TOSHIBA

32bit TX System RISC
TX19 Family

TMP19A44FDA/FE/F10XBG

Rev1.3　01-Apr-2010

Contents

32-bit RISC Microprocessor - TX19 Family
TMP19A44F10XBG/FEXBG /FDAXBG

1. Overview and Features

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

TMP19A44 is a 32-bit RISC microprocessor with a TX19A/H1 processor core and various peripheral functions integrated into one package. It can operate at low voltage with low power consumption.

Features of TMP19A44 are as follows:

(1) TX19A/H1 processor core (refer to the separate volume “19A/ H1 Architecture” for details.

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 MIPS16^{TM}ASE 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 dissipation have been achieved.

- High performance
  - Almost all instructions can be executed with one clock.
  - High performance is possible via a three-operand operation instruction.
  - 3-stage pipeline
  - Built-in high-speed memory
  - DSP function: A 32-bit multiplication and accumulation operation can be executed with one clock.
  - On-board floating-point unit (FPU)
- 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) Internal program memory and data memory

Product name	Built-in Flash ROM	Built-in RAM * （Backup Ram）
TMP19A44F10XBG	1024Kbyte	64Kbyte (16K)
TMP19A44FEXBG	768Kbyte	64Kbyte (16K)
TMP19A44FDAXBG	512Kbyte	32Kbyte (8K)

*: Includes back-up RAM.

- ROM correction function: 8 word × 12 blocks

(3) External memory expansion

- Expandable to 16 megabytes (for both programs and data)
- External data bus:
  Separate bus/ multiplexed bus : Coexistence of 8- and 16-bit widths is possible.
  Chip select/wait controller : 4 channels

(4) DMA controller : 8 channels (8 interrupt factors)

- Activated by an interrupt or software
- Data to be transferred to internal memory, internal I/O, external memory, and external I/O

(5) 16-bit timer : 18 channels

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG output (every 4 channels, synchronous outputs are possible)
- Input capture mode
- 2-phase pulse input counter function (4 channels assigned to perform this function): Multiplication-by-4 mode

(6) 2-phase pulse input counter : 6 channels

- Interrupt function (quadruple/ normal/ compare). Counting available in normal and stop modes

(7) 32-bit timer

- 32-bit input capture register : 4 channels
- 32-bit compare register : 8 channels
- 32-bit time base timer : 1 channel

(8) Real time clock (RTC) : 1 channel

- Clock (hour, minute and second)
- Calendar (Month, week, date and leap year)
- Time correction ＋ or － 30 seconds (by software)
- Alarm (Alarm output)
- Alarm interrupt

(9) Watchdog timer : 1 channel

- 26-stage binary counter

(10) General-purpose serial interface : 3 channels

- Selectable between the UART mode and the synchronization mode (with 4-byte FIFO)

(11) High-speed serial interface : 3 channels

- Selectable between the UART mode and the high-speed synchronization mode (with 32 byte FIFO)

(12) Serial bus interface : 1 channel

- Selectable between the I^2C bus mode and the clock synchronization mode

(13) 10-bit A/D converter (with S/H) : 16 channels

- Start by an external trigger, and the internal timer activated by a trigger
- Fixed channel/scan mode
- Single/repeat mode
- Top-priority conversion mode
- AD monitor function: 6 channels (2ch for each unit)
- Conversion time 1.15 μsec(@ 40/ 80MHz)
- 3 units: 4ch, 4ch, 8ch
- Synchronous start of 2 units: available by external trigger

(14) Key-on wakeup : 32 channels

- Dynamic pull-up

(15) Interrupt function

- CPU: 2 factors ....software interrupt instruction
- Internal: 68 factors.............The order of precedence can be set over 7 levels (except for the watchdog timer interrupt).
- External: 64 factors ....The order of precedence can be set over 7 levels. Because 32 factors are associated with KWUP, the number of interrupt factors is one.

(16) Input and output ports ....160 terminals

(17) Standby function

- Four standby modes (IDLE, SLEEP, STOP and BACKUP SLEEP, BACKUP STOP)
- Sub clock: Slow, sleep and backup modes (32.768kHz)

(18) Clock generator

- Built-in PLL (multiplied by 8)
- Clock gear function: The high-speed clock can be divided into 8/8, 4/8, 2/8, 1/8, or 1/16.
- ϕT0: fperiph/2, fperiph/4, fperiph/8, fperiph/16, fperiph/32

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

(20) Maximum operating frequency

- 80 MHz (PLL multiplied by 8)

(21) Operating voltage range

- Regulator equipped, single power supply, 2.7-3.6V input.
- I/O and ADC: 2.7 V to 3.6 V

(22) Temperature range

- -20~85 degrees (operating range)
- 0~70 degrees (during Flash writing/ erasing)

(23) Package

P-FBGA241-1212-0.65 (12 mm × 12 mm, 0.65 mm pitch)

(24) Backup mode

Enable to operate:

- RAM 16/ 8KB
- I/O
- 2-phase counter: 6 channels
- KWUP (with dynamic pull-up): 32 channels
- External interrupt INT
- Clock count

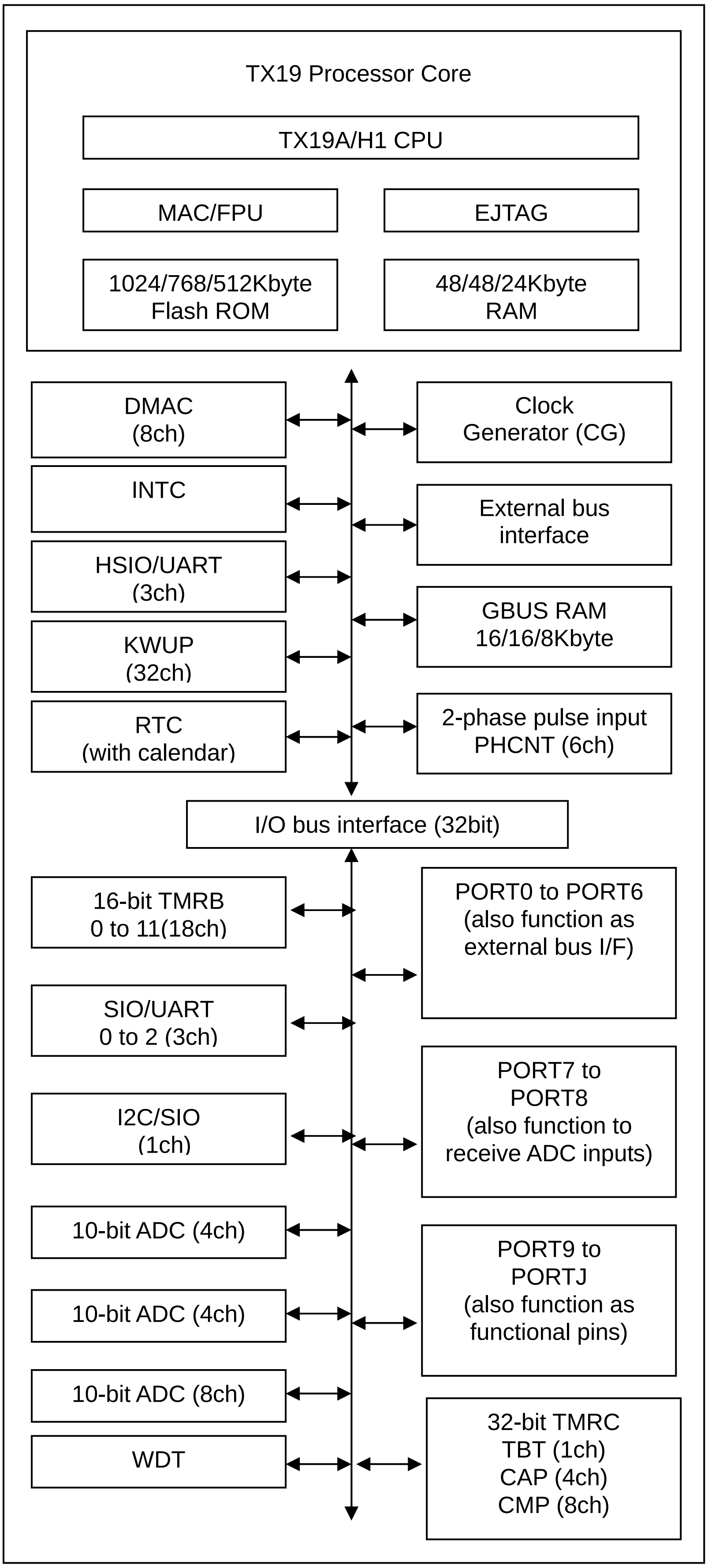

TX19 Processor Core
TX19A/H1 CPU
MAC/FPU
EJTAG
1024/768/512Kbyte Flash ROM
48/48/24Kbyte RAM
DMAC (8ch)
INTC
HSIO/UART (3ch)
KWUP (32ch)
RTC (with calendar)
Clock Generator (CG)
External bus interface
GBUS RAM 16/16/8Kbyte
2-phase pulse input PHCNT (6ch)
I/O bus interface (32bit)
16-bit TMRB 0 to 11(18ch)
SIO/UART 0 to 2 (3ch)
I2C/SIO (1ch)
10-bit ADC (4ch)
10-bit ADC (4ch)
10-bit ADC (8ch)
WDT
PORT0 to PORT6 (also function as external bus I/F)
PORT7 to PORT8 (also function to receive ADC inputs)
PORT9 to PORTJ (also function as functional pins)
32-bit TMRC TBT (1ch) CAP (4ch) CMP (8ch)

Fig. 1-1 TMP19A44 Block Diagram

2. Pin Layout and Pin Functions

This section shows the pin layout of TMP19A44 and describes the names and functions of input and output pins.

2.1 Pin Layout (Top view)

Fig. 2-1 shows the pin layout of TMP19A44.

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
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17
F1	F2	F3	F4	F5	F6							F13	F14	F15	F16	F17
G1	G2	G3	G4	G5								G13	G14	G15	G16	G17
H1	H2	H3	H4	H5								H13	H14	H15	H16	H17
J1	J2	J3	J4	J5								J13	J14	J15	J16	J17
K1	K2	K3	K4	K5								K13	K14	K15	K16	K17
L1	L2	L3	L4	L5								L13	L14	L15	L16	L17
M1	M2	M3	M4	M5								M13	M14	M15	M16	M17
N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15	N16	N17
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17
U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17

Fig. 2-1 Pin Layout (P-FBGA193)

2.2 Pin Numbers and Names

Table 2-1 shows the pin numbers and names of TMP19A44.

Table 2-1 Pin numbers and names

Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name
A1	NC	C15	DVSS	G1	VREFLB	L16	PJ6/INT6	R3	TEST0
A2	P84/AINC4	C16	PH5/INT1D/TBBIN1	G2	AVSSB	L17	PJ5/INT17	R4	P02/D2/AD2
A3	P81/AINC1	C17	TDI/RXD0	G3	DVSS	M1	PC0/TBTIN/KEY30	R5	P05/D5/AD5
A4	P83/AINC3	D1	P73/INT11/AINA3	G4	P90/HTXD0	M2	PC1/TCOUT0	R6	P10/D8/AD8/A8
A5	VREFHC	D2	P72/INT10/AINA2	G5	NC	M3	PC2/TCOUT1	R7	P14/D12/AD12/A12
A6	PF3/KEY19/DACK4	D3	P70/AINA0	G13	NC	M4	PC3/TCOUT2	R8	P20/A16/A0/TB1IN0
A7	PE6/KEY14	D4	AVSSC	G14	PI4/ADTRGC	M5	NC	R9	P23/A19/A3/TB2IN1
A8	PE1/KEY09	D5	DVSS	G15	PI3/PHC5IN1	M13	DVCC3	R10	P27/A23/A7/TB5IN1
A9	PD4/TBCOUT	D6	PF6/KEY22/TCOUT6	G16	PI2/PHC5IN0	M14	PJ4/INT16	R11	P42/CS2/KEY26
A10	PD0/HTXD2	D7	PF1/KEY17/DACK0	G17	DINT	M15	PJ3/INT15	R12	P45/BUSMD
A11	PA4/INT4/TB6IN0	D8	PE4/KEY12	H1	DVCC3	M16	PJ2/INT14	R13	P50/A0/INTC
A12	BVCC3	D9	PD7/ADTRGB	H2	P91/HRXD0	M17	PJ1/TB11IN1	R14	P53/A3/INTF
A13	XT2	D10	PD3/TBBOUT	H3	P92/HSCLK0/HCTS0	N1	PC4/SO/SDA	R15	TEST3
A14	XT1	D11	PA7/PHC2IN1	H4	P93/TB9OUT	N2	PC5/SI/SCL	R16	P61/A9/RXD0/INTA
A15	X2	D12	PA3/INT3/PHC1IN1	H5	NC	N3	PC6/SCK	R17	P60/A8/TXD0
A16	X1	D13	PH4/INT1C/TBBIN0	H13	NC	N4	DVCC3	T1	P34/BUSRQ/TBEOUT
A17	NC	D14	TEST4	H14	NC	N5	DVSS	T2	TEST1
B1	P85/AINC5	D15	PH3/INT1B/TBAIN1	H15	PI1/PHC4IN1	N6	NC	T3	P36/RW/TC2IN
B2	VREFLC	D16	RESET	H16	PI0/PHC4IN0	N7	NC	T4	P01/D1/AD1
B3	P82/AINC2	D17	TDO/TXD0	H17	EJE	N8	NC	T5	P04/D4/AD4
B4	P80/AINC0	E1	P75/AINB1	J1	P94/TXD2	N9	DVSS	T6	P07/D7/AD7
B5	AVCC3C	E2	P74/AINB0	J2	P95/RXD2	N10	NC	T7	P13/D11/AD11/A11
B6	PF4/KEY20/TCOUT4	E3	AVCC3B	J3	P96/SCLK2/CTS2	N11	NC	T8	P17/D15/AD15/A15
B7	PE7/KEY15	E4	AVCC3A	J4	P97/TBAOUT	N12	REGTEST3	T9	P22/A18/A2/TB2IN0
B8	PE2/KEY10	E5	VREFLA	J5	DVCC3	N13	DVSS	T10	P26/A22/A6/TB5IN0
B9	PD5/TBDOUT	E6	PF7/KEY23/TCOUT7	J13	DVCC3	N14	PJ0/TB11IN0	T11	P41/CS1/KEY25
B10	PD2/HSCLK2/HCTS2	E7	PF2/KEY18/DREQ4	J14	PG7/KEY07	N15	P67/A15/TB5OUT	T12	P44/SCOUT
B11	PA5/INT5/TB6IN1	E8	PE5/KEY13	J15	PG6/KEY06	N16	P66/A14/SCLK1/CTS1	T13	P47/TBFOUT
B12	PA1/INT1/PHC0IN1	E9	PE0/KEY08	J16	PG5/KEY05	N17	P65/A13/RXD1/INTB	T14	P52/A2/INTE
B13	PA0/INT0/PHC0IN0	E10	DVSS	J17	DVCC3	P1	P30/RD	T15	P55/A5/TB1OUT
B14	DVCC3	E11	REGTEST0	K1	PB2/TB6OUT	P2	P31/WR	T16	DVSS
B15	REGTEST2	E12	DVCC3	K2	PB4/HTXD1	P3	PC7/TCOUT3	T17	P57/A7/TB3OUT/KEY29
B16	CVSS	E13	DVSS	K3	PB3/TB7OUT	P4	DVSS	U1	NC
B17	TRST	E14	PH2/INT1A/TBAIN0	K4	PB1/PHC3IN1	P5	DVCC3	U2	P35/BUSAK/TC1IN
C1	P86/AINC6/INT8	E15	PH1/INT19/TB9IN1	K5	NC	P6	P11/D9/AD9/A9	U3	P37/ALE/TC3IN
C2	P71/AINA1	E16	PH0/INT18/TB9IN0	K13	NC	P7	P15/D13/AD13/A13	U4	P00/D0/AD0
C3	AVSSA	E17	TMS	K14	PG4/KEY04	P8	P21/A17/A1/TB1IN1	U5	P03/D3/AD3
C4	P87/AINC7/INT9	F1	P77/AINB3/INT13	K15	PG3/KEY03	P9	P24/A20/A4/TB3IN0	U6	P06/D6/AD6
C5	DVCC3	F2	P76/AINB2/INT12	K16	PG2/KEY02	P10	DVCC3	U7	P12/D10/AD10/A10
C6	PF5/KEY21/TCOUT5	F3	VREFHB	K17	PG1/KEY01	P11	DVCC3	U8	P16/D14/AD14/A14
C7	PF0/KEY16/DREQ0	F4	VREFHA	L1	PB5/HRXD1	P12	BOOT	U9	DVSS
C8	PE3/KEY11	F5	NC	L2	PB7/TB8OUT	P13	REGTEST1	U10	P25/A21/A5/TB3IN1
C9	PD6/KEY31/ADTRGA	F6	NC	L3	PB6/HSCLK1/HCTS1	P14	TEST2	U11	P40/CS0/KEY24
C10	PD1/HRXD2	F13	NC	L4	PB0/PHC3IN0	P15	P64/A12/TXD1	U12	P43/CS3/KEY27
C11	PA6/PHC2IN0	F14	PI7/ADTRGSNC	L5	NC	P16	P63/A11/TB4OUT	U13	P46/ENDIAN
C12	PA2/INT2/PHC1IN0	F15	PI6/TB11OUT	L13	NC	P17	P62/A10/SCLK0/CTS0	U14	P51/A1/INTD
C13	PH7/INT1F/TBDIN1	F16	PI5/TB10OUT	L14	PG0/KEY00	R1	P32/HWR/TC0IN	U15	P54/A4/TB0OUT
C14	PH6/INT1E/TBDIN0	F17	TCK	L15	PJ7/INT7	R2	P33/WAIT/RDY	U16	P56/A6/TB2OUT/KEY28
								U17	NC

