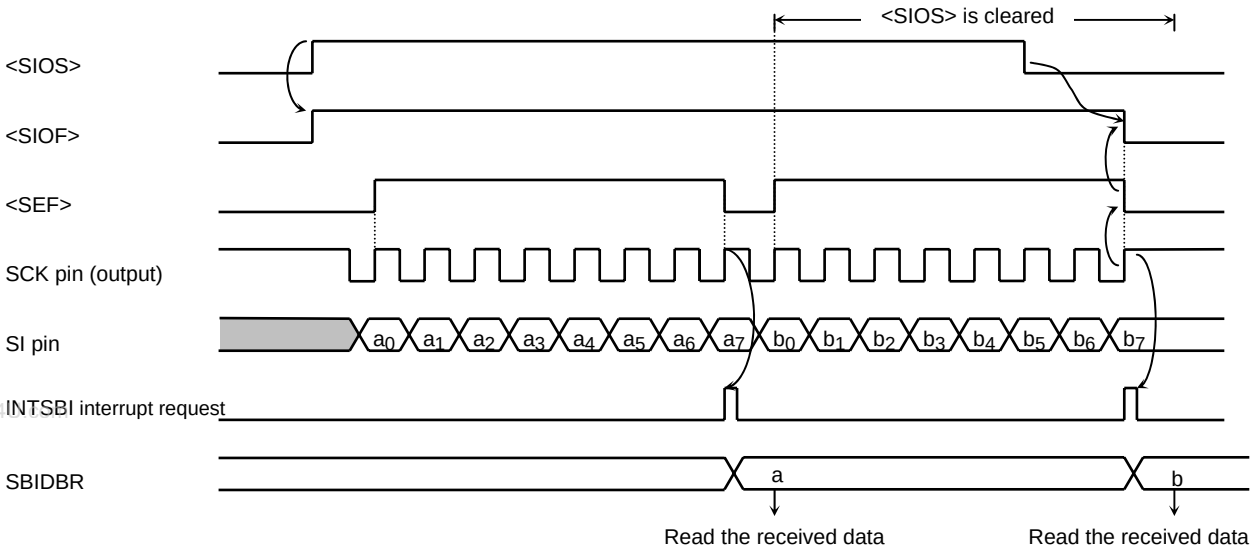


Fig. 14.7.2.3 Receive Mode (Example: Internal Clock)

③ 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBIDBR and setting SBICR1 <SIOS> to "1" enables transmission and reception. The transmit data is output through the SO pin at the falling of the serial clock, and the received data is taken in through the SI pin at the rising of the serial clock, with the least-significant bit (LSB) first. Once the shift register is loaded with the 8-bit data, it transfers the received data to SBIDBR and the INTSBI interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBIDBR is shared between transmit and receive operations, the received data must be read before the next transmit data is written.

In the internal clock operation, the serial clock will be automatically in the wait state until the received data is read and the next transmit data is written.

In the external clock mode, shift operations are executed in synchronization with the external serial clock. Therefore, the received data must be read and the next transmit data must be written before the next shift operation is started. The maximum data transfer rate for the external clock operation varies depending on the maximum latency between generating the interrupt request and reading the received data and writing the transmit data.

At the beginning of transmission, the same value as in the last bit of the previously transmitted data is output in a period from setting <SIOF> to "1" to the falling edge of SCK.

Transmission and reception can be terminated by clearing <SIOS> to "0" or setting SBICR1 <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBIDBR. The program checks SBISR <SIOF> to determine whether transmission and reception have come to an end. <SIOF> is cleared to "0" at the end of transmission and reception. If <SIOINH> is set, the transmission and reception are aborted immediately and <SIOF> is cleared to "0."

(Note) The contents of SBIDBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

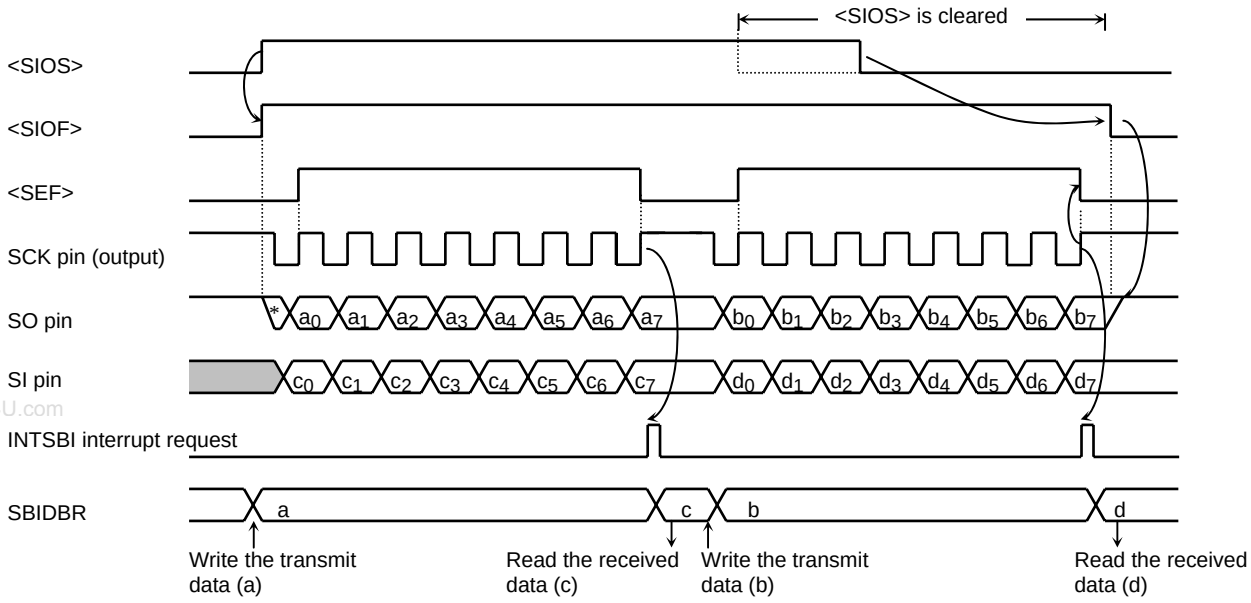


Fig. 14.7.2.4 Transmit/Receive Mode (Example: Internal Clock)

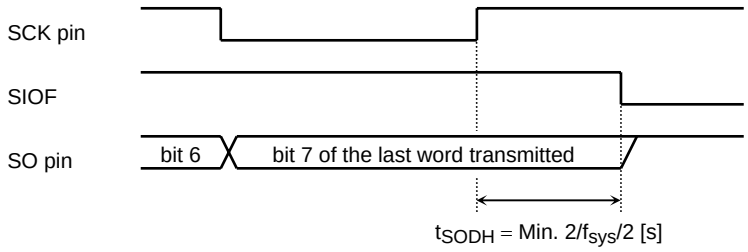


Fig. 14.7.2.5 Transmit Data Retention Time at the End of Transmission/Reception (In the Transmit/Receive Mode)

```
                  7 6 5 4 3 2 1 0
SBICR1      ← 0 1 1 0 0 X X X     Selects the transmit mode.

SBIDBR      ← X X X X X X X X     Writes the transmit data.
SBICR1      ← 1 0 1 0 0 X X X     Starts reception/transmission.
```

INTSBI interrupt

```
Reg.        ← SBIODBR             Reads the received data.
SBIDBR      ← X X X X X X X X     Writes the transmit data.
```

15. Analog/Digital Converter

A 10-bit, sequential-conversion analog/digital converter (A/D converter) is built into the TMP19A64. This A/D converter is equipped with 24 analog input channels.

Fig. 15.1 shows the block diagram of this A/D converter.

These 24 analog input channels (pins AN0 through AN23) are also used as input ports.

(Note) If it is necessary to reduce a power current by operating the TMP19A64 in IDLE, SLEEP, SLOW or STOP mode and if either case shown below is applicable, you must first stop the A/D converter and then execute the instruction to put the TMP19A64 into standby mode:

1) The TMP19A64 must be put into IDLE mode when ADMOD1<I2AD> is "0."
2) The TMP19A64 must be put into SLEEP, SLOW or STOP mode.

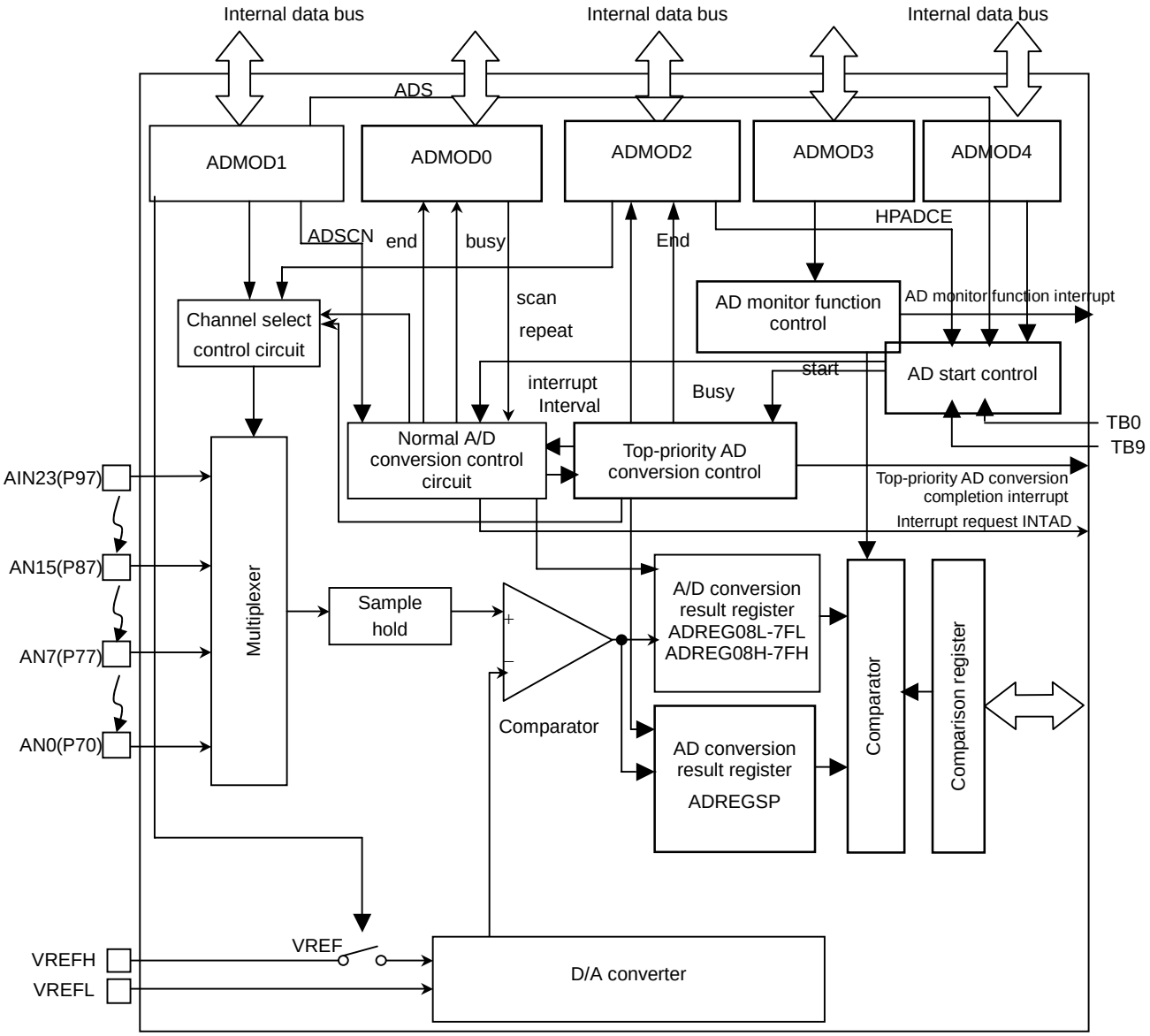


Fig. 15.1 A/D Converter Block Diagram

15.1 Control Register

The A/D converter is controlled by A/D mode control registers (ADMOD0, ADMOD1, ADMOD2, ADMOD3 and ADMOD4). Results of A/D conversion are stored in 16 upper and lower A/D conversion result registers ADREG08H/L through ADREG7FH/L. Results of High-priority conversion are stored in ADREGSPH/L.

Fig. 15.1.1 shows the registers related to the A/D converter.

A/D Mode Control Register 0

ADMOD0 (0xFFFF_F314)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFN	ADBFN		ITM1	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R	R/W				
After reset	0	0	0	0	0	0	0	0
Function	Normal A/D conversion completion flag 0: Before or during conversion 1: Completion	Normal A/D conversion BUSY flag 0: Conversion stop 1: During conversion	"0" is read.	Specify interrupt in fixed channel repeat conversion mode	Specify interrupt in fixed channel repeat conversion mode	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode	Specify scan mode 0: Fixed channel mode 1: Channel scan mode	Start A/D conversion 0: Don't care 1: Start conversion "0" is always read.

Specify A/D conversion interrupt in fixed channel repeat conversion mode

	Fixed channel repeat conversion mode <SCAN> = "0," <REPEAT> = "1"
00	Generate interrupt once every single conversion
01	Generate interrupt once every 4 conversions
10	Generate interrupt once every 8 conversions
11	Setting prohibited

Fig. 15.1.1 Registers related to the A/D Converter

A/D Mode Control Register 1

ADMOD1 (0xFFFF_F315)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON	I2AD	ADSCN	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	VREF application control 0: OFF 1: ON	IDLE 0: Stop 1: Activate	Specify operation mode for channel scanning 0: 4ch scan 1: 8ch scan	Select analog input channel				

Select analog input channel

<SCAN> / <ADCH4,3,2,1,0>	0 Fixed channel	1 4 channel scan (ADSCN=0)	1 8 channel scan (ADSCN=1)
00000	AN0	AN0	AN0
00001	AN1	AN0 to AN1	AN0 to AN1
00010	AN2	AN0 to AN2	AN0 to AN2
00011	AN3	AN0 to AN3	AN0 to AN3
00100	AN4	AN4	AN0 to AN4
00101	AN5	AN4 to AN5	AN0 to AN5
00110	AN6	AN4 to AN6	AN0 to AN6
00111	AN7	AN4 to AN7	AN0 to AN7
01000	AN8	AN8	AN8
01001	AN9	AN8 to AN9	AN8 to AN9
01010	AN10	AN8 to AN10	AN8 to AN10
01011	AN11	AN8 to AN11	AN8 to AN11
01100	AN12	AN12	AN8 to AN12
01101	AN13	AN12 to AN13	AN8 to AN13
01110	AN14	AN12 to AN14	AN8 to AN14
01111	AN15	AN12 to AN15	AN8 to AN15
10000	AN16	AN16	AN16
10001	AN17	AN16 to AN17	AN16 to AN17
10010	AN18	AN16 to AN18	AN16 to AN18
10011	AN19	AN16 to AN19	AN16 to AN19
10100	AN20	AN20	AN16 to AN20
10101	AN21	AN20 to AN21	AN16 to AN21
10110	AN22	AN20 to AN22	AN16 to AN22
10111	AN23	AN20 to AN23	AN16 to AN23

(Note 1) Before starting AD conversion, write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

(Note 2) To go into standby mode upon completion of AD conversion, set <VREFON> to "0."

Fig. 15.1.2 Registers related to the A/D Converter

A/D Mode Control Register 2

ADMOD2 (0xFFFF_F316)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFHP	ADBFHP	HPADCE	HPADCH4	HPADCH3	HPADCH2	HPADCH1	HPADCH0
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	Top-priority AD conversion completion flag 0: Before or during conversion 1: Upon completion	Top-priority AD conversion BUSY flag 0: During conversion halts 1: During conversion	Activate top-priority AD conversion 0:Don't care 1:Start conversion "0" is always read.	Select analog input channel when activating top-priority AD conversion				

<HPADCH4,3,2,1,0>	Analog input channel when executing top-priority AD conversion
00000	AN0
00001	AN1
00010	AN2
00011	AN3
00100	AN4
00101	AN5
00110	AN6
00111	AN7
01000	AN8
01001	AN9
01010	AN10
01011	AN11
01100	AN12
01101	AN13
01110	AN14
01111	AN15
10000	AN16
10001	AN17
10010	AN18
10011	AN19
10100	AN20
10101	AN21
10110	AN22
10111	AN23

Fig. 15.1.3 Registers related to the A/D Converter

A/D Mode Control Register 3

ADMOD3 (0xFFFF_F317)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC	REGS3	REGS2	REGS1	REGS0	ADOBSV
Read/Write	R/W	R	R/W	R/W				R/W
After reset	0	0	0	0	0	0	0	0
Function	Write "0."	"0" is read.	Make AD monitor function interrupt setting 0: Smaller than comparison Regi 1: Larger than comparison Regi	BIT for selecting the AD conversion result storage Regi that is to be compared with the comparison Regi if the AD monitor function is enabled				AD monitor function 0: Disable 1: Enable

<REGS3,2,1,0>	AD conversion result storage Regi to be compared
0000	ADREG08
0001	ADREG19
0010	ADREG2A
0011	ADREG3B
0100	ADREG4C
0101	ADREG5D
0110	ADREG6E
0111	ADREG7F
1XXX	ADREGSP

A/D Mode Control Register 4

ADMOD4 (0xFFFF_F318)

	7	6	5	4	3	2	1	0
bit Symbol	HADHS	HADHTG	ADHS	ADHTG			ADRST1	ADRST0
Read/Write	R/W				R		W	W
After reset	0	0	0	0	0		0	0
Function	HW source for activating top-priority A/D conversion 0: INTTB90 1: INTTB91	HW for activating top-priority A/D conversion 0: Disable 1: Enable	HW source for activating normal A/D conversion 0: INTTB00 1: INTTB01	HW for activating normal A/D conversion 0: Disable 1: Enable	"0" is read.		Overwriting 10 with 01 allows ADC to be software reset. All registers except the ADCLK register are initialized.	

(Note 1) If AD conversion is executed with the match triggers <ADHTG> and <HADHTG> of a 16-bit timer set to "1" by using a source for triggering H/W, A/D conversion can be activated at specified intervals by performing three steps shown below when the timer is idle:

① Select a source for triggering HW: <ADHS>, <HADHS>
② Enable H/W activation of AD conversion: <ADHTG>, <HADHTG>
③ Start the timer.

(Note 2) Do not make a High-priority AD conversion setting and a normal AD conversion setting simultaneously.

Fig. 15.1.4 Registers related to the A/D Converter

Lower A/D Conversion Result Register 08

ADREG08L (0xFFFF_F300)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00					OVR0	ADR0RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 08

ADREG08H (0xFFFF_F301)

	7	6	5	4	3	2	1	0
bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 19

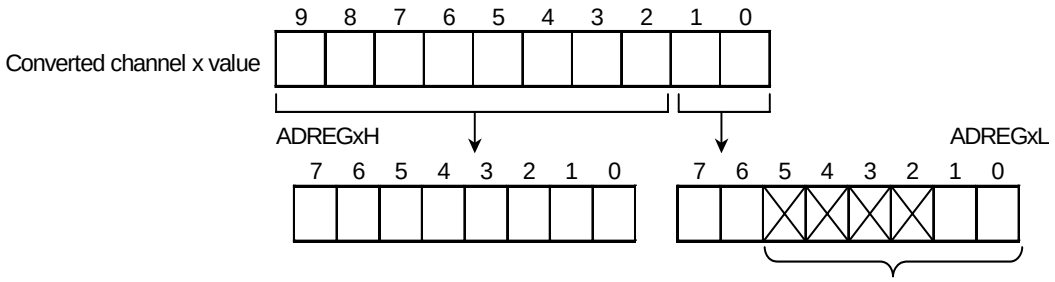
ADREG19L (0xFFFF_F302)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10					OVR1	ADR1RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUNflag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 19

ADREG19H (0xFFFF_F303)

	7	6	5	4	3	2	1	0
bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 of registers ADREG08L and ADREG19L are always "1."
- Bit 0 of registers ADREG08L and ADREG19L is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) clears this bit to "0."
- Bit 1 of registers ADREG08L and ADREG19L is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- **When reading conversion result storage registers, first read upper registers and then lower registers.**

Fig. 15.1.5 Registers related to the A/D Converter

Lower A/D Conversion Result Register 2A

ADREG2AL (0xFFFF_F304)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20					OVR2	ADR2RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 2A

ADREG2AH (0xFFFF_F305)

	7	6	5	4	3	2	1	0
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 3B

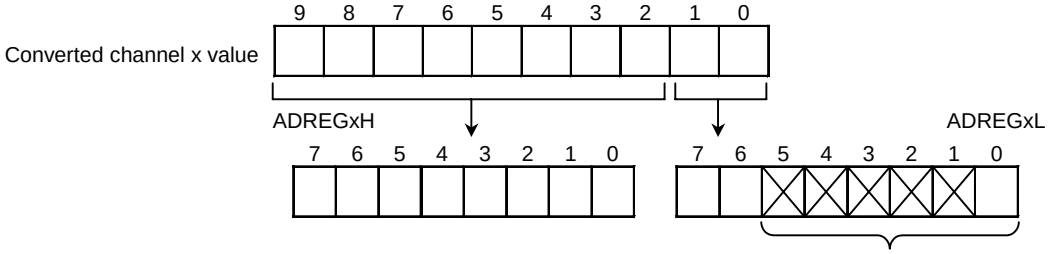
ADREG3BL (0xFFFF_F306)

	7	6	5	4	3	2	1	0
bit Symbol	ADR31	ADR30					OVR3	ADR3RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 3B

ADREG3BH (0xFFFF_F307)

	7	6	5	4	3	2	1	0
bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 of registers ADREG2AL and ADREG3BL are always "1."
- Bit 0 of registers ADREG2AL and ADREG3BL is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) clears this bit to "0."
- Bit 1 of registers ADREG2AL and ADREG3BL is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- **When reading conversion result storage registers, first read upper registers and then lower registers.**

Fig. 15.1.6 Registers related to the A/D Converter

Lower A/D Conversion Result Register 4C

ADREG4CL (0xFFFF_F308)

	7	6	5	4	3	2	1	0
bit Symbol	ADR41	ADR40					OVR4	ADR4RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 4C

ADREG4CH (0xFFFF_F309)

	7	6	5	4	3	2	1	0
bit Symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 5D

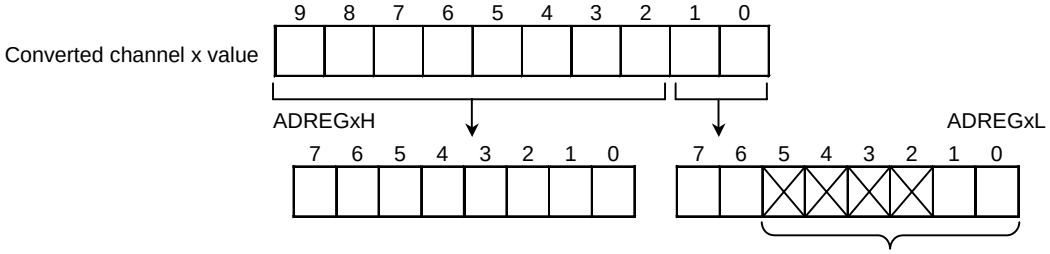
ADREG5DL (0xFFFF_F30A)

	7	6	5	4	3	2	1	0
bit Symbol	ADR51	ADR50					OVR5	ADR5RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 5D

ADREG5DH (0xFFFF_F30B)

	7	6	5	4	3	2	1	0
bit Symbol	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 of registers ADREG4CL and ADREG5DL are always "1."
- Bit 0 of registers ADREG4CL and ADREG5DL is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) clears this bit to "0."
- Bit 1 of registers ADREG4CL and ADREG5DL is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- **When reading conversion result storage registers, first read upper registers and then lower registers.**

Fig. 15.1.7 Registers related to the A/D Converter

Lower A/D Conversion Result Register 6E

ADREG6EL (0xFFFF_F30C)

	7	6	5	4	3	2	1	0
bit Symbol	ADR61	ADR60					OVR6	ADR6RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 6E

ADREG6EH (0xFFFF_F30D)

	7	6	5	4	3	2	1	0
bit Symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 7F

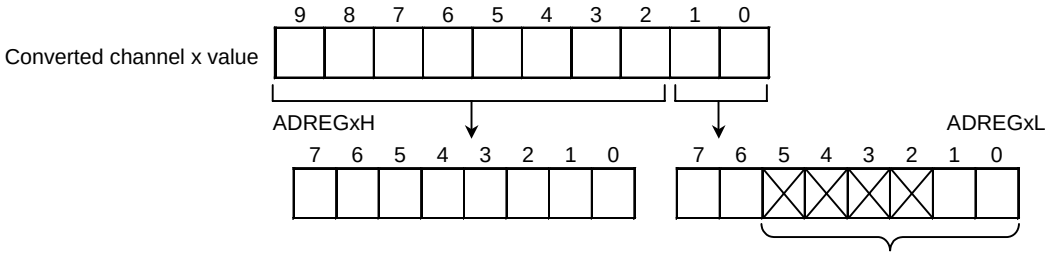
ADREG7FL (0xFFFF_F30E)

	7	6	5	4	3	2	1	0
bit Symbol	ADR71	ADR70					OVR7	ADR7RF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUNflag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 7F

ADREG7FH (0xFFFF_F30F)

	7	6	5	4	3	2	1	0
bit Symbol	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 of registers ADREG6EL and ADREG7FL are always "1."
- Bit 0 of registers ADREG6EL and ADREG7FL is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) clears this bit to "0."
- Bit 1 of registers ADREG6EL and ADREG7FL is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- **When reading conversion result storage registers, first read upper registers and then lower registers.**

Fig. 15.1.8 Registers related to the A/D Converter

Lower A/D Conversion Result Register SP

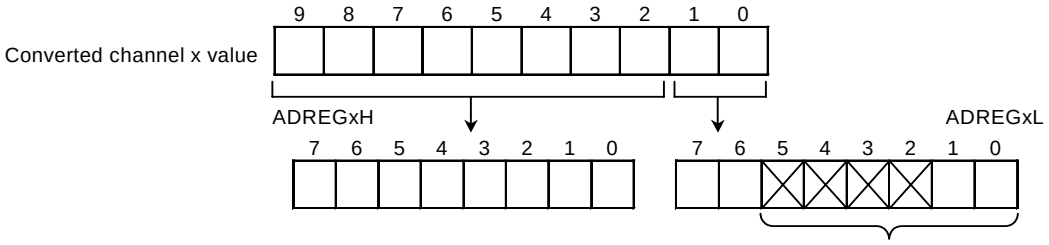
ADREGSPL (0xFFFF_F310)

	7	6	5	4	3	2	1	0
bit Symbol	ADRSP1	ADRSP0					OVRSP	ADRSPRF
Read/Write	R		R				R	R
After reset	0	0	1	1	1	1	0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register SP

ADREGSPH (0xFFFF_F311)

	7	6	5	4	3	2	1	0
bit Symbol	ADRSP9	ADRSP8	ADRSP7	ADRSP6	ADRSP5	ADRSP4	ADRSP3	ADRSP2
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 of register ADREGSPL are always "1."
- Bit 0 of register ADREGSPL is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) clears this bit to "0."
- Bit 1 of register ADREGSPL is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- **When reading conversion result storage registers, first read upper registers and then lower registers.**

Fig. 15.1.9 Registers related to the A/D Converter

Lower A/D Conversion Result Comparison Register

ADCOMREG L (0xFFFF_F312)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20						
Read/Write	R/W		R					
After reset	0	0	0	0	0	0	0	0
Function	Store lower 2 bits of A/D conversion result comparison		"0" is read.					

Upper A/D Conversion Result Comparison Register

ADCOMREGH (0xFFFF_F313)

	7	6	5	4	3	2	1	0
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Store upper 8 bits of A/D conversion result comparison							

(Note) To set or change a value in this register, the AD monitor function must be disabled (ADMOD3<ADOBSV>="0").

Fig. 15.1.10 Registers related to the A/D Converter

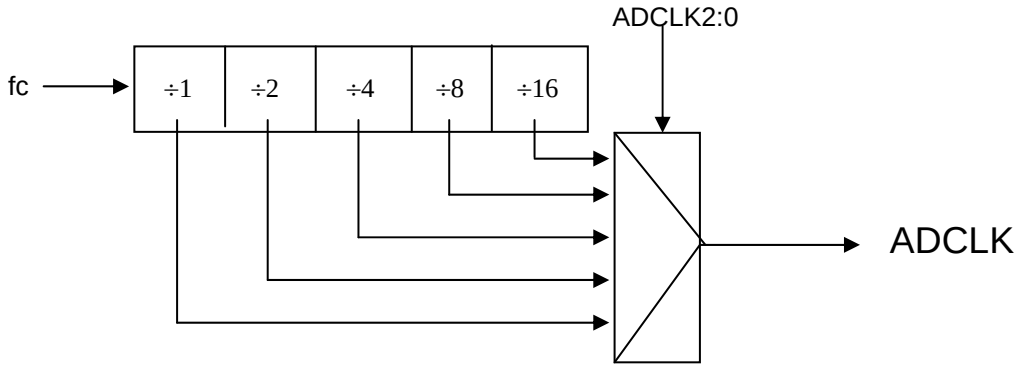
15.2 Conversion Clock

- The conversion time is calculated based on the 41 conversion clock and the sample hold time.