2.3 Pin Names and Functions

Table 2-2 through Table 2-9 show the names and functions of input and output pins.

Table 2-2 Pin names and functions (1/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
P00~P07 D0~D7 AD0~D7	8	I/O I/O I/O	Port 0: Input/output port that allows input/output to be set in units of bits Data (lower): Data bus 0 to 7 (separate bus mode) Address data (lower): Address data bus 0 to 7 (multiplexed bus mode)	P-up		
P10~P17 D8~D15 AD8~AD15 A8~A15	8	I/O I/O I/O O	Port 1: Input/output port that allows input/output to be set in units of bits Data (upper): Data bus 8 to 15 (separate bus mode) Address data (upper): Address data bus 8 to 15 (multiplexed bus mode) Address: Address bus 8 to 15 (multiplexed bus mode)	P-up		
P20~P27 A16~A23 A0~A7 TB1IN0,TB1IN1 TB2IN0,TB2IN1 TB3IN0,TB3IN1 TB5IN0,TB5IN1	8	I/O O O I I I I	Port 2: Input/output port that allows input/output to be set in units of bits Address: Address bus 15 to 23 (separate bus mode) Address: Address bus 0 to 7 (multiplexed bus mode) 16-bit timer 1 input 0,1: For inputting the count/capture trigger of a 16-bit timer 1 16-bit timer 2 input 0,1: For inputting the count/capture trigger of a 16-bit timer 2 16-bit timer 3 input 0,1: For inputting the count/capture trigger of a 16-bit timer 3 16-bit timer 5 input 0,1: For inputting the count/capture trigger of a 16-bit timer 5	P-up		
P30 $\overline{\text{RD}}$	1	I/O O	Port 30: Port used exclusively for output RD Output Read: Strobe signal for reading external memory	P-up		
P31 $\overline{\text{WR}}$	1	I/O O	Port 31: Port used exclusively for output Write: Strobe signal for writing data of D0 to D7 pins	P-up		
P32 $\overline{\text{HWR}}$ TC0IN	1	I/O O I	Port 32: Input/output port Write upper-pin data: Strobe signal for writing data of D8 to D15 pins For inputting the capture trigger for 32-bit timer	P-up		
P33 $\overline{\text{WAIT}}$ RDY	1	I/O I I	Port 33: Input/output port Wait: Pin for requesting CPU to put a bus in a wait state Ready: Pin for notifying CPU that a bus is ready	P-up		
P34 $\overline{\text{BUSRQ}}$ TBEOUT	1	I/O I O	Port 34: Input/output port Bus request: Signal requesting CPU to allow an external master to take the bus control authority 16-bit timer E output: Pin for outputting 16-bit timer E	P-up		
P35 $\overline{\text{BUSAK}}$ TC1IN	1	I/O O I	Port 35: Input/output port Bus acknowledge: Signal notifying that CPU has released the bus control authority in response to $\overline{\text{BUSRQ}}$ For inputting the capture trigger for 32-bit timer	P-up		
P36 R/$\overline{\text{W}}$ TC2IN	1	I/O O I	Port 36: Input/output port Read/write: "1" shows a read cycle or a dummy cycle. "0" shows a write cycle. For inputting the capture trigger for 32-bit timer	P-up		
P37 ALE TC3IN	1	I/O O I	Port 37: Input/output port Address latch enable (address latch is enabled only if access to external memory (multiplexed bus mode) is taking place). For inputting the capture trigger for 32-bit timer	P-up		

Table 2-3 Pin names and functions (2/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
P40 $\overline{\text{CS0}}$ KEY24	1	I/O O I	Port 40: Input/output port Chip select 0: "0" is output if the address is in a designated address area. KEY on wake up input 24: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
P41 $\overline{\text{CS1}}$ KEY25	1	I/O O I	Port 41: Input/output port Chip select 1: "0" is output if the address is in a designated address area. KEY on wake up input 25: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
P42 $\overline{\text{CS2}}$ KEY26	1	I/O O I	Port 42: Input/output port Chip select 2: "0" is output if the address is in a designated address area. KEY on wake up input 26: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
P43 $\overline{\text{CS3}}$ KEY27	1	I/O O I	Port 43: Input/output port Chip select 3: "0" is output if the address is in a designated address area. KEY on wake up input 27: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
P44 SCOUT	1	I/O O	Port 44: Input/output port System clock output: Selectable between high- and low-speed clock outputs, as in the case of CPU	P-up		
P45 BUSMD	1	I/O I	Port 45: Input/output port Pin for setting an external bus mode: This pin functions as a multiplexed bus by sampling the "H (DVCC3) level" at the rise of a reset signal. It also functions as a separate bus by sampling "L" at the rise of a reset signal. When performing a reset operation, pull it up or down according to a bus mode to be used. Input with Schmitt trigger. (After a reset operation is performed, it can be used as a port.)	P-up	○ w/ noise filter	
P46 ENDIAN	1	I/O I	Port 46: Input/output port This pin is used to set a mode. It performs a big-endian operation by sampling the "H (DVCC3) level" at the rise of a reset signal, and performs a little-endian operation by sampling "L" at the rise of a reset signal. When performing a reset operation, pull it up or down according to the type of endian to be used. (After a reset operation is performed, it can be used as a port.)	P-up	○ w/ noise filter	
P47 TBFOUT	1	I/O O	Port 47: Input/output port 16-bit timer F output: Pin for outputting a 16-bit timer F	P-up		
P50~P53 A0~A3 INTC~INTF	4	I/O O I	Port 5: Input/output port that allows input/output to be set in units of bits Address: Address buses 0 to 3 (separate bus mode) Interrupt request pins C to F: Selectable between "H" level, "L" level, rising edge, and falling edge	P-up	○ w/ noise filter	
P54,P55 A4,A5 TB0OUT TB1OUT	2	I/O O O O	Port 5: Input/output port that allows input/output to be set in units of bits Address: Address buses 4 and 5 (separate bus mode) 16-bit timer 0 output: Pin for outputting a 16-bit timer 0 16-bit timer 1 output: Pin for outputting a 16-bit timer 1	P-up		
P56,P57 A6,A7 TB2OUT TB3OUT KEY28,KEY29	2	I/O O O O I	Port 5: Input/output port that allows input/output to be set in units of bits Address: Address buses 6 and 7 (separate bus mode) 16-bit timer 2 output: Pin for outputting a 16-bit timer 2 16-bit timer 3 output: Pin for outputting a 16-bit timer 3 KEY on wake up input 28 and 29: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	

Table 2-4 Pin names and functions (3/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
P60 A8 TXD0	1	I/O O O	Port 60: Input/output port Address: Address bus 8 (separate bus mode) Sending serial data 0: Open drain output pin depending on the program used	P-up		○
P61 A9 RXD0 INTA	1	I/O O I I	Port 61: Input/output port Address: Address bus 9 (separate bus mode) Receiving serial data 0 Interrupt request pin A: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
P62 A10 SCLK0 $\overline{\text{CTS0}}$	1	I/O O I/O I	Port 62: Input/output port Address: Address bus 10 (separate bus mode) Serial clock input/output 0 Handshake input pin Open drain output pin depending on the program used	P-up		○
P63 A11 TB4OUT	1	I/O O O	Port 63: Input/output port that allows input/output to be set in units of bits Address: Address bus 11 (separate bus mode) 16-bit timer 4 output: Pin for outputting a 16-bit timer 4	P-up		
P64 A12 TXD1	1	I/O O O	Port 64: Input/output port Address: Address bus 12 (separate bus mode) Sending serial data 1	P-up		○
P65 A13 RXD1 INTB	1	I/O O I I	Port 65: Input/output port Address: Address bus 13 (separate bus mode) Receiving serial data 1 Interrupt request pin B: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
P66 A14 SCLK1 $\overline{\text{CTS1}}$	1	I/O O I/O I	Port 6: Input/output port Address: Address buses 14 (separate bus mode) Serial clock input/output 1 Handshake input pin	P-up		○
P67 A15 TB5OUT	1	I/O O O	Port 67: Input/output port that allows input/output to be set in units of bits Address: Address buses 15 (separate bus mode) 16-bit timer 5 output: Pin for outputting a 16-bit timer 5	P-up		
P70,P71 AINA0,AINA1	2	I I	Port 7: Port used exclusively for input Analog input: Input from A/D converter (unit A)	P-up		
P72,P73 AINA2,AINA3 INT10,11	2	I I I	Port 7: Port used exclusively for input Analog input: Input from A/D converter (unit A) Interrupt request pins 10 and 11: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
P74,P75 AINB0,AINB1	2	I I	Port 7: Port used exclusively for input Analog input: Input from A/D converter (unit B)	P-up		
P76,P77 AINB2,AINB3 INT12,13	2	I I I	Port 7: Port used exclusively for input Analog input: Input from A/D converter (unit B) Interrupt request pins 12 and 13: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
P80~P85 AINC0~ AINC5	6	I I	Port 8: Port used exclusively for input Analog input: Input from A/D converter (unit C)	P-up		

Table 2-5 Pin names and functions (4/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
P86,P87 AINC6,AINC7 INT8,9	2	I I	Port 8: Port used exclusively for input Analog input: Input from A/D converter (unit C) Interrupt request pins 8 and 9: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
P90 HTXD0	1	I/O O	Port 90: Input/output port Sending serial data 0 at high speeds	P-up		○
P91 HRXD0	1	I/O I	Port 91: Input/output port Receiving serial data 0 at high speeds	P-up	○ w/ noise filter	
P92 HSCLK0 HCTS0	1	I/O I/O I	Port 91: Input/output port High-speed serial clock input/output 0 Handshake input pin	P-up		○
P93 TB9OUT	1	I/O O	Port 93: Input/output port that allows input/output to be set in units of bits 16-bit timer 9 output: Pin for outputting a 16-bit timer 9	P-up		
P94 TXD2	1	I/O O	Port 94: Input/output port Sending serial data 2	P-up		○
P95 RXD2	1	I/O I	Port 95: Input/output port Receiving serial data 2	P-up	○ w/ noise filter	
P96 SCLK2 CTS2	1	I/O I/O I	Port 96: Input/output port Serial clock input/output 2 Handshake input pin	P-up		○
P97 TBAOUT	1	I/O O	Port 97: Input/output port that allows input/output to be set in units of bits 16-bit timer A output: Pin for outputting a 16-bit timer A	P-up		
PA0 PHC0IN0 INT0	1	I/O I I	Port A0: Input/output port 2-phase pulse input counter 0 input 0 Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
PA1 PHC0IN1 INT1	1	I/O I I	Port A1: Input/output port 2-phase pulse input counter 0 input 1 Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
PA2 PHC1IN0 INT2	1	I/O I I	Port A2: Input/output port 2-phase pulse input counter 1 input 0 Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
PA3 PHC1IN1 INT3	1	I/O I I	Port A3: Input/output port 2-phase pulse input counter 1 input 1 Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
PA4 TB6IN0 INT4	1	I/O I I	Port A4: Input/output port 16-bit timer 6 input 0: For inputting the capture trigger of a 16-bit timer 6 Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
PA5 TB6IN1 INT5	1	I/O I I	Port A5: Input/output port 16-bit timer 6 input 1: For inputting the capture trigger of a 16-bit timer 6 Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges.	P-up	○ w/ noise filter	
PA6 PHC2IN0	1	I/O I	Port A6: Input/output port 2-phase pulse input counter 2 input 0	P-up		
PA7 PHC2IN1	1	I/O I	Port A7: Input/output port 2-phase pulse input counter 2 input 1	P-up		

Table 2-6 Pin names and functions (5/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
PB0 PHC3IN0	1	I/O I	Port B0: Input/output port 2-phase pulse input counter 3 input 0	P-up	○ w/ noise filter	
PB1 PHC3IN1	1	I/O I	Port B1: Input/output port 2-phase pulse input counter 3 input 1	P-up	○ w/ noise filter	
PB2,PB3 TB6OUT TB7OUT	2	I/O O O	Port B: Input/output port that allows input/output to be set in units of bits 16-bit timer 6 output: Pin for outputting a 16-bit timer 6 16-bit timer 7 output: Pin for outputting a 16-bit timer 7	P-up		
PB4 HTXD1	1	I/O O	Port B4: Input/output port Sending serial data 1 at high speeds	P-up		○
PB5 HRXD1	1	I/O I	Port B5: Input/output port Receiving serial data 0 at high speeds	P-up	○ w/ noise filter	
PB6 HSCLK1 $\overline{\text{HCTS}}$1	1	I/O I/O I	Port B6: Input/output port Sending/ receiving serial data 1 at high speeds Handshake input pin	P-up		○
PB7 TB8OUT	1	I/O O	Port B: Input/output port that allows input/output to be set in units of bits 16-bit timer 8 output: Pin for outputting a 16-bit timer 8	P-up		
PC0 TBTIN KEY30	1	I/O I	Port C0: Input/output port 32-bit time base timer input: For inputting a 32-bit time base timer KEY on wake up input 30: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
PC1~PC3 TCOUT0~ TCOUT2	3	I/O O O	Port C: Input/output port that allows input/output to be set in units of bits Outputting 32-bit timer if the result of a comparison is a match	P-up		
PC4 SO SDA	1	I/O O I/O	Port C4: 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 I2C mode	P-up	○ w/ noise filter	○
PC5 SI SCL	1	I/O I I/O	Port C5: 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 I2C mode	P-up	○ w/ noise filter	○
PC6 SCK	1	I/O I/O	Port C6: Input/output port Pin for inputting and outputting a clock if the serial bus interface operates in the I2C mode	P-up		○
PC7 TCOUT3	1	I/O O	Port C: Input/output port that allows input/output to be set in units of bits Outputting 32-bit timer if the result of a comparison is a match	P-up		
PD0 HTXD2	1	I/O O	Port D0: Input/output port Sending serial data 2 at high speeds	P-up		○
PD1 HRXD2	1	I/O I	Port D1: Input/output port receiving serial data 2 at high speeds	P-up	○ w/ noise filter	
PD2 HSCLK2 $\overline{\text{HCTS}}$2	1	I/O I/O I	Port D2: Input/output port High-speed serial clock input/output 2 Handshake input pin	P-up		○
PD3~PD5 TBBOUT~ TBDOUT	3	I/O O	Port D3 to D5: Input/output port that allows input/output to be set in units of bits 16-bit timer B/ C/ D output: Pin for outputting a 16-bit timer B/ C/ D	P-up		