A/D Conversion Clock Setting Register

ADCLK (0xFFFF_F31C)

	7	6	5	4	3	2	1	0
bit Symbol		TSH2	TSH1	TSH0		ADCLK2	ADCLK1	ADCLK0
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function	Write "0."	Select the A/D sample hold time 000:12 conversion clock 001:12×2 conversion clock 010: 12×3 conversion clock 011: 12×4 conversion clock 100: 12×16 conversion clock 101: 12×64 conversion clock 110: 12×256 conversion clock 111: 12×1024 conversion clock				Select the A/D prescaler output 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 111:reserved		



Conversion clock	Sample hold time	tconv.
6.75 MHz	Conversion clk*12*1 (1.78 us)	7.85 us
	Conversion clk*12*2 (3.56 us)	9.63 us
	Conversion clk*12*3 (5.33 us)	11.4 us
	Conversion clk*12*4 (7.11 us)	13.2 us
	Conversion clk*12*16 (28.4 us)	34.5 us
	Conversion clk*12*64 (114 us)	120 us
	Conversion clk*12*256 (455 us)	461 us
	Conversion clk*12*1024 (1.82 ms)	1.83 ms

15.3 Description of Operations

15.3.1 Analog Reference Voltage

The "H" level of the analog reference voltage shall be applied to the VREFH pin, and the "L" level shall be applied to the VREFL pin. By writing "0" to the ADMOD1<VREFON> bit, a switched-on state of VREFH-VREFL can be turned into a switched-off state. To start AD conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

15.3.2 Selecting the Analog Input Channel

How the analog input channel is selected is different depending on A/D converter operation mode used.

(1) Normal AD conversion mode

- If the analog input channel is used in a fixed state (ADMOD0<SCAN>="0"):

  One channel is selected from analog input pins AIN0 through AIN23 by setting ADMOD1<ADCH4 to 0> to an appropriate setting.

- If the analog input channel is used in a scan state (ADMOD0<SCAN>="1"):

  One scan mode is selected from 24 scan modes by setting ADMOD1<ADCH4 to 0> and ADSCN to appropriate settings.

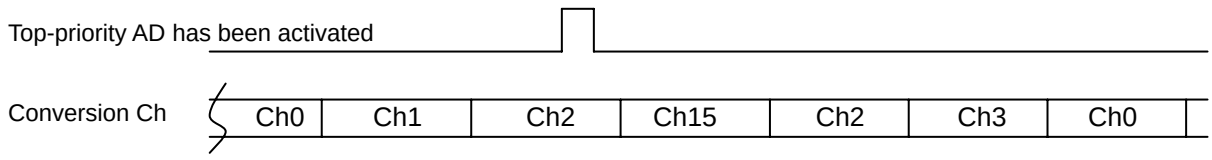
(2) High-priority AD conversion mode

One channel is selected from analog input pins AIN0 through AIN23 by setting ADMOD2<HPADCH4 to 0> to an appropriate setting.

After a reset, ADMOD0<SCAN> is initialized to "0" and ADMOD1<ADCH3:0> is initialized to "0000." This initialization works as a trigger to select a fixed channel input through the AN0 pin. The pins that are not used as analog input channels can be used as ordinary input ports.

If High-priority AD conversion is activated during normal AD conversion, normal AD conversion is discontinued, High-priority AD conversion is executed and completed, and then normal AD conversion is resumed.

Example: A case in which repeat-scan conversion is ongoing at channels AIN0 through AIN3 with ADMOD0<REPEAT:SCAN> set to "11" and ADMOD1<ADCH4:0> set to 00011, and High-priority AD conversion has been activated at AIN15 with ADMOD2<HPADCH4:0>=01111:



15.3.3 Starting A/D Conversion

Two types of A/D conversion are supported: normal AD conversion and High-priority AD conversion. Normal AD conversion is software activated by setting ADMOD0<ADS> to "1." High-priority AD conversion is software activated by setting ADMOD2<HPADCE> to "1." 4 operation modes are made available to normal AD conversion. In performing normal AD conversion, one of these operation modes must be selected by setting ADMOD0<2:1> to an appropriate setting. For High-priority AD conversion, only one operation mode can be used: fixed channel single conversion mode. Normal AD conversion can be activated using the HW activation source selected by ADMOD4<ADHS>, and High-priority AD conversion can be activated using the HW activation source selected by ADMOD4<HADHS>. If this bit is "0," normal AD conversion is activated in response to INTTB00 generated by the 16-bit timer 0, and High-priority AD conversion is activated in response to INTTB90 generated by the 16-bit timer 9. If this bit is "1," normal AD conversion is activated in response to INTTB01 generated by the 16-bit timer 0, and High-priority AD conversion is activated in response to INTTB91 generated by the 16-bit timer 9. Software activation is still valid even after H/W activation has been authorized.

When normal A/D conversion starts, the A/D conversion Busy flag (ADMOD0<ADBF>) showing that A/D conversion is under way is set to "1." When High-priority A/D conversion starts, the A/D conversion Busy flag (ADMOD2<ADBFHP>) showing that A/D conversion is under way is set to "1." If normal A/D conversion is interrupted by High-priority A/D conversion, the value of the Busy flag for normal A/D conversion before the start of High-priority A/D conversion is retained. The value of the conversion completion flag EOCFN for normal A/D conversion before the start of High-priority A/D conversion can also be retained.

(Note) Normal A/D conversion must not be activated when High-priority A/D conversion is under way. If activated when High-priority A/D conversion is under way, the High-priority A/D conversion completion flag cannot be set, and the flag for previous normal A/D conversion cannot be cleared.

To reactivate normal A/D conversion, a software reset (ADMOD4<ADRST1:0>) must be performed before starting A/D conversion. The HW activation method must not be used to reactivate normal A/D conversion.

If ADMOD2<HPADCE> is set to "1" during normal A/D conversion, ongoing A/D conversion is discontinued and High-priority A/D conversion starts; specifically, A/D conversion (fixed channel single conversion) is executed for a channel designated by ADMOD2<3:0>. After the result of this High-priority A/D conversion is stored in the storage register ADREGSP, normal A/D conversion is resumed.

If HW activation of High-priority A/D conversion is authorized during normal A/D conversion, ongoing A/D conversion is discontinued when requirements for activation using a resource are met, and High-priority A/D conversion (fixed channel single conversion) starts for a channel designated by ADMOD2<3:0>. After the result of this High-priority A/D conversion is stored in the storage register ADREGSP, normal A/D conversion is resumed.

15.3.4 A/D Conversion Modes and A/D Conversion Completion Interrupts

For A/D conversion, the following four operation modes are supported. For normal A/D conversion, an operation mode can be selected by setting ADMOD0<2:1> to an appropriate setting. For High-priority A/D conversion, the fixed channel single conversion mode is automatically selected, irrespective of the ADMOD0<2:1> setting.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

(1) Normal A/D conversion

An operation mode is selected with ADMOD0<REPEAT, SCAN>. As A/D conversion starts, ADMOD0<ADBFN> is set to "1." When specified A/D conversion is completed, the A/D conversion completion interrupt (INTAD) is generated, and ADMOD0<EOCF> showing the completion of A/D conversion is set to "1." If <REPEAT>="0," <ADBFN> returns to "0" concurrently with the setting of EOCF. If <REPEAT> is set to "1," <ADBFN> remains at "1" and A/D conversion continues.

① Fixed channel single conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "00," A/D conversion is performed in the fixed channel single conversion mode.

In this mode, A/D conversion is performed once for one channel selected. After A/D conversion is completed, ADMOD0<EOCF> is set to "1," ADMOD0<ADBF> is cleared to "0," and the interrupt request INTAD is generated. <EOCF> is cleared to "0" upon read.

② Channel scan single conversion mode

If ADMOD0 <REPET,SCAN> is set to "01," A/D conversion is performed in the channel scan single conversion mode.

In this mode, A/D conversion is performed once for each scan channel selected. After A/D scan conversion is completed, ADMOD0<EOCF> is set to "1," ADMOD0<ADBF> is cleared to "0," and the interrupt request INTAD is generated. <EOCF> is cleared to "0" upon read.

③ Fixed channel repeat conversion mode

If ADMOD0<REPEAT,SCAN> is set to "10," A/D conversion is performed in fixed channel repeat conversion mode.

In this mode, A/D conversion is performed repeatedly for one channel selected. After A/D conversion is completed, ADMOD <EOCF> is set to "1." ADMOD0 <ADBF> is not cleared to "0." It remains at "1." The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD0 <ITM1:0> to an appropriate setting. <EOCF> is set with the same timing as this interrupt INTAD is generated.

<EOCF> is cleared to "0" upon read.

With <ITM1:0> set to "00," an interrupt request is generated each time one A/D conversion is completed. In this case, the conversion results are always stored in the storage register ADREG08. After the conversion result is stored, EOCF changes to "1."

With <ITM1:0> set to "01," an interrupt request is generated each time four A/D conversion are completed. In this case, the conversion results are sequentially stored in storage registers ADREG08 through ADREG3B. After the conversion results are stored in ADREG3B, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADREG08. <EOCF> is cleared to "0" upon read.

With <ITM1:0> set to "10," an interrupt request is generated each time eight A/D conversions are completed. In this case, the conversion results are sequentially stored in storage registers ADREG08 through ADREG7F. After the conversion results are stored in ADREG7F, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADREG08.

<EOCF> is cleared to "0" upon read.

④ Channel scan repeat conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "11," A/D conversion is performed in the channel scan repeat conversion mode.

In this mode, A/D conversion is performed repeatedly for a scan channel selected. Each time one A/D scan conversion is completed, ADMOD0 <EOCF> is set to "1," and the interrupt request INTAD is generated. ADMOD0 <ADBF> is not cleared to "0." It remains at "1." <EOCF> is cleared to "0" upon read.

To stop the A/D conversion operation in the repeat conversion mode (modes described in ③ and ④ above), write "0" to ADMOD0 <REPEAT>. When ongoing A/D conversion is completed, the repeat conversion mode terminates, and ADMOD0 <ADBF> is set to "0."

(2) High-priority A/D conversion

High-priority A/D conversion is performed only in fixed channel single conversion mode. The ADMOD0<REPEAT, SCAN> setting has no relevance to the High-priority A/D conversion operations or preparations. As activation requirements are met, A/D conversion is performed only once for a channel designated by ADMOD2<HPADCH3:0>. After the A/D conversion is completed, the High-priority A/D conversion completion interrupt is generated, ADMOD2<EOCFHP> is set to "1," and <ADBFHP> returns to "0." The EOCFHP Flag is cleared upon read.

Relationships between A/D Conversion Modes, Interrupt Generation Timings and Flag Operations

Conversion mode	Interrupt generation timing	EOCF setting timing (see Note)	ADBF (after the interrupt is generated)	ADMOD0		
				ITM1:0	REPEAT	SCAN
Fixed channel single conversion	After conversion is completed	After conversion is completed	0	—	0	0
Fixed channel repeat conversion	Each time one conversion is completed	After one conversion is completed	1	00	1	0
	Each time four conversions are completed	After four conversions are completed	1	01		
	Each time eight conversions are completed	After eight conversions are completed	1	10		
Channel scan single conversion	After scan conversion is completed	After scan conversion is completed	1	—	0	1
Channel scan repeat conversion	Each time one scan conversion is completed	After one scan conversion is completed	1	—	1	1

(Note) EOCF is cleared upon read.

Fig. 15.3.4.1 Relationships between A/D Conversion Modes, Interrupt Generation Timings and Flag Operations

15.3.5 High-priority Conversion Mode

By interrupting ongoing normal A/D conversion, High-priority A/D conversion can be performed. High-priority A/D conversion can be software activated by setting ADMOD2<HPADCE> to "1" or it can be activated using the HW resource by setting ADMOD4<7:6> to an appropriate setting. If High-priority A/D conversion has been activated during normal A/D conversion, ongoing normal A/D conversion is interrupted, and single conversion is performed for a channel designated by ADMOD2<3:0>. The result of single conversion is stored in ADREGSP, and the High-priority A/D conversion interrupt is generated. After High-priority A/D conversion is completed, normal A/D conversion is resumed; the status of normal A/D conversion immediately before being interrupted is maintained. High-priority A/D conversion activated while High-priority A/D conversion is under way is ignored.

For example, if channel repeat conversion is activated for channels AN0 through AN8 and if <HPADCE> is set to "1" during AN3 conversion, AN3 conversion is suspended, and conversion is performed for a channel designated by <HPADC3:0>. After the result of conversion is stored in ADREGSP, channel repeat conversion is resumed, starting from AN3.

15.3.6 A/D Monitor Function

If ADMOD3<ADOBSV> is set to "1," the A/D monitor function is enabled. If the value of the conversion result storage register specified by REGS<3:0> becomes larger or smaller ("larger" or "smaller" to be designated by ADOBIC) than the value of a comparison register, the A/D monitor function interrupt is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result storage register, and the interrupt is generated if the conditions are met. Because storage registers assigned to perform the A/D monitor function are usually not read by software, overrun flag <OVRn> is always set and the conversion result storage flag <ADRnRF> is also set. To use the A/D monitor function, therefore, a flag of a corresponding conversion result storage register must not be used.

15.3.7 A/D Conversion Time

By setting ADCLK<ADCLK2:0> to an appropriate setting, one A/D conversion clock can be selected for fc, fc/2, fc/4, fc/8 and fc/16 (AD prescaler outputs). To achieve the guaranteed accuracy, the A/D conversion clock must be 6.75 MHz or less, that is, the A/D conversion time must be 7.85 μs or longer.

15.3.8 Storing and Reading A/D Conversion Results

A/D conversion results are stored in upper and lower A/D conversion result registers for normal A/D conversion (ADREG08H/L through ADRG7FH/L).

In fixed channel repeat conversion mode, A/D conversion results are sequentially stored in ADREG08H/L through ADREG7FH/L. If <ITM1:0> is so set as to generate the interrupt each time one A/D conversion is completed, conversion results are stored only in ADREG08H/L. If <ITM1:0> is so set as to generate the interrupt each time four A/D conversions are completed, conversion results are sequentially stored in ADREG08H/L through ADREG3BH/L.

Table 15.3.8.1 shows analog input channels and related A/D conversion result registers.

Table 15.3.8.1 Analog Input Channels and Related A/D Conversion Result Registers

Analog input channel	A/D conversion result register			
	Conversion modes other than shown to the right	Fixed channel repeat conversion mode (every one conversion)	Fixed channel repeat conversion mode (every four conversions)	Fixed channel repeat conversion mode (every eight conversions)
AN0	ADREG08H/L	ADREG08H/L fixed	ADREG08H/L ↔ ADREG3BH/L	ADREG08H/L ↔ ADREG7FH/L
AN1	ADREG19H/L			
AN2	ADREG2AH/L			
AN3	ADREG3BH/L			
AN4	ADREG4CH/L			
AN5	ADREG5DH/L			
AN6	ADREG6EH/L			
AN7	ADREG7FH/L			
AN8	ADREG08H/L			
AN9	ADREG19H/L			
AN10	ADREG2AH/L			
AN11	ADREG3BH/L			
AN12	ADREG4CH/L			
AN13	ADREG5DH/L			
AN14	ADREG6EH/L			
AN15	ADREG7FH/L			
AN16	ADREG08H/L			
AN17	ADREG19H/L			
AN18	ADREG2AH/L			
AN19	ADREG3BH/L			
AN20	ADREG4CH/L			
AN21	ADREG5DH/L			
AN22	ADREG6EH/L			
AN23	ADREG7FH/L			

15.3.9 Data Polling

To process A/D conversion results without using interrupts, ADMOD0<EOCF> must be polled. If this flag is set, conversion results are stored in a specified A/D conversion result register. After confirming that this flag is set, read that conversion result storage register. In reading the register, make sure that you first read upper bits and then lower bits to detect an overrun. If OVRn is "0" and ADRnRF is "1" in lower bits, a correct conversion result has been obtained.

16. Watchdog Timer (Runaway Detection Timer)

The TMP19A64 has a built-in watchdog timer for detecting runaways.

The watchdog timer (WDT) is for detecting malfunctions (runaways) of the CPU caused by noises or other disturbances and remedying them to return the CPU to normal operation. If the timer detects a runaway, it generates a non-maskable interrupt to notify the CPU.

By connecting the output of the watchdog timer to a reset pin (inside the chip), it is possible to force the watchdog timer to reset itself.

16.1 Configuration

Fig. 16.1 shows the block diagram of the watchdog timer.

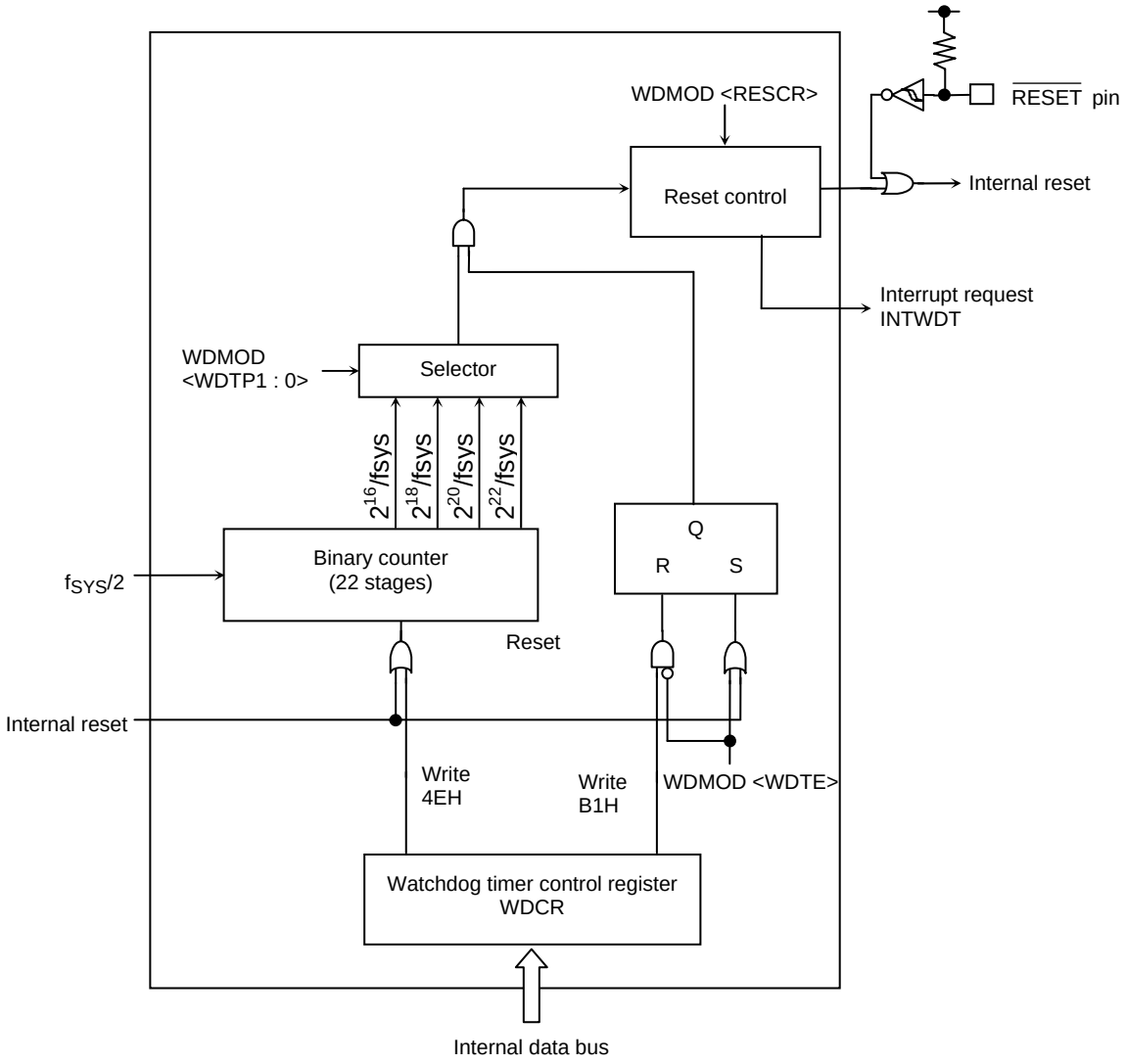


Fig. 16.1 Block Diagram of the Watchdog Timer

16.2 Watchdog Timer Interrupt

The watchdog timer consists of the binary counters that are arranged in 22 stages and work using the $f_{SYS/2}$ system clock as an input clock. The outputs produced by these binary counters are 2^{15}, 2^{17}, 2^{19} and 2^{21}. By selecting one of these outputs with WDMOD <WDTP1:0>, a watchdog timer interrupt can be generated when an overflow occurs, as shown in Fig. 16.2.1.

Because the watchdog timer interrupt is a non-maskable interrupt factor, NMIFLG <WDT> at the INTC performs a task of identifying it.

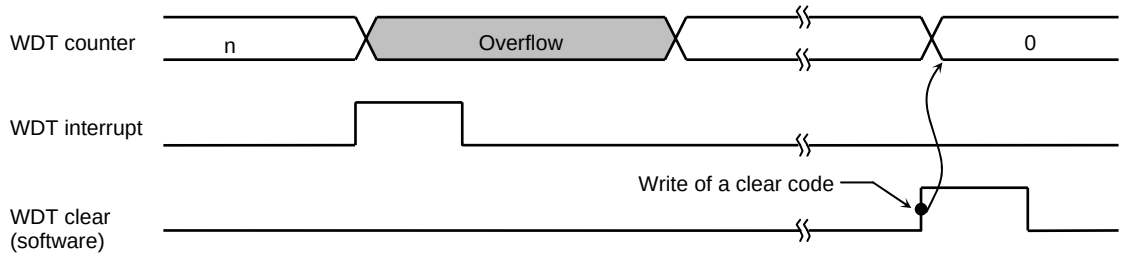


Fig. 16.2.1 Normal Mode

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is effected for a 32-state time, as shown in Fig. 16.2.2. If this reset is effected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 8 is used as an input clock $f_{SYS/2}$.

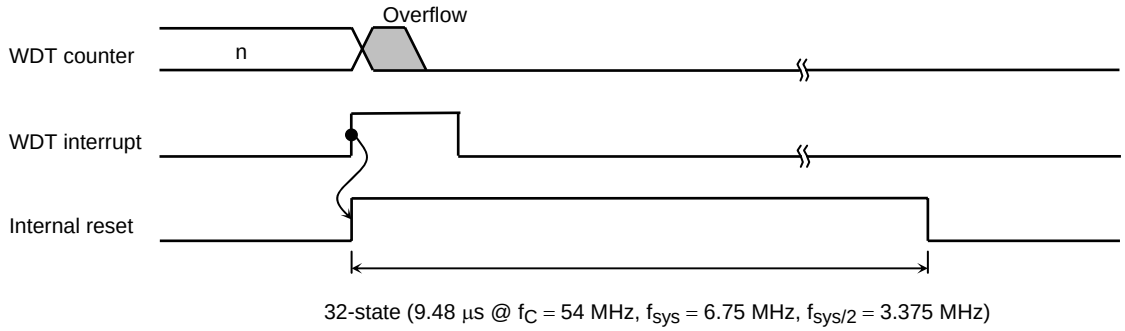


Fig. 16.2.2 Reset Mode

(Note 1) When the watchdog timer functions to effect a reset, sampling of the status of the $\overline{\text{PLLOFF}}$ pin still continues. Therefore, use the $\overline{\text{PLLOFF}}$ pin at the level fixed to "H."

(Note 2) If the watchdog timer is operated when the high-frequency oscillator is idle, the system reset operation initiated by the watchdog timer becomes erratic due to the unstable oscillation of the high-frequency oscillator. Therefore, do not operate the watchdog timer when the high-frequency oscillator is idle.

16.3 Control Registers

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

16.3.1 Watchdog Timer Mode Register (WDMOD)

① Specifying the detection time of the watchdog timer <WDTP1: 0>

This is a 2-bit register for specifying the watchdog timer interrupt time for runaway detection. When a reset is effected, this register is initialized to WDMOD <WDTP1, 0> = "00." Fig. 16.3.1.1 shows the detection time of the watchdog timer.

② Enabling/disabling the watchdog timer <WDTE>

When reset, WDMOD <WDTE> is initialized to "1" and the watchdog timer is enabled.

To disable the watchdog timer, this bit must be set to "0" and, at the same time, the disable code (B1H) must be written to the WDCR register. This dual setting is intended to minimize the probability that the watchdog timer may inadvertently be disabled if a runaway occurs.

To change the status of the watchdog timer from "disable" to "enable," set the <WDTE> bit to "1."

③ Watchdog timer out reset connection <RESCR>

This register is used to make a non-maskable interrupt (INTWDT) setting associated with the detection of a runaway or to make a connection setting after an internal reset. After a reset, WDMOD <RESCR> is initialized to "0," and a non-maskable interrupt setting is established. For information on the status of non-maskable interrupts, refer to the NMIFLG register which is described in Chapter 6 "Interrupts."

WDMOD (0xFFFF_F090)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	—
Read/Write	R/W	R/W		R	R	R/W		R/W
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0: Disable 1: Enable	Selects WDT detection time 00: $2^{16}/f_{SYS}$ 01: $2^{18}/f_{SYS}$ 10: $2^{20}/f_{SYS}$ 11: $2^{22}/f_{SYS}$				IDLE 0: Stop 1: Start	Selects internal RESET 0: Connects NMI 1: Connects internal reset	Write "0."

Watchdog timer out control

0	NMI interrupt
1	Connects WDT out to internal reset

Detection time of watchdog timer @ fc = 54 MHz

SYSCR1 clock gear value <GEAR2:0>	Detection Time of Watchdog Timer			
	WDMOD<WDTP1, 0>			
	00	01	10	11
000 (fc)	1.2 ms	4.9 ms	19.4 ms	77.7 ms
100 (fc/2)	2.4 ms	9.7 ms	38.8 ms	155.3 ms
110 (fc/4)	4.9 ms	19.4 ms	77.7 ms	310.7 ms
111 (fc/8)	9.7 ms	38.8 ms	155.3 ms	621.4 ms

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Fig. 16.3.1.1 Watchdog Timer Mode Register

16.3.2 Watchdog Timer Control Register (WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

- Disabling control

  By writing the disable code (B1H) to this WDCR register after setting WDMOD <WDTE> to "0," the watchdog timer can be disabled.

```
WDMOD  ← 0 _ _ _ _ _ _ _   Clears WDTE to "0."
WDCR   ← 1 0 1 1 0 0 0 1   Writes the disable code (B1H).
```


- Enabling control

  Set WDMOD <WDTE> to "1."

- Watchdog timer clearing control

  Writing the clear code (4EH) to the WDCR register clears the binary counter and allows it to resume counting.

```
WDCR   ← 0 1 0 0 1 1 1 0   Writes the clear code (4EH).
```

(Note) Writing the disable code (BIH) clears the binary counter.

WDCR (0xFFFF_F091)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	—							
Function	B1H : WDT disable code 4EH : WDT clear code Others: Disabled This is a write-only register. If each bit is read, "0" is returned.							

→ Disable & clear of WDT

B1H	Disable code
4EH	Clear code
Others	—

Fig. 16.3.2.1 Watchdog Timer Control Register

16.4 Operation Description

The watchdog timer generates the INTWD interrupt after a lapse of the detection time specified by the WDMOD <WDTP1, 0> register. Before generating the INTWD interrupt, the binary counter for the watchdog timer must be cleared to "0" using software (instruction). If the CPU malfunctions (runs away) due to noise or other disturbances and cannot execute the instruction to clear the binary counter, the binary counter overflows and the INTWD interrupt is generated. The CPU is able to recognize the occurrence of a malfunction (runaway) by identifying the INTWD interrupt and to restore the faulty condition to normal by using a malfunction (runaway) countermeasure program. Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

The watchdog timer begins operation immediately after a reset is cleared.

In STOP mode, the watchdog timer is reset and in an idle state. When the bus is open ($\overline{\text{BUSAK}}$ = "L"), it continues counting. In IDLE mode, its operation depends on the WDMOD <I2WDT> setting. Before putting it in IDLE mode, WDMOD <I2WDT> must be set to an appropriate setting, as required.

Examples:

① To clear the binary counter

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Writes the clear code (4EH)

② To set the detection time of the watchdog timer to $2^{18}/f_{SYS}$

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	1	–	–	–	–	–

③ To disable the watchdog timer

		7	6	5	4	3	2	1	0	
WDMOD	←	0	–	–	–	–	–	–	–	Clears WDTE to "0"
WDCR	←	1	0	1	1	0	0	0	1	Writes the disable code (B1H)

Note: If the watchdog timer is operated when the high-frequency oscillator is idle, the system reset operation initiated by the watchdog timer becomes erratic due to the unstable oscillation of the high-frequency oscillator. Therefore, do not operate the watchdog timer when the high-frequency oscillator is idle.

17. Backup Module (Clock Timer, Backup RAM)

17.1 Features

The TMP19A64 has a backup module (backup mode) with a built-in timer dedicated to clock operations and a built-in backup RAM. Using this backup module, the TMP19A64 can operate in low-power-consumption operation modes. Specifically, power to all blocks (CPU, peripheral I/Os, etc.) except the backup module is disconnected; because only the backup module is supplied with power, it is possible to reduce the amount of consumption current greatly.

17.2 Block Diagram

Fig. 17.2 shows the block diagram of the backup module.