Table 2-7 Pin names and functions (6/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
PD6 ADTRGA KEY31	1	I/O I I	Port D6: Input/output port that allows input/output to be set in units of bits Pin for starting A/D trigger or A/D converter (unit A) from an external source KEY on wake up input 31: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
PD7 ADTRGB	1	I/O I	Port D6: Input/output port that allows input/output to be set in units of bits Pin for starting A/D trigger or A/D converter (unit B) from an external source	P-up	○ w/ noise filter	
PE0~PE7 KEY08~KEY15	8	I/O I	Port E: Input/output port that allows input/output to be set in units of bits KEY on wake up input 08 to 15: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
PF0, PF2 $\overline{\text{DREQ0,4}}$ KEY16, KEY18	2	I/O I I	Port F: Input/output port that allows input/output to be set in units of bits DMA request signals 0 and 4: For inputting the request to transfer data by DMA from an external I/O device to DMAC0 or DMAC4 KEY on wake up input 16 to 19: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
PF1, PF3 $\overline{\text{DACK0,4}}$ KEY17, KEY19	2	I/O O I	Port F: Input/output port that allows input/output to be set in units of bits DMA request signals 0 and 4: For inputting the request to transfer data by DMA from an external I/O device to DMAC0 or DMAC4 KEY on wake up input 16 to 19: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
PF4-PF7 KEY20~KEY23 TCOUT4~TCOUT7	4	I/O I O	Port F: Input/output port that allows input/output to be set in units of bits KEY on wake up input 20 to 23: (Dynamic pull up is selectable) Outputting 32-bit timer if the result of a comparison is a match	P-up	○ w/ noise filter	
PG0~PG7 KEY00-KEY07	8	I/O I	Port G: Input/output port that allows input/output to be set in units of bits KEY on wake up input 00 to 07: (Dynamic pull up is selectable)	P-up	○ w/ noise filter	
PH0~PH7 INT18~INT1F TB9IN0, TB9IN1 TBAIN0,TBAIN1 TBBIN0,TBBIN1 TBDIN0,TBDIN1	8	I/O I I I I I	Port H: Input/output port that allows input/output to be set in units of bits Interrupt request pins 18 to 1F: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges 16-bit timer 9 input 0,1: For inputting the count/capture trigger of a 16-bit timer 9 16-bit timer A input 0,1: For inputting the count/capture trigger of a 16-bit timer A 16-bit timer B input 0,1: For inputting the count/capture trigger of a 16-bit timer B 16-bit timer D input 0,1: For inputting the count/capture trigger of a 16-bit timer D	P-up	○ w/ noise filter	
PI0 PHC4IN0	1	I/O I	Port I0: Input/output port 2-phase pulse input counter 4 input 0	P-up	○ w/ noise filter	
PI1 PHC4IN1	1	I/O I	Port I1: Input/output port 2-phase pulse input counter 4 input 1	P-up	○ w/ noise filter	
PI2 PHC5IN0	1	I/O I	Port I2: Input/output port 2-phase pulse input counter 5 input 0	P-up	○ w/ noise filter	
PI3 PHC5IN1	1	I/O I	Port I3: Input/output port 2-phase pulse input counter 5 input 1	P-up	○ w/ noise filter	
PI4 ADTRGC	1	I/O I	Port I4: Input/output port Pin for starting A/D trigger or A/D converter from an external source	P-up	–	

Table 2-8 Pin names and functions (7/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
PI5,6 TB10OUT TB11OUT	2	I/O O O	Port I5, I6: Input/output port 16-bit timer 10 output: Pin for outputting a 16-bit timer 10 16-bit timer 11 output: Pin for outputting a 16-bit timer 11	P-up		
PI7 ADTRGSNC	1	I/O I	Port I7: Input/output port Pin for starting A/D trigger or A/D converter from an external source	P-up	–	
PJ0,1 TB11IN0,TB11IN1	2	I/O I	Port I5, I6: Input/output port 16-bit timer 11 input 0,1: For inputting the count/capture trigger of a 16-bit timer 11	P-up	○ w/ noise filter	
PJ2~PJ7 INT14-17,6,7	6	I/O	Port J2to J7 : Input/output port Interrupt request pins 6, 7 and 14 to 17: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges	P-up	○ w/ noise filter	
$\overline{\text{EJE}}$	1	I	EJTAG enable: Signal for DSU-ICE	P-up	○ w/ noise filter	
TCK SCLK	1	I I/O	Test clock input: Signal for testing DSU-ICE Serial clock input/output	P-up	○ w/ noise filter	
TMS	1	I	Test mode select input: Signal for testing DSU-ICE	P-up	○	
DINT	1	I	Signal for DSU-ICE	P-up	○	
TDI RXD0	1	I I	Test data input: Signal for DSU-ICE Receiving serial data 0	P-up	○	
TDO TXD0	1	O O	Test data output: Signal for testing DSU-ICE Sending serial data 0			○
TRST	1	I	Test data input: Signal for testing DSU-ICE	P-down	○	
$\overline{\text{RESET}}$	1	I	Reset: Initializing LSI	P-up	○ w/ noise filter	
X1	1	I	Pin for connecting a high-speed oscillator (X1: Input with Schmitt trigger)		○	
X2	1	O	Pin for connecting a high-speed oscillator			
XT1	1	I	Pin for connecting a low-speed oscillator (XT1: Input with Schmitt trigger)		○	
XT2	1	O	Pin for connecting a low-speed oscillator			
$\overline{\text{BOOT}}$	1	I	Pin for setting a single boot mode: This pin goes into single boot mode by sampling "L" at the rise of a reset signal. It is used to overwrite internal flash memory. By sampling "H (DVCC3) level" at the rise of a reset signal, it performs a normal operation. This pin should be pulled up under normal operating conditions. Pull it up when resetting.	P-up	○	
VREFHA~C	3	I	Pin (H) for supplying the A/D converter with a reference power supply Connect this pin to AVCCA3 to 3C if the A/D converter is not used.			
VREFLA~C	3	I	Pin (L) for supplying the A/D converter with a reference power supply Connect this pin to GND if the A/D converter is not used.			
AVCC3A~C	3	–	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.(DVCC3)			
AVSSA~C	3	–	A/D converter GND pin (0 V). Connect this pin to GND even if the A/D converter is not used. Pin (L) for supplying the A/D converter with a reference power supply			

Table 2-9 Pin names and functions (8/8)

Pin name	No. of pins	Input/ output	Function	Programmable Pull up/ Pull down	Schmitt trigger	Programmable Open Drain Output
TEST0	1	I	TEST pin: Set to OPEN.		○	
TEST1	1		TEST pin: Set to OPEN.			
TEST2	1		TEST pin: Set to OPEN.			
TEST3	1		TEST pin: Set to OPEN. (Please do not put the positive voltage.)			
TEST4	1	I	TEST pin: To be fixed to DVCC3		○	
CVSS	1	―	Oscillator GND pin (0 V)			
DVCC3	13	―	Power supply pin: 3 V power supply			
DVSS	11	―	Power supply pin: GND pin (0 V)			
NC	26	―	Non-connection pin			

2.4 Pin Names and Power Supply Pins

Table 2-10 Pin names and Power Supplies

Pin name	Power supply	Pin name	Power supply
P00-P07	DVCC3	PF0-PF7	DVCC3
P10-P17	DVCC3	PG0-PG7	DVCC3
P20-P27	DVCC3	PH0-PH7	DVCC3
P30-P37	DVCC3	PI0-PI7	DVCC3
P40-P47	DVCC3	PJ0-PJ7	DVCC3
P50-P57	DVCC3	$\overline{\text{EJE}}$	DVCC3
P60-P67	DVCC3	$\overline{\text{TRST}}$	DVCC3
P70-P73	AVCC3A	TDI	DVCC3
P74-P77	AVCC3B	TDO	DVCC3
P80-P87	AVCC3C	TMS	DVCC3
P90-P97	DVCC3	TCK	DVCC3
PA0-PA7	DVCC3	$\overline{\text{DINT}}$	DVCC3
PB0-PB7	DVCC3	$\overline{\text{RESET}}$	DVCC3
PC0-PC7	DVCC3	$\overline{\text{BOOT}}$	DVCC3
PD0-PD7	DVCC3	X1、X2	1.5V (internally)
PE0-PE7	DVCC3	XT1, XT2	DVCC3

2.5 Pin Numbers and Power Supply Pins

Table 2-11 Pin Numbers and Power Supplies

Power supply	Pin number	Voltage range
DVCC3	A12, B14, C5, E12, H1, J5, J13, J17, M13, N4, P5, P10, P11	2.7V-3.6V
AVCC3A	E4	
AVCC3B	E3	
AVCC3C	B5	

3. Processor Core

The TMP19A44 has a high-performance 32-bit processor core (TX19A/H1 processor core). For information on the operations of this processor core, please refer to the "TX19A/H1 Architecture."

This chapter describes the functions unique to the TMP19A44 that are not explained in that document.

3.1 Reset Operation

3.1.1 Initial state

The internal circuits, register settings and pin status of the TMP19A44 are undefined right after the power-on. The state continues until the RESET pin receives low level input after all the power supply voltage is applied.

3.1.2 Operation

As the precondition, ensure that an internal high-frequency oscillator provides stable oscillation while power supply voltage is in the operating range. To reset the TMP19A44, input RESET signal at low level “0” for a minimum duration of 12 system clocks (1.2ms with external 10MHz oscillator).

3.1.3 Cancellation

When the reset is canceled, the system control coprocessor (CP0) and the internal I/O register of the TX19A/H1 processor core are initialized. Note that the clock gear enters 1/1 mode and the PLL multiplication circuit stops after canceling the reset. Therefore, setting for the PLL operation is required.

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_0000 (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) After turning the power on, make sure that the power supply voltage and oscillation have stabilized, wait for 500 μs or longer, and perform the reset.

(Note 3) In the FLASH program, the reset period of 0.5 uS or longer is required independently of the system clock.

(Note 4) The reset operation can alter the internal RAM state, but does not alter data in the backup RAM except for backup RAM.

4. Memory Map

4.1 Memory Map

4.1.1 TMP19A44F10XBG

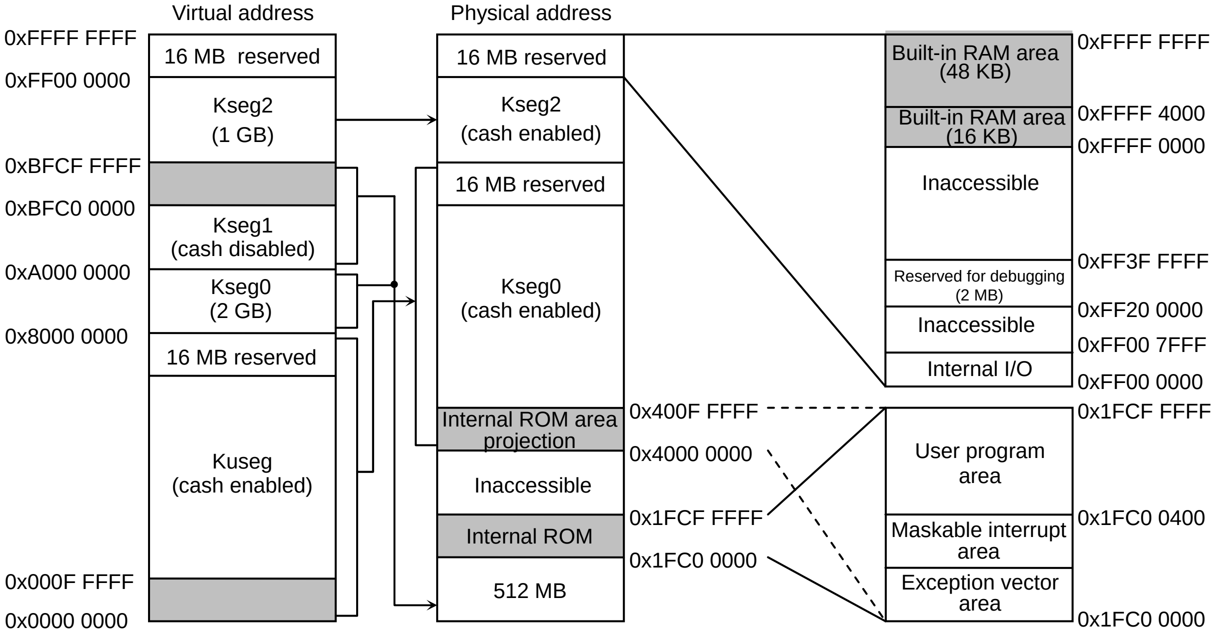


Fig. 4-1 Memory Map

(Note 1) The internal ROM is mapped to:
0x1FC0_0000-0x1FCF_FFFF (1024 KB)
The internal RAM is mapped to:
0xFFFF_4000-0xFFFF_FFFF (48 KB)

(Note 2) The amount of back up RAM installed is 16KB.
0xFFFF_0000-0xFFFF_3FFF (Back up RAM)

(Note 3) Do not place an instruction in the last four words of a physical area.
Internal ROM: 0x1FCF_FFF0-0x1FCF_FFFF (256 KB)

(Note 4) The mirror region cannot be used for ROM correction.

4.1.2 TMP19A44FEXBG

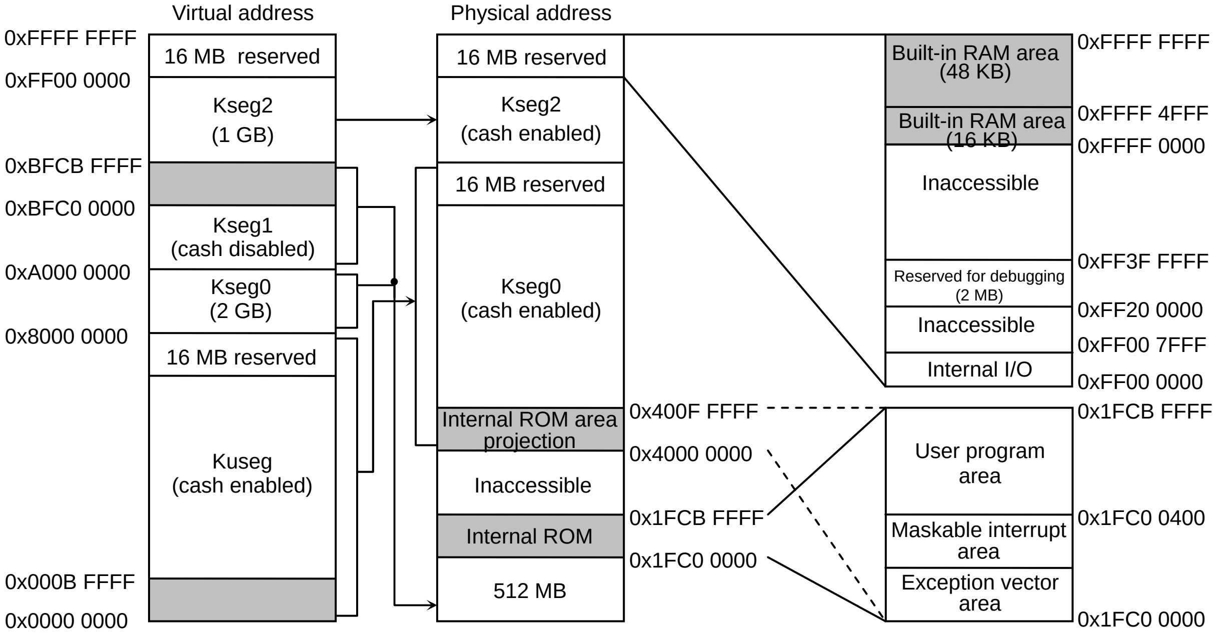


Fig. 4-2 Memory Map

(Note 1) The internal ROM is mapped to:
0x1FC0_0000-0x1FCB_FFFF (768 KB)
The internal RAM is mapped to:
0xFFFF_4000-0xFFFF_FFFF (48 KB)

(Note 2) The amount of back up RAM installed is 16KB.
0xFFFF_0000-0xFFFF_3FFF (Back up RAM)

(Note 3) Do not place an instruction in the last four words of a physical area.
Internal ROM: 0x1FCB_FFF0-0x1FCB_FFFF (256 KB)

(Note 4) The mirror region cannot be used for ROM correction.

4.1.3 TMP19A44FDAXBG

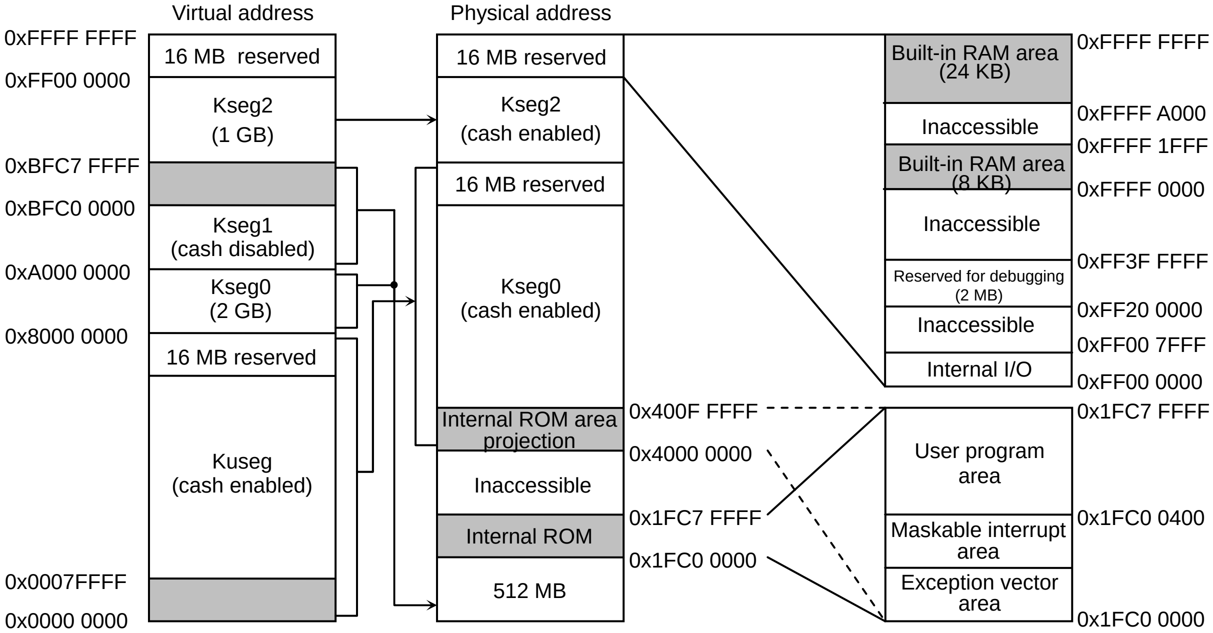


Fig. 4-3 Memory Map

(Note 5) The internal ROM is mapped to:
0x1FC0_0000~0x1FC7_FFFF (512KB)
The internal RAM is mapped to:
0xFFFF_A000~0xFFFF_FFFF (24KB)

(Note 6) The amount of back up RAM installed is 8KB.
0xFFFF_0000～0xFFFF_1FFF (Back up RAM)

(Note 7) Do not place an instruction in the last four words of a physical area.
Internal ROM: 0x1FC7_FFF0-0x1FC7_FFFF (256 KB)

(Note 8) The mirror region cannot be used for ROM correction.

4.2 Internal RAM

TMP19A44 is equipped with accessible work RAM (48K/24K) and back-up RAM (16K/8KB).

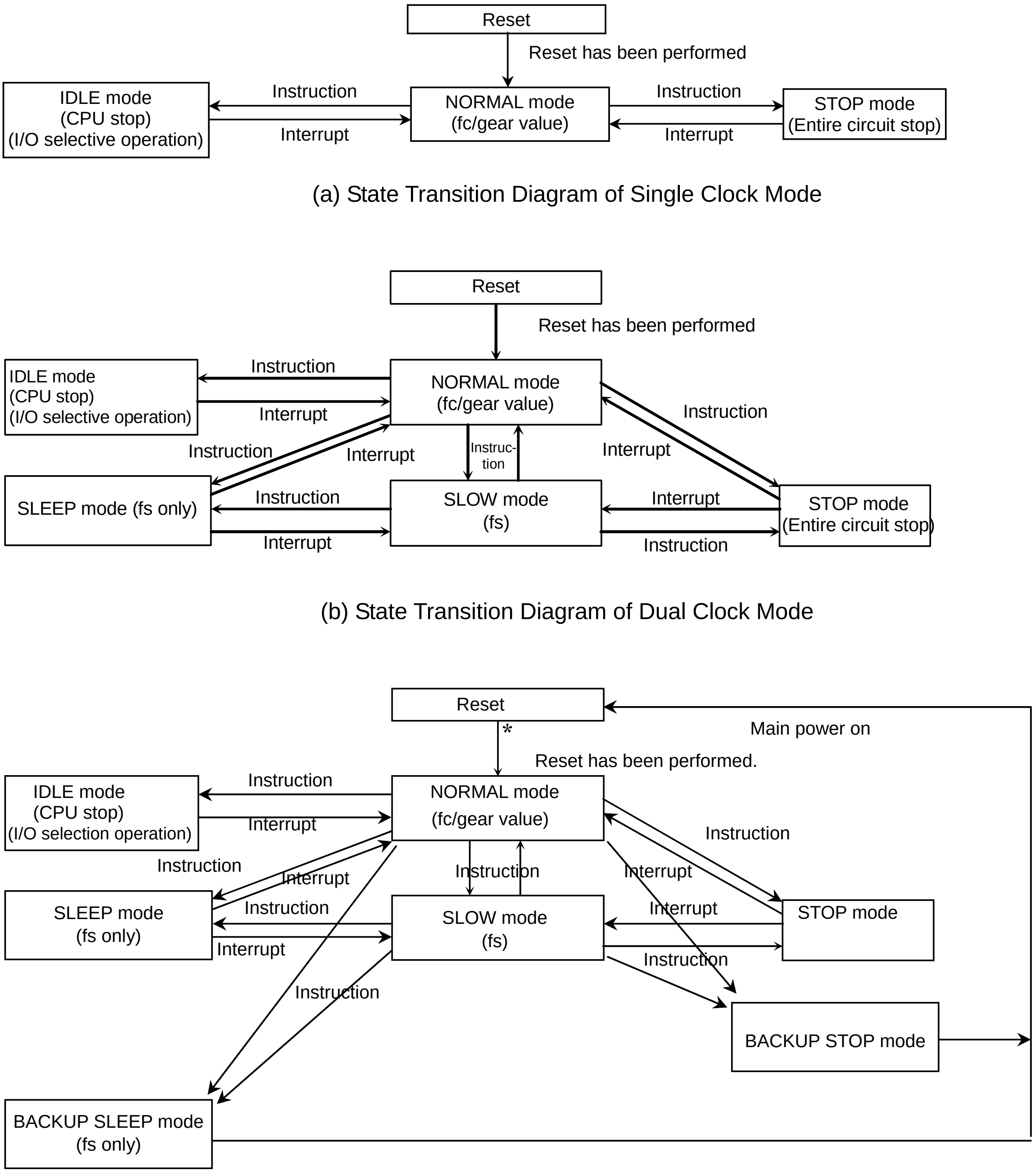
The work RAM and back-up RAM are available for program area and data area. The data in the work RAM is initialized by reset. The data in the back-up RAM remains intact as long as stable electrical potential is provided from a power supply unit (BVCC3) even when reset is executed.

To access the back-up RAM in normal mode, 4 clocks are required.

5. Clock/Standby Control (CG)

The system operation modes contain the standby modes in which the processor core operations are stopped to reduce power consumption. (c) State Transition Diagram of BACKUP Mode

Fig. 5.1 State Transition Diagram of Each Operation Mode is shown below.



(a) State Transition Diagram of Single Clock Mode

(b) State Transition Diagram of Dual Clock Mode

(c) State Transition Diagram of BACKUP Mode

Fig. 5.1 State Transition Diagram of Each Operation Mode

*: CG is not initialized when the mode is shifted from backup to normal.

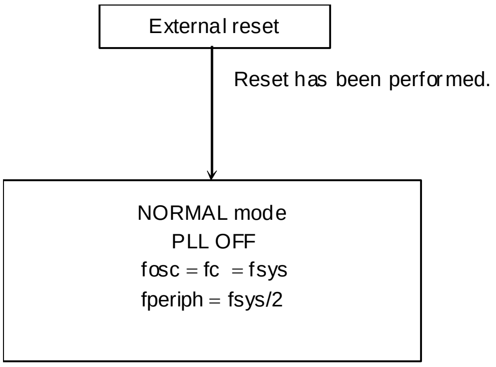


Fig. 5.2 Default State of the System Clock

fosc	**: Clock frequency to be input via the X1 and X2 pins**
fpll	**: Clock frequency multiplied (multiplied by 8) by the PLL**
fc	**: High-frequency clock frequency**
fs	**: Low-frequency clock frequency**
fgear	**: Clock frequency selected by the system control register SYSCR1<GEAR2:0> in the clock generator**
fsys	**: System clock frequency**
fperiph	**: Clock frequency selected by SYSCR1<FPSEL> (Clock to be input to the peripheral I/O prescaler)**

5.1 Clock System Block Diagram

5.1.1 Main System Clock

- Allows for oscillator connection or external clock input.
- Clock gear (1/2, 1/4, 1/8 and 1/16)
  (After reset: 1/1)
- Input frequency

Input frequency	Maximum operating frequency	Minimum operating frequency
8~10（MHz）	80 MHz	4 MHz *

* PLL is on: Clock gear 1/8 (default) is used when 8 MHz (MIN) is input.
PLL is off: 1/2 or 1/1 is selectable.

Input frequency (low frequency)

Input frequency range	Maximum operating frequency
30KHz~34 KHz	34 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.

5.2 Clock Gear

- The high-speed clock is divided into 3/4, 1/2, 1/4 or 1/8.
- The internal I/O prescaler clock ϕT0: fperiph/2, fperiph/4, fperiph/8, fperiph/16 and fperiph/32 （After reset: fperiph/32）
- The PLLSEL register controls PLL on/off. PLL stops after reset.

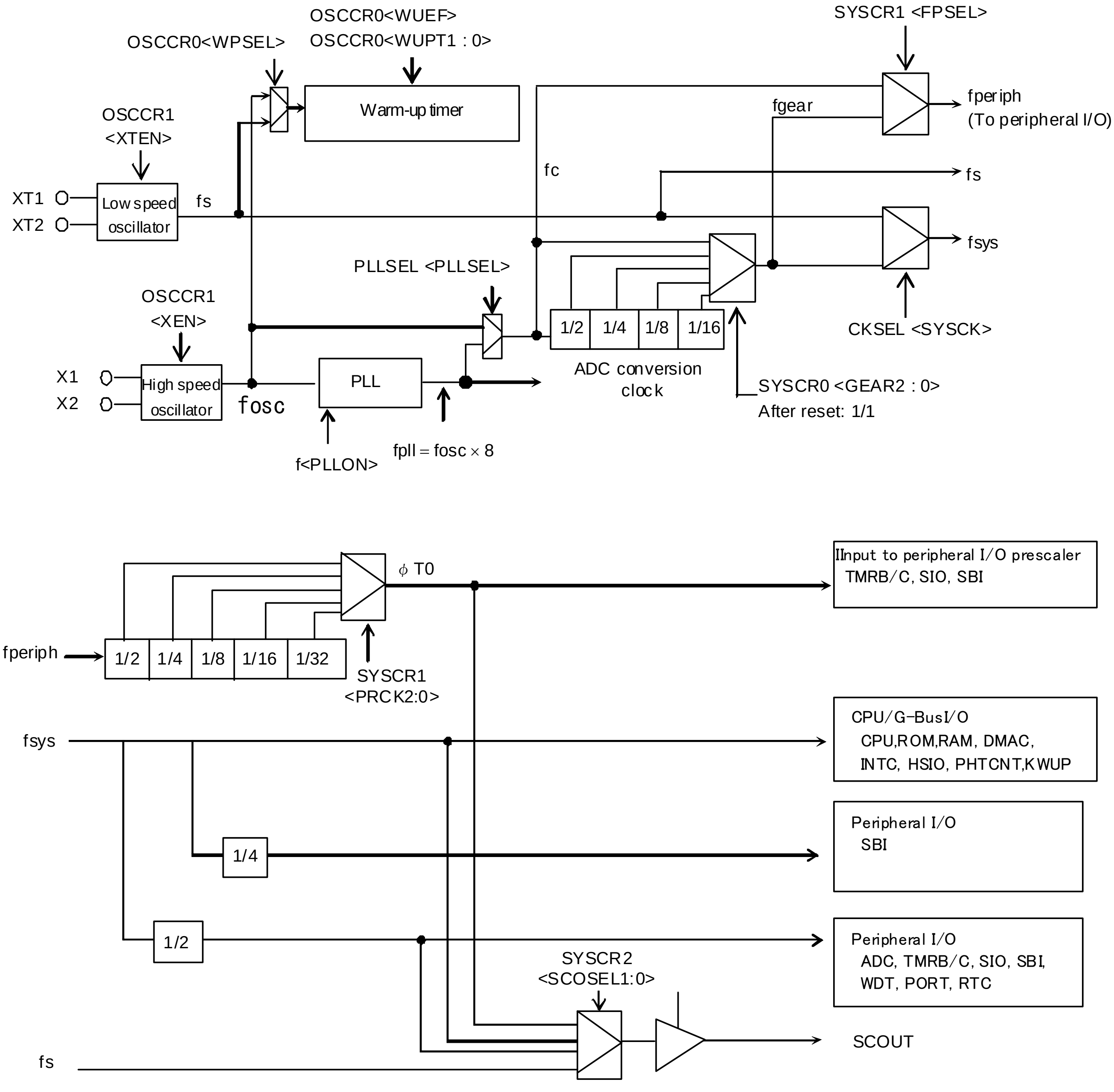


Fig. 5.3 Clock and Standby Related Block Diagram

5.3 CG Registers

5.3.1 System Control Registers

			7	6	5	4	3	2	1	0
LITTLE BIG	SYSCR0 (0xFF00_1900) (0xFF00_1903)	Bit symbol						GEAR2	GEAR1	GEAR0
		Read/Write	R					R/W	R/W	R/W
		After reset	0	0	0	0	0	0	0	0
		Function	This can be read as “0”.					Select gear of high-speed clock (fc) 000: fc 100: fc/2 001: reserved 101: fc/4 010: reserved 110: fc/8 011: reserved 111: fc/16		
			15	14	13	12	11	10	9	8
LITTLE BIG	SYSCR1 (0xFF00_1901) (0xFF00_1902)	Bit symbol				FPSEL		PRCK2	PRCK1	PRCK0
		Read/Write	R			R/W	R	R/W	R/W	R/W
		After reset	0	0	0	0	0	0	0	0
		Function	This can be read as “0”.			Select fperiph 0:fgear 1:fc	This can be read as “0”.	Select prescaler clock 000: fperiph/2 100: fperiph/32 001: fperiph/4 101: Reserved 010: fperiph/8 110: Reserved 011: fperiph/16 111: Reserved		
			23	22	21	20	19	18	17	16
LITTLE BIG	SYSCR2 (0xFF00_1902) (0xFF00_1901)	Bit symbol							SCOSEL1	SCOSEL0
		Read/Write	R						R/W	R/W
		After reset	0	0	0	0	0	0	0	1
		Function	This can be read as “0”.						Select SCOUT output 00:fs 01:fsys/2 10:fsys 11:φT0	
			31	30	29	28	27	26	25	24
		Bit symbol								
		Read/Write	R							
		After reset	0							
		Function	This can be read as “0”.							