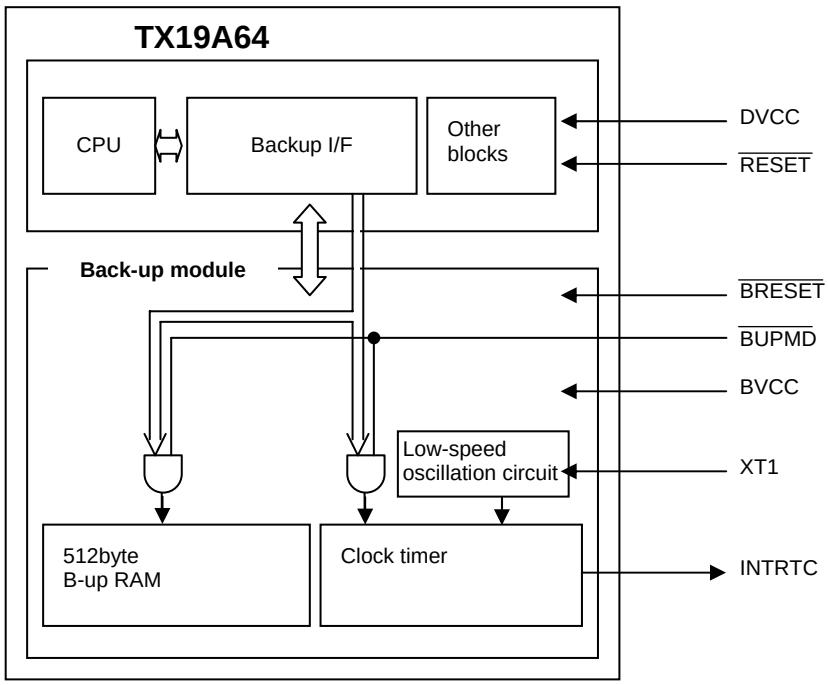


Fig. 17.2 Block Diagram of the Backup Module

Precautions for the use of the backup module:

- Low-speed oscillation starts when the backup module (BVCC) is powered on. The software start or stop of low-speed oscillation is not permitted.
- To put the TMP19A64 in backup mode or normal operation mode, necessary settings must be made.
- When the backup module is operating in SLOW mode, access to the backup RAM is prohibited.
- The functions that can be initialized with $\overline{\text{BRESET}}$ are as follows:

  Clock timer: Initialize

  Backup RAM: Undefined

  Backup module reset flag: Initialize

  Registers in the backup module (RTCFLG, RTCCR, RTCREG)

  Low-speed oscillator: Continued oscillation

- If the backup module and the low-frequency oscillator are not used, the following settings must be made:

  Power supply level: BVCC, $\overline{\text{BRESET}}$

  GND level: XT1, $\overline{\text{BUPMD}}$

17.3 Backup Mode

A backup mode is provided as a system operation mode. In backup mode, the power to all blocks except the backup module is disconnected so that the TMP19A64 can operate with low power consumption.

Fig. 17.3 is the state transition diagram showing a transition to the backup mode.

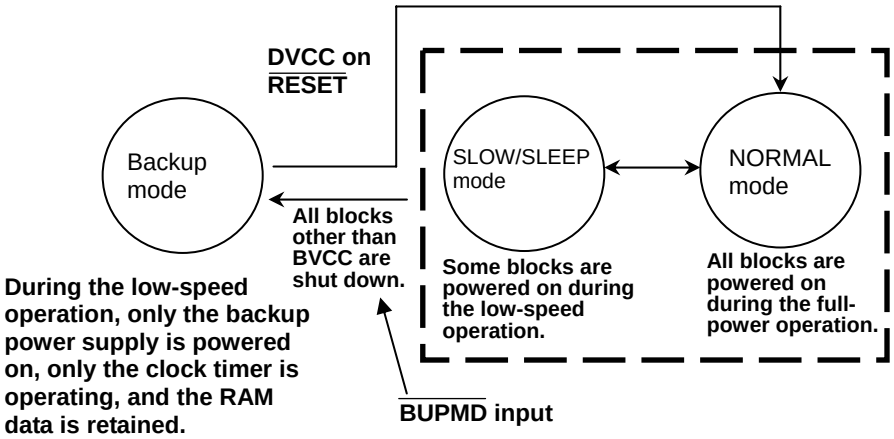
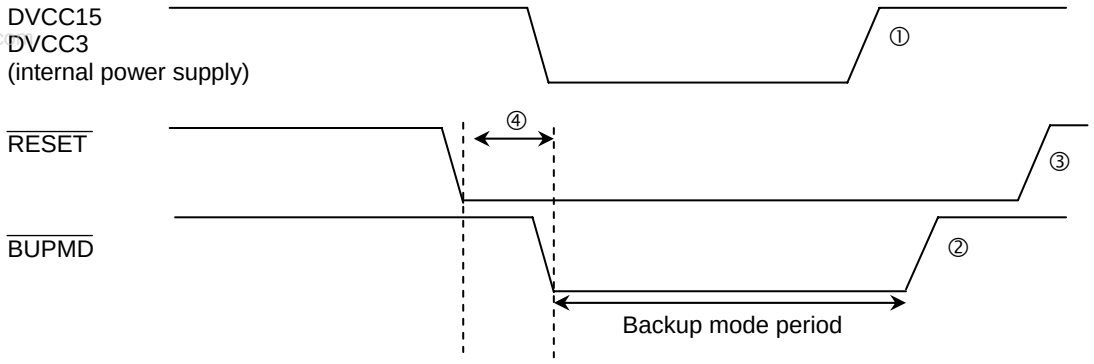


Fig. 17.3 Block Diagram of the Backup Module

17.4 Backup Mode Operation

17.4.1 Transition to Backup Mode

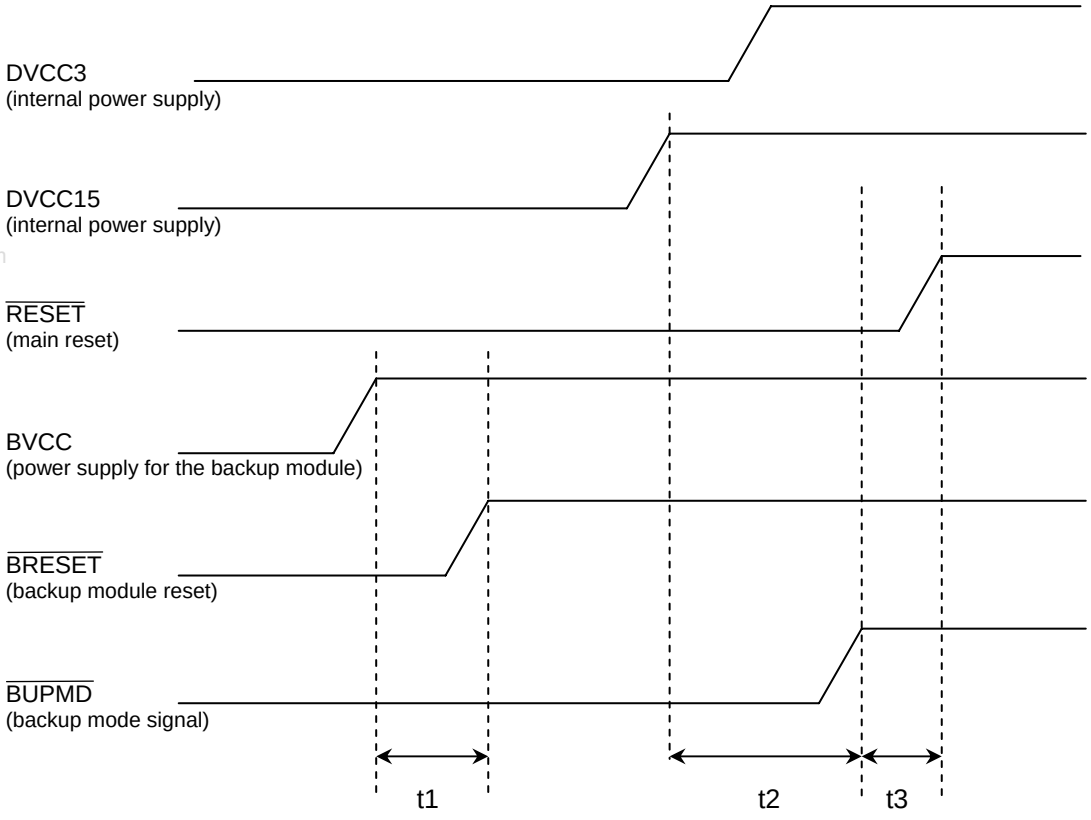
To put the TMP19A64 into backup mode, first set the backup mode trigger pin (BUPMD) to "0," and then cut off the main power supply (DVCC3, DVCC15). When performing these two steps, caution must be used because there is the possibility that data is being written to the backup RAM. Therefore, steps must be performed according to the sequence shown below. Additionally, to recover from backup mode, the power must be turned on and signals must be processed according to the sequence shown below.



- To recover from backup mode, steps ①, ② and ③ must be performed in this order.
- If data is being written to the backup RAM in the backup module, the period (4) must be more than 50 clocks (1 µsec (@54 MHz)) in order to guarantee the integrity of data.

17.4.2 Power-on (Recovery from Backup Mode)

Example: If the DVCC15 power and the BVCC power are activated with different timings



- t1: As BVCC stabilizes, $\overline{\text{BRESET}}$ is maintained at "L" for more than 2 ms*. (* This time length differs depending on the characteristics of the oscillator.)
- t2: $\overline{\text{BUPMD}}$ is set to "H" after a lapse of the warming-up time for the high-speed oscillator.
- t3: $\overline{\text{RESET}}$ is cleared after the level of $\overline{\text{BUPMD}}$ changes to "H." (The backup module is initialized according to the initial routine.)

<Precautions for the transition from normal operation mode to STOP mode>

Even if the instruction to move to STOP mode has been executed, low-speed oscillation continues as long as BVCC (power supply for the backup module) is supplying power. Therefore, after the instruction to move to STOP mode is executed, BVCC must be shut down. To recover from STOP mode, first start BVCC, $\overline{\text{BRESET}}$ and $\overline{\text{BUPMD}}$ in the same sequence as they are powered on, and then clear STOP mode.

17.5 Backup RAM

17.5.1 Features

The backup module has a built-in backup RAM (512 bytes) to be used when the TMP19A64 operates in low-power-consumption operation mode. This RAM holds data when the TMP19A64 is operating in backup mode. The data held in the RAM remains intact even if a reset is executed.

- Backup RAM area (512 bytes): 0xFFFF_E800 through 0xFFFF_E9FF
- Data in the backup RAM area is retained when the TMP19A64 is operating in backup mode.
- The data held in the backup RAM area is retained even if a reset (/RESET) is executed.
- The /BRESET pin is used to initialize (undefined value) the backup RAM area.

Note: Concerning the access to the backup RAM area for a read or write, a time length equal to 10 system clocks is required to process one such access.

17.6 Clock Timer

17.6.1 Features

The backup module has a built-in clock timer to be used when the TMP19A64 operates in low-power consumption operation mode. This clock timer using 32.768 kHz as a low clock frequency can generate interrupts at time intervals of 0.125s, 0.250s, 0.500s and 1.000s so that the TMP19A64 is able to use the clock function when operating in low-power-consumption operation modes.

This clock timer can be operated in all operation modes of low-frequency oscillation. The interrupt generated by the clock timer allows the TMP19A64 to recover from standby mode (except STOP mode). To use the clock timer interrupt (INTRTC), the IMCGD register in the CG must be set to an appropriate setting.

Fig. 17.6.1 shows the block diagram of the clock timer.

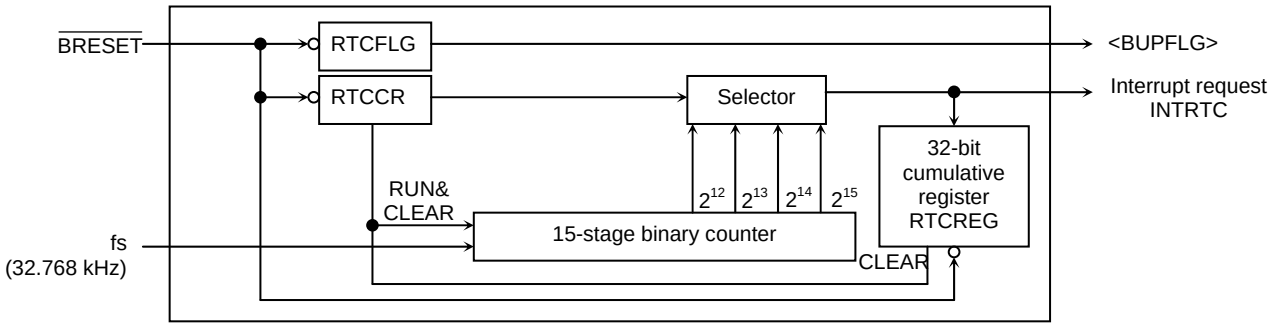


Fig. 17.6.1 Block Diagram of the Clock Timer

17.6.2 Registers

The clock timer is controlled by the clock timer control register (RTCCR), backup mode flag register (RTCFLG), and clock timer count cumulative register (RTCREG). These registers are the 32-bit registers that can be initialized by /BRESET.

Fig. 17.6.2.1 shows the clock timer control register.

(fs = 32.768 kHz)

RTCCR (0xFFFF_E704)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After BRESET	0	0	0	0	0	0	0	0
Function								

	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After BRESET	0	0	0	0	0	0	0	0
Function								

	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After BRESET	0	0	0	0	0	0	0	0
Function								

	7	6	5	4	3	2	1	0
Bit Symbol					RTCRCLR	RTCSEL1	RTCSEL0	RTCRUN
Read/Write	R/W	R/W	R		W	R/W		R/W
After BRESET	0	0	0	0	0	0	0	0
Function	Write "0."	Write "0."			Clear cumulative register 0: Clear 1: Don't Care	Interrupt generation cycle 00: 2^{15}/fs (1.000 s) 01: 2^{14}/fs (0.500 s) 10: 2^{13}/fs (0.250 s) 11: 2^{12}/fs (0.125 s)		Binary counter 0: Stop & clear 1: Count

Fig. 17.6.2.1 Clock Timer Control Register

(Note) To access this register, 32-bit access is required.
(Note) Values read from the registers are undefined until /BRESET is activated.
(Note) Values read from RTCCR<RTCRCLR> are always "1."
(Note) Before changing the RTCCR<RTCSSEL1:0> setting, make sure that RTCCR<RTCRUN> is "0" and that the RTC interrupt is disabled.

The backup mode flag register RTCFLG is a 32-bit register that has the <BUPFLG> bit for monitoring the activation of /BRESET and can be initialized by /BRESET. By writing "1" to the <BUPFLG> bit after /BRESET when starting the backup module, this register can be used as a /BRESET activation monitor.

Fig. 17.6.2.2 shows the clock timer control register.

RTCFLG (0xFFFF_E700)

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After BRESET	0	0	0	0	0	0	0	0
Function	See Note							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After BRESET	0	0	0	0	0	0	0	0
Function	See Note							
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After BRESET	0	0	0	0	0	0	0	0
Function	See Note							
	7	6	5	4	3	2	1	0
Bit Symbol								BUPFLG
Read/Write	R							R/W
After BRESET	0	0	0	0	0	0	0	0
Function	See Note							BRESET Monitor flag 0: After BRESET See Notes

Fig. 17.6.2.2 Backup Mode Flag Register

(Note) Values read from this register are undefined until /BRESET is activated.
(Note) For this register, 32-bit access is required.
(Note) Only "1" can be written to the <BUPFLG> bit.
(Note) After /BRESET, the <BUPFLG> bit changes to "0." Therefore, this register can be used as a /BRESET activation monitor by writing "1" after /BRESET when starting the backup module.

The clock timer is provided with a clock count cumulative register (RTCREG) for counting the number of times interrupts are generated. If 1.0s is selected as an interrupt generation cycle, a maximum of 4294967296 seconds can be retained (136 years, 70 days, 6 hours, 28 minutes, and 16 seconds).

Clock Count Cumulative Register

RTCREG
(0xFFFF_E708)

	31	30	29	28	27	26	25	24
Bit Symbol	RUI31	RUI30	RUI29	RUI28	RUI27	RUI26	RUI25	RUI24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							
	23	22	21	20	19	18	17	16
Bit Symbol	RUI23	RUI22	RUI21	RUI20	RUI19	RUI18	RUI17	RUI16
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							
	15	14	13	12	11	10	9	8
Bit Symbol	RUI15	RUI14	RUI13	RUI12	RUI11	RUI10	RUI9	RUI8
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							
	7	6	5	4	3	2	1	0
Bit Symbol	RUI7	RUI6	RUI5	RUI4	RUI3	RUI2	RUI1	RUI0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							

Fig. 17.6.2.3 Clock Count Cumulative Register

(Note) Values read from this register are undefined until /BRESET is activated.
(Note) To access this register, 32-bit access is required.
(Note) A write to this cumulative register clears the prescaler.
(Note) Interrupts must be disabled during a read.

Example of the clock timer interrupt setting:

Initialization

		7	6	5	4	3	2	1	0	
IMCD	←	0	0	1	0	0	0	0	0	Disables the interrupt INTRTC Sets the bit <23:16> of a 32-bit register
RTCCR	←	0	0	0	0	X	X	X	0	Stops the RTC timer count Sets the bit <7:0> of a 32-bit register
IMCGD	←	0	0	1	1	0	0	0	1	Sets the bit <15:8> of a 32-bit register
EICRCG	←	0	0	0	0	1	1	0	1	Clears the interrupt request for the CG block Set the bit <7:0> of a 32-bit register
INTCLR	←	0	1	1	1	1	0	0	0	Clears the interrupt request for the INTC block Sets the bit <8:0> of a 32-bit register
RTCCR	←	0	0	0	0	1	X	X	1	Starts the timer count Sets the bit <7:0> of a 32-bit register
IMCD	←	0	0	1	0	0	X	X	X	Sets the interrupt level Set the bit <23:16> of a 32-bit register

INTRTC interrupt

		7	6	5	4	3	2	1	0	
EICRCG	←	0	0	0	0	1	1	0	1	Clears the interrupt request for the CG block Sets the bit <7:0> of a 32-bit register
INTCLR	←	0	1	1	1	1	0	0	0	Clears the interrupt request for the INTC block Sets the bit <8:0> of a 32-bit register

Processing
Interruption finished

(Note 1) X means "don't care."
(Note 2) To disable the interrupt generated in standby mode, IMCD must be first set and then IMCGD.

18. Key-on Wakeup

18.1 Outline

- The TMP19A64 has 8 key inputs, KEY0 to KEY7, which can be used for releasing the STOP/SLEEP mode or for external interrupts. Note that interrupt processing is executed with one interrupt factor for the 8 inputs. Each key input can be configured to be used or not, by programming (KWUPSTn).
- The active state of each input can be configured to the rising edge, the falling edge, the high level or the low level, by programming (KWUPSTn).
- An interrupt request is cleared by reading the key interrupt state register KWUPST in the interrupt processing.
- The key input pins have pull-up functions, which can be enabled or disabled by programming the key pull-up control register KUPPUP.

18.2 Key-on Wakeup Operation

The TMP19A64 has 8 key input pins, KEY0 to KEY7. Program the IMCGC0<KWUPEN> register in the CG to determine whether to use the key inputs for releasing the STOP/SLEEP mode or for normal interrupts. Setting <KWUPEN> to "1" causes all the key inputs, KEY0 to KEY7, to be used for interrupts for releasing the STOP/SLEEP mode. Program KWUPSTn<KEYnEN> to enable or disable interrupt inputs for each key input pin. Also, program KWUPSTn<KEYn1: KEYn0> to define the active state of each key input pin to be used. Detection of key inputs is carried out in the KWUP block, and the detection results are notified to the IMCGD register in the CG as the active high level. Therefore, program IMCGD<EMCGC1:C0> to "01" to determine the detection level to the high level. The results of detection in the CG are also notified to the interrupt controller INTC as the active high level. Therefore, program the INTC to "01" to define the corresponding interrupt as the high level. Setting IMCGD<KWUPEN> to 0 (default) configures all the input pins, KEY0 to KEY7 to the normal interrupts. In this case, you don't have to make settings at the CG, but just specify the INTC detection level to the high level. Program KWUPSTn in the same way to enable or disable each key input and define their active states. Reading KWUPST during interrupt processing clears the generated key interrupt requests.

(Note) If two or more key inputs are generated, the interrupt requests, which have been generated before the sequence of clearing the interrupt requests carried out in the interrupt processing routine that corresponds to the first key input, will be cleared at the same time. Key interrupts are generated again for the interrupt requests that are generated after the said sequence of clearing the interrupt requests.

18.3 Pull-up Function

Each key input has the pull-up function. Pull-up can be enabled for each bit of key inputs KEY0 to KEY7 by setting KUPPUP<KEYPUP0:7> to "1." The pull-up function does not work for the key inputs that are disabled at KWUPSTn<KEYnEN>, independently of the KUPPUP<KEYPUP> setting.

Cautions on use of key inputs with pull-up enabled

A) When you make the first setting after turning the power ON
   1) Set KUPPUP (<KEYPUPn> ="1").
   2) Set KWUPSTn<KEYnEN> to "1" for the KEYn input to be used.
   3) Wait until the pull-up operation is completed.
   4) Set KWUPSTn to define the active state of the KEYn input to be used.
   5) Clear interrupt requests by reading KWUPST.
   6) Set CG and the INTC. (Refer to Chapter 6, "Interrupt Settings" for the details of setting methods.)

B) To change the active state of a key input during operation
   1) Disable key interrupts by setting IMC3<ILD2:D0> to "000" at the INTC.
   2) Change the active state by setting KWUPSTn for the KEYn input to be changed.
   3) Clear interrupt requests by reading KWUPST.
   4) Enable the key interrupt at the INTC. (Set IMC3<ILD2:D0> to a desired level.)

C) To enable a key input during operation
   1) Disable key interrupts by setting IMC3<ILD2:D0> to "000" at the INTC.
   2) Set KWUPSTn<KEYnEN> to "1" for the key input to be used.
   3) Wait until the pull-up operation is completed.
   4) Define the active state of the key input to be used at the corresponding KWUPSTn.
   5) Clear interrupt requests by reading KWUPST.
   6) Enable key interrupts at the INTC. (Set IMC3< ILD2:D0> to a desired level.)

Cautions on use of key inputs with pull-up disabled

A) When you make the first setting after turning the power ON
   1) Set KUPPUP (<KEYPUPn> ="0")
   2) Set KWUPSTn to define the active state of the KEYn input to be used.
   3) Clear interrupt requests by reading KWUPST.
   4) Set KWUPSTn<KEYnEN> to "1" for the KEYn input to be used.
   5) Set CG and the INTC. (Refer to Chapter 6, "Interrupt Settings" for the details of setting methods.)

B) To change the active state of a key input during operation
   1) Disable key interrupts by setting IMC3<ILD2:D0> to "000" at the INTC.
   2) Change the active state by setting KWUPSTn for the key input to be changed.
   3) Clear interrupt requests by reading KWUPST.
   4) Enable key interrupts at the INTC. (Set IMC3< ILD2:D0> to a desired level.)

C) To enable a key input during operation
   1) Disable key interrupts by setting IMC3<ILD2:D0> to "000" at the INTC.
   2) Define the active state by setting KWUPSTn for the key input to be used.
   3) Clear interrupt requests by reading KWUPST.
   4) Set KWUPSTn<KEYnEN> to "1" for the key input to be used.
   5) Enable key interrupts at the INTC. (Set IMC3<ILD2:D0> to a desired level.)

Key pull-up control register: KUPPUP

KUPPUP (0xFFFF_F371)

	7	6	5	4	3	2	1	0
bit Symbol	KEYPUP7	KEYPUP6	KEYPUP5	KEYPUP4	KEYPUP3	KEYPUP2	KEYPUP1	KEYPUP0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled	0: Pull-up disabled 1: Pull-up enabled

18.4 Key Input Detection

1) <KEYPUPn> Pull-up disabled/enabled

The active state of each KEYn input can be defined to the high or low level or to the rising and/or falling edges by setting KWUPSTn<KEYn1:0>. The active states of KEYn inputs are continuously detected.

		7	6	5	4	3	2	1	0
KWUPST0	bit Symbol			KEY01	KEY00				KEY0EN
(0xFFFF_F360)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY0 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY0 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST1	bit Symbol			KEY11	KEY10				KEY1EN
(0xFFFF_F361)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY1 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY1 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST2	bit Symbol			KEY21	KEY20				KEY2EN
(0xFFFF_F362)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY2 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY2 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST3	bit Symbol			KEY31	KEY30				KEY3EN
(0xFFFF_F363)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY3 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY3 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST4	bit Symbol			KEY41	KEY40				KEY4EN
(0xFFFF_F364)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY4 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY4 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST5	bit Symbol			KEY51	KEY50				KEY5EN
(0xFFFF_F365)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY5 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY5 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST6	bit Symbol			KEY61	KEY60				KEY6EN
(0xFFFF_F366)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY6 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY6 interrupt input 0: Disable 1: Enable
		7	6	5	4	3	2	1	0
KWUPST7	bit Symbol			KEY71	KEY70				KEY7EN
(0xFFFF_F367)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY7 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY7 interrupt input 0: Disable 1: Enable

18.5 Detection of Key Input Interrupts and Clearance of Requests

When KEYnEN is set to 1 and an active signal is input to KEYn, the KEYINTn channel that corresponds to KWUPST is set to "1," indicating that an interrupt is generated. The KWUPST is the read-only register. Reading this register clears the corresponding bit that has been set to "1."

If the active state is set to the high or low level, the corresponding bit of the KWUPST register remains "1" after it is read, unless the external input is withdrawn.

KEY interrupt state register: KWUPST

KWUPST (0xFFFF_F370)

	7	6	5	4	3	2	1	0
bit Symbol	KEYINT7	KEYINT6	KEYINT5	KEYINT4	KEYINT3	KEYINT2	KEYINT1	KEYINT0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	KEY7 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY6 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY5 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY4 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY3 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY2 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY1 interrupt state 0: No interrupt generated 1: Interrupt generated	KEY0 interrupt state 0: No interrupt generated 1: Interrupt generated

19. ROM Correction Function

This chapter describes the ROM correction function built into the TMP19A64.

19.1 Features

- Using this function, eight pieces of one-word data or four pieces of eight-word data can be replaced.
- If an address (lower 5 or 2 bits are "don't care" bits) written to the address register matches an address generated by the PC or DMAC, ROM data is replaced by data generated by the ROM correction data register which is established in a RAM area assigned to the above address register.
- ROM correction is automatically authorized by writing an address to each address register.

19.2 Description of Operations

By setting in the address register ADDREGn a physical address (including a projection area) of the ROM area to be corrected, ROM data can be replaced by data generated by a data register in a RAM area assigned to ADDREGn. The ROM correction function is automatically enabled when an address is set in ADDREGn, and it cannot be disabled. After a reset, the ROM correction function is disabled. Therefore, to execute ROM correction with the initial setting after a reset is cleared, it is necessary to set an address in ADDREG. As an address is set in ADDREG, the ROM correction function is enabled for this register. If the CPU has the bus right, ROM data is replaced when the value generated by the PC matches that of the address register. If the DMAC has the bus right, ROM data is replaced when a source or destination address generated by the DMAC matches the value of the address register. For example, if an address is set in ADDREG0 and ADDREG3, the ROM correction function is enabled for this area; match detection is performed on these registers, and data replacement is executed if there is a match. Data replacement is not executed for ADDREG1, ADDREG2, and ADDREG4 through ADDREG7. Although the bit <31:5> exists in address registers, match detection is performed on A<20:5>. Internal processing is that data replacement is executed by multiplying the ROMCS signal showing a ROM area by the result of a match detection operation performed by ROM correction circuitry.

If eight-word data is replaced, an address for ROM correction can be established only on an eight-word boundary, and data is replaced in units of 32 bytes. If only part of 32-byte data must be replaced with different data, the addresses that do not need to be replaced must be overwritten with the same data as the one existing prior to data replacement.

ADDREGn registers and RAM areas assigned to them are as follows:

Register	Address	RAM area	Number of words
ADDREG0	0xFFFF_E540	0xFFFD_FF60 - 0xFFFD_FF7F	8
ADDREG1	0xFFFF_E544	0xFFFD_FF80 - 0xFFFD_FF9F	8
ADDREG2	0xFFFF_E548	0xFFFD_FFA0 - 0xFFFD_FFBF	8
ADDREG3	0xFFFF_E54C	0xFFFD_FFC0 - 0xFFFD_FFDF	8
ADDREG4	0xFFFF_E550	0xFFFD_FFE0 - 0xFFFD_FFE3	1
ADDREG5	0xFFFF_E554	0xFFFD_FFE4 - 0xFFFD_FFE7	1
ADDREG6	0xFFFF_E558	0xFFFD_FFE8 - 0xFFFD_FFEB	1
ADDREG7	0xFFFF_E55C	0xFFFD_FFEC - 0xFFFD_FFEF	1
ADDREG8	0xFFFF_E560	0xFFFD_FFF0 - 0xFFFD_FFE3	1
ADDREG9	0xFFFF_E564	0xFFFD_FFF4 - 0xFFFD_FFE7	1
ADDREGA	0xFFFF_E568	0xFFFD_FFF8 - 0xFFFD_FFEB	1
ADDREGB	0xFFFF_E56C	0xFFFD_FFFC - 0xFFFD_FFEF	1

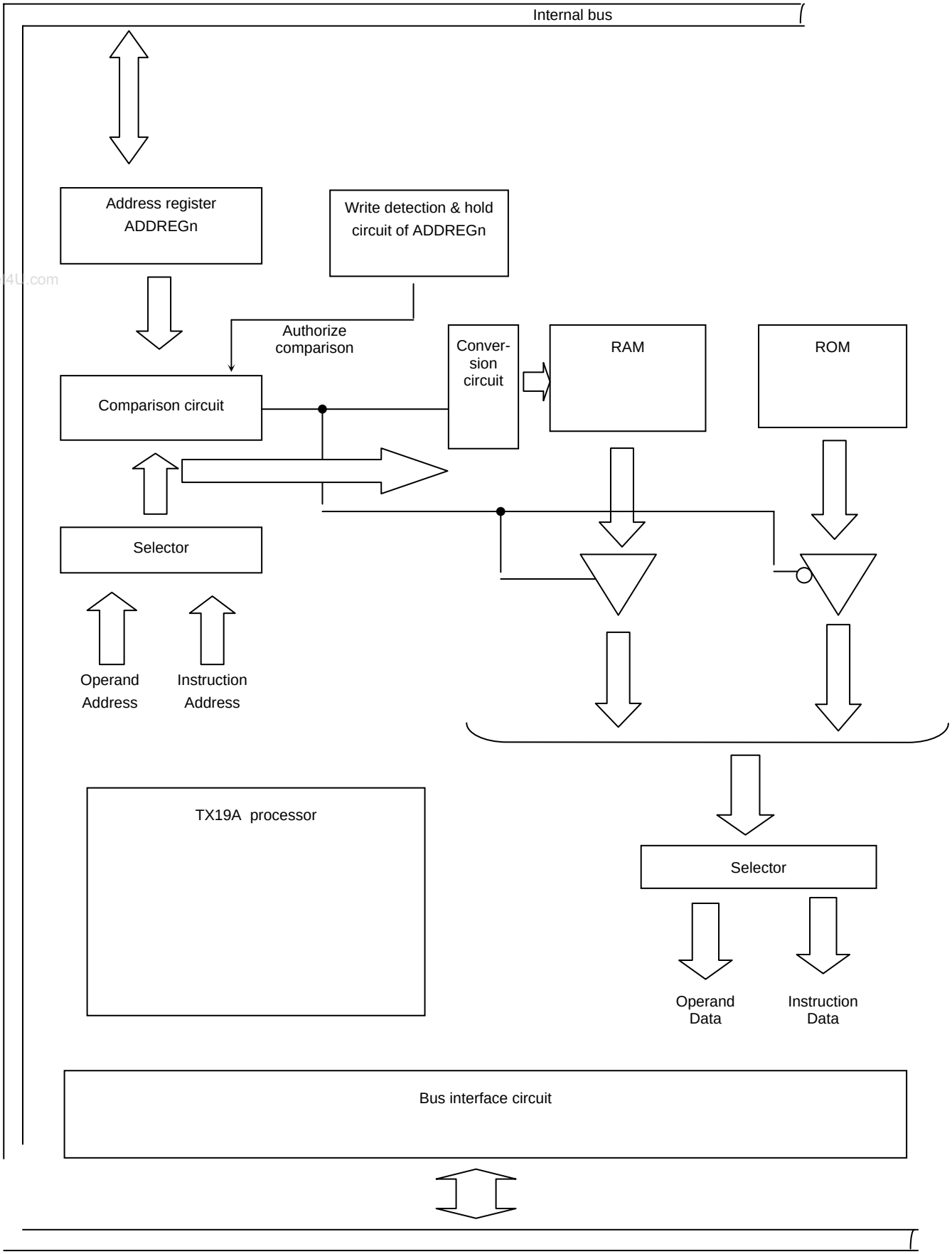


Fig. 19.2.1 ROM Correction System Diagram

19.3 Registers

(1) Address registers

ADDREG0 (0xFFFF_E540)

	7	6	5	4	3	2	1	0
bit Symbol	ADD07	ADD06	ADD05					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD015	ADD014	ADD013	ADD012	ADD011	ADD010	ADD09	ADD08
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD023	ADD022	ADD021	ADD020	ADD019	ADD018	ADD017	ADD016
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD031	ADD030	ADD029	ADD028	ADD027	ADD026	ADD025	ADD024
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG1 (0xFFFF_E544)

	7	6	5	4	3	2	1	0
bit Symbol	ADD17	ADD16	ADD15					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD115	ADD114	ADD113	ADD112	ADD111	ADD110	ADD19	ADD18
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD123	ADD122	ADD121	ADD120	ADD119	ADD118	ADD117	ADD116
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD131	ADD130	ADD129	ADD128	ADD127	ADD126	ADD125	ADD124
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG2 (0xFFFF_E548)

	7	6	5	4	3	2	1	0
bit Symbol	ADD27	ADD26	ADD25					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD215	ADD214	ADD213	ADD212	ADD211	ADD210	ADD29	ADD28
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD223	ADD222	ADD221	ADD220	ADD219	ADD218	ADD217	ADD216
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD231	ADD230	ADD229	ADD228	ADD227	ADD226	ADD225	ADD224
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG3 (0xFFFF_E54C)

	7	6	5	4	3	2	1	0
bit Symbol	ADD37	ADD36	ADD35					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD315	ADD314	ADD313	ADD312	ADD311	ADD310	ADD39	ADD38
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD323	ADD322	ADD321	ADD320	ADD319	ADD318	ADD317	ADD316
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD331	ADD330	ADD329	ADD328	ADD327	ADD326	ADD325	ADD324
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG4 (0xFFFF_E550)

	7	6	5	4	3	2	1	0
bit Symbol	ADD47	ADD46	ADD45	ADD44	ADD43	ADD42		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD415	ADD414	ADD413	ADD412	ADD411	ADD410	ADD49	ADD48
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD423	ADD422	ADD421	ADD420	ADD419	ADD418	ADD417	ADD416
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD431	ADD430	ADD429	ADD428	ADD427	ADD426	ADD425	ADD424
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG5 (0xFFFF_E554)

	7	6	5	4	3	2	1	0
bit Symbol	ADD57	ADD56	ADD55	ADD54	ADD53	ADD52		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD515	ADD514	ADD513	ADD512	ADD511	ADD510	ADD59	ADD58
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD523	ADD522	ADD521	ADD520	ADD519	ADD518	ADD517	ADD516
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD531	ADD530	ADD529	ADD528	ADD527	ADD526	ADD525	ADD524
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG6 (0xFFFF_E558)

	7	6	5	4	3	2	1	0
bit Symbol	ADD67	ADD66	ADD65	ADD64	ADD63	ADD62		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD615	ADD614	ADD613	ADD612	ADD611	ADD610	ADD69	ADD68
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD623	ADD622	ADD621	ADD620	ADD619	ADD618	ADD617	ADD616
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD631	ADD630	ADD629	ADD628	ADD627	ADD626	ADD625	ADD624
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG7 (0xFFFF_E55C)

	7	6	5	4	3	2	1	0
bit Symbol	ADD77	ADD76	ADD75	ADD74	ADD73	ADD72		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD715	ADD714	ADD713	ADD712	ADD711	ADD710	ADD79	ADD78
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD723	ADD722	ADD721	ADD720	ADD719	ADD718	ADD717	ADD716
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD731	ADD730	ADD729	ADD728	ADD727	ADD726	ADD725	ADD724
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG8 (0xFFFF_E560)

	7	6	5	4	3	2	1	0
bit Symbol	ADD87	ADD86	ADD85	ADD84	ADD83	ADD82		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD815	ADD814	ADD813	ADD812	ADD811	ADD810	ADD89	ADD88
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD823	ADD822	ADD821	ADD820	ADD819	ADD818	ADD817	ADD816
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD831	ADD830	ADD829	ADD828	ADD827	ADD826	ADD825	ADD824
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG9 (0xFFFF_E564)

	7	6	5	4	3	2	1	0
bit Symbol	ADD97	ADD96	ADD95	ADD94	ADD93	ADD92		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD915	ADD914	ADD913	ADD912	ADD911	ADD910	ADD99	ADD98
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD923	ADD922	ADD921	ADD920	ADD919	ADD918	ADD917	ADD916
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD931	ADD930	ADD929	ADD928	ADD927	ADD926	ADD925	ADD924
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREGA (0xFFFF_E568)

	7	6	5	4	3	2	1	0
bit Symbol	ADDA7	ADDA6	ADDA5	ADDA4	ADDA3	ADDA2		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADDA15	ADDA14	ADDA13	ADDA12	ADDA11	ADDA10	ADDA9	ADDA8
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADDA23	ADDA22	ADDA21	ADDA20	ADDA19	ADDA18	ADDA17	ADDA16
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADDA31	ADDA30	ADDA29	ADDA28	ADDA27	ADDA26	ADDA25	ADDA24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREGB (0xFFFF_E56C)

	7	6	5	4	3	2	1	0
bit Symbol	ADDB7	ADDB6	ADDB5	ADDB4	ADDB3	ADDB2		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADDB15	ADDB14	ADDB13	ADDB12	ADDB11	ADDB10	ADDB9	ADDB8
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADDB23	ADDB22	ADDB21	ADDB20	ADDB19	ADDB18	ADDB17	ADDB16
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADDB31	ADDB30	ADDB29	ADDB28	ADDB27	ADDB26	ADDB25	ADDB24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(Note 1) Data cannot be transferred by DMA to the address register. However, data can be transferred by DMA to the RAM area where data for replacement is placed. The ROM correction function supports data replacement for both CPU and DMA access.

(Note 2) Writing back the initial value "0x00" allows data at the reset address to be replaced.

20. Security Function

20.1 General

This device is implemented with the ROM security function for the internal ROM (flash) area as well as the DSU security function to inhibit use of DSU (DSU-Probes). The following three security functions are available:

- Flash security
- ROM security
- DSU security

20.2 Features

20.2.1 Flash Security

The flash security function refers to the condition where all the memory areas are protected through the automatic protection bit programming function to use the FLCS <PROTECT3:0> bits inhibiting write and erase operations of the internal ROM data for individual protection areas (in 512 kB blocks). In this case, the flash memory cannot be read from any area outside the flash memory such as the internal RAM areas where the protection bit erase command cannot be accepted. After this, no command writing can be performed normally. The flash security function is also a function to be necessary in enabling the ROM security and DSU security functions.

When the automatic protection bit erase command is executed while the system is in a secure condition, the flash memory is automatically initialized within the device. Therefore, be sufficiently careful in making a transition to a secure state.

20.2.2 ROM Security

The ROM security function can inhibit data write/read operations to/from the internal ROM. This function is used together with the flash security function.

Although the PC of RAM area instructions that have been replaced from the ROM area through the ROM correction function indicates an address in the flash ROM area, it is actually in the RAM area and thus data cannot be read in the condition ROM security is in place. For reading data using an instruction in the RAM area that has been replaced from the ROM area, some special method such as to use a program in the ROM area to write the data value into RAM will be necessary.

When the ROM security is applied to the ROM area, the following operations are inhibited:

- Operation to load or store ROM area data using an instruction placed outside the ROM area
- DMAC data transfer of ROM area data
- EJTAG based operation to load or store ROM area data
- Boot ROM operation to load or store ROM area data
- Flash writer operation to load or store ROM area data
- Access to security related registers (ROMSEC1 and ROMSEC2) in the ROM area using an instruction placed outside the ROM area.
- Execution of any flash command sequence other than the automatic block protection clear command and automatic block security clear command in the writer mode and any flash command sequence in the single mode or boot mode that specifies an address in the ROM area

Even when the ROM security is applied to the ROM area, the following operations can be performed:

- Loading of ROM area data using an instruction placed in the ROM area
- Loading of data outside the ROM area to use an instruction placed in any area
- Branch instruction to jump to the ROM area to use an instruction placed in any area
- PC trace (with some limitations) and break operations in the ROM area to use EJTAG

20.2.3 DSU Security

The DSU security function prevents easy reading of the internal flash memory by a third party other than the authorized user when an onboard DSU probe is used. By enabling the DSU security function, it becomes impossible to read the internal flash memory from a DSU probe. This function is used together with the flash security function.

20.3 Outline Security Configuration and Correspondence Table

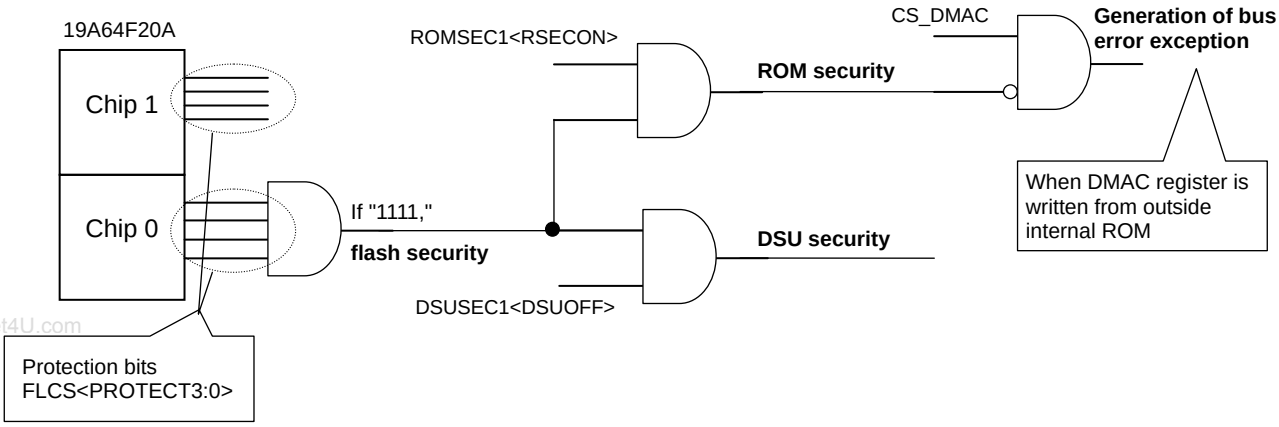


Fig. 21.3.1 Various Security Conditions (Outline)

Table 21.3.1 Various Security Conditions in Each Mode

	Protection bit setting, FLCS <PROTECT3:0>	1111				≠1111
	ROM security enable bit, ROMSEC1<RSECON>	1		0		Don't Care
	DSU security enable bit, DSUSEC1<DSUOFF>	1	0	1	0	Don't Care
	Flash security state	ON				OFF
	ROM security state	ON		OFF		OFF
	DSU security state	ON	OFF	ON	OFF	OFF
Single/Single Boot mode	Flash read from the internal ROM	○	○	○	○	○
	Flash read from outside the internal ROM	× *1	× *1	○	○	○
	ROM security enable clear (from ROM)	○	○			○
	ROM security enable clear (from outside ROM)	× *2	× *2			○
	DSU security enable clear (from ROM)	○		○		○
	DSU security enable clear (from outside ROM)	× *3		○		○
	Generation of protection bit erase command	× *4	× *4	○ *8	○ *8	○
	Generation of command other than protection bit erase command	× *5	× *5	× *7	× *7	Δ *9
	Write to DMAC configuration register (from ROM)	○	○	○	○	○
	Write to DMAC configuration register (from outside ROM)	× *6	× *6	○	○	○
Writer mode	Flash read	× *1	× *1	× *1	× *1	○
	Generation of protection bit erase command	○ *8	○ *8	○ *8	○ *8	Δ *9
	Generation of command other than protection bit erase command	× *7	× *7	× *7	× *7	Δ *9

*1 : Always reads "0x00000098."
*2 : Masks the stored data (Register cannot be written or cleared.)
*3 : Masks the stored data (Register cannot be written or cleared.)
*4 : Command address is masked and the flash memory cannot recognize the command.
*5 : Command address is masked and the flash memory cannot recognize the command.
*6 : Bus error exceptions are generated. (When set to DMAC register.)
*7 : Commands are not recognized because of the flash security state.
*8 : Commands result in flash area erase and protection bit erase operations because of the flash security state.
*9 : Command conversion is performed in the flash interface according to the protection bit status and input command.

20.4 Register

Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the block protection status.

Table 21.3.2 Flash Control Register

FLCS (0xFFFF_E520)

	7	6	5	4	3	2	1	0
Bit Symbol	PROTECT3	PROTECT2	PROTECT1	PROTECT0		ROMTYPE	PRGB	RDY/BSY
Read/Write	R				R	R	R/W	R
After power on reset	0	0	0	0	0	0	0	1
Function	Protection area setting (in 512 kB blocks) 0000: No blocks are protected xxx1: Area 0 is protected xx1x: Area 1 is protected x1xx: Area 2 is protected 1xxx: Area 3 is protected					ROM ID bit 0: Flash 1: MROM	Programming bit 0: Already issued 1: Issue	Ready/Busy 0: In operation 1: Finished operation
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After power on reset	0	0	0	0	0	0	0	0
Function								
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After power on reset	0	0	0	0	0	0	0	0
Function								
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After power on reset	0	0	0	0	0	0	0	0
Function								

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. It returns to "1" upon power on.

(Note) Be sure to confirm the ready status whenever a command is to be issued.
Issuing a command while the device is busy may result in a situation where any further command inputs are rejected in addition to the fact that the command cannot be transferred correctly. In such a case, restore the system by using system reset or a reset command.

Bit 1: Programming bit

This bit notifies the flash interface that a command is to be issued to the flash memory.
Be sure to set this bit to "1" whenever a command is to be issued to the internal flash memory. Also, when all commands have been issued, set this bit to "0" after confirming that the <RDY/BSY> bit has been set to "1."

Bit 2: ROM type identification bit

This bit is read after reset to identify whether the ROM is a flash ROM or a mask ROM.
Flash ROM: "0"
Mask ROM: "1"

Bits [7:4]: Protection bits (x: can be set to any combination of areas)

Each of the protection bits (4 bits) represents the protection status of the corresponding area. When a bit is set to "1," it indicates that the area corresponding to the bit is protected. When the area is protected, data cannot be written into it.

Table 21.3.3 ROM Security Register

ROMSEC1 (0xFFFF_E518)

	7	6	5	4	3	2	1	0
Bit Symbol								RSECON
Read/Write	R							R/W
After power on reset	0							1
Function	Always reads "0."							ROM ROM security 1: ON 0: OFF (Note)
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After power on reset	0							
Function	Always reads "0."							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After power on reset	0							
Function								
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After power on reset	0							
Function	Always reads "0."							

(Note) This register can be initialized only by a power on reset. Normal reset inputs cannot reset the register.

(Note) This register must be 32-bit accessed.

Table 21.3.4 Security Lock Register

ROMSEC2 (0xFFFF_E51C)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
After reset	Undefined							
Function	Refer to the note.							
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	W							
After reset	Undefined							
Function	Refer to the note.							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	W							
After reset	Undefined							
Function	Refer to the note.							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	W							
After reset	Undefined							
Function	Refer to the note.							

(Note) After setting ROMSEC1 <RSECON>, setting "0x0000_003D" to this register sets the value to ROMSEC1 <RSECON>.
(Note) When ROM security is applied to a ROM area, the ROMSEC1 and ROMSEC2 registers can be accessed only from an instruction placed in the ROM area.
(Note) This register must be 32-bit accessed.
(Note) This register is a write-only register. Any value read is undefined.

Table 21.3.5 DSU Security Mode Register

DSUSEC1 (0xFFFF_E510)

	7	6	5	4	3	2	1	0
Bit Symbol								DSUOFF
Read/Write	R							R/W
After power on reset	0							1
Function	Always reads "0."							1: DSU disable 0: DSU enable
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After power on reset	0							
Function	Always reads "0."							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After power on reset	0							
Function								
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After power on reset	0							
Function	Always reads "0."							

(Note) This register can be initialized only by a power on reset. Normal reset inputs cannot reset the register.
(Note) This register must be 32-bit accessed.

Table 21.3.6 DSU Security Control Register

DSUSEC2 (0xFFFF_E514)

	7	6	5	4	3	2	1	0
Bit Symbol	DSECODE07	DSECODE06	DSECODE05	DSECODE04	DSECODE03	DSECODE02	DSECODE01	DSECODE00
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							
	15	14	13	12	11	10	9	8
Bit Symbol	DSECODE15	DSECODE14	DSECODE13	DSECODE12	DSECODE11	DSECODE10	DSECODE09	DSECODE08
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							
	23	22	21	20	19	18	17	16
Bit Symbol	DSECODE23	DSECODE22	DSECODE21	DSECODE20	DSECODE19	DSECODE18	DSECODE17	DSECODE16
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							
	31	30	29	28	27	26	25	24
Bit Symbol	DSECODE31	DSECODE30	DSECODE29	DSECODE28	DSECODE27	DSECODE26	DSECODE25	DSECODE24
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							

(Note) This register must be 32-bit accessed.
(Note) This register is a write-only register. Any value read is undefined.

20.5 Setting Security Configuration

If it is necessary to rewrite the flash memory or protection bits while the device is in a secure state, either perform the automatic protection bit erase operation or clear the ROM security function. While the DSU security is applied, any DSU cannot be used.

The setting is necessary to make DSU-probe available beforehand if an automatic protection bit programming is executed to result in a flash security state.

When the automatic protection bit erase command is executed while the system is in the flash security mode, the flash memory is automatically initialized within the device. Therefore, be sufficiently careful in making a transition to a secure state.

20.5.1 Flash Security

Setting or clearing of flash security is made using a command sequence to the flash memory to use the protection bit programming command. Refer to command sequence descriptions in the section describing flash memory operation for more details.

20.5.2 ROM Security

In order to prevent the ROM security function from being accidentally removed by system runaway, etc., a double action method is used to set or clear the ROM security function. To make ROM security functional, first set the ROM security register ROMSEC1 <RSECON> to "1" and then write the security code "0x0000_003D" to the ROM security lock register ROMSEC2. Similarly, when the ROM security function is to be cleared, first set the ROM security register ROMSEC1 <RSECON> to "0" and then write the security code "0x0000_003D" to the ROM security lock register ROMSEC2.

(Note) The ROM security register has a power on reset circuit and the <RSECON> bit is set to "1" after power is turned on. If the flash security function is in place at this point, the ROM security function is automatically enabled to inhibit data write/read operations to/from the internal ROM.

20.5.3 DSU Security

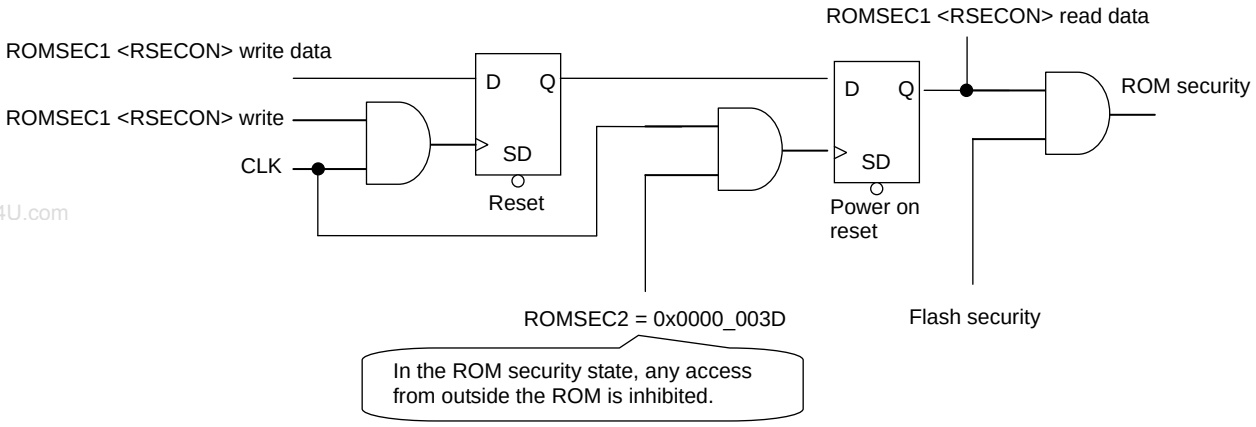
DSU enable/disable (Enables or disables use of DSU probes for debugging)

In order to prevent the DSU inhibit function from being accidentally removed by system runaway, etc., a double action method is used to clear the DSU inhibit function. So, first set the DSU security mode register DSUSEC1 <DSUOFF> to "0" and then write the security code "0x0000_00C5" to the DSU security control register DSUSEC2. Then, debugging to use a DSU probe is allowed. While power to the device is still applied, setting DSUSEC1 <DSUOFF> to "1" and writing "0x0000_00C5" to the DSUSEC2 register will enable the security function again.

(Note) The DSU security mode register has a power on reset circuit and the <DSUOFF> bit is set to "1" after power is turned on. If the flash security function is in place at this point, the DSU security function is automatically enabled and it becomes impossible to read the internal flash memory from any DSU probe.

20.5.4 ROM Security Register: ROMSEC1 <RSECON>

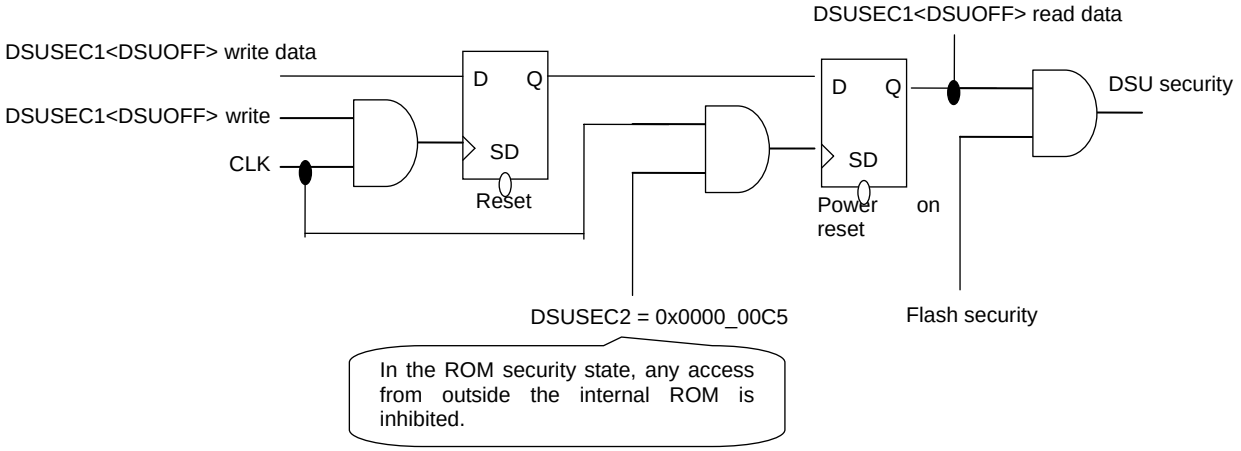
The ROM security register is provided with a power on reset circuit. Note that the data to be read from the ROMSEC1 <RSECON> bit is different from the original data written to the register. The outline schematic diagram is shown below:



20.5.5 DSU Security Mode Register: DSUSEC1 <DSUOFF>

The DSU security mode register is provided with a power on reset circuit.

Note that the data to be read from the DSUSEC1 <DSUOFF> bit is different from the original data written to the register. The outline schematic diagram is shown below:



21. Table of Special Function Registers

Special function registers are allocated to an 8K-byte address space from FFFFE000H to FFFFFFFFH.

[1] Port registers
[2] Watchdog timer
[3] 16-bit timer
[4] I^2CBUS/serial channel
[5] UART/serial channel
[6] 10-bit A/D converter
[7] Key-on wake-up
[8] 32-bit input capture
[9] 32-bit compare
[10] Interrupt controller
[11] DMA controller
[12] Chip select/wait controller
[13] Access control
[14] Security control
[15] FLASH control
[16] ROM correction
[17] Clock timer
[18] Clock generator

(Note) 0xFFFF_F000 to 0xFFFF_FFFF are a little-endian area.
0xFFFF_E000 to 0xFFFF_EFFF are a bi-endian area.

(Note) For continuous 8-bit long registers, 16- or 32-bit access is possible. The use of 16- or 32-bit access requires that an even-number address be accessed and that an even-number address does not contain undefined areas.