<Bit 2:0><GEAR 2:0> : Selects gear of high-speed clock (fc)

<Bit 10:8><PRCK 2:0> : Selects prescaler clock to peripheral I/O.

<Bit 12><FPSEL> : Selects fperiph source clock.

<Bit 16:17><SCOSEL1:0> : Enables to output specified clock from SCOUT pin（P44）.

5.3.2 Oscillator Control Register

OSCCR0
LITTLE 0xFF00_1904)
BIG 0xFF00_1907)

OSCCR1
LITTLE 0xFF00_1905)
BIG 0xFF00_1906)

	7	6	5	4	3	2	1	0
Bit symbol		WUPT2	WUPT1	WUPT0	WUPSEL	PLLON	WUEF	WUEON
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R	W
After reset	0	0	0	1	0	0	0	0
Function	This can be read as “0”.	Select oscillator warm-up time X1 is selected （XT1 is selected） 000: No warm-up 001: Setting prohibited （2^6/ Input frequency） 010: 2^{13}/Input frequency（2^7/ Input frequency 011: 2^{14}/Input frequency 2^8/ Input frequency 100: 2^{15}/Input frequency 2^{15}/Input frequency 101: 2^{16}/Input frequency 2^{16}/Input frequency 110,111: Setting prohibited			Warm-up counter 0: X1 1: XT1	PLL operation 0: Stop 1: Oscillating	Status of warm-up timer (WUP) for oscillator 0: warm-up completed 1: Warm-up is in progress	Control of warm-up timer (WUP) for oscillator 0: don’t care 1: Starting warm-up
	15	14	13	12	11	10	9	8
Bit symbol			DRVOSCL	DRVOSCH			XTEN	XEN
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	This can be read as “0”.	This can be read as “0”.	Low-speed oscillator current control 0: High capability 1:Low capability	High-speed oscillator current control 0: High capability 1:Low capability	This can be read as “0”.	This can be read as “0”.	Low-speed oscillator 0: Stop 1: Oscillating	High-speed oscillator 0: Stop 1 : Oscillating
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

<Bit 0><WUEON> : Enables to start oscillator warm-up timer.
*Set this bit independently after setting other bits in the register.

<Bit 1><WUEF> : Indicates status of warm-up timer for oscillator.

<Bit 2><PLLON> : Selects PLL（multiplying circuit）operation.
It stops after reset. Reconfiguration is required.

<Bit 3><WUPSEL> : Selects oscillator to warm-up.

<Bit 6:4><WUT2:0> : Selects oscillator warm-up time.

<Bit 8><XEN> : Selects high-speed oscillator operation.

<Bit 8><XTEN> : Selects low-speed oscillator operation.

<Bit 12><DRVOSCH> : Selects current for high-speed oscillator.
The <DRVOSCH> bit is cleared to “0” (large current) when the mode is shifted from STOP to normal even if the bit is set to ”1” (small current) in shifting to STOP mode.
Reconfigure the bit if needed.

<Bit 13><DRVOSCL> : Selects current for low-speed oscillator.
The <DRVOSCL> bit is cleared to “0” (large current) when the mode is shifted from STOP to normal even if the bit is set to ”1” (small current) in shifting to STOP mode.
Reconfigure the bit if needed.

5.3.3 Standby Control Register

STBYCR0
LITTLE (0xFF00_1908)
BIG (0xFF00_190B)

	7	6	5	4	3	2	1	0
Bit symbol						STBY2	STBY1	STBY0
Read/Write	R					R/W	R/W	R/W
After reset	0					0	1	1
Function	This can be read as “0”.					Select standby mode 000: Reserved 001: STOP 010: SLEEP 011: IDLE 100: Reserved 101: Backup STOP 110: Backup SLEEP 111: Reserved		

STBYCR1
LITTLE (0xFF00_1909)
BIG (0xFF00_190A)

	15	14	13	12	11	10	9	8
Bit symbol							RXTEN	RXEN
Read/Write	R						R/W	R/W
After reset	0						0	1
Function	This can be read as “0”.						Low-speed oscillator after the STOP mode is released 0: Stop 1: Oscillating	High-speed oscillator after the STOP mode is released 0: Stop 1: Oscillating

STBYCR2
LITTLE (0xFF00_190A)
BIG (0xFF00_1909)

	23	22	21	20	19	18	17	16
Bit symbol							PTKEEP	DRVE
Read/Write	R						R/W	R/W
After reset	0						0	0
Function	This can be read as “0”.						0:Varies depending on the port condition 1: Fixed to the port condition shifted from 0 to 1	0: Not to drive the pin even in the STOP mode. 1: Drive the pin even in the STOP mode.

	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	This can be read as “0”.							

<Bit 2:0><STBY2:0> : Selects standby mode.

<Bit 8><RXEN> : Selects high-speed oscillator operation after releasing STOP mode.

<Bit 9><RXTEN> : Selects low-speed oscillator operation after releasing STOP mode.

<Bit 16><DRVE> : Selects pin drive condition in STOP mode.
This setting is invalid in backup mode.

<Bit 17><PTKEEP> : Specifies port condition.

5.3.4 PLL Select Register

		7	6	5	4	3	2	1	0
PLLSEL	Bit symbol								PLLSEL
(0xFF00_190C)	Read/Write	R							R/W
	After reset	0	0	0	0	0	0	0	0
	Function	This can be read as “0”.							Select PLL 0: X1 1: PLL
		15	14	13	12	11	10	9	8
	Bit symbol								
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	This can be read as “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	This can be read as “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	This can be read as “0”.							

<Bit 0><PLLSEL> : Enable or disable clock multiplied by PLL.
“X1” is used after reset. Reconfiguration is required if you use PLL..

5.3.5 System Clock Select Register

CKSEL
(0xFF00_1910)

	7	6	5	4	3	2	1	0
Bit symbol							SYSCK	SYSCKFLG
Read/Write	R						R/W	R
After reset	0	0	0	0	0	0	0	0
Function	This can be read as “0”.						Select system clock 0: High-speed (fc) 1: Low-speed (fs)	System clock status flag 0: High-speed (fc) 1: Low-speed (fs) (stable and identical to <SYSCK>)
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	This can be read as “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	This can be read as “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	This can be read as “0”.							

<Bit 0><SYSCKFLG> : Indicates status when system clock is switched.
To make the <SYSCK> setting for switching oscillator valid, there is a time lag. If <SYSCLKFLG> recognizes the oscillator specified in <SYSCK>, the switching is completed.

<Bit 1><SYSCK> : Enables to select system clock.
To change <SYSCK> setting, set <XEN> and <XTEN> of the OSCCR1 register to “1” beforehand.

5.3.6 Reset Flag Register

RSTFLG
(0xFF00_191C)

	7	6	5	4	3	2	1	0
Bit symbol				PrRSTF	BUPRSTF	WDTRSTF	PINRSTF	PONRSTF
Read/Write	R			R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	This can be read as “0”.			Pr reset flag 0: “0” is written 1:Reset from Pr reset	Backup reset flag 0: “0” is written 1:Reset from backup reset	WDT reset flag 0: “0” is written 1:Reset from WDT	RESET pin flag 0: “0” is written 1:Rest from RESET pin	Power-on reset flag 0: “0” is written 1:Rest from power-on reset
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	This can be read as “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	This can be read as “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	This can be read as “0”.							

<Bit 0><PONRSTF> : Power on reset sets this bit to “1”.

<Bit 1><PINRSTF> : Reset from RESET pin sets this bit to “1”.

<Bit 2><WDTRSTF> : Reset by WDT sets this bit to “1”.

<Bit 3><BUPRSTF> : Returning from backup mode sets this bit to “1”.

This register is initialized only by power-on reset in.

5.4 System Clock Controller

5.4.1 Initial Values after Reset

Reset initializes the system clock controller as follows.

High-speed oscillator	: ON (oscillating)
Low-speed oscillator	: ON (oscillating)
PLL (phase locked loop circuit)	: OFF (stop)
High-speed clock gear	: fc (no frequency dividing)

For example, when a 10-MHz oscillator is connected to the X1 or X2 pin, fsys becomes 10MHz after reset.

5.4.2 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 OSCCR0<WUPT2:0> depending on the characteristics of the oscillator. The OSCCR0<WUEON><WUEF> are 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 (CKSEL<SYSCK>).

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

(c) State Transition Diagram of BACKUP Mode

Fig. 5.1 shows the warm-up time when switching occurs.

Table 5.1 Warm-up Time (fosc=10 MHz, fs=32.768 kHz)

Warm-up time options OSCCR0<WUPT 2:0>	High-speed clock (fosc) OSCCR0<WUPSEL>="0"		Low-speed clock (fs) OSCCR0<WUPSEL>="1"	
000	–	No warm-up	–	No warm-up
001	Setting prohibited	409.6 (μs)	2^6/ input frequency	1.953 (ms)
010	2^{13}/ input frequency	819.2 (μs)	2^7/ input frequency	3.906 (ms)
011	2^{14}/ input frequency	1.638 (ms)	2^8/ input frequency	7.813 (ms)
100	2^{15}/ input frequency	3.277 (ms)	2^{15}/ input frequency	1.0 (s)
101	2^{16}/ input frequency	6.554 (ms)	2^{16}/ input frequency	2.0 (s)
110	Setting prohibited	–	2^{17}/ input frequency	4.0 (s)
111	Setting prohibited	–	Setting prohibited	–

(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.

<Example 1> Transition from the NORMAL mode to the SLOW mode
OSCCR0<WUPT2:0>＝"xx": Select the warm-up time
OSCCR0<WUPSEL>＝"1": Warm-up time XT1
OSCCR1<XTEN>＝"1": Enable the low-speed oscillation (fs)
OSCCR0<WUEON>＝"1": Start the warm-up timer (WUP)
OSCCR0<WUEF>Read: Wait until the state becomes "0" (WUP is finished)
CKSEL<SYSCK>＝"1": Switch the system clock to low speed (fs)
CKSEL<SYSCKFLG>Read: Confirm that the current state is "1" (the current system clock is fs)
OSCCR1<XEN>＝"0": Disable the high-speed oscillation (fosc)

<Example 2> Transition from the SLOW mode to the NORMAL mode
OSCCR0<WUPT2:0>＝"xx": Select the warm-up time
OSCCR0<WUPSEL>＝"0": Warm-up time X1
OSCCR1<XEN>＝"1": Enable the high-speed oscillation (fosc)
OSCCR0<WUEF>＝"1": Start the warm-up timer (WUP)
OSCCR0<WUEF>Read: Wait until the state becomes "0" (WUP is finished)
CKSEL<SYSCK>＝"0": Switch the system clock to high speed (fgear)
CKSEL<SYSCKFLG>Read: Confirm that the current state is "0" (the current system clock is fgear)
OSCCR1<XTEN>＝"0": Disable the low-speed oscillation (fs)

(Note 1) In the SLOW mode, the CPU operate with the low-speed clock, and the INTC, the real time clock timer, the IO port, the EBIF (external bus interface) and the PHTCNT,KWUP are operable. Stop other internal peripheral functions before the system enters the SLOW mode.

(Note 2) Before switching the system clock, ensure that the clock is properly switched by reading the CKSEL<SYSCKFLG> bit.

5.4.3 System Clock Pin Output Function

The system clock fsys and fsys/2, prescaler input clock for peripheral I/O ϕT0 and low-speed clock fs can be output from the P44/SCOUT pin. By setting the port 4 related registers, P4CR<P44C> to "1" and P4FC1<P44F> to "1," the P44/SCOUT pin becomes the SCOUT output pin. The output clock is selected by setting the SYSCR2<SCOSEL1:0>.

Table 5.2 shows the pin states in each standby mode when the P44/SCOUT pin is set to the SCOUT output.

Table 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.				Fixed to "0" or "1".
<SCOSEL1:0> = "01"	Output the fsys/2 clock.				
<SCOSEL1:0> = "10"	Output the fsys clock.				
<SCOSEL1:0> = "11"	Output the ϕT0 clock.				

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

(Note) The system clock cannot be output from SCOUT in backup mode.

5.4.4 Reducing the Oscillator Driving Capability

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

Setting the OSCCR1<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

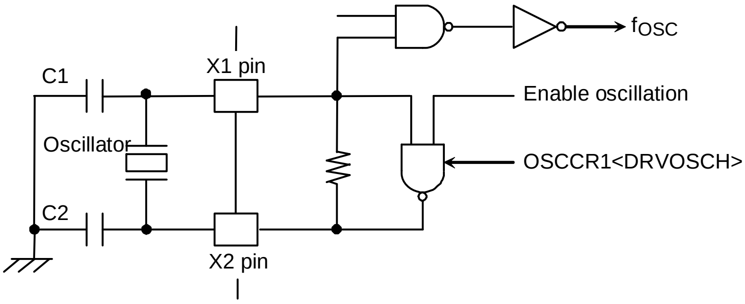


Fig. 5.4 Oscillator Driving Capability

5.5 Prescaler Clock Controller

Each internal I/O (TMRB0-11, SIO0-2, HSIO0-2 and SBI0) 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 then dividing the clock according to the setting of SYSCR1<PRCK2:0>. After the controller is reset, fperiph/2 is selected as ϕT0.

5.6 Clock Multiplication Circuit (PLL)

This circuit outputs the fpll clock that is multiplied by eight of the high-speed oscillator output clock, fosc. This lowers the oscillator input frequency while increasing the internal clock speed.

5.7 Standby Controller

The TX19A/ H1 core has several low-consumption modes. To shift to the STOP, SLEEP, IDLE (Halt or Doze) or backup 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 STBYCR0<STBY2:0>.

The features of the IDLE, SLEEP, STOP and backup modes are described below.