Little-endian

[1] PORT registers

ADR	Register name
FFFFF000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF010H	
1H	
2H	P2
3H	
4H	P2CR
5H	P2FC
6H	
7H	
8H	P3
9H	
AH	P3CR
BH	P3FC
CH	
DH	
EH	P4
FH	

ADR	Register name
FFFFF020H	P4CR
1H	P4FC
2H	
3H	
4H	
5H	
6H	
7H	
8H	P5
9H	P6
AH	
BH	
CH	P5CR
DH	P5FC
EH	P6CR
FH	P6FC

ADR	Register name
FFFFF040H	P7
1H	P8
2H	P9
3H	PA
4H	
5H	
6H	
7H	PACR
8H	P7FC
9H	P8FC
AH	P9FC
BH	PAFC
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF050H	PB
1H	PC
2H	PD
3H	PE
4H	PBCR
5H	PCCR
6H	PDCR
7H	PECR
8H	PBFC
9H	PCFC
AH	PDFC
BH	PEFC
CH	
DH	PCODE
EH	PDODE
FH	PEODE

ADR	Register name
FFFFF060H	PF
1H	PG
2H	PH
3H	PI
4H	PFCR
5H	PGCR
6H	PHCR
7H	PICR
8H	PFFC
9H	PGFC
AH	PHFC
BH	PIFC
CH	PFODE
DH	
EH	
FH	

ADR	Register name
FFFFF070H	PJ
1H	PK
2H	
3H	
4H	PJCR
5H	PKCR
6H	
7H	
8H	PJFC
9H	PKFC
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0C0H	PL
1H	PM
2H	PN
3H	PO
4H	PLCR
5H	PMCR
6H	PNCR
7H	POCR
8H	
9H	
AH	
BH	POFC
CH	
DH	
EH	
FH	POODE

ADR	Register name
FFFFF0D0H	PP
1H	PQ
2H	
3H	
4H	PPCR
5H	PQCR
6H	
7H	
8H	PPFC
9H	
AH	
BH	
CH	PPFC2
DH	PQFC2
EH	
FH	

Little-endian

[2] WDT

ADR	Register name
FFFFF090H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] 16-bit timer

ADR	Register name
FFFFF140H	TB0RUN
1H	TB0CR
2H	TB0MOD
3H	TB0FFCR
4H	TB0ST
5H	
6H	TB0UCL
7H	TB0UCH
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

ADR	Register name
FFFFF150H	TB1RUN
1H	TB1CR
2H	TB1MOD
3H	TB1FFCR
4H	TB1ST
5H	
6H	TB1UCL
7H	TB1UCH
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

ADR	Register name
FFFFF160H	TB2RUN
1H	TB2CR
2H	TB2MOD
3H	TB2FFCR
4H	TB2ST
5H	
6H	TB2UCL
7H	TB2UCH
8H	TB2RG0L
9H	TB2RG0H
AH	TB2RG1L
BH	TB2RG1H
CH	TB2CP0L
DH	TB2CP0H
EH	TB2CP1L
FH	TB2CP1H

ADR	Register name
FFFFF170H	TB3RUN
1H	TB3CR
2H	TB3MOD
3H	TB3FFCR
4H	TB3ST
5H	
6H	TB3UCL
7H	TB3UCH
8H	TB3RG0L
9H	TB3RG0H
AH	TB3RG1L
BH	TB3RG1H
CH	TB3CP0L
DH	TB3CP0H
EH	TB3CP1L
FH	TB3CP1H

ADR	Register name
FFFFF180H	TB4RUN
1H	TB4CR
2H	TB4MOD
3H	TB4FFCR
4H	TB4ST
5H	
6H	TB4UCL
7H	TB4UCH
8H	TB4RG0L
9H	TB4RG0H
AH	TB4RG1L
BH	TB4RG1H
CH	TB4CP0L
DH	TB4CP0H
EH	TB4CP1L
FH	TB4CP1H

ADR	Register name
FFFFF190H	TB5RUN
1H	TB5CR
2H	TB5MOD
3H	TB5FFCR
4H	TB5ST
5H	
6H	TB5UCL
7H	TB5UCH
8H	TB5RG0L
9H	TB5RG0H
AH	TB5RG1L
BH	TB5RG1H
CH	TB5CP0L
DH	TB5CP0H
EH	TB5CP1L
FH	TB5CP1H

ADR	Register name
FFFFF1A0H	TB6RUN
1H	TB6CR
2H	TB6MOD
3H	TB6FFCR
4H	TB6ST
5H	
6H	TB6UCL
7H	TB6UCH
8H	TB6RG0L
9H	TB6RG0H
AH	TB6RG1L
BH	TB6RG1H
CH	TB6CP0L
DH	TB6CP0H
EH	TB6CP1L
FH	TB6CP1H

ADR	Register name
FFFFF1B0H	TB7RUN
1H	TB7CR
2H	TB7MOD
3H	TB7FFCR
4H	TB7ST
5H	
6H	TB7UCL
7H	TB7UCH
8H	TB7RG0L
9H	TB7RG0H
AH	TB7RG1L
BH	TB7RG1H
CH	TB7CP0L
DH	TB7CP0H
EH	TB7CP1L
FH	TB7CP1H

Little-endian

ADR	Register name
FFFFF1C0H	TB8RUN
1H	TB8CR
2H	TB8MOD
3H	TB8FFCR
4H	TB8ST
5H	
6H	TB8UCL
7H	TB8UCH
8H	TB8RG0L
9H	TB8RG0H
AH	TB8RG1L
BH	TB8RG1H
CH	TB8CP0L
DH	TB8CP0H
EH	TB8CP1L
FH	TB8CP1H

ADR	Register name
FFFFF1D0H	TB9RUN
1H	TB9CR
2H	TB9MOD
3H	TB9FFCR
4H	TB9ST
5H	
6H	TB9UCL
7H	TB9UCH
8H	TB9RG0L
9H	TB9RG0H
AH	TB9RG1L
BH	TB9RG1H
CH	TB9CP0L
DH	TB9CP0H
EH	TB9CP1L
FH	TB9CP1H

ADR	Register name
FFFFF1E0H	TBARUN
1H	TBACR
2H	TBAMOD
3H	TBAFFCR
4H	TBAST
5H	
6H	TBAUCL
7H	TBAUCH
8H	TBARG0L
9H	TBARG0H
AH	TBARG1L
BH	TBARG1H
CH	TBACP0L
DH	TBACP0H
EH	TBACP1L
FH	TBACP1H

[4] I2C/SIO

ADR	Register name
FFFFF250H	SBICR1
1H	SBIDBR
2H	I2CAR
3H	SBICR2/SR
4H	SBIBR0
5H	
6H	
7H	SBICR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] UART/SIO

ADR	Register name
FFFFF260H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	SC0MOD2
7H	SC0EN
8H	SC0RFC
9H	SC0TFC
AH	SC0RST
BH	SC0TST
CH	SC0FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF270H	SC1BUF
1H	SC1CR
2H	SC1MOD0
3H	BR1CR
4H	BR1ADD
5H	SC1MOD1
6H	SC1MOD2
7H	SC1EN
8H	SC1RFC
9H	SC1TFC
AH	SC1RST
BH	SC1TST
CH	SC1FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF280H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	SC2MOD2
7H	SC2EN
8H	SC2RFC
9H	SC2TFC
AH	SC2RST
BH	SC2TST
CH	SC2FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF290H	SC3BUF
1H	SC3CR
2H	SC3MOD0
3H	BR3CR
4H	BR3ADD
5H	SC3MOD1
6H	SC3MOD2
7H	SC3EN
8H	SC3RFC
9H	SC3TFC
AH	SC3RST
BH	SC3TST
CH	SC3FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF2A0H	SC4BUF
1H	SC4CR
2H	SC4MOD0
3H	BR4CR
4H	BR4ADD
5H	SC4MOD1
6H	SC4MOD2
7H	SC4EN
8H	SC4RFC
9H	SC4TFC
AH	SC4RST
BH	SC4TST
CH	SC4FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF2B0H	SC5BUF
1H	SC5CR
2H	SC5MOD0
3H	BR5CR
4H	BR5ADD
5H	SC5MOD1
6H	SC5MOD2
7H	SC5EN
8H	SC5RFC
9H	SC5TFC
AH	SC5RST
BH	SC5TST
CH	SC5FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF2C0H	SC6BUF
1H	SC6CR
2H	SC6MOD0
3H	BR6CR
4H	BR6ADD
5H	SC6MOD1
6H	SC6MOD2
7H	SC6EN
8H	SC6RFC
9H	SC6TFC
AH	SC6RST
BH	SC6TST
CH	SC6FCNF
DH	
EH	
FH	

Little-endian

[6] 10-bit ADC

ADR	Register name
FFFFF300H	ADREG08L
1H	ADREG08H
2H	ADREG19L
3H	ADREG19H
4H	ADREG2AL
5H	ADREG2AH
6H	ADREG3BL
7H	ADREG3BH
8H	ADREG4CL
9H	ADREG4CH
AH	ADREG5DL
BH	ADREG5DH
CH	ADREG6EL
DH	ADREG6EH
EH	ADREG7FL
FH	ADREG7FH

ADR	Register name
FFFFF310H	ADREGSPL
1H	ADREGSPH
2H	ADCOMREGL
3H	ADCOMREGH
4H	ADMOD0
5H	ADMOD1
6H	ADMOD2
7H	ADMOD3
8H	ADMOD4
9H	
AH	
BH	
CH	ADCLK
DH	
EH	
FH	

[7] KWUP

ADR	Register name
FFFFF360H	KWUPST0
1H	KWUPST1
2H	KWUPST2
3H	KWUPST3
4H	KWUPST4
5H	KWUPST5
6H	KWUPST6
7H	KWUPST7
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF370H	KWUPST
1H	KUPPUP
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[8] 32-bit input capture

ADR	Register name
FFFFF400H	TCCR
1H	TBTRUN
2H	TBTCR
3H	
4H	TBTCAP0
5H	TBTCAP1
6H	TBTCAP2
7H	TBTCAP3
8H	TBTRDCAP0
9H	TBTRDCAP1
AH	TBTRDCAP2
BH	TBTRDCAP3
CH	TCGIM
DH	TCGST
EH	
FH	

ADR	Register name
FFFFF410H	CAP0CR
1H	
2H	
3H	
4H	TCCAP0LL
5H	TCCAP0LH
6H	TCCAP0HL
7H	TCCAP0HH
8H	CAP1CR
9H	
AH	
BH	
CH	TCCAP1LL
DH	TCCAP1LH
EH	TCCAP1HL
FH	TCCAP1HH

ADR	Register name
FFFFF420H	CAP2CR
1H	
2H	
3H	
4H	TCCAP2LL
5H	TCCAP2LH
6H	TCCAP2HL
7H	TCCAP2HH
8H	CAP3CR
9H	
AH	
BH	
CH	TCCAP3LL
DH	TCCAP3LH
EH	TCCAP3HL
FH	TCCAP3HH

[9] 32-bit output compare

ADR	Register name
FFFFF440H	TCCMP0LL
1H	TCCMP0LH
2H	TCCMP0HL
3H	TCCMP0HH
4H	TCCMP1LL
5H	TCCMP1LH
6H	TCCMP1HL
7H	TCCMP1HH
8H	TCCMP2LL
9H	TCCMP2LH
AH	TCCMP2HL
BH	TCCMP2HH
CH	TCCMP3LL
DH	TCCMP3LH
EH	TCCMP3HL
FH	TCCMP3HH

ADR	Register name
FFFFF450H	TCCMP4LL
1H	TCCMP4LH
2H	TCCMP4HL
3H	TCCMP4HH
4H	TCCMP5LL
5H	TCCMP5LH
6H	TCCMP5HL
7H	TCCMP5HH
8H	TCCMP6LL
9H	TCCMP6LH
AH	TCCMP6HL
BH	TCCMP6HH
CH	TCCMP7LL
DH	TCCMP7LH
EH	TCCMP7HL
FH	TCCMP7HH

ADR	Register name
FFFFF460H	TCCMP8LL
1H	TCCMP8LH
2H	TCCMP8HL
3H	TCCMP8HH
4H	TCCMP9LL
5H	TCCMP9LH
6H	TCCMP9HL
7H	TCCMP9HH
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF470H	CMPCTL0
1H	CMPCTL1
2H	CMPCTL2
3H	CMPCTL3
4H	CMPCTL4
5H	CMPCTL5
6H	CMPCTL6
7H	CMPCTL7
8H	CMPCTL8
9H	CMPCTL9
AH	
BH	
CH	
DH	
EH	
FH	

Little-endian

[10] INTC

ADR	Register name
FFFFE000H	IMC0
1H	ditto
2H	ditto
3H	ditto
4H	IMC1
5H	ditto
6H	ditto
7H	ditto
8H	IMC2
9H	ditto
AH	ditto
BH	ditto
CH	IMC3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE010H	IMC4
1H	ditto
2H	ditto
3H	ditto
4H	IMC5
5H	ditto
6H	ditto
7H	ditto
8H	IMC6
9H	ditto
AH	ditto
BH	ditto
CH	IMC7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE020H	IMC8
1H	ditto
2H	ditto
3H	ditto
4H	IMC9
5H	ditto
6H	ditto
7H	ditto
8H	IMCA
9H	ditto
AH	ditto
BH	ditto
CH	IMCB
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE030H	IMCC
1H	ditto
2H	ditto
3H	ditto
4H	IMCD
5H	ditto
6H	ditto
7H	ditto
8H	IMCE
9H	ditto
AH	ditto
BH	ditto
CH	IMCF
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE040H	IVR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE060H	INTCLR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE100H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	ILEV
DH	ditto
EH	ditto
FH	ditto

Little-endian

[11] DMAC

ADR	Register name
FFFFE200H	CCR0
1H	ditto
2H	ditto
3H	ditto
4H	CSR0
5H	ditto
6H	ditto
7H	ditto
8H	SAR0
9H	ditto
AH	ditto
BH	ditto
CH	DAR0
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE210H	BCR0
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR0
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE220H	CCR1
1H	ditto
2H	ditto
3H	ditto
4H	CSR1
5H	ditto
6H	ditto
7H	ditto
8H	SAR1
9H	ditto
AH	ditto
BH	ditto
CH	DAR1
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE230H	BCR1
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR1
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE240H	CCR2
1H	ditto
2H	ditto
3H	ditto
4H	CSR2
5H	ditto
6H	ditto
7H	ditto
8H	SAR2
9H	ditto
AH	ditto
BH	ditto
CH	DAR2
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE250H	BCR2
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR2
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE260H	CCR3
1H	ditto
2H	ditto
3H	ditto
4H	CSR3
5H	ditto
6H	ditto
7H	ditto
8H	SAR3
9H	ditto
AH	ditto
BH	ditto
CH	DAR3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE270H	BCR3
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR3
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE280H	CCR4
1H	ditto
2H	ditto
3H	ditto
4H	CSR4
5H	ditto
6H	ditto
7H	ditto
8H	SAR4
9H	ditto
AH	ditto
BH	ditto
CH	DAR4
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE290H	BCR4
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR4
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2A0H	CCR5
1H	ditto
2H	ditto
3H	ditto
4H	CSR5
5H	ditto
6H	ditto
7H	ditto
8H	SAR5
9H	ditto
AH	ditto
BH	ditto
CH	DAR5
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2B0H	BCR5
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR5
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

Little-endian

ADR	Register name
FFFFE2C0H	CCR6
1H	ditto
2H	ditto
3H	ditto
4H	CSR6
5H	ditto
6H	ditto
7H	ditto
8H	SAR6
9H	ditto
AH	ditto
BH	ditto
CH	DAR6
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2D0H	BCR6
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR6
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2E0H	CCR7
1H	ditto
2H	ditto
3H	ditto
4H	CSR7
5H	ditto
6H	ditto
7H	ditto
8H	SAR7
9H	ditto
AH	ditto
BH	ditto
CH	DAR7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2F0H	BCR7
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR7
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE300H	DCR
1H	ditto
2H	ditto
3H	ditto
4H	RSR
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	DHR
DH	ditto
EH	ditto
FH	ditto

Little-endian

[12] CS/WAIT controller

ADR	Register name
FFFFE400H	BMA0
1H	ditto
2H	ditto
3H	ditto
4H	BMA1
5H	ditto
6H	ditto
7H	ditto
8H	BMA2
9H	ditto
AH	ditto
BH	ditto
CH	BMA3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE410H	BMA4
1H	ditto
2H	ditto
3H	ditto
4H	BMA5
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE480H	B01CS
1H	ditto
2H	ditto
3H	ditto
4H	B23CS
5H	ditto
6H	ditto
7H	ditto
8H	B45CS
9H	ditto
AH	ditto
BH	ditto
CH	BEXCS
DH	ditto
EH	ditto
FH	ditto

[13] Access control

ADR	Register name
FFFFE500H	PFBWAIT
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] Security control

ADR	Register name
FFFFE510H	DSUSEC1
1H	ditto
2H	ditto
3H	ditto
4H	DSUSEC2
5H	ditto
6H	ditto
7H	ditto
8H	ROMSEC1
9H	
AH	
BH	
CH	ROMSEC2
DH	
EH	
FH	

[15] FLASH control

ADR	Register name
FFFFE520H	FLCS
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] ROM correction

ADR	Register name
FFFFE540H	ADDREG0
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG1
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG2
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE550H	ADDREG4
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG5
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG6
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE560H	ADDREG8
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG9
5H	ditto
6H	ditto
7H	ditto
8H	ADDREGA
9H	ditto
AH	ditto
BH	ditto
CH	ADDREGB
DH	ditto
EH	ditto
FH	ditto

Little-endian

[17] Clock timer

ADR	Register name
FFFFE700H	RTCFLG
1H	ditto
2H	ditto
3H	ditto
4H	RTCCR
5H	ditto
6H	ditto
7H	ditto
8H	RTCREG
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE710H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[18] CG

ADR	Register name
FFFFEE00H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	SYSCR3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFEE10H	IMCGA
1H	ditto
2H	ditto
3H	ditto
4H	IMCGB
5H	ditto
6H	ditto
7H	ditto
8H	IMCGC
9H	ditto
AH	ditto
BH	ditto
CH	IMCGD
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFEE20H	EICRCG
1H	ditto
2H	ditto
3H	ditto
4H	NMIFLG
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-endian

[1] PORT registers

ADR	Register name
FFFFF000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF010H	
1H	
2H	P2
3H	
4H	P2CR
5H	P2FC
6H	
7H	
8H	P3
9H	
AH	P3CR
BH	P3FC
CH	
DH	
EH	P4
FH	

ADR	Register name
FFFFF020H	P4CR
1H	P4FC
2H	
3H	
4H	
5H	
6H	
7H	
8H	P5
9H	P6
AH	
BH	
CH	P5CR
DH	P5FC
EH	P6CR
FH	P6FC

ADR	Register name
FFFFF040H	P7
1H	P8
2H	P9
3H	PA
4H	
5H	
6H	
7H	PACR
8H	P7FC
9H	P8FC
AH	P9FC
BH	PAFC
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF050H	PB
1H	PC
2H	PD
3H	PE
4H	PBCR
5H	PCCR
6H	PDCR
7H	PECR
8H	PBFC
9H	PCFC
AH	PDFC
BH	PEFC
CH	
DH	PCODE
EH	PDODE
FH	PEODE

ADR	Register name
FFFFF060H	PF
1H	PG
2H	PH
3H	PI
4H	PFCR
5H	PGCR
6H	PHCR
7H	PICR
8H	PFFC
9H	PGFC
AH	PHFC
BH	PIFC
CH	PFODE
DH	
EH	
FH	

ADR	Register name
FFFFF070H	PJ
1H	PK
2H	
3H	
4H	PJCR
5H	PKCR
6H	
7H	
8H	PJFC
9H	PKFC
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0C0H	PL
1H	PM
2H	PN
3H	PO
4H	PLCR
5H	PMCR
6H	PNCR
7H	POCR
8H	
9H	
AH	
BH	POFC
CH	
DH	
EH	
FH	POODE

ADR	Register name
FFFFF0D0H	PP
1H	PQ
2H	
3H	
4H	PPCR
5H	PQCR
6H	
7H	
8H	PPFC
9H	
AH	
BH	
CH	PPFC2
DH	PQFC2
EH	
FH	

Big-endian

[2] WDT

ADR	Register name
FFFFF090H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] 16-bit timer

ADR	Register name
FFFFF140H	TB0RUN
1H	TB0CR
2H	TB0MOD
3H	TB0FFCR
4H	TB0ST
5H	
6H	TB0UCL
7H	TB0UCH
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

ADR	Register name
FFFFF150H	TB1RUN
1H	TB1CR
2H	TB1MOD
3H	TB1FFCR
4H	TB1ST
5H	
6H	TB1UCL
7H	TB1UCH
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

ADR	Register name
FFFFF160H	TB2RUN
1H	TB2CR
2H	TB2MOD
3H	TB2FFCR
4H	TB2ST
5H	
6H	TB2UCL
7H	TB2UCH
8H	TB2RG0L
9H	TB2RG0H
AH	TB2RG1L
BH	TB2RG1H
CH	TB2CP0L
DH	TB2CP0H
EH	TB2CP1L
FH	TB2CP1H

ADR	Register name
FFFFF170H	TB3RUN
1H	TB3CR
2H	TB3MOD
3H	TB3FFCR
4H	TB3ST
5H	
6H	TB3UCL
7H	TB3UCH
8H	TB3RG0L
9H	TB3RG0H
AH	TB3RG1L
BH	TB3RG1H
CH	TB3CP0L
DH	TB3CP0H
EH	TB3CP1L
FH	TB3CP1H

ADR	Register name
FFFFF180H	TB4RUN
1H	TB4CR
2H	TB4MOD
3H	TB4FFCR
4H	TB4ST
5H	
6H	TB4UCL
7H	TB4UCH
8H	TB4RG0L
9H	TB4RG0H
AH	TB4RG1L
BH	TB4RG1H
CH	TB4CP0L
DH	TB4CP0H
EH	TB4CP1L
FH	TB4CP1H

ADR	Register name
FFFFF190H	TB5RUN
1H	TB5CR
2H	TB5MOD
3H	TB5FFCR
4H	TB5ST
5H	
6H	TB5UCL
7H	TB5UCH
8H	TB5RG0L
9H	TB5RG0H
AH	TB5RG1L
BH	TB5RG1H
CH	TB5CP0L
DH	TB5CP0H
EH	TB5CP1L
FH	TB5CP1H

ADR	Register name
FFFFF1A0H	TB6RUN
1H	TB6CR
2H	TB6MOD
3H	TB6FFCR
4H	TB6ST
5H	
6H	TB6UCL
7H	TB6UCH
8H	TB6RG0L
9H	TB6RG0H
AH	TB6RG1L
BH	TB6RG1H
CH	TB6CP0L
DH	TB6CP0H
EH	TB6CP1L
FH	TB6CP1H

ADR	Register name
FFFFF1B0H	TB7RUN
1H	TB7CR
2H	TB7MOD
3H	TB7FFCR
4H	TB7ST
5H	
6H	TB7UCL
7H	TB7UCH
8H	TB7RG0L
9H	TB7RG0H
AH	TB7RG1L
BH	TB7RG1H
CH	TB7CP0L
DH	TB7CP0H
EH	TB7CP1L
FH	TB7CP1H

Big-endian

ADR	Register name
FFFFF1C0H	TB8RUN
1H	TB8CR
2H	TB8MOD
3H	TB8FFCR
4H	TB8ST
5H	
6H	TB8UCL
7H	TB8UCH
8H	TB8RG0L
9H	TB8RG0H
AH	TB8RG1L
BH	TB8RG1H
CH	TB8CP0L
DH	TB8CP0H
EH	TB8CP1L
FH	TB8CP1H

ADR	Register name
FFFFF1D0H	TB9RUN
1H	TB9CR
2H	TB9MOD
3H	TB9FFCR
4H	TB9ST
5H	
6H	TB9UCL
7H	TB9UCH
8H	TB9RG0L
9H	TB9RG0H
AH	TB9RG1L
BH	TB9RG1H
CH	TB9CP0L
DH	TB9CP0H
EH	TB9CP1L
FH	TB9CP1H

ADR	Register name
FFFFF1E0H	TBARUN
1H	TBACR
2H	TBAMOD
3H	TBAFFCR
4H	TBAST
5H	
6H	TBAUCL
7H	TBAUCH
8H	TBARG0L
9H	TBARG0H
AH	TBARG1L
BH	TBARG1H
CH	TBACP0L
DH	TBACP0H
EH	TBACP1L
FH	TBACP1H

[4] I2C/SIO

ADR	Register name
FFFFF250H	SBICR1
1H	SBIDBR
2H	I2CAR
3H	SBICR2/SR
4H	SBIBR0
5H	
6H	
7H	SBICR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] UART/SIO

ADR	Register name
FFFFF260H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	SC0MOD2
7H	SC0EN
8H	SC0RFC
9H	SC0TFC
AH	SC0RST
BH	SC0TST
CH	SC0FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF270H	SC1BUF
1H	SC1CR
2H	SC1MOD0
3H	BR1CR
4H	BR1ADD
5H	SC1MOD1
6H	SC1MOD2
7H	SC1EN
8H	SC1RFC
9H	SC1TFC
AH	SC1RST
BH	SC1TST
CH	SC1FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF280H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	SC2MOD2
7H	SC2EN
8H	SC2RFC
9H	SC2TFC
AH	SC2RST
BH	SC2TST
CH	SC2FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF290H	SC3BUF
1H	SC3CR
2H	SC3MOD0
3H	BR3CR
4H	BR3ADD
5H	SC3MOD1
6H	SC3MOD2
7H	SC3EN
8H	SC3RFC
9H	SC3TFC
AH	SC3RST
BH	SC3TST
CH	SC3FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF2A0H	SC4BUF
1H	SC4CR
2H	SC4MOD0
3H	BR4CR
4H	BR4ADD
5H	SC4MOD1
6H	SC4MOD2
7H	SC4EN
8H	SC4RFC
9H	SC4TFC
AH	SC4RST
BH	SC4TST
CH	SC4FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF2B0H	SC5BUF
1H	SC5CR
2H	SC5MOD0
3H	BR5CR
4H	BR5ADD
5H	SC5MOD1
6H	SC5MOD2
7H	SC5EN
8H	SC5RFC
9H	SC5TFC
AH	SC5RST
BH	SC5TST
CH	SC5FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF2C0H	SC6BUF
1H	SC6CR
2H	SC6MOD0
3H	BR6CR
4H	BR6ADD
5H	SC6MOD1
6H	SC6MOD2
7H	SC6EN
8H	SC6RFC
9H	SC6TFC
AH	SC6RST
BH	SC6TST
CH	SC6FCNF
DH	
EH	
FH	

Big-endian

[6] 10-bit ADC

ADR	Register name
FFFFF300H	ADREG08L
1H	ADREG08H
2H	ADREG19L
3H	ADREG19H
4H	ADREG2AL
5H	ADREG2AH
6H	ADREG3BL
7H	ADREG3BH
8H	ADREG4CL
9H	ADREG4CH
AH	ADREG5DL
BH	ADREG5DH
CH	ADREG6EL
DH	ADREG6EH
EH	ADREG7FL
FH	ADREG7FH

ADR	Register name
FFFFF310H	ADREGSPL
1H	ADREGSPH
2H	ADCOMREGL
3H	ADCOMREGH
4H	ADMOD0
5H	ADMOD1
6H	ADMOD2
7H	ADMOD3
8H	ADMOD4
9H	
AH	
BH	
CH	ADCLK
DH	
EH	
FH	

[7] KWUP

ADR	Register name
FFFFF360H	KWUPST0
1H	KWUPST1
2H	KWUPST2
3H	KWUPST3
4H	KWUPST4
5H	KWUPST5
6H	KWUPST6
7H	KWUPST7
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF370H	KWUPST
1H	KUPPUP
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[8] 32-bit input capture

ADR	Register name
FFFFF400H	TCCR
1H	TBTRUN
2H	TBTCR
3H	
4H	TBTCAP0
5H	TBTCAP1
6H	TBTCAP2
7H	TBTCAP3
8H	TBTRDCAP0
9H	TBTRDCAP1
AH	TBTRDCAP2
BH	TBTRDCAP3
CH	TCGIM
DH	TCGST
EH	
FH	

ADR	Register name
FFFFF410H	CAP0CR
1H	
2H	
3H	
4H	TCCAP0LL
5H	TCCAP0LH
6H	TCCAP0HL
7H	TCCAP0HH
8H	CAP1CR
9H	
AH	
BH	
CH	TCCAP1LL
DH	TCCAP1LH
EH	TCCAP1HL
FH	TCCAP1HH

ADR	Register name
FFFFF420H	CAP2CR
1H	
2H	
3H	
4H	TCCAP2LL
5H	TCCAP2LH
6H	TCCAP2HL
7H	TCCAP2HH
8H	CAP3CR
9H	
AH	
BH	
CH	TCCAP3LL
DH	TCCAP3LH
EH	TCCAP3HL
FH	TCCAP3HH

[9] 32-bit output compare

ADR	Register name
FFFFF440H	TCCMP0LL
1H	TCCMP0LH
2H	TCCMP0HL
3H	TCCMP0HH
4H	TCCMP1LL
5H	TCCMP1LH
6H	TCCMP1HL
7H	TCCMP1HH
8H	TCCMP2LL
9H	TCCMP2LH
AH	TCCMP2HL
BH	TCCMP2HH
CH	TCCMP3LL
DH	TCCMP3LH
EH	TCCMP3HL
FH	TCCMP3HH

ADR	Register name
FFFFF450H	TCCMP4LL
1H	TCCMP4LH
2H	TCCMP4HL
3H	TCCMP4HH
4H	TCCMP5LL
5H	TCCMP5LH
6H	TCCMP5HL
7H	TCCMP5HH
8H	TCCMP6LL
9H	TCCMP6LH
AH	TCCMP6HL
BH	TCCMP6HH
CH	TCCMP7LL
DH	TCCMP7LH
EH	TCCMP7HL
FH	TCCMP7HH

ADR	Register name
FFFFF460H	TCCMP8LL
1H	TCCMP8LH
2H	TCCMP8HL
3H	TCCMP8HH
4H	TCCMP9LL
5H	TCCMP9LH
6H	TCCMP9HL
7H	TCCMP9HH
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF470H	CMPCTL0
1H	CMPCTL1
2H	CMPCTL2
3H	CMPCTL3
4H	CMPCTL4
5H	CMPCTL5
6H	CMPCTL6
7H	CMPCTL7
8H	CMPCTL8
9H	CMPCTL9
AH	
BH	
CH	
DH	
EH	
FH	

Big-endian

[10] INTC

ADR	Register name
FFFFE000H	IMC0
1H	ditto
2H	ditto
3H	ditto
4H	IMC1
5H	ditto
6H	ditto
7H	ditto
8H	IMC2
9H	ditto
AH	ditto
BH	ditto
CH	IMC3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE010H	IMC4
1H	ditto
2H	ditto
3H	ditto
4H	IMC5
5H	ditto
6H	ditto
7H	ditto
8H	IMC6
9H	ditto
AH	ditto
BH	ditto
CH	IMC7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE020H	IMC8
1H	ditto
2H	ditto
3H	ditto
4H	IMC9
5H	ditto
6H	ditto
7H	ditto
8H	IMCA
9H	ditto
AH	ditto
BH	ditto
CH	IMCB
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE030H	IMCC
1H	ditto
2H	ditto
3H	ditto
4H	IMCD
5H	ditto
6H	ditto
7H	ditto
8H	IMCE
9H	ditto
AH	ditto
BH	ditto
CH	IMCF
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE040H	IVR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE060H	INTCLR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE100H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	ILEV
DH	ditto
EH	ditto
FH	ditto

Big-endian

[11] DMAC

ADR	Register name
FFFFE200H	CCR0
1H	ditto
2H	ditto
3H	ditto
4H	CSR0
5H	ditto
6H	ditto
7H	ditto
8H	SAR0
9H	ditto
AH	ditto
BH	ditto
CH	DAR0
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE210H	BCR0
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR0
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE220H	CCR1
1H	ditto
2H	ditto
3H	ditto
4H	CSR1
5H	ditto
6H	ditto
7H	ditto
8H	SAR1
9H	ditto
AH	ditto
BH	ditto
CH	DAR1
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE230H	BCR1
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR1
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE240H	CCR2
1H	ditto
2H	ditto
3H	ditto
4H	CSR2
5H	ditto
6H	ditto
7H	ditto
8H	SAR2
9H	ditto
AH	ditto
BH	ditto
CH	DAR2
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE250H	BCR2
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR2
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE260H	CCR3
1H	ditto
2H	ditto
3H	ditto
4H	CSR3
5H	ditto
6H	ditto
7H	ditto
8H	SAR3
9H	ditto
AH	ditto
BH	ditto
CH	DAR3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE270H	BCR3
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR3
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE280H	CCR4
1H	ditto
2H	ditto
3H	ditto
4H	CSR4
5H	ditto
6H	ditto
7H	ditto
8H	SAR4
9H	ditto
AH	ditto
BH	ditto
CH	DAR4
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE290H	BCR4
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR4
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2A0H	CCR5
1H	ditto
2H	ditto
3H	ditto
4H	CSR5
5H	ditto
6H	ditto
7H	ditto
8H	SAR5
9H	ditto
AH	ditto
BH	ditto
CH	DAR5
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2B0H	BCR5
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR5
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

Big-endian

ADR	Register name
FFFFE2C0H	CCR6
1H	ditto
2H	ditto
3H	ditto
4H	CSR6
5H	ditto
6H	ditto
7H	ditto
8H	SAR6
9H	ditto
AH	ditto
BH	ditto
CH	DAR6
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2D0H	BCR6
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR6
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2E0H	CCR7
1H	ditto
2H	ditto
3H	ditto
4H	CSR7
5H	ditto
6H	ditto
7H	ditto
8H	SAR7
9H	ditto
AH	ditto
BH	ditto
CH	DAR7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2F0H	BCR7
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR7
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE300H	DCR
1H	ditto
2H	ditto
3H	ditto
4H	RSR
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	DHR
DH	ditto
EH	ditto
FH	ditto

Big-endian

[12] CS/WAIT controller

ADR	Register name
FFFFE400H	BMA0
1H	ditto
2H	ditto
3H	ditto
4H	BMA1
5H	ditto
6H	ditto
7H	ditto
8H	BMA2
9H	ditto
AH	ditto
BH	ditto
CH	BMA3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE410H	BMA4
1H	ditto
2H	ditto
3H	ditto
4H	BMA5
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE480H	B01CS
1H	ditto
2H	ditto
3H	ditto
4H	B23CS
5H	ditto
6H	ditto
7H	ditto
8H	B45CS
9H	ditto
AH	ditto
BH	ditto
CH	BEXCS
DH	ditto
EH	ditto
FH	ditto

[13]Access control

ADR	Register name
FFFFE500H	
1H	
2H	
3H	PFBWAIT
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] Security control

ADR	Register name
FFFFE510H	DSUSEC1
1H	ditto
2H	ditto
3H	ditto
4H	DSUSEC2
5H	ditto
6H	ditto
7H	ditto
8H	ROMSEC1
9H	
AH	
BH	
CH	ROMSEC2
DH	
EH	
FH	

[15] FLASH control

ADR	Register name
FFFFE520H	FLCS
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] ROM correction

ADR	Register name
FFFFE540H	ADDREG0
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG1
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG2
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE550H	ADDREG4
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG5
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG6
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE560H	ADDREG8
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG9
5H	ditto
6H	ditto
7H	ditto
8H	ADDREGA
9H	ditto
AH	ditto
BH	ditto
CH	ADDREGB
DH	ditto
EH	ditto
FH	ditto

Big-endian

[17] Clock timer

ADR	Register name
FFFFE700H	RTCFLG
1H	ditto
2H	ditto
3H	ditto
4H	RTCCR
5H	ditto
6H	ditto
7H	ditto
8H	RTCREG
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE710H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[18] CG

ADR	Register name
FFFFEE00H	SYSCR3
1H	SYSCR2
2H	SYSCR1
3H	SYSCR0
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFEE10H	IMCGA
1H	ditto
2H	ditto
3H	ditto
4H	IMCGB
5H	ditto
6H	ditto
7H	ditto
8H	IMCGC
9H	ditto
AH	ditto
BH	ditto
CH	IMCGD
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFEE20H	EICRCG
1H	ditto
2H	ditto
3H	ditto
4H	NMIFLG
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

22. Electrical Characteristics

The letter x in equations presented in this chapter represents the cycle period of the fsys clock selected through the programming of the SYSCR1.SYSCK bit. The fsys clock may be derived from either the high-speed or low-speed crystal oscillator. The programming of the clock gear function also affects the fsys frequency. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.SYSCK = 0) and a clock gear factor of 1/fc (SYSCR1.GEAR[2:0] = 000).

22.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		Vcc2 (Core)	− 0.3 to 3.0	V
		Vcc3（I/O）	− 0.3 to 3.9	
		AVCC（A/D）	− 0.3 to 3.9	
		BVCC	− 0.3 to 3.9	
Supply voltage		V_{IN}	− 0.3 to V_{CC}+0.3	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	-5	
	Total	ΣI_{OH}	50	
Power dissipation (Ta = 85°C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	℃
Storage temperature		T_{STG}	−40 to 125	℃
Operating temperature	Except during flash W/E	T_{OPR}	-20 to 85	℃
	During flash W/E		0 to 70	
Write/erase cycles		N_{EW}	100	cycle

V_{CC15}＝DVCC15＝CVCC15＝FVCC15、V_{CC}3＝DVCC3n（n＝0 to 4）、AVCC＝AVCC3m（m＝1 to 2） V_{SS}＝DVSS＊＝AVSS＊＝CVSS＝FVSS

Note: The Absolute Maximum Rating is a rating that must never be exceeded, even for an instant. Not a single Absolute Maximum Rating value can be exceeded. If any Absolute Maximum Rating value is exceeded, the product may be damaged or weakened, or damage or combustion may cause personal injury. Always be sure to design your application devices so the Absolute Maximum Rating is never exceeded.

22.2 DC Electrical Characteristics (1/3)

Ta＝－20 to 85℃

Parameter		Symbol	Conditions	Min	Typ (Note 1)	Max	Unit
Supply voltage CVCC15＝DVCC15 CVSS＝DVSS＝0V		DVCC15	fosc = 8 to 13.5MHz fs = 30kHz to 34kHz fsys = 30kHz to 54MHz PLLOFF="1"	1.35		1.65	V
		BVCC	fsys = 16kHz to 54MHz	1.8		3.3	
		DVCC3n (n＝0 to 4)	fsys = 4 to 54MHz	1.65		3.3	
Low-level input voltage	P7 to P9 (Used as a port)	V_{IL1}	2.7V≦AVCC32≦AVCC31≦3.3V	−0.3		0.3AVCC31 0.3AVCC32	V
	Normal port	V_{IL2}	1.65V≦DVCC3n≦3.3V （n=0 to 4）			0.3DVCC3n	
			1.8V≦BVCC≦3.3V			0.3BVCC	
	Schmitt-Triggered port	V_{IL3}	1.65V≦DVCC3n≦3.3V（n=0 to 4） 1.8V≦BVCC≦3.3V			0.2DVCC3n 0.2BVCC	
			1.35V≦DVCC15≦1.65V			0.1DVCC15	
	X1	V_{IL4}	1.35V≦CVCC15≦1.65V			0.1CVCC	
	XT1	V_{IL5}	1.8V≦BVCC≦3.3V			0.1CVCC	

Note1: BVCC ：Normal mode 2.3V to 3.3V,BACKUP mode 1.8V to 3.3V

Ta＝－20 to 85℃

Parameter		Symbol	Conditions		Min.	Typ (Note 1)	Max.	Unit
High-level input voltage	P7 to P9 (Used as a port)	V_{IH1}	2.7V≦AVCC32≦AVCC31≦3.3V		0.7AVCC31 0.7AVCC32		DVCC3n+0.3 BVCC+0.3 DVCC15+0.2 CVCC+0.2	V
	Normal port	V_{IH2}	1.65V≦DVCC3n≦3.3V（n=0 to 4）		0.7DVCC3n			
			1.8V≦BVCC≦3.3V		0.7BVCC			
	Schmitt-Triggered port	V_{IH3}	1.65V≦DVCC3n≦3.3V（n=0 to 4） 1.8V≦BVCC≦3.3V		0.8DVCC3n 0.8BVCC			
			1.35V≦DVCC15≦1.65V		0.9DVCC15			
	X1	V_{IH4}	1.35V≦CVCC≦1.65V		0.9CVCC			
	XT2	V_{IH4}	1.8V≦BVCC≦3.3V		0.9BVCC			
Low-level output voltage		V_{OL}	I_{OL} = 2mA	DVCC3n≧2.7V			0.4	V
			I_{OL} = 500μA	DVCC3n ＜ 2.7V			0.2DVCC3n≦0.4	
High-level output voltage		V_{OH}	I_{OH} = −2mA	DVCC3n ≧ 2.7V	2.4			
			I_{OH} = −500μA	DVCC3n ＜ 2.7V	0.8DVCC3n			

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3n =3.0V, BVCC=3.0V, AVCC3m=3.3V, unless otherwise noted

22.3 DC Electrical Characteristics (2/3)

Ta = −20 to 85°C

Parameter	Symbol	Conditions	Min.	Typ (Note 1)	Max.	Unit
Input leakage current	I_{LI}	$0.0 \leqq V_{IN} \leqq$ DVCC15 $0.0 \leqq V_{IN} \leqq$ BVCC $0.0 \leqq V_{IN} \leqq$ DVCC3n (n=0 to 4) $0.0 \leqq V_{IN} \leqq$ AVCC31 $0.0 \leqq V_{IN} \leqq$ AVCC32		0.02	±5	μA
Output leakage current	I_{LO}	$0.2 \leqq V_{IN} \leqq$ DVCC15−0.2 $0.2 \leqq V_{IN} \leqq$ BVCC−0.2 $0.2 \leqq V_{IN} \leqq$ DVCC3n−0.2 (n=0 to 4) $0.2 \leqq V_{IN} \leqq$ AVCC31−0.2 $0.2 \leqq V_{IN} \leqq$ AVCC32−0.2		0.05	±10	
Power-down voltage (STOP mode RAM backup)	V_{STOP} (DVCC15)		1.35		1.65	V
	V_{STOP1} (BVCC)		1.8		3.3	
	V_{STOP2} (AVCC3)	V_{IL1} = 0.3AVCC31,32 V_{IH1} = 0.7AVCC31,32	2.7		3.6	
	V_{STOP3} (DVCC3)	V_{IL2} = 0.3DVCC3n, V_{IL3} = 0.1DVCC3n V_{IH2} = 0.7DVCC3n, V_{IH3} = 0.9DVCC3n (n=0 to 4)	1.65		3.3	
Pull-up resister at Reset	RRST	DVCC15 = 1.5V ± 0.15V	20	50	150	kΩ
Schmitt-Triggered port	VTH	1.65V≦DVCC3n≦3.3V (n=0 to 4) 1.8V≦BVCC≦3.3V 1.35V≦DVCC15≦1.65V	0.3	0.6		V
Programmable pull-up/ pull-down resistor	PKH	DVCC3n = 1.65V to 3.3V (n=0 to 4) DVCC15 = 1.35V to 1.65V BVCC = 1.8V to 3.3V	20	50	150	kΩ
Pin capacitance (Except power supply pins)	C_{IO}	Fc = 1MHz			10	pF

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3n =3.0V, BVCC=3.0V, AVCC3m=3.3V, unless otherwise noted

22.4 DC Electrical Characteristics (3/3)

DVCC15＝CVCC15＝FVCC15＝1.35V to 1.65V, DVCC3n＝FVCC3＝2.7V to 3.3V, AVCC3m＝2.7V to 3.3V, BVCC=1.8V to 3.3V

Ta＝－20 to 85℃ （n＝0 to 4、m＝1,2）

Parameter	Symbol	Conditions	Min.	Typ. (Note 1)	Max.	Unit
NORMAL(Note 2): Gear = 1/1	ICC	Fsys = 54 MHz (fosc = 13.5 MHz, PLLOFF="DVCC15")		50	60	mA
IDLE(Doze)				18	28	
IDLE(Halt)				14	23	
SLOW		Fsys = 32.768kHz (fs = 32.768kHz)		300	970	μA
SLEEP		Fsys = 32.768kHz (fs = 32.768kHz)		100	950	μA
STOP		DVCC15 = CVCC15 =1.35 to 1.65V BVCC = 1.8 to 3.3V DVCC3n = 1.65 to 3.3V AVCC3m = 2.7 to 3.3V		90	900	μA
BACKUP		BVCC = 1.8 to 3.3V		3	5	μA

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3n =3.0V, BVCC=3.0V, AVCC3m=3.3V, unless otherwise noted

Note 2: Measured with the CPU dhrystone operating, all I/O peripherals channel on, and 16-bit external bus operated with 4 system clocks.

Note 3: The supply current flowing through the DVCC15、BVCC、DVCC3n、CVCC15 and AVCC3m pins is included in the digital supply current parameter (ICC).

22.5 10-bit ADC Electrical Characteristics

DVCC15=CVCC15=1.35V to 1.65V, AVCC3m=2.7V to 3.3V,
AVSS=DVSS, Ta=－20 to 85℃

Parameter		Symbol	Conditions	Min	Typ	Max	Unit
Analog reference voltage (+)		VREFH		2.7 AVCC3m−0.3	 AVCC	3.3 AVCC3m+0.3	V
Analog reference voltage (−)		VREFL		AVSS	AVSS	AVSS+0.2	V
Analog input voltage		VAIN		VREFL		VREFH	V
Analog supply current	A/D conversion	IREF	AVCC3m = VREFH = 3.0V ± 0.3V DVSS = AVSS = VREFL		1.15	1.8	mA
	Non-A/D conversion		AVCC3m = VREFH = 2.7 to 3.3V DVSS = AVSS = VREFL		0.1	10.0	μA
Analog input capacitance		—			1.0	2.0	pF
Analog input impedance		—			2.0	3.5	kΩ
INL error		—	AVCC3m = VREFH = 3.0 V ±0.3 V DVSS = AVSS = VREFL AIN resistance < 1.3kΩ AIN load capacitance < 20 pF AVCCm load capacitance ≥ 10 μF VREFH load capacitance ≥ 10 μF Conversion time ≥ 7.85 μs		±2	3	LSB
DNL error		—			±1	3	LSB
Offset error		—			±2	3	LSB
Gain error		—			±2	4	LSB

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

Note 2: The supply current flowing through the AVCC3m pin is included in the digital supply current parameter (ICC).

22.6 AC Electrical Characteristics

[1] Separate Bus mode

(1)DVCC15＝CVCC15＝FVCC15＝1.35V to 1.65V, DVCC3n＝FVCC3＝2.3V to 3.3V

SYSCR3<ALESEL> = “0”, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	18.5				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	(1+ALE)x-20		17		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-14		4.5		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x(2+TW+ALE)-42		50.5	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x(1+TW)-28		27.5	ns
6	$\overline{RD}$ width low	t_{RR}	x(1+TW)-10		45.5		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x-15		3.5		ns
9	$\overline{WR}$ /$\overline{HWR}$ width low	t_{WW}	x(1+TW)-10		45.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		12.3		12.3	ns
11	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x(1+TW)-18		37.5		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x-15		3.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x+(ALE)x+(TW-1)x -30		25.5	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x(TW-3)+7	x(TW-1)-17	25.5	38.5	ns

Note 1: No. 1 to 13:

Internal 2 wait insertion , ALE “1” Clock, @54MHz

TW = (Auto wait insertion + 2N)

No. 14 :

Conditions (Auto wait insertion + 2N)

TW = 2 + 2*1 = 4

AC measurement conditions:

Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

(2) DVCC15＝CVCC15＝FVCC15＝1.35V to 1.65V, DVCC3n＝FVCC3＝1.65V to 1.95V

SYSCR3<ALESEL> = “0”, 2programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	18.5				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	(1+ALE)x-20		17		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-7		11.5		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x(2+TW+ALE)-42		50.5	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x(1+TW)-28		27.5	ns
6	$\overline{RD}$ width low	t_{RR}	x(1+TW)-10		45.5		ns
7	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0-A23 output	t_{RAE}	x-15		3.5		ns
9	$\overline{WR}$ /$\overline{HWR}$ width low	t_{WW}	x(1+TW)-10		45.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		12.3		12.3	ns
11	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x(1+TW)-18		37.5		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x-15		3.5		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x+(ALE)x+(TW-1)x -30		25.5	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x(TW-3)+7	x(TW-1)-17	25.5	38.5	ns

Note 1: No. 1 to 13:

Internal 2 wait insertion，ALE“1”Clock, @54MHz

TW = (Auto wait + 2N)

No. 14 :

Conditions (Auto wait insertion + 2N)

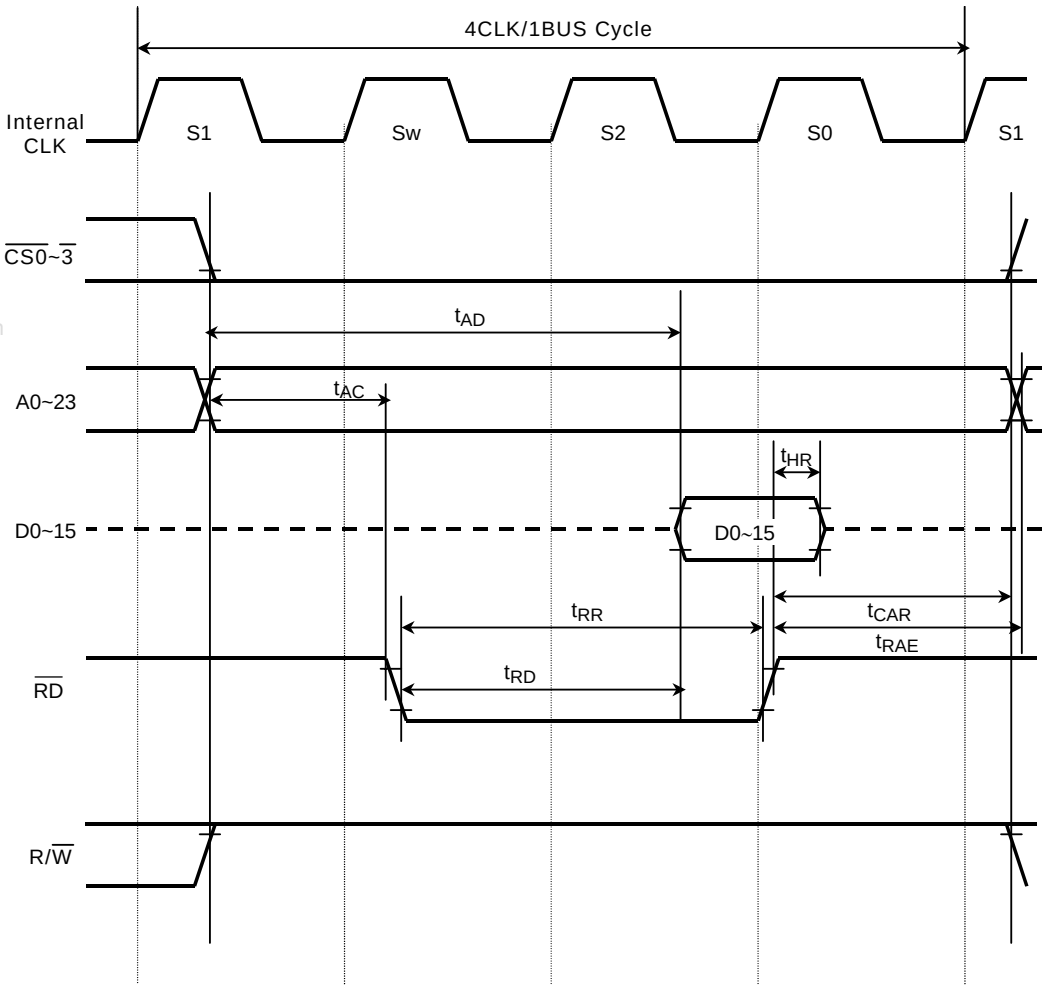
TW = 2 + 2*1 = 4

AC measurement conditions:

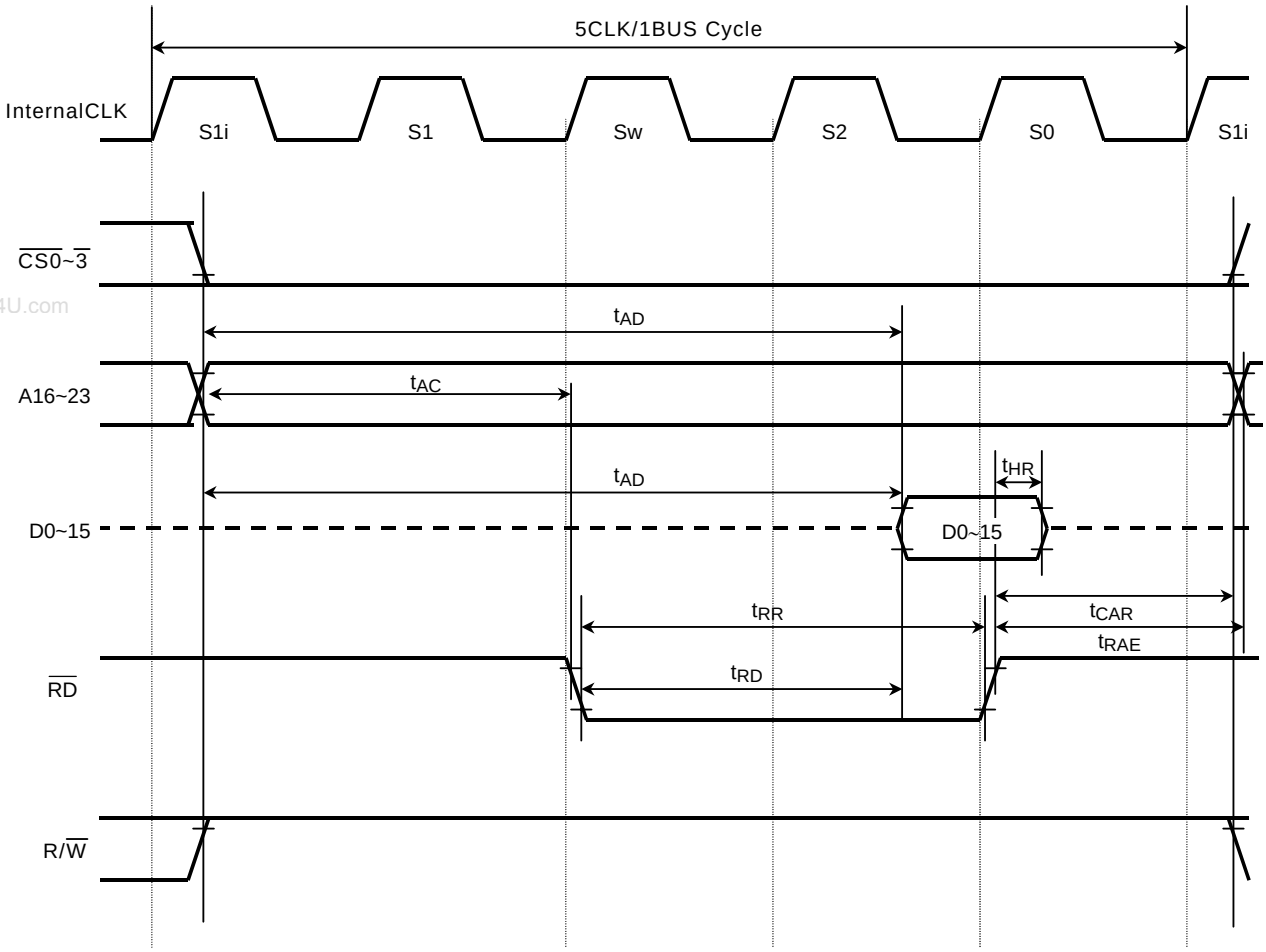
Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

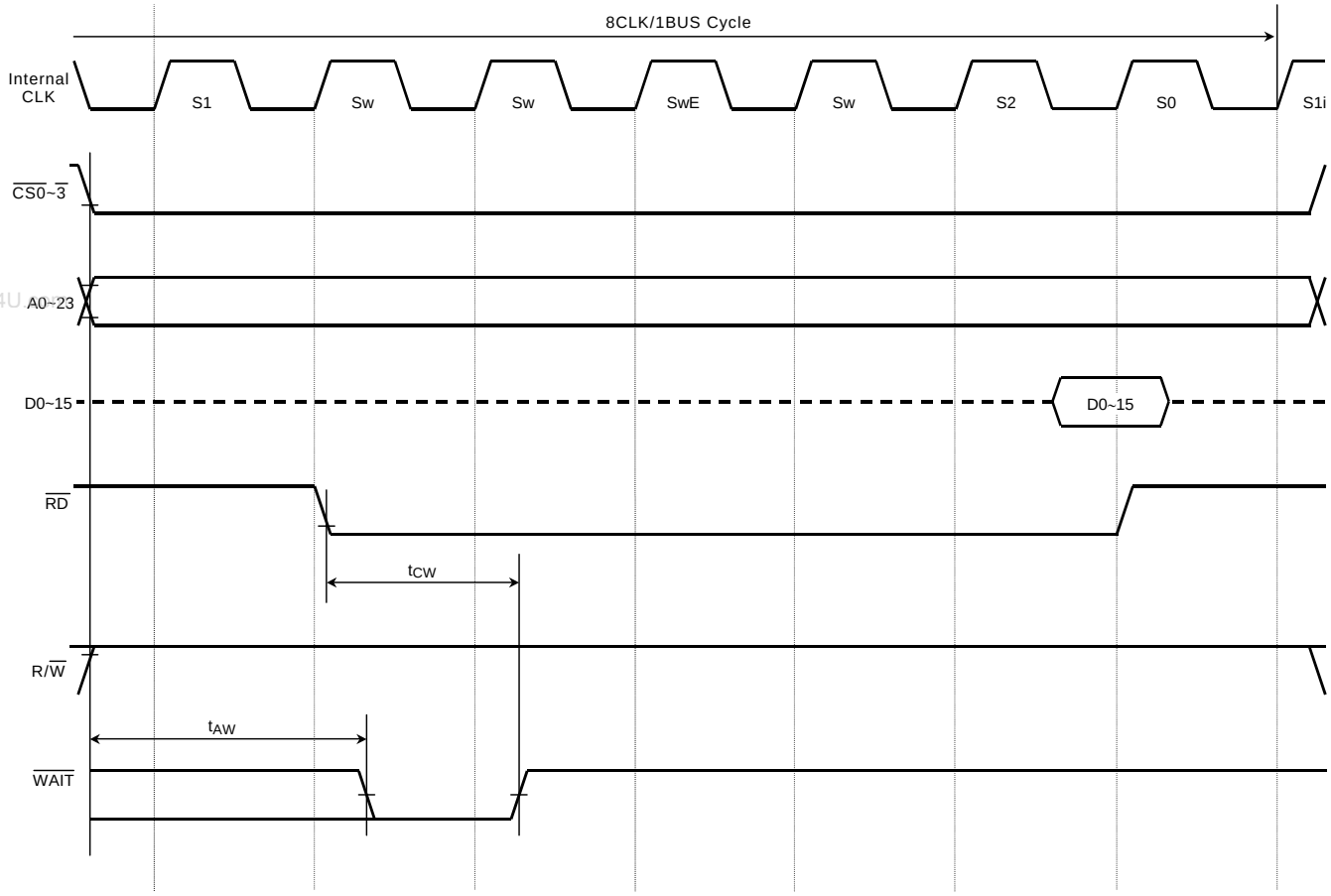
(1) Read cycle timing (SYSCR3<ALESEL> = 0, 1 programmed wait state)



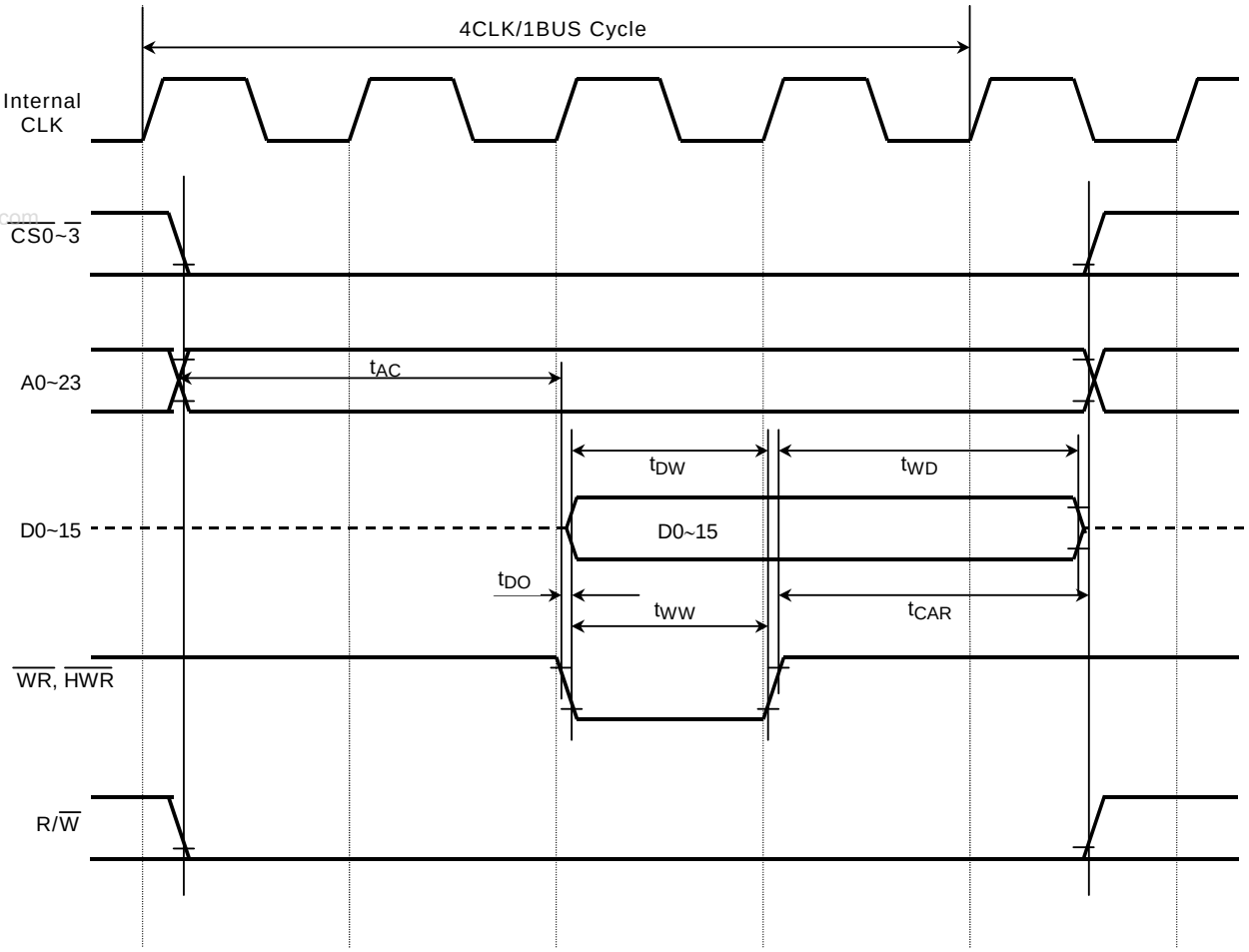
(2) Read cycle timing (SYSCR3<ALESEL> = 1, 1 programmed wait state)