5.7.1 Standby Mode

5.7.1.1 IDLE Mode

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 5.3 Internal I/O setting registers for the IDLE mode

Internal I/O	IDLE Mode Setting Register
TMRB0~11	TBxRUN<I2TBx>
TMRC	TCCR<I2TBT>
SIO0~3	SCxMOD1<I2Sx>
HSIO0~3	HSCxMOD1<I2Sx>
I2C/SIO(SBI)	SBIBR1<I2SBIx>
A/D converter A~C	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/ H1 processor core stops the processer operation while holding the status of the pipeline. The TX19A/H1 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/ H1 processor core stops the processer operation while holding the status of the pipeline. The TX19A/ H1 can respond to the bus control authority request given from the outside of the processor core.

SLEEP: Only the internal low-speed oscillator, the clock timer, the 2-phase pulse input counter and the KWUP (dynamic pull-up circuit) operate.

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

The standby mode selection Status<RP > of CP0 is selected by the combination.
Please do not execute the WAIT instruction in the setting of "X" in the following table.

	STBY	HALT	DOZE
	2:0	RP=0	RP=1
RESERVED	OTHER	X	X
STOP	001	STOP	X
SLEEP	010	SLEEP	X
IDLE	011	HALT	DOZE
BACKUP STOP	101	BACKUP STOP	X
BACKUP SLEEP	110	BACKUP SLEEP	X

5.7.2 CG Operations in Each Mode

Table 5.4 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/ Backup	Fs only	×	Clock timer, 2-phase pulse input counter and KWUP	×
	Stop/ Backup	×	×	×	×

○: 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, clock timer, 2-phase pulse input counter and KWUP

5.7.3 Block Operations in Each Mode

Table 5.5 Block Operating Status in Each Operation Mode

Block	NORMAL	SLOW	IDLE (Doze)	IDLE (Halt)	SLEEP	STOP	SLEEP Backup	STOP Backup
TX19A/H1 processor core	○	○	×	×	×	×	×	×
DMAC	○	○	○	×	×	×	×	×
INTC	○	○	○	○	×	×	×	×
External bus I/F	○	○	○	×	×	×	×	×
IO port	○	○	○	×	×	×	×	×
ADC	○	×	ON/OFF selectable for each module		×	×	×	×
SIO	○	×			×	×	×	×
HSIO	○	×			×	×	×	×
I2C	○	×			×	×	×	×
TMRB	○	×			×	×	×	×
TMRC	○	×			×	×	×	×
WDT	○	×			×	×	×	×
2-phase pulse input counter	○	○			○	○	○ (fs only)	○
Dynamic pull-up (KWUP)	○	○	○	○	○	○（Static pull-up）	○	○（Static pull-up）
RTC	○	○	○	○	○	×	○	×
CG	○	○	○	○	○	×	○	×
High-speed oscillator (fc)	○	Δ (Note)	○	○	×	×	×	×
Low-speed oscillator (fs)	○	○	○	○	○	×	○	×

○:ON ×:OFF

(Note) When the system enters the SLOW mode, the high-speed oscillator must be stopped by setting the OSCCR1<XEN>.

5.7.4 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 / H1 processor core. Details are shown in Table 5.6.

- 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 non-maskable 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.

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

Table 5.6 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/ B-Sleep	Standby mode	IDLE (programmable)	SLEEP/ B-Sleep	Standby mode
Standby release source	Interrupt	INTWDT	◎	×	×	◎	—	—
		INT0~B, INT10~1B	◎	◎	◎	○	○	○
		KWUP00~31	◎	◎	◎	○	○	○
		INTRTC	◎	◎	×	○	○	×
		PHCNT0~5	◎	◎	◎	○	○	○
		INTTB0~11	◎	×	×	○	×	×
		INTRX0~2,INTTX0~2	◎	×	×	○	×	×
		HINTRX0~2,HINTTX0~2	◎	×	×	○	×	×
		INTS0	◎	×	×	○	×	×
		INTAD/INTADHP/INTADM	◎	×	×	○	×	×
	RESET		◎	◎	◎	◎	◎	◎

◎: 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

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

(Note 2) 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.

(Note 3) 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.

5.7.5 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 STBYCR2<DRVE>. Table 5.7 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 OSCCR0<WUPT2:0>.

(Note 1) To shift from the NORMAL mode to the STOP mode on the TMP19A44, do not set the OSCCR0<WUPT2:0> to "000" for the warm-up time setting. The internal system recovery time cannot be satisfied when the system recovers from the STOP mode. To shift from the SLOW mode to the STOP mode (high-speed clock is stopped) be sure to set OSCCR0<WUPT2:0>＝"110" during warm-up time.

(Note 2) The setting in <DRVE> is valid in STOP mode. It is invalid in backup mode.

Table 5.7 Warm-up Settings for Transitions of Operation Modes

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

5.7.6 Recovery from the STOP or SLEEP Mode

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

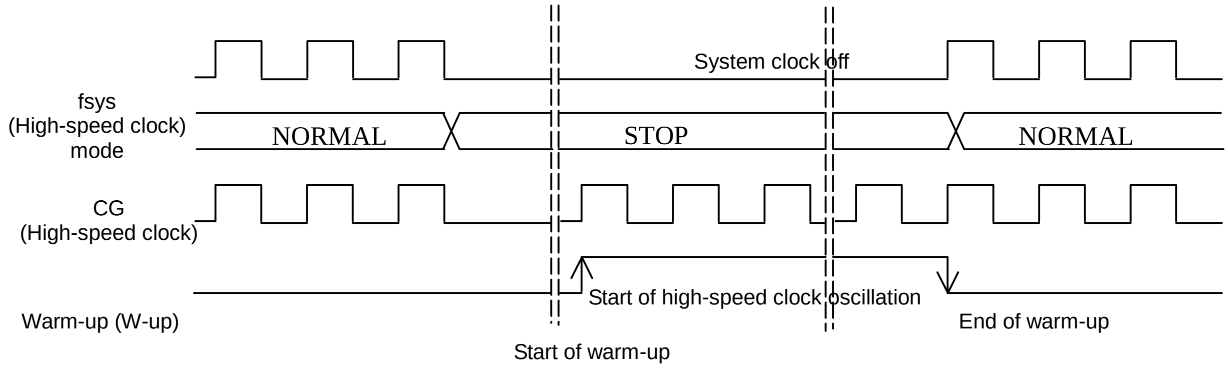


Table 5.8 Warm-up time

Selection of warm-up time OSCCR0<WUPT 2:0>	Warm-up time (fosc = 10.0MHz)
000	No warm-up
001(2^{12}/inputfrequency)	409.6（μs）
010(2^{13}/inputfrequency)	819.2（μs）
011(2^{14}/inputfrequency)	1.638（ms）
100(2^{15}/inputfrequency)	3.277（ms）
101(2^{16}/inputfrequency)	6.554（ms）
110 (Setting prohibited)	—
111 (Setting prohibited)	—

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

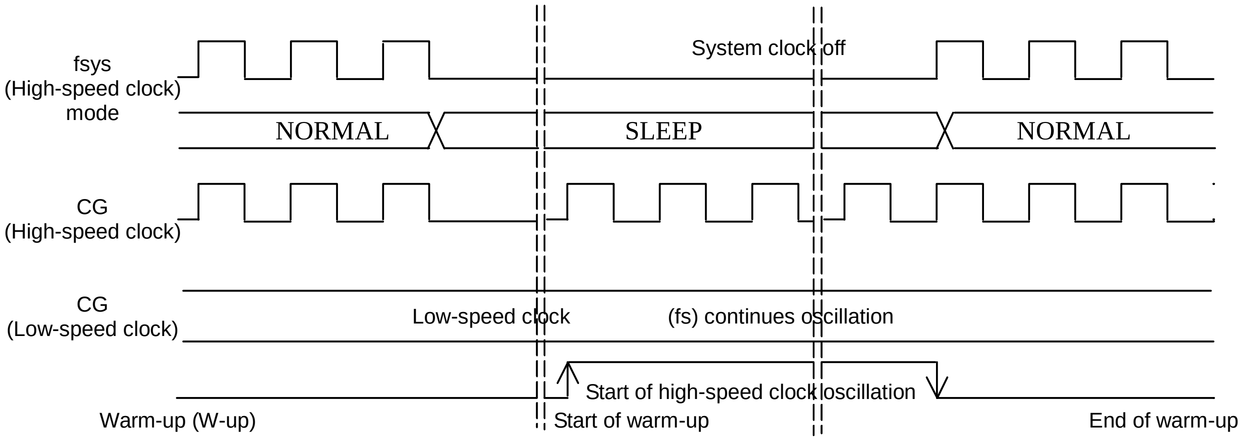


Table 5.9 Warm-up Time

Selection of warm-up time OSCCR0<WUPT 2:0>	Warm-up time (fosc = 10.0MHz)
000	No warm-up
001 (2^{12}/ input frequency)	409.6（μs）
010 (2^{13}/ input frequency)	819.2（μs）
011 (2^{14}/ input frequency)	1.638（ms）
100 (2^{15}/ input frequency)	3.277（ms）
101 (2^{16}/ input frequency)	6.554（ms）
110 (Setting prohibited)	—
111 (Setting prohibited)	—

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

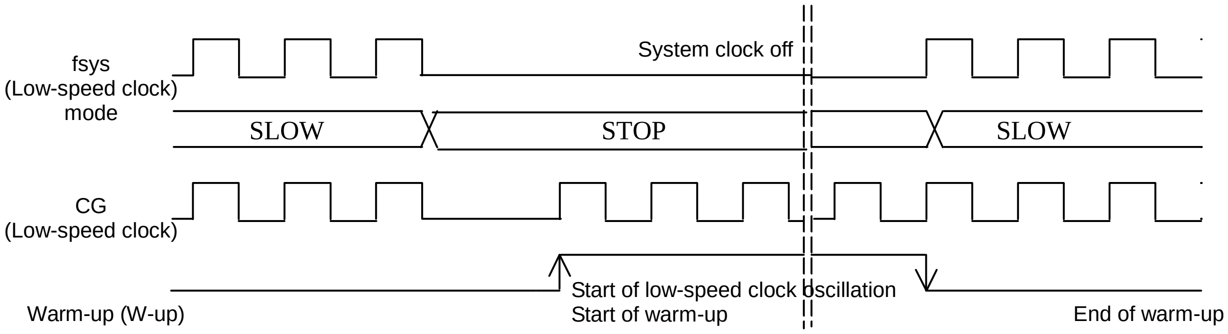
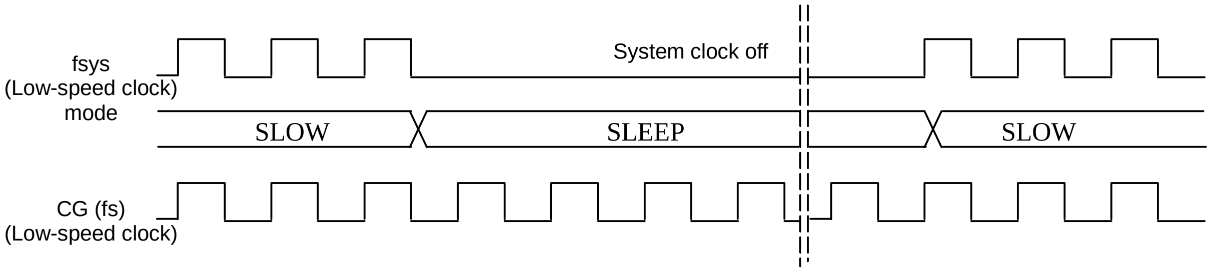


Table 5.10 Warm-up Time

Selection of warm-up time OSCCR0<WUPT 2:0>	Warm-up time (fs = 32.768kHz)
000	No warm-up
001 (2^{12}/ input frequency)	1.953（ms）
010 (2^{13}/ input frequency)	3.906（ms）
011 (2^{14}/ input frequency)	7.813（ms）
100 (2^{15}/ input frequency)	1.0（s）
101 (2^{16}/ input frequency)	2.0（s）
110 (2^{17}/ input frequency)	4.0（s）
111 (Setting prohibited)	—

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.

6. Exceptions/Interrupts

This chapter describes types, factors and process flow of exceptions/ interrupts.

“6.1 Overview” describes types of factors, generation mechanism and process flow. The section 6.2 or later describes details unique to each factor.

Exceptions/ interrupts have close relation to the CPU core architecture. Refer to “TX19A/H1 architecture” if needed.

6.1 Overview

This section roughly describes features, types and process flow of exceptions and interrupts. Please refer to the section 6.2 or later for details.

6.1.1 Exceptions and Interrupts

Exceptions and interrupts request the CPU to stop the ongoing processing and execute other processing. Interrupts are classified into general interrupt and debug interrupt.

6.1.1.1 Exceptions features

General exception is caused by an abnormal condition or an instruction to generate an exception.

Debug exception is used when debugging.

6.1.1.2 Interrupts features

Interrupts are classified into two types. One is maskable hardware interrupt of which factor is hardware-generated (i.e. interrupt request signal from external pin or peripheral IP). Another is Maskable software interrupt that is caused by software. They are generically named as maskable interrupts.

The features of the mast able interrupts are as follows:

・Prioritization according to interrupt level

The 19A44 can handle multiple interrupts according to the seven programmable interrupt priority levels. By setting a mask level, an interrupt of which level is lower than the specified can be masked.

・Shadow register

A register bank ”Shadow Register Set” enables high-speed response to an interrupt without storing the contents to the genera purpose register (GPR).

・DMAC activation

An interrupt can be used as DMAC activator. It requires settings in the interrupt controller.

Hereinafter general interrupts and debug interrupts are referred to as “exception”, maskable interrupts are referred to as “interrupt”.

If maskable interrupt factors are divided into software-generated or hardware-generated, we call them “software interrupt” or “hardware interrupt” respectively.

6.1.2 Types

The following lists general exceptions, debug exceptions and interrupts.

For more details, see “TX19A/H1architecture”.

(1) General exceptions

Reset exception
Non-maskable interrupt (NMI)
Address error exception (instruction fetch)
Address error exception (load/store)
Bus error exception (instruction fetch)
Bus error exception (load/ store)
Co-processor unusable exception
Reserved instruction exception
Integer overflow exception
Trap exception
System call exception
Breakpoint exception

(2) Debug exception

Single step exception
Debug breakpoint exception

(3) Interrupts

Software interrupts
Hardware interrupts

6.1.3 Process Flow

The following shows how an exception/ interrupt is handled.

▣ indicates hardware handling. □ indicates software handling.

Details are described in the later section.

Processing	Details	See
▣ Exception/ interrupt request	Exception/ interrupt request is made.	6.2.3.1
▣ Detecting exception/ interrupt	The CPU detects the exception/ interrupt.	6.2.3.2
▣ Handling exception/ interrupt	The CPU/ INTC/ CG handle the exception/ interrupt.	6.2.3.3
▣ CPU branches off to exception handler	The CPU branches off to an appropriate exception vector address according to the exception/ interrupt detected.	6.2.3.4
□ Executing exception handler program	Program for the exception handler.	6.2.3.5
□ Returning to normal program	Configure to return to the formerly-processed program from the exception handler.	6.2.3.6

6.1.3.1 Generation of exception/ interrupt request

An exception or an interrupt request is generated by various factors (i. e. instruction executed by the CPU, external interrupt pin and peripheral I/O)

(1) Exception request

An exception request is mainly caused by an abnormal condition detected during executing an instruction.

The following list categorized the types of the exceptions into four groups: "Requests not from the CPU", "Abnormal conditions during instructions", "Instructions to generate exceptions" and "Debug register setting".