(2) Read cycle timing SYSCR3<ALESEL> = 1, 4 externally generated wait states with N = 1)



(4) Write cycle timing (SYSCR3<ALESEL> = 1, zero wait sate)



[2] Multiplex Bus mode

(1) DVCC15=CVCC15=FVCC15=1.35V to 1.65V, DVCC3n=FVCC3=2.3V to 3.3V

1. ALE width = 1 clock cycle, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	18.5				ns
2	A0-A15 valid to ALE low	t_{AL}	(ALE)x-12		6.5		ns
3	A0-A15 hold after ALE low	t_{LA}	x-8		10.5		ns
4	ALE pulse width high	t_{LL}	(ALE)x-6		12.5		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x-8		10.5		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x-15		3.5		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x-20		17.0		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x-20		17.0		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-14		4.5		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x(2+TW+ALE)-42		50.5	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x(2+TW+ALE)-42		50.5	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x(1+TW)-28		27.5	ns
13	$\overline{RD}$ width low	t_{RR}	x(1+TW)-10		45.5		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x-15		3.5		ns
16	$\overline{WR}/\overline{HWR}$ width low	t_{WW}	x(1+TW)-10		45.5		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x(1+TW)-18		37.5		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x-15		3.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x+(ALE)x+(TW-1)x-30		25.5	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x+(ALE)x+(TW-1)x-30		25.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x(TW-3)+7	x(TW-1)-17	25.5	38.5	ns

Note 1: No. 1 to 20:

Internal 2 wait insertion , ALE “1” Clock, @54MHz

TW = (Auto wait insertion + 2N)

No. 21 :

Conditions (Auto wait + 2N)

TW = 2 + 2*1 = 4

AC measurement conditions:

Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

(2) DVCC15＝CVCC15＝FVCC15＝1.35V to 1.65V, DVCC3n＝FVCC3＝1.65V to 1.95V

ALE width = 1 clock cycles, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	18.5				ns
2	A0-A15 valid to ALE low	t_{AL}	(ALE)x-12		6.5		ns
3	A0-A15 hold after ALE low	t_{LA}	x-8		10.5		ns
4	ALE pulse width high	t_{LL}	(ALE)x-6		12.5		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x-8		10.5		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x-15		3.5		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x-20		17.0		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x-20		17.0		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-7		11.5		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x(2+TW+ALE)-42		50.5	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x(2+TW+ALE)-42		50.5	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x(1+TW)-28		27.5	ns
13	$\overline{RD}$ width low	t_{RR}	x(1+TW)-10		45.5		ns
14	D0-D15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-A15 output	t_{RAE}	x-15		3.5		ns
16	$\overline{WR}/\overline{HWR}$ width low	t_{WW}	x(1+TW)-10		45.5		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x(1+TW)-18		37.5		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x-15		3.5		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x+(ALE)x+(TW-1)x-30		25.5	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x+(ALE)x+(TW-1)x-30		25.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x(TW-3)+7	x(TW-1)-17	25.5	38.5	ns

Note 1: No. 1 to 20:

Internal 2 wait insertion , ALE “1” Clock, @54MHz

TW = (Auto insert wait + 2N)

No. 21 :

Conditions (Auto 2 waits insertion + 2N)

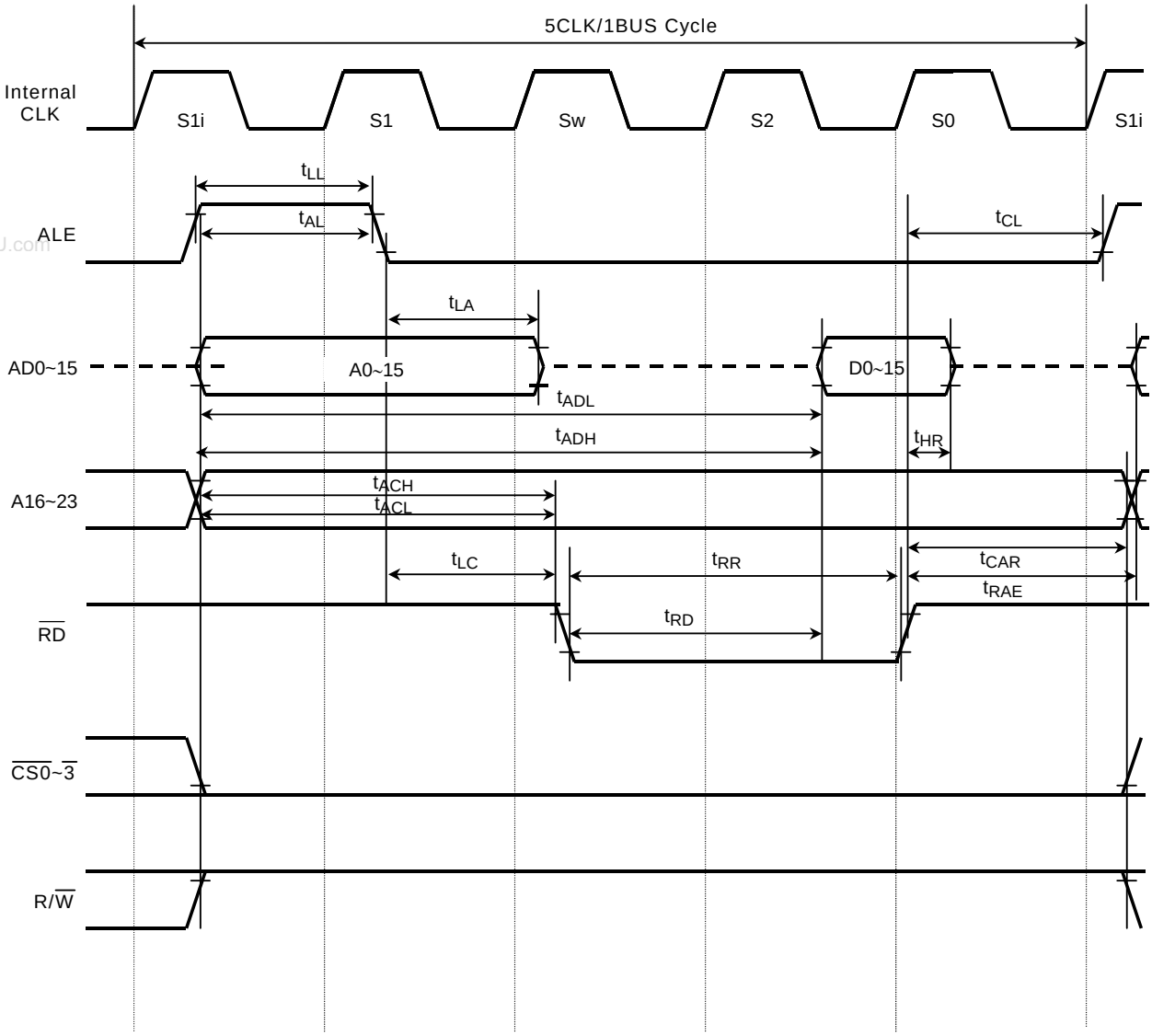
TW = 2 + 2*1 = 4

AC measurement conditions:

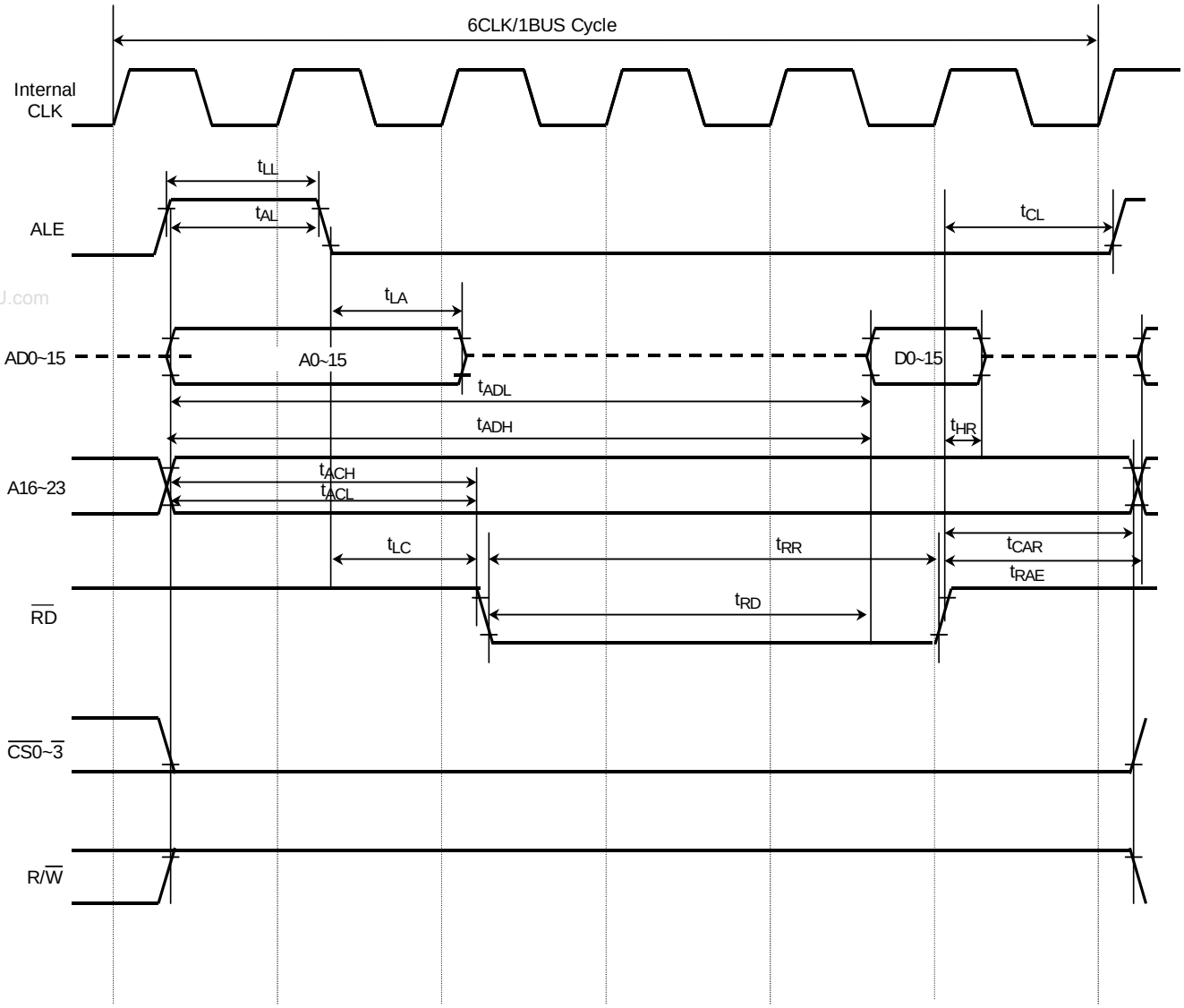
Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

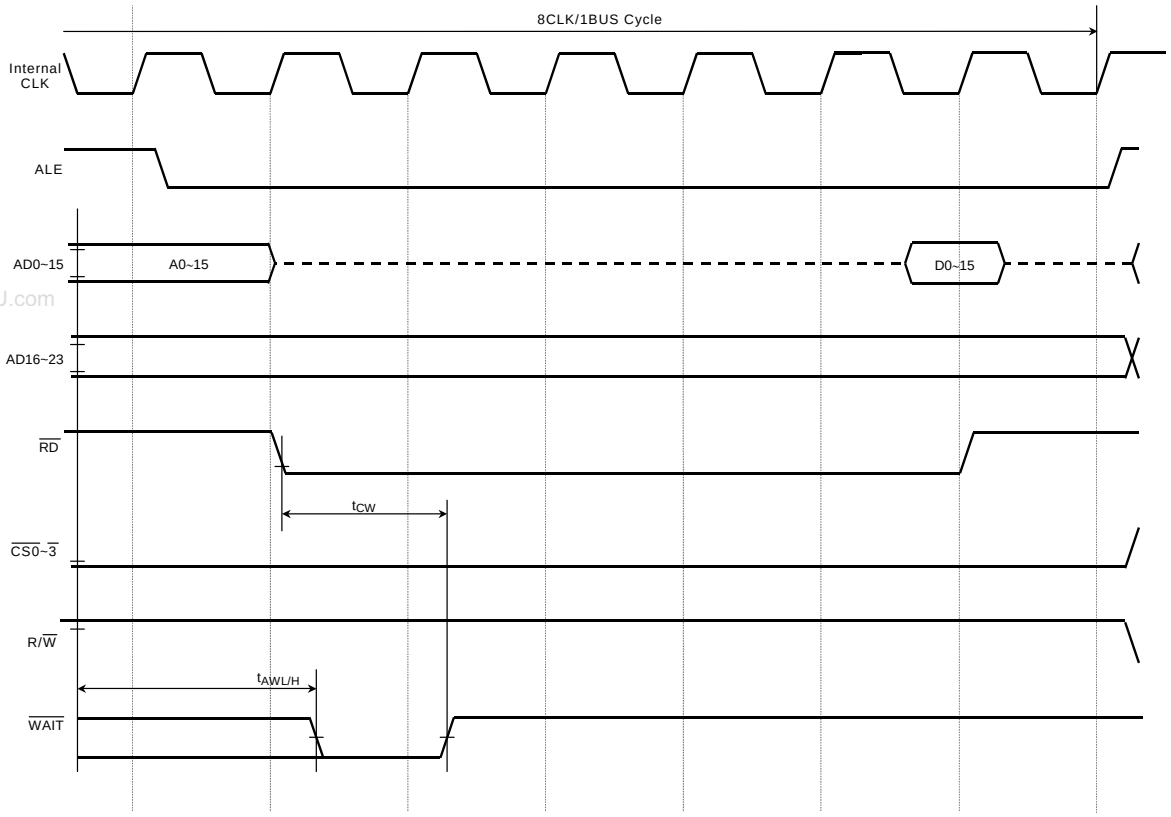
(1) Read cycle timing, ALE width = 1 clock cycle, 1 programmed wait state



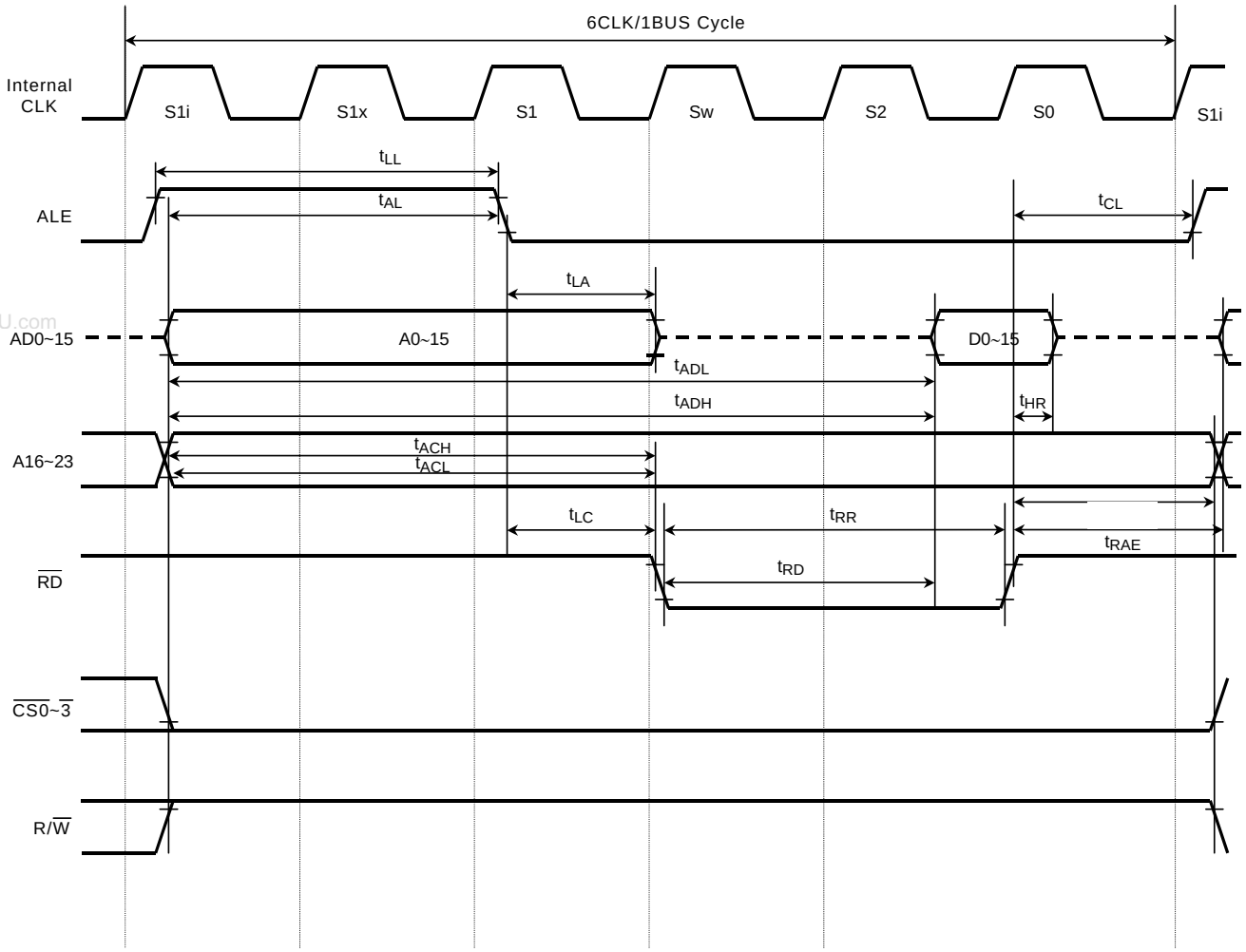
(2) Read cycle timing, ALE width = 1 clock cycle, 2 programmed wait state



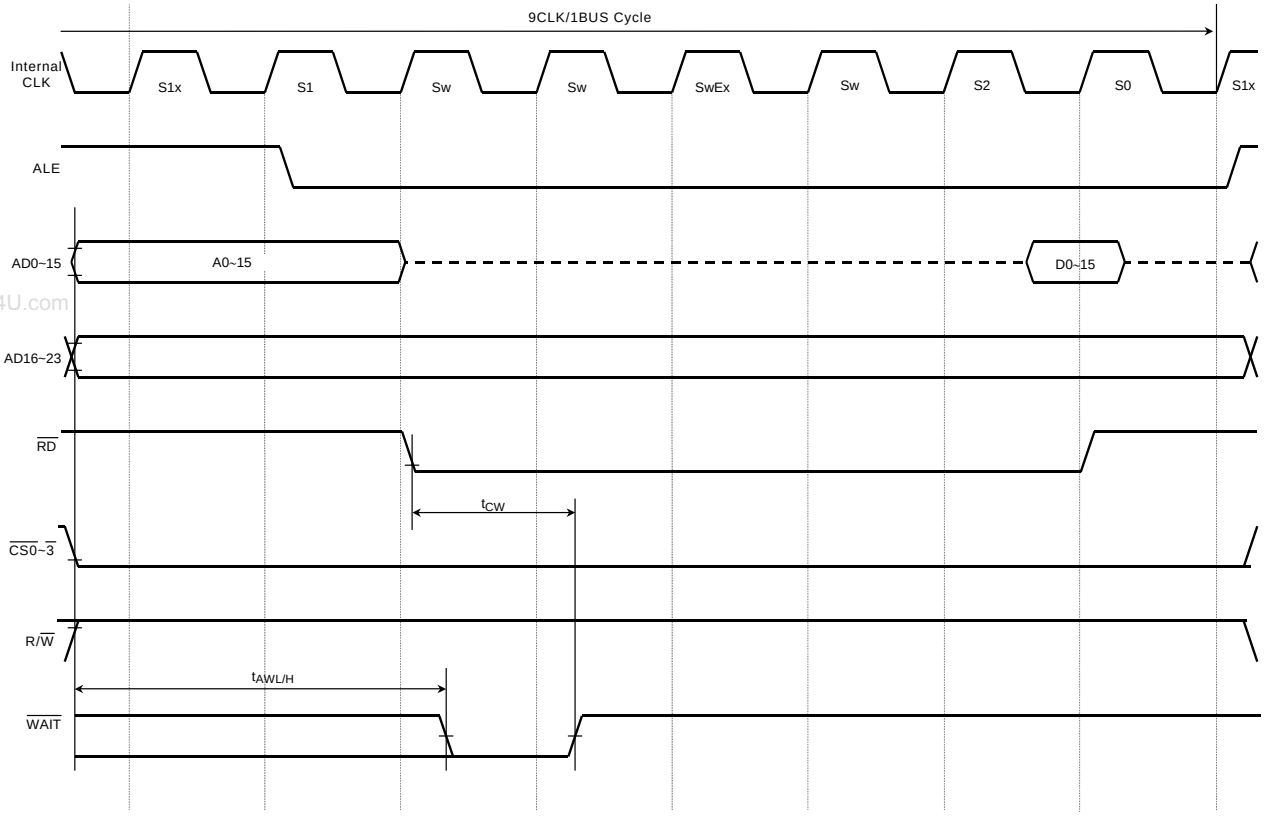
(3) Read cycle timing, ALE width = 1 clock cycle, 4 programmed wait state



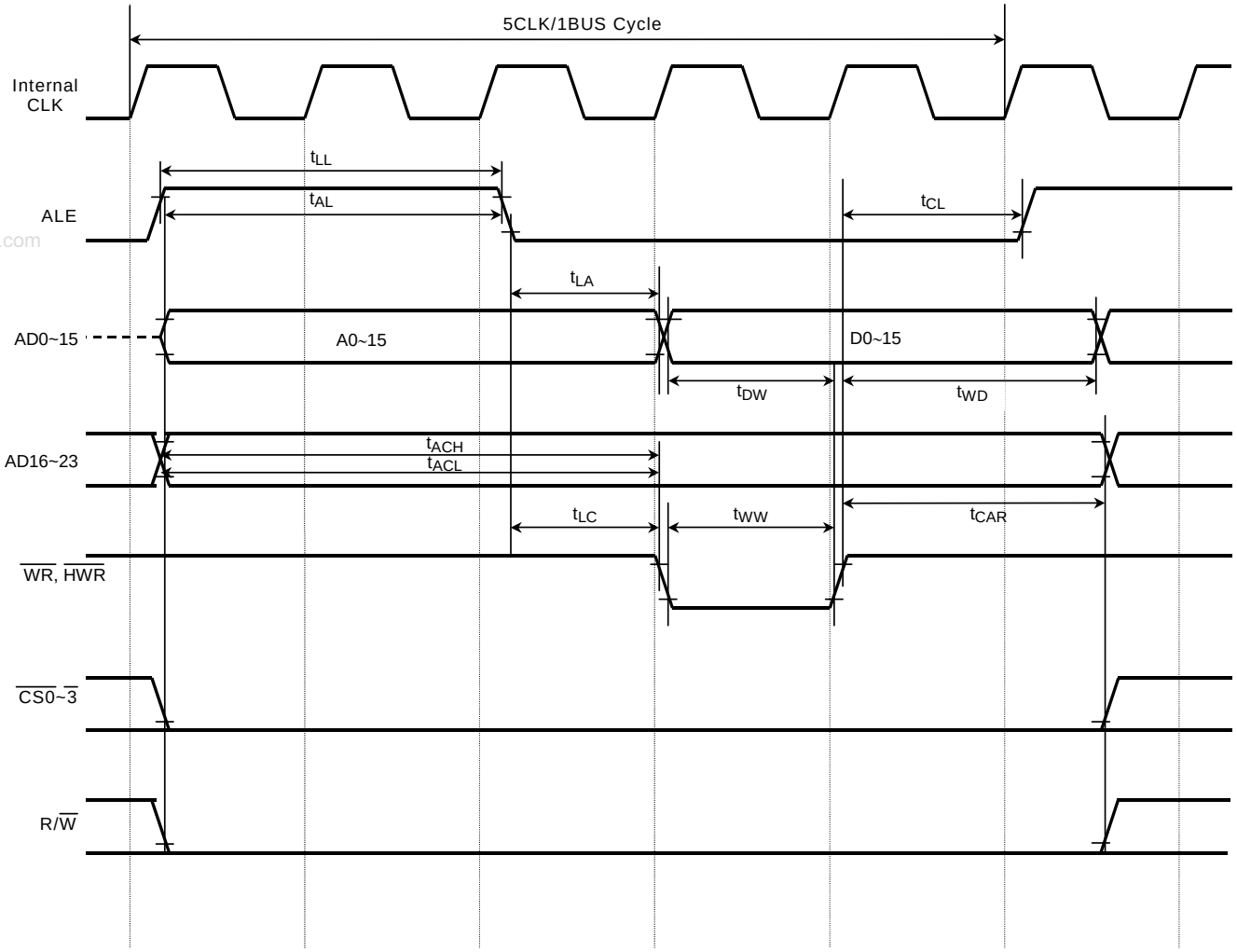
(4) Read cycle timing, ALE width = 2 clock cycle, 1 programmed wait state



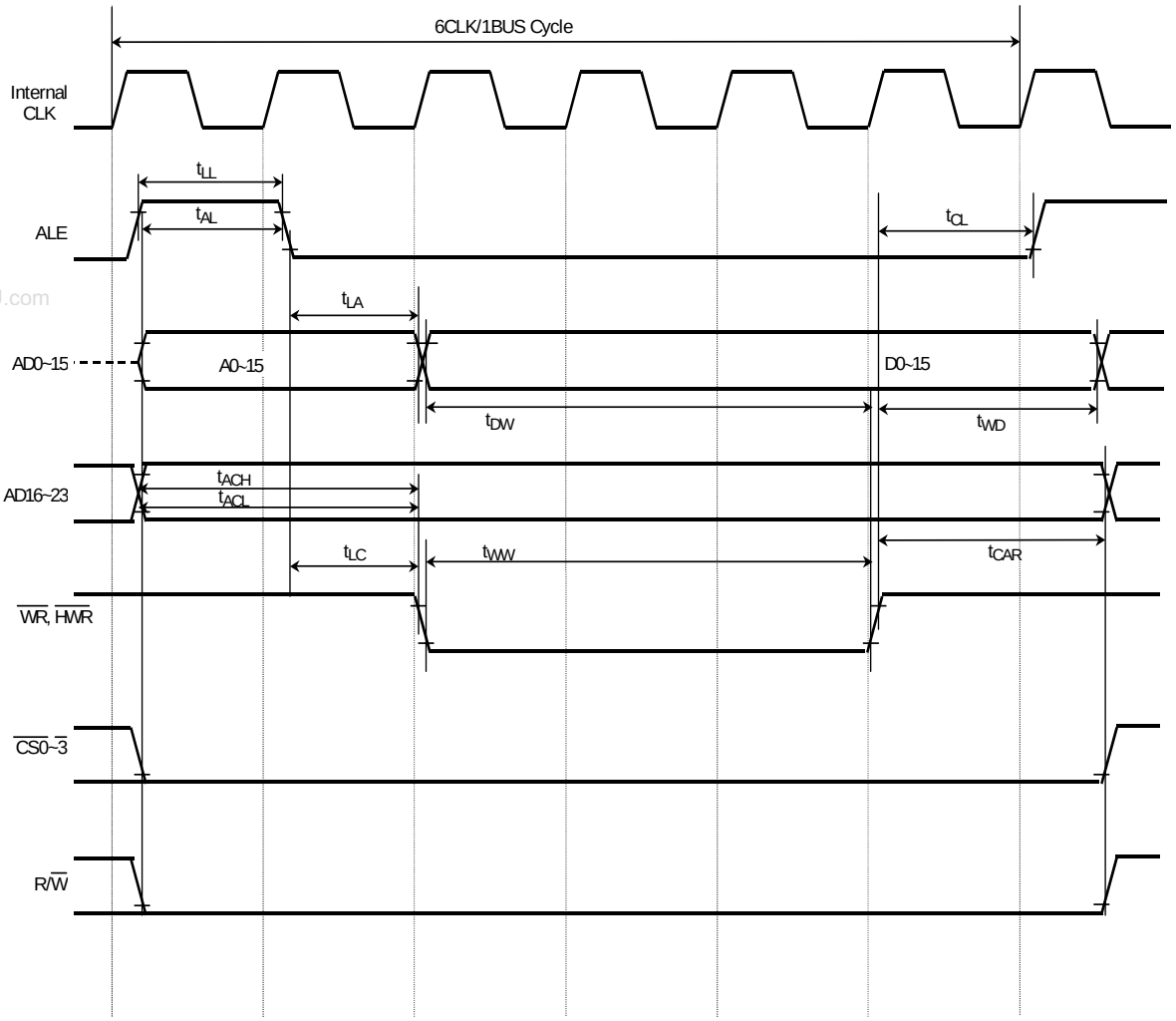
(5) Read cycle timing, ALE width = 2 clock cycle, 4 programmed wait state



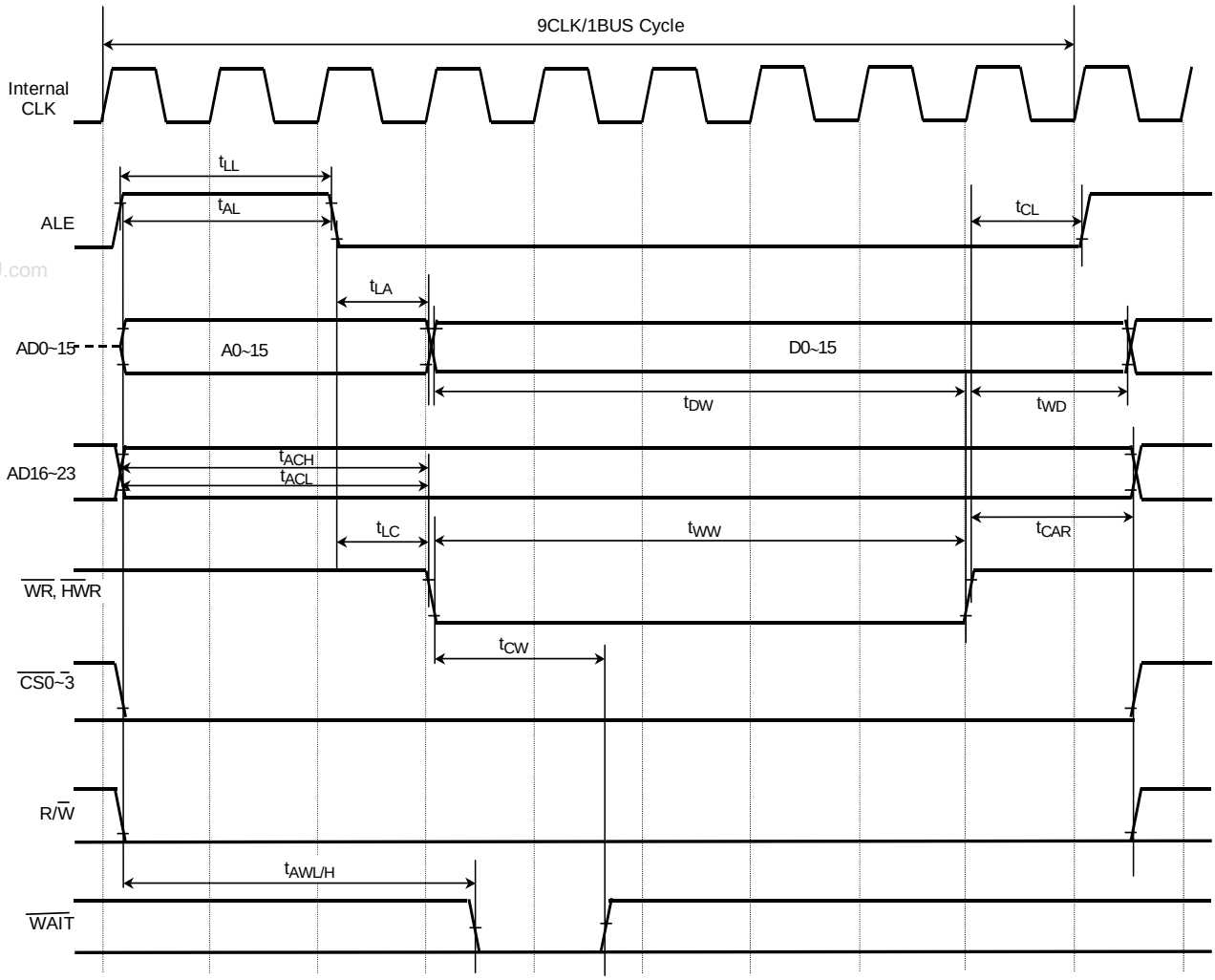
(6) Write cycle timing, ALE width = 2 clock cycles, zero wait state



(7) Write cycle timing, ALE width = 1 clock cycles, 2 wait state



(8) Write cycle timing, ALE width = 2 clock cycles, 4 wait state

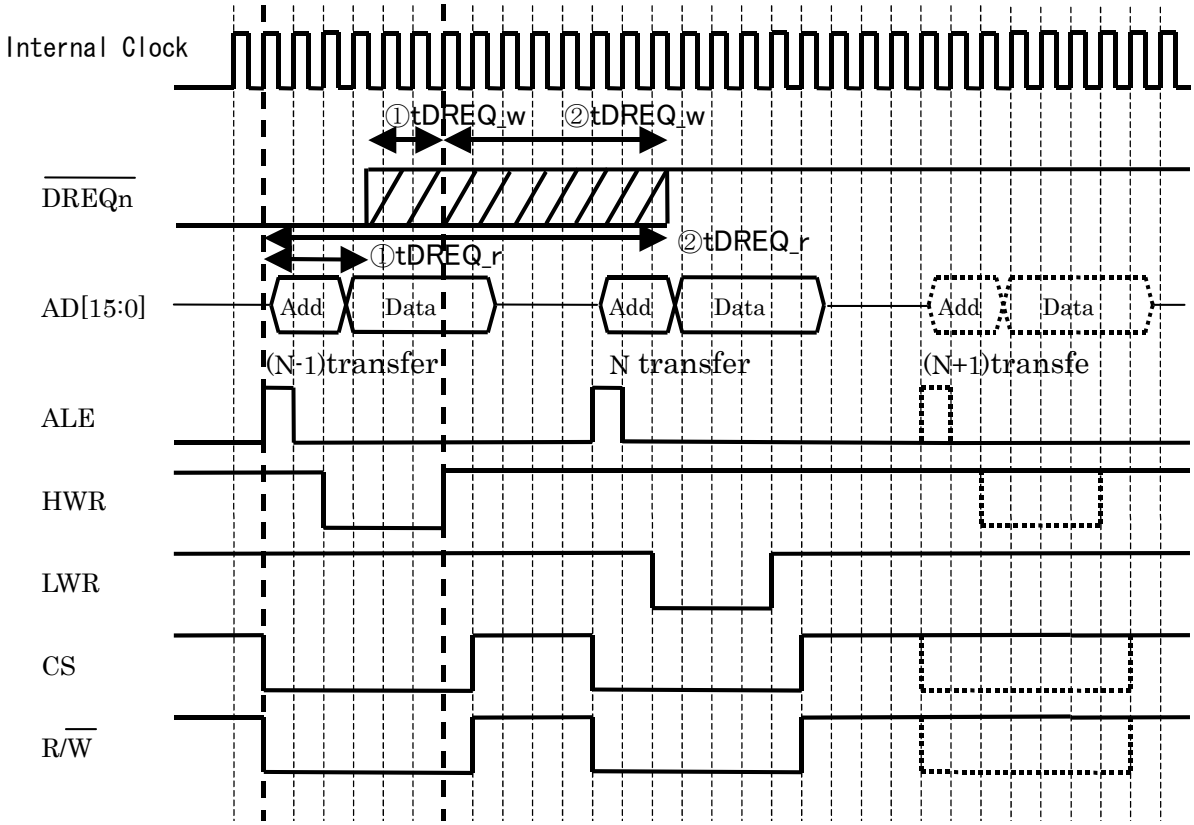


22.7 Transfer with DMA Request

The following shows an example of a transfer between the on-chip RAM and an external device in multiplex bus mode.

- 16-bit data bus width, non-recovery time
- Level data transfer mode
- Transfer size of 16 bits, device port size (DPS) of 16 bits
- Source/destination: on-chip RAM/external device

The following shows transfer operation timing of the on-chip RAM to an external bus during write operation (memory-to-memory transfer).



(1) Indicates the condition under which Nth transfer is performed successfully.
(2) Indicates the condition under which (N + 1)th transfer is not performed.

(1) DVCC15＝CVCC15＝FVCC15＝ 1.35V to 1.65V, AVCC3m＝FVCC3＝2.7V to 3.3V
DVCC33＝2.3V to 3.3V, DVCC30/31/32/34＝1.65V to 3.3V, Ta＝ －20 to 85°C （m＝1 to 2）

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			(1)Min	(2)Max	Min	Max	
2	$\overline{RD}$ asserted to $\overline{DREQn}$ negated (external device to on-chip RAM transfer)	tDREQ_r	(W+1)x	(2W＋ALE＋8)x －51	37	152.5	ns
3	$\overline{WR}$ / $\overline{HWR}$ rising to $\overline{DREQn}$ negated (on-chip RAM to external device transfer)	tDREQ_w	−(W+2)x	(5+WAIT)x－51.8	−55.5	59.2	ns

(2) DVCC15＝CVCC15＝FVCC15＝1.35V to 1.65V, AVCC3m =FVCC3＝2.7V to 3.3V
DVCC33＝1.65V to 1.95V, DVCC30/31/32/34＝1.65V to 3.3V, Ta＝－20 to 85°C （m＝1 to 2）

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			(1)Min	(2)Max	Min	Max	
2	$\overline{RD}$ asserted to $\overline{DREQn}$ negated (external device to on-chip RAM transfer)	tDREQ_r	(W+1) x	(2W＋ALE＋8) x－56	37	147.5	ns
3	$\overline{WR}$ / $\overline{HWR}$ rising to $\overline{DREQn}$ negated (on-chip RAM to external device transfer)	tDREQ_w	−(W+2)x	(5+WAIT)x－56.8	−55.5	54.2	ns

W: Number of wait-state cycles inserted. In the case of (2 + N) externally generated wait states with N = 1, W becomes 4

ALE: Apply ALE = ALE 1 clock, ALE = 1 for ALE 2 clock. The values in the above table are obtained with W = 1, ALE = 1.

22.8 Serial Channel Timing

(1) I/O Interface mode (DVCC3n = 1.65V to 3.3V)

In the table below, the letter x represents the fsys cycle period, which varies depending on the programming of the clock gear function.

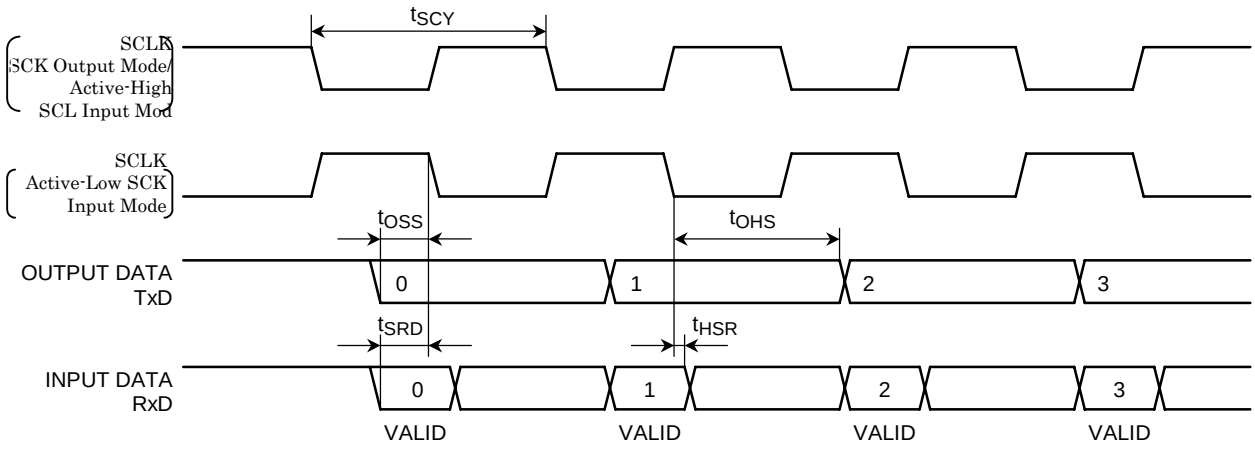
(1) SCLK input mode (SIO0 to SIO6)

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	12x		222		ns
SCLK Clock High width(input)	TscH	6x		111		ns
SCLK Clock Low width (input)	TscL	6x		111		ns
TxD data to SCLK rise or fall*	t_{OSS}	2x−30		6		ns
TxD data hold after SCLK rise or fall*	t_{OHS}	8x−15		129		ns
RxD data valid to SCLK rise or fall*	t_{SRD}	30		30		ns
RxD data hold after SCLK rise or fall*	t_{HSR}	2x+30		66		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

2. SCLK output mode (SIO0 to SIO6)

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	8x		222		ns
TxD data to SCLK rise or fall*	t_{OSS}	4x−10		62		ns
TxD data hold after SCLK rise or fall*	t_{OHS}	4x−10		62		ns
RxD data valid to SCLK rise or fall*	t_{SRD}	45		45		ns
RxD data hold after SCLK rise or fall*	t_{HSR}	0		0		ns



22.9 SBI Timing

(1) I2C mode

In the table below, the letters x represent the fsys periods, respectively.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBI0CR1.

Parameter	Symbol	Equation		Standard mode		Fast mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	t_{SCL}	0		0	100	0	400	kHz
Hold time for START condition	$t_{HD:STA}$			4.0		0.6		μs
SCL clock low width (Input) (Note 1)	t_{LOW}			4.7		1.3		μs
SCL clock high width (Output) (Note 2)	t_{HIGH}			4.0		0.6		μs
Setup time for a repeated START condition	$t_{SU:STA}$	(Note 5)		4.7		0.6		μs
Data hold time (Input) (Note 3, 4)	$t_{HD:DAT}$			0.0		0.0		μs
Data setup time	$t_{SU:DAT}$			250		100		ns
Setup time for STOP condition	$t_{SU:STO}$			4.0		0.6		μs
Bus free time between STOP and START conditions	t_{BUF}	(Note 5)		4.7		1.3		μs

Note 1: SCL clock low width (output) is calculated with: $(2^{n-1}+58)/(fsys/2)$

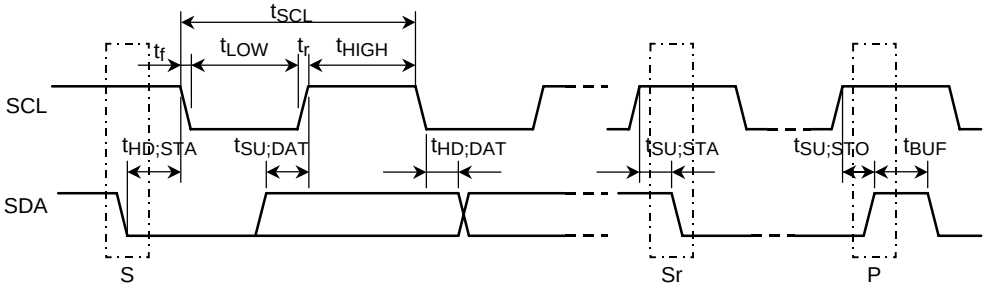
Note 2: SCL clock high width (output) is calculated with $(2^{n-1}+12)/(fsys/2)$

Notice: On I^2C-bus specification, Maximum Speed of Standard mode is 100KHz ,Fast mode is 400Khz. Internal SCL clock Frequency setting should be shown above Note1 & Note2.

Note 3: The output data hold time is equal to 12x

Note 4: The Philips I^2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the fall edge of SCL. However, the 19A64 SBI does not satisfy this requirement. Also, the output buffer for SCL does not incorporate slope control of the falling edges; therefore, the equipment manufacturer should design so that the input data hold time shown in the table is satisfied, including tr/tf of the SCL and SDA lines.

Note 5: Software-dependent



S: START condition
Sr: Repeated START condition
P: STOP condition

(2) Clock-Synchronous 8-Bit SIO mode

In the tables below, the letters x represent the fsys cycle periods, respectively. The letter n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBI0CR1.

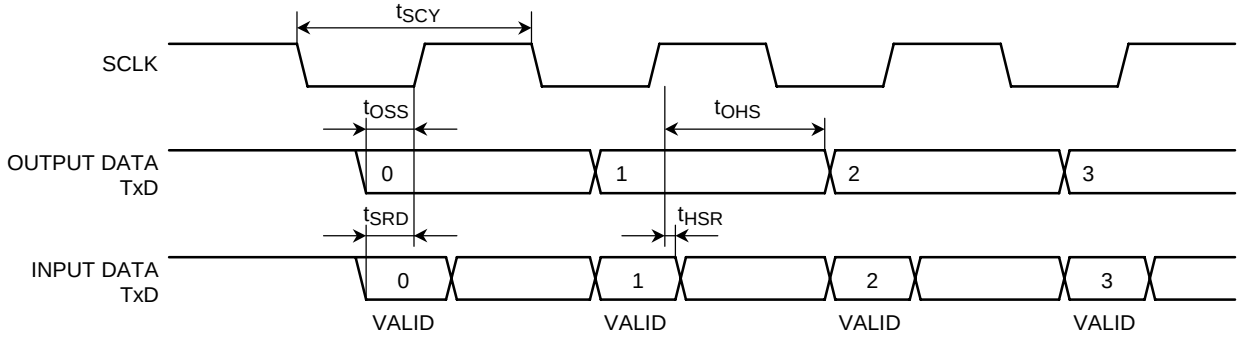
The electrical specifications below are for an SCK signal with a 50% duty cycle.

③ SCK Input mode

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
SCK period	t_{SCY}	16x		296		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - (6x + 30)$		7		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) + 4x$		222		ns
SI data valid to SCK rise	t_{SRD}	0		0		ns
SI data hold after SCK rise	t_{HSR}	4x + 10		84		ns

④ SCK Output mode

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
SCK period (programmable)	t_{SCY}	16x		296		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - 20$		128		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) - 20$		128		ns
SI data valid to SCK rise	t_{SRD}	2x + 30		67		ns
SI data hold after SCK rise	t_{HSR}	0		0		ns



22.10 Event Counter

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2X + 100		137		ns
Clock high pulse width	t_{VCKH}	2X + 100		137		ns

22.11 Timer Capture

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2X + 100		137		ns
High pulse width	t_{CPH}	2X + 100		137		ns

22.12 General Interrupts

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTA	t_{INTAL}	X + 100		118.5		ns
High pulse width for INT0-INTA	t_{INTAH}	X + 100		118.5		ns

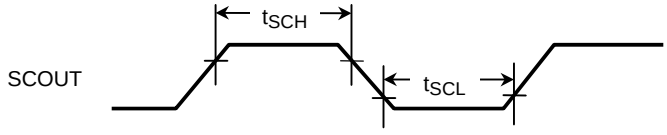
22.13 $\overline{\text{NMI}}$ and STOP /SLEEP Wake-up Interrupts

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for $\overline{\text{NMI}}$ and INT0-INT4	t_{INTBL}	100		100		ns
High pulse width for INT0-INT4	t_{INTBH}	100		100		ns

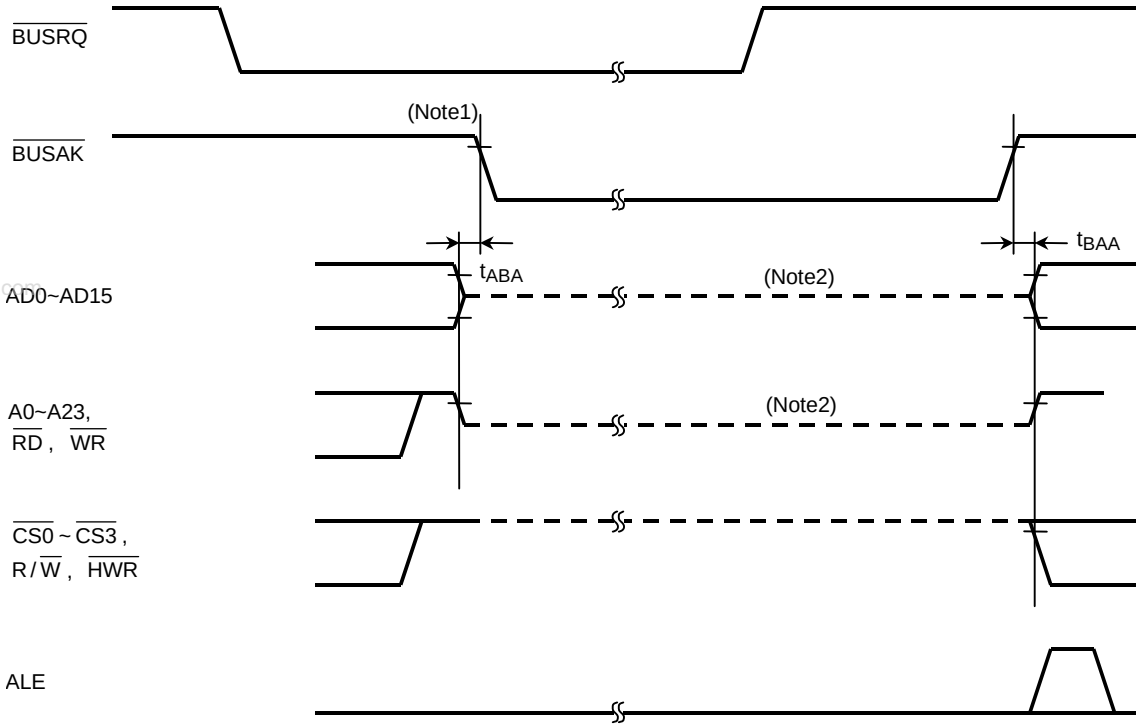
22.14 SCOUT Pin

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Clock high pulse width	t_{SCH}	0.5T − 5		4.25		ns
Clock low pulse width	t_{SCL}	0.5T − 5		4.25		ns

Note: In the above table, the letter T represents the cycle period of the SCOUT output clock.



22.15 Bus Request and Bus Acknowledge Signals



Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Bus float to BUSAK asserted	t_{ABA}	0	80	0	80	ns
Bus float after BUSAK negated	t_{BAA}	0	80	0	80	ns

Note 1: If the current bus cycle has not terminated due to wait-state insertion, the TMP19A64F20BXBG does not respond to BUSRQ until the wait state ends.

Note 2: This broken line indicates that output buffers are disabled, not that the signals are at indeterminate states. The pin holds the last logic value present at that pin before the bus is relinquished. This is dynamically accomplished through external load capacitances. The equipment manufacturer may maintain the bus at a predefined state by means of off-chip restores, but he or she should design, considering the time (determined by the CR constant) it takes for a signal to reach a desired state. The on-chip, integrated programmable pullup/pulldown resistors remain active, depending on internal signal states.

22.16 KWUP Input

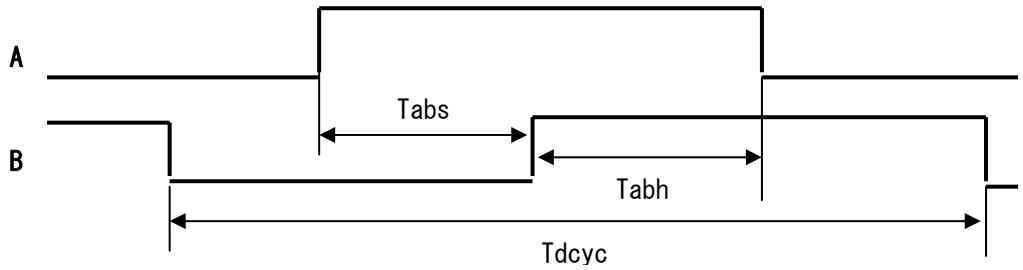
Pull-up Register Active

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY0-D	tky_{TBL}	X+100		118		ns
High pulse width for KEY0-D	tky_{TBH}	X+100		118		ns

22.17 Dual Pulse Input

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Dual input pulse period	Tdcyc	8Y		296		ns
Dual input pulse setup	Tabs	Y+20		57		ns
Dual input pulse hold	Tabh	Y+20		57		ns

Y: Sampling clock (fsys/2)



23. Notations, Precautions and Restrictions

23.1 Notations and Terms

(1) I/O register fields are often referred to as <*register_mnemonic*>.<*field_name*> for the interest of brevity. For example, TRUN.T0RUN means the T0RUN bit in the TRUN register.

(2) fc, fsys, state

fosc: Clock supplied from the X1 and X2 pins

fpll: Clock generated by the on-chip PLL

fc: Clock selected by the $\overline{\text{PLLOFF}}$ pin

fgear: Clock selected by the SYSCR1.GEAR[1:0] bits

fsys: Clock selected by the SYSCR1.SYSCK bit

The fsys cycle is referred to as a state.

In addition, the clock selected by the SYSCR1.FPSEL bit and the prescaler clock source selected by the SYSCR0.PRCK[1:0] bits are referred to as fperiph and ϕT0 respectively.

23.2 Precautions and Restrictions

(1) Processor Revision Identifier

The Process Revision Identifier (PRId) register in the TX19A core of the TMP19A64C1D contains 0x0000_2CA1.

(2) BW0–BW1 Pins

The BW0 and BW1 pins must be connected to the DVCC2 pin to ensure that their signal levels do not fluctuate during chip operation.

(3) Oscillator Warm-Up Counter

If an external crystal is utilized, an interrupt signal programmed to bring the TMP1940CYAF out of STOP mode triggers the on-chip warm-up counter. The system clock is not supplied to the on-chip logic until the warm-up counter expires.

(4) Programmable Pullup Resistors

When port pins are configured as input ports, the integrated pull-up resistors can be enabled and disabled under software control. The pull-up resistors are not programmable when port pins are configured as output ports.

The relevant port registers are programmed with the data resister.

(5) External Bus Mastership

The pin states while the bus is granted to an external device are described in Chapter 7, *I/O Ports*.

(6) Watchdog Timer (WDT)

Upon reset, the WDT is enabled. If the watchdog timer function is not required, it must be disabled after reset. When relevant pins are configured as bus arbitration signals, the I/O peripherals including the WDT can operate during external bus mastership.

(7) A/D Converter (ADC)

The ladder resistor network between the VREFH and VREFL pins can be disconnected under software control. This helps to reduce power dissipation, for example, in STOP mode.

(8) Undefined Bits in I/O Registers

Undefined I/O register bits are read as undefined states. Therefore, software must be coded without relying on the states of any undefined bits.

(9) Electrostatic Discharge (ESD) Sensitivity

The following shows ESD sensitivity. Protect the device from static damage during device development or production stage. For a detailed description on ESD, see General Safety Precautions and Usage Considerations.

- **TMP19A64C1DXBG**

Specification	Sensitivity
Machine Model: MM	±200 V
Human Body Model: HBM	−1750V ～+2000 V

- **TMP19A64F20AXBG**

Specification	Sensitivity
Machine Model: MM	±200 V
Human Body Model: HBM	−2000V ～+2000 V

(10) Bus Access of Debug Mode (Mask product only)

Bus Accessing is abnormal for external function with SREQ mode in Debug mode,

Which means Debug<DM>="1" in CP0 register. Of Mask Type MCU ,TMP19A64C1DXBG.

Pls don't access to external function with SREQ Mode in debug mode.

(11) Notations, Precautions and Restrictions

Overflow Exception

Problem:

If an overflow exception caused a jump to the exception handler and the first instruction in that exception handler caused another exception, the EPC register should point to the address of the first instruction in the exception handler. However, the EPC register might contain the address that caused the overflow exception.

- **Problem-Causing Situation:**

  When, with the instruction pipeline full, an overflow exception was taken at the following sequence of instructions and then the first instruction in the overflow exception handler causes another exception

  ADD, ADDI or SUB <= # Instruction that causes an overflow

  Jump or branch instruction <= # Instruction with a delay slot

  Delay slot

Note: Toshiba's compiler uses no instructions that could cause an overflow. Therefore, this problem does not occur.

Workaround:

Don't place a jump or branch instruction immediately following an instruction that could cause an overflow (ADD, ADDI or SUB).

LWL and LWR Instructions

Problem:

The LWL or LWR instruction might provide incorrect results.

- **Problem-Causing Situation #1:**
  a. The destination of a load instruction (LB, LBU, LH, LHU, LW, LWL or LWR) is identical to that of the LWL or LWR instruction.
  b. The instruction pipeline is full. (The load instruction and the LWL or LWR instruction will be executed consecutively.)
  c. The DMAC is programmed for data cache snooping. Once the load instruction is executed, the DMAC initiates a DMA transaction. After it has been serviced, the LWL or LWR instruction is executed.

  This problem occurs when all of these conditions are true.

- **Problem-Causing Situation #2:**
  a. The destination of a load instruction (LB, LBU, LH, LHU, LW, LWL or LWR) is identical to that of the LWL or LWR instruction.
  b. The Doze or Halt bit in the Config register is set to 1 immediately before the load instruction.
  c. The instruction pipeline is full. (The load instruction and the LWL or LWR instruction will be executed consecutively.)
  d. After the load instruction is executed, the processor is put in the STOP, SLEEP or IDLE mode.
  e. After an interrupt signaling brings the processor out of the STOP, SLEEP or IDLE mode, the LWL or LWR instruction is executed.

Note: This applies to the case in which an interrupt signaling does not generate an interrupt upon exit from STOP or IDLE mode. In other words, either the IEc bit in the Status register is cleared (interrupts disabled), or if the IEc bit is set, the priority level of the incoming interrupt signaling is lower than the mask level programmed in the CMask field in the Status register. (Exit from STOP, SLEEP or IDLE mode can be accomplished even with such settings.)

This problem occurs when all of these conditions are true.

Workarounds:

To use the LWL or LWR instruction,

1) Place a NOP between a load instruction and the LWL or LWR instruction, or
2) Disable the data cache snooping of the DMAC before the LWL or LWR instruction is executed. Also, do not put the processor in STOP, SLEEP or IDLE mode before the LWL or LWR instruction is executed.

Overflow Exception When a DSU Probe Is Used

Problem:

It looks as if an overflow exception caused a jump to the reset and nonmaskable exception vector address (0xBFC0_0000).

- **Problem-Causing Situation:**

  When an overflow exception occurs, with the processor connected to a DSU probe

Note: Toshiba's compiler uses no instructions that could cause an overflow. Therefore, this problem does not occur.

Workaround:

Don't place a jump or branch instruction immediately following an instruction that could cause an overflow (ADD, ADDI or SUB).

Malfunction of using BUSREQ signl in External Bus Access mode

[Condition]

1. In External Bus mode, using Auto WAIT insert function (as same as +N wait)
2. Use External Bus request signal Function (BUSREQ).
3. For each target product,Bus setting mode (Multiplex/ separate)、ALE width(short/long)

. Please refer to following table.

(Exp: ALE Band =1.5CLK, Auto wait = 3)