Table 6.1 Exception types and factors

●Requests not from the CPU	
Reset exception	Reset signal WDT
Non-maskable interrupt (NMI）	External NMI pin
●Abnormal conditions during instructions	
Address error exception (instruction fetch/ load/ store）	Instruction fetch from the address undesignated. Reading operand in the address undesignated.
Bus error exception (instruction fetch/ load/ store）	Instruction fetch from the unused area Reading operand in the unused area
Co-processor unusable exception	Executing co-processor instruction without setting the CU bit of the status register in the co-processor.
Integer overflow exception	Overflow of ADD, ADDI or SUB instruction result
Reserved instruction exception	Executing undefined instruction code
●Instructions to generate exceptions	
System call exception	Executing SYSCALL instruction
Debug breakpoint exception	Executing SDBBP instruction
Breakpoint exception	Executing BREAK instruction
Trap exception	Executing TGE, TGEU, TLT, TLTU, TEQ TNE, TGEI, TGEIU, TLTI, TLTIU, TEQI and TNEI instructions.
●Debug register setting	
Single step exception	Setting SSt bit in the Debug register

(2) Interrupt request

An interrupt request is caused by software (software interrupt), an external pin or a peripheral IP (hardware interrupt)

To make interrupt request, setting of the CP0 register, CG and interrupt controller are required. Details are described in "6.5 Hardware interrupt".

6.1.3.2 Exceptions/ interrupts detection

(1) Priority

There are cases that multiple exceptions and interrupts are detected simultaneously. To handle the situation, the CPU executes an exception of an interrupt with the highest priority.

The priority of the exceptions/ interrupts is shown below.

Table 6.2 Exceptions/ interrupts priority (when detected simultaneously)

Priority	Exceptions/ interrupts
Higher	Reset exception
	Single step exception
	Non-maskable interrupt (NMI)
	Hardware interrupts
	Software interrupts
	Address error exception (instruction fetch)
	Bus error exception (instruction fetch/ store)
	Debug breakpoint exception
	Co-processor unusable exception
	Reserved instruction exception
	Integer overflow exception
	Trap exception
	System call exception
	Breakpoint exception
	Address error exception (load/store)
Lower	Bus error exception (data access)

(Note) When a hardware interrupt and a software interrupt are detected simultaneously, a maskablehardware interrupt is executed.

6.1.3.3 Exceptions/ interrupt handling

(1) Filed change

When an exception/ interrupt is detected, related fields in the CP0 register of the CPU, the interrupt controller registers and the CG registers are changed.

The condition after the change differs depending on the exception/ interrupt detected as shown in the table below.

Table 6.3 Register filed change

		Factor				
Register	Field	Reset	Non-maskable interrupt (NMI)	Other exceptions	Interrupts	Meaning
●CP0 register						
Status	RP	0	-	-	-	Low-power consumption mode is set（select Halt/Doze）.
	BEV	1	-	-	-	Exception handler address has been changed.
	NMI	0	1	-	-	NMI is generated.
	ERL	1	1	-	-	Reset/ NMI is generated. Interrupt prohibited when this bit is set.
	EXL	-	-	1	1	Exception (other than reset/NMI) is generated. Interrupt prohibited when this bit is set.
Cause	BD	-	-	1/0	1/0	1: exception/ interrupt is generated in the slot of branch instruction. 0: others （This bit changes only when the EXL bit of the status register is “0”.）
	CE[1:0]	-	-	(Cop No)	-	Co-processor number referred when co-processor unusable exception is generated.
	ExcCode	-	-	Code	Code	Code according to exception/ interrupt.
Error EPC	-	PC	PC	-	-	Program counter of an instruction executed when a factor is detected. It indicates program counter of branch instruction when a factor is detected in the branch instruction slot.
EPC	-	-	-	PC	PC	
Bad VAddr	-	-	-	(Addr)	-	Virtual address considered to be an error when an address error exception is generated.
SSCR	CSS	-	-	-	Reg Set No	Register set number. When the SSD bit of the SSCR register（bit to enable the shadow register）is “0” (enabled), an interrupt changes the CSS bit to select The shadow register corresponding to the detected interrupt level. (the shadow register switching) CSS contents set before an interrupt is stored in PSS.
	PSS	Reg Set No	Reg Set No	Reg Set No	Reg Set No	
● Interrupt controller						
ILEV	CMASK	-	-	-	Level	CMASK: detected interrupt level PMASK: CMASK value before interrupt generation
	PMASK	－	－	－	Level	

“-“ indicates no change. Factors shown with（）are changed by specific exceptions.

Other than the registers shown above, the NMIFLG register setting changes when non-maskable interrupt is detected and the IVR register setting of the interrupt controller is changed when maskable interrupt is detected. These changes are described in “6.2 Reset exception/ non-maskable interrupt” and “6.5 Hardware interrupt”.

This table excludes the changes related to debug exceptions. See “TX19A/H1 architecture” for these changes.

6.1.3.4 Branch to Exception handler

The CPU branches off to the exception vector address (exception handler) in detecting an exception/ interrupt.

The exception vector addresses differ depending on the exceptions/ interrupts or the settings in the CP0 register Status<BEV> bit and Cause<IV> bit.

The Status<BEV> bit is set to "1" after reset. If you set the exception vector address to the internal ROM, do not change the bit from "1".

The Cause<IV> bit is undefined after reset. Setting this bit to "1" is required. By setting this bit to "1" the vector addresses of interrupt exceptions and other exceptions can be distinguished.

The following table lists the vector addresses.

Table 6.4 Exception vector address (virtual)

Exceptions/ interrupts	Status<BEV>=0	Status<BEV>=1
RESET,NMI	0xBFC0_0000	0xBFC0_0000
Debug exception	0xBFC0_0480	0xBFC0_0480
Interrupt (Cause<IV>=0)	0x8000_0180	0xBFC0_0380
Interrupt (Cause<IV>=1)	0x8000_0200	0xBFC0_0400
Other exceptions	0x8000_0180	0xBFC0_0380

6.1.3.5 Execution of Exception Handler Program

The exception handler executes exception/ interrupt processing. The processing varies depending on the exception/ interrupt generated. You need to program the exception handler.

When using the exception handler for interrupts, processing (i.e. clearing interrupt request) may be required in order to prevent the same interrupt from occurring.

See "6.5 hardware interrupt" for details.

6.1.3.6 Returning from Exceptions/ interrupts

You can refer to the exception/ interrupt generation program counter stored in the ErrorEPC/ EPC of the CP0 register when returning from the exception handler to the program previously executed. Please note that the same exception/ interrupt may occur once again at that time.

(1) Returning with ERET instruction

To return from an exception handler of general exception/ interrupt, you can use the ERET instruction. By executing the ERET instruction, the normal operation restarts from the address stored in the Error EPC／EPC of the CP0 register, and the CP register automatically goes back to the pre-exception/interrupt state. If you set the rewritable ErrorEPC／EPC with the return address in advance, the normal operation restarts from the desired address by executing the ERET instruction.

The ERET instruction operates differently according to the setting of Status<ERL> and <EXL> bits as shown below.

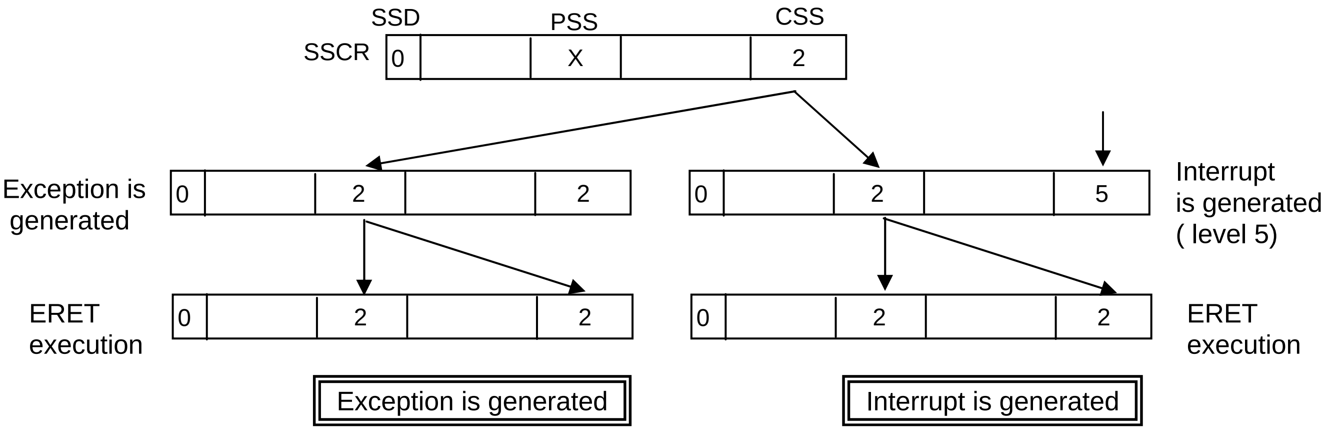
・When ERL=1 (Reset/ NMI is generated)

Status<ERL>	←	“0”
SSCR<CSS>	←	“PSS”
Branch off to	←	Address stored in ErrorEPC

・EXL=1 (general exception/ interrupt other than reset/NMI is generated)

Status<EXL>	←	“0”
SSCR<CSS>	←	“PSS”
Branch off to	←	Address store in EPC

(Note) The ERET instruction copies the values in SSCR<CSS> to SSCR<PSS> regardless of the exception/interrupt to be executed.
（If SSD bit is “0” and a shadow register is available.）
When an exception is generated, “CSS” is copied to “PSS”, however, CSS remains intact. That is, “CSS” values remain intact during and after handling exception.



(2) Interrupt Level Register (ILEV)

The following is required only for a hardware interrupt.

If the interrupt occurs, its interrupt level is set to ILEV<CSS> of the ILEV register in INTC as a mask level. Write “0” to ILEV<MLEV> to shift to the level of the previous interrupt because it does not automatically shift by an ERET instruction. If you want to set a new value to ILEV<CMASK>, set “1” to ILEV<MLEV> and write a value to ILEV<CMASK> simultaneously.

(3) Returning without ERET instruction

You can return from exception handler to a normal program without using an ERET instruction. In this case, settings of in Status<ERL>, Status<EXL> and SSCR<CSS> are the same as when the exception/ interrupt is generated. Change the settings if needed.

6.2 Reset Exception/ Non-maskable Interrupt (NMI)

A reset exception and non-maskable interrupt are branch off to the same reset exception vector address. When both a reset exception and NMI can be used, check the CP0 register to distinguish which one is generated.

6.2.1 Factors

Reset exception and NMI have factors shown below.

6.2.1.1 Factors of Reset Exception

・External reset pin

Setting an external reset pin to "L" and shift it to "H" generates a reset exception.

・WDT

WDT can generate a reset exception. See "Chapter 18 Watchdog timer" for details.

6.2.1.2 Factors of NMI

・External NMI pin

Setting an external NMI pin to "L" generates NMI.
(TMP19A44 has no external NMI pin.）

・WDT

WDT can generate NMI. See "Chapter 18 Watchdog timer" for details.

・Write bus error

When an operand write bus cycle results in a bus error, NMI is generated instead of a bus error exception.

6.2.2 Reset Exception/ NMI Handling

6.2.2.1 Flowchart

The following shows how a reset exception and NMI are handled.

⧈ indicates hardware handling. □ indicates software handling.

Processing	Details
Reset exception/ NMI factor is generated	A reset exception/ NMI factor is generated.
⇩	
CPU detects reset exception/ NMI	The CPU detects reset exception/ NMI.
⇩	
CPU/ CG handles reset exception/ NMI.	The CPU/ CG handles the reset exception/ NMI. ○Reset exception All the special function registers of the peripherals are initialized. The following bits in the CP0 register are automatically set. CP0: Status<ERL> ← "1" CP0: ErrorEPC ← "PC of exception"(Note) CP0: Status<BEV> ← "1" CP0: Status<NMI> ← "0" CP0: Status<RP> ← "0" CP0: SSCR<PSS> ← Register set number (when exception occurs) ○NMI The following bits in the CP0 register and CG are automatically set. CP0: Status<ERL> ← "1" CP0: ErrorEPC ← "PC of exception" (Note) CP0: Status<NMI> ← "1" CP0: Cause<BD> ← "1/0" (Note) CG: NMIFLG ← "1 is set to corresponding bit" CP0: SSCR<PSS> ← Register set number (when exception occurs)
⇩	
CPU branches off to exception handler	The CPU branches off to the reset exception vector address (0xBFC0_0000).
⇩	

Processing	Details
⇩	
Execute exception handler program	Program for the exception handler.
⇩	
Returning to preceding program	Configure to return to the preceding program from the exception handler.

(Note) Cause<BD> bit of the CP0 register is set to
-“1”: when an exception is generated by an instruction in a jump/ branch slot.
-“0”: other than the above.
ErrorEPC normally stores PC of the instruction causes an exception. It stores PC of a jump/ branch instruction when an exception is generated by an instruction in a jump/ branch slot.

6.2.2.2 How to Distinguish Reset Exception and Non-maskable Interrupt

A reset exception and non-maskable interrupt are branch off to the same reset exception vector address. To distinguish which one is generated, refer to the Status<NMI> bit of the CP0 register. Indicating “0” means reset exception is generated. Indicating “1” means non-maskable interrupt is generated. This bit is not changed by other exceptions/ interrupts.

A non-maskable interrupt has multiple interrupt factors. To distinguish which factor is used, refer to the NMIFLG register in the CG register.

6.2.2.3 Status<EXL> Bit of the CP0 Register

If “1” is set to the Status<EXL> bit, interrupts are prohibited.

“1” is set to the Status<EXL> bit due to exception handling of the CPU when a reset exception/ non-maskable interrupt occurs. This bit is automatically cleared to “0” when returning from an exception handler by executing the ERET instruction.

To execute an interrupt while handling an exception, set “0”.

6.2.2.4 PC Stored in the ErrorEPC

The ErrorEPC register in the CP0 register stores PC of the instruction causes an exception. As for a reset exception right after power-on, a value stored in the register is undefined since the PC value is undefined.

6.2.2.5 Return from Exception Handler

See “6.1.3.6 Returning from Exceptions/ interrupts”.

6.3 General Exceptions

This section describes general exceptions other than reset exceptions/ NMIs.

General exceptions are as follows:

Address error exception (instruction fetch)
Address error exception (load/store)
Bus error exception (instruction fetch)
Bus error exception (load)
Co-processor unusable exception
Reserved instruction exception
Integer overflow exception
Trap exception
System call exception
Breakpoint exception

6.3.1 Factors

General exceptions are generated when a certain instructions are executed or errors (i. e. in correct instruction fetch) are detected . See TX19A/H1 architecture for details of conditions that each exception is generated.

6.3.2 General Exception Handling

6.3.2.1 Flowchart

The following shows how a hardware interrupt is handled.

▣ indicates hardware handling. □ indicates software handling.

Processing	Details
General exception factor is generated.	The interrupt factor is generated.
⇩	
The CPU detects interrupt	The CPU detects the interrupt.
⇩	
CPU Handles interrupt	The CPU handles the interrupt. The following bits in the CP0 register are automatically set. CP0: Status<EXL> ← “1” CP0: EPC ← “PC of exception”(Note 1) CP0: Cause<BD> ← “1/0”(Note 1) CP0: Cause<ExcCode> ← Exception factor code CP0: BadVAddr ← Error address (Note 2) CP0: SSCR<PSS> ← Register set number (when exception occurs)
⇩	
CPU branches off to exception handler	The CPU branches off to the exception vector address (0xBFC0_0380).
⇩	
Execute exception handler program	Program for the exception handler.
⇩	
Returning to preceding program	Configure to return to the preceding program from the exception handler.

(Note 1) Cause<BD> bit of the CP0 register is set to

-“1”: when an exception is generated by an instruction in a jump/ branch slot.

-“0”: other than the above.

ErrorEPC normally stores PC of the instruction causes an exception. It stores PC of a jump/ branch instruction when an exception is generated by an instruction in a jump/ branch slot.

(Note 2) An error virtual address is stored when an address error exception occurs.

6.3.2.2 When General Exception Occurs

Generation of general exceptions other than a trap exception, a system call exception and a break point exception means abnormal condition is detected. When abnormal condition is detected, reset process is taken in most cases.

When a general exception (excluding reset exceptions/ NMI) other than a bus error exception (instruction fetch and data access) occurs, EPC stores PC that causes the exception. Therefore, if ERET is used for returning to normal operation, the same exception may occur.

6.3.2.3 Return from Exception Handler

See section "6.1.3.6 Returning from Exceptions/ Interrupts".

6.4 Software Interrupt

Software interrupt, which has two factors, occurs by setting the CP0 register.

A software interrupt has the same priority as a hardware interrupt. These interrupts may occur simultaneously and use the same exception handler.

6.4.1 Factors

6.4.1.1 Condition of Generating Interrupt

A software interrupt factor is recognized when the following three settings are configured in the CP0 register. At least 3 clocks are required to generate the interrupt after the factor is generated.

1) Status<IM>[1:0] (interrupt mask) is "1".

2) Cause<IP>[1:0] (interrupt request) is "1".

3) Status<IE> (interrupt enable bit) is "1".

The Status<IM>[1:0] and Cause<IP>[1:0] need to be used together. If IM is "1", IP must be "1", and if IM is "0", IP must be "0".

IE bit is to enable an interrupt and used for both software and hardware interrupts.

6.4.1.2 Condition of not Generating Interrupt

Even if the three settings shown above are completed, a software interrupt cannot be generated under the following conditions. The factor is suspended until the condition is changed as interrupt-acceptable.

・Status<ERL> or Status<EXL> bit is set

Status<ERL> bit is set to "1" when reset or NMI occurs. Status<EXL> bit is set to "1" when an interrupt or a general exception other than reset / NMI occurs. After an exception or an interrupt is generated, software interrupts are prohibited.

Both Status<ERL> and Status<EXL> bits are rewritable. Writing these bits to "0"using exception handler program enables software interrupts. These bits automatically return to "0" when returning from exception handler by ERET instruction.

・In debug mode

In debug mode, which is defined as duration from debug exception generation to returning by DRET instruction, a software interrupt is ignored.

・The CPU is stalled

When the CPU is stalled for any reason, a software interrupt is not generated.

6.4.1.3 Clearing Factors

A software interrupt factor is held unless the register settings are rewritten. In the exception handler, Status<EXL> of the CP0 register is set to "1". Therefore, another software interrupt is not generated. A software interrupt is recognized again if Status<EXL> is cleared to "0" when EXL bit is rewritten during exception handling or returning from exception handler by ERET.

To clear a software interrupt factor, clear one of the Status<IM>, Cause<IP> and Status<IE> bits to "0". Status<IE> is a bit to enable both software and hardware interrupts. Clearing this bit to "0" disables both interrupts.

6.4.1.4 Reading IVR Register

Read the IVR register of INTC after a software interrupt is generated. "4" can be read. Until the IVR is read, no hardware interrupt from INTC is accepted.

6.4.1.5 Return from Exception Handler

See "6.1.3.6 Returning from Exception/ Interrupts".

6.4.2 Software Interrupt Handling

6.4.2.1 Flowchart

The following shows how a software interrupt is handled.

▣ indicates hardware handling. □ indicates software handling.

Processing	Details
Interrupt factor is generated	Set the following bits in the CP0 register. Status<IM>[1:0] Cause<IP>[1:0] Status<IE>
CPU detects interrupt	The CPU detects the interrupt. Detection is unavailable in the following conditions. -Status<ERL> or Status<EXL> bit is set -In debug mode -The CPU is stalled. At least 3 clocks are required to detect the interrupt after the factor is generated.
CPU handles interrupt	The CPU handles the interrupt. The following bits in the CP0 register are automatically set. CP0: Status[EXL] ← "1" CP0: ErrorEPC ← "PC of exception"(Note) CP0: Cause[BD] ← "1/0"(Note)
CPU branches off to exception handler	The CPU branches off to the exception vector address. An interrupt vector address differs depending on combination of the BEV bit in the Status register and the IV bit in the Cause register.

	BEV=0	BEV=1
IV=0	0x8000_0180	0xBFC0_0380
IV=1	0x8000_0200	0xBFC0_0400

Processing	Details
Execute exception handler program	Program for the exception handler. Clear the interrupt factor if needed. Read INTC IVR register.
Returning to preceding program	Configure to return to the preceding program from the exception handler.

(Note) Cause<BD> bit of the CP0 register is set to
-"1": when an exception is generated by an instruction in a jump/ branch slot.
-"0": other than the above.
ErrorEPC normally stores PC of the instruction causes an exception. It stores PC of a jump/ branch instruction when an exception is generated by an instruction in a jump/ branch slot.

6.4.2.2 Interrupt Factor Indication

A software interrupt uses the same exception handler as a hardware interrupt. These interrupts may occur simultaneously. In this case, a hardware interrupt has a priority if a shadow register is available.

The Cause<IP>[4:0] and Status<IM>[4:0] of the CP0 register indicate an interrupt factor. If any of the Cause<IP>[1:0] indicates "1", it is the software interrupt factor. If any of the Cause<IP>[4:2] indicates "1", it is the hardware interrupt factor. Detecting both interrupts simultaneously enables both factors.

The Status<IM>[4:0] and Cause<IP>[4:0] can mask a factor. They need to be used together. A factor is not recognized even if the Cause<IP> is set to "1" when corresponding Status<IM> is "0".

The Cause<IP>[4:2] bits do not indicate an individual interrupt factor. It indicates interrupt level of a hardware interrupt. See "6.5 Hardware Interrupt" for details of hardware interrupt.

6.5 Hardware Interrupt

A hardware interrupt is generated when INTC notifies the CPU of an interrupt level.

INTC controls interrupt factors, prioritizes them and notifies the CPU of the level of the highest priority.

Interrupt factors that can be used for clearing standby mode is transmitted to INTC via CG. Therefore, setting of CG is also required.

This section describes route, factors, and settings of hardware interrupts. How to handle multiple interrupts and how to use an interrupt as a DMAC factor are also described.

6.5.1 Interrupt Factors

6.5.1.1 Interrupt Route

Fig. 6.1 shows an interrupt request route.

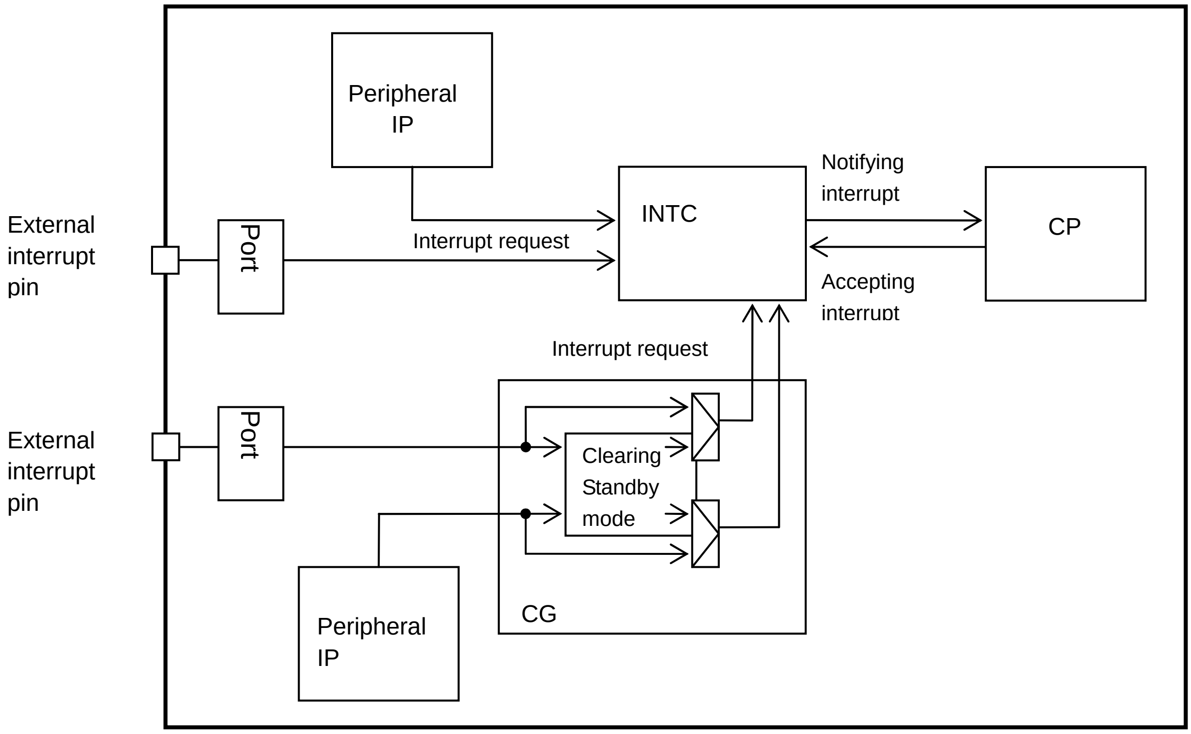


Fig. 6.1 Interrupt Route

6.5.1.2 Generation of Factors

A hardware interrupt is requested from external pins assigned to interrupt factors, peripheral IP and INTC.

・External pins

To use external pins as interrupt pins, configure the port control register.

・Peripheral IP

To generate an interrupt from peripheral IP, configure the appropriate peripheral IP so that it can output an interrupt.

See chapters of each IP for details.

・INTC

INTC can generate an interrupt factor by setting a register. This function is caller software set. To generate an interrupt, set the active state to “L” level and any desired interrupt level. To clear the factor, set the active state other than “L” level and set the interrupt level to “0”. See “6.5.2.2 Preparation (5) Preconfiguration 3 (software set)” for details.

6.5.1.3 Transmission of Factors

An interrupt request that is not used to clear standby mode is directly transmitted to INTC.

An interrupt request that is used to clear standby mode is transmitted to INTC via CG. Using or not using a factor for clearing standby mode can be set in the CG control register.

6.5.1.4 Request Acceptance

Upon detecting interrupt factors, INTC notifies the CPU of the interrupt factor with the highest priority according to the priority order.

The CPU recognizes the incoming interrupt level as a hardware interrupt factor, and notifies INTC that the interrupt request is accepted unless other exceptions with higher priorities are received.

6.5.1.5 List of Interrupt Factors

Table 6.5 shows the list of interrupt factors.

Table 6.5 List of Hardware Interrupt Factors

No.	Interrupt Factors	IVR	Activation trigger	Control register		Interrupt pin (Returning from standby)
				Interrupt controller	CG	
0	– (No interrupt factor)	0x000	–	IMC00		
1	Software interrupt	0x004	–			
2	INT0	0x008	Selectable (Note 1)		IMCGA	PA0
3	INT1	0x00C				PA1
4	INT2	0x010		IMC01		PA2
5	INT3	0x014				PA3
6	INT4	0x018			IMCGB	PA4
7	INT5	0x01C				PA5
8	INT6	0x020		IMC02		PJ6
9	INT7	0x024				PJ7
10	INT8	0x028			IMCGC	P86
11	INT9	0x02C				P87
12	INTA	0x030		IMC03		P61
13	INTB	0x034				P65
14	INTC	0x038				
15	INTD	0x03C				
16	INTE	0x040		IMC04		
17	INTF	0x044				
18	KWUP : Key-on wake-up	0x048	"H" level		IMCGD	32 pins
19	INT10	0x04C	Selectable (Note 1)		IMCGF	P72
20	INT11	0x050		IMC05		P73
21	INT12	0x054				P76
22	INT13	0x058				P77
23	INT14	0x05C			IMCG10	PJ2
24	INT15	0x060		IMC06		PJ3
25	INT16	0x064				PJ4
26	INT17	0x068				PJ5
27	INT18	0x06C			IMCG11	PH0
28	INT19	0x070		IMC07		PH1
29	INT1A	0x074				PH2
30	INT1B	0x078				PH3
31	INT1C	0x07C				
32	INT1D	0x080		IMC08		
33	INT1E	0x084				
34	INT1F	0x088				
35	INTRX0 : Serial reception (channel.0)	0x08C	Rising edge (Note 1)			
36	INTTX0 : Serial transmission (channel.0)	0x090		IMC09		
37	INTRX1 : Serial reception (channel.1)	0x094				
38	INTTX1 : Serial transmission (channel.1)	0x098				
39	INTRX2 : Serial reception (channel.2)	0x09C				
40	INTTX2 : Serial transmission (channel.2)	0x0A0		IMC0A		
41	HINTRX0 : High-speed serial reception (Hchannel.0)	0x0A4				
42	HINTTX0 :High-speed serial transmission (Hchannel.0)	0x0A8				
43	HINTRX1 : High-speed serial reception (Hchannel.1)	0x0AC				

No.	Interrupt Factors		IVR	Activation trigger	Control register		Interrupt pin (Returning from standby)
					Interrupt controller	CG	
44	HINTTX1	: High-speed serial transmission (Hchannel.1)	0x0B0	Rising edge (Note 1)	IMC0B		
45	HINTRX2	: High-speed serial reception (Hchannel.2)	0x0B4				
46	HINTTX2	: High-speed serial transmission (Hchannel.2)	0x0B8				
47	INTSBI0	: Serial bus interface 0	0x0BC				
48	INTADHPA	: Highest priority AD conversion complete interrupt A	0x0C0		IM0C		
49	INTADMA	: AD conversion monitoring function interrupt A	0x0C4				
50	INTADHPB	: Highest priority AD conversion complete interrupt B	0x0C8				
51	INTADMB	: AD conversion monitoring function interrupt B	0x0CC				
52	INTADHPC	: Highest priority AD conversion complete interrupt C	0x0D0		IM0D		
53	INTADMC	: AD conversion monitoring function interrupt C	0x0D4				
54	INTTB0	: 16bitTMRB 0	0x0D8				
55	INTTB1	: 16bitTMRB 1	0x0DC				
56	INTTB2	: 16bitTMRB 2	0x0E0		IM0E		
57	INTTB3	: 16bitTMRB 3	0x0E4				
58	INTTB4	: 16bitTMRB 4	0x0E8				
59	INTTB5	: 16bitTMRB 5	0x0EC				
60	INTTB6	: 16bitTMRB 6	0x0F0		IM0F		
61	INTTB7	: 16bitTMRB 7	0x0F4				
62	INTTB8	: 16bitTMRB 8	0x0F8				
63	INTTB9	: 16bitTMRB 9	0x0FC				
64	INTTBA	: 16bitTMRB A	0x100		IM10		
65	INTTBB	: 16bitTMRB B	0x104				
66	INTTBC	: 16bitTMRB C	0x108				
67	INTTBD	: 16bitTMRB D	0x10C				
68	INTTBE	: 16bitTMRB E	0x110		IM11		
69	INTTBF	: 16bitTMRB F	0x114				
70	INTADA	: A/D conversion completion A	0x118				
71	INTADB	: A/D conversion completion B	0x11C				
72	INTADC	: A/D conversion completion C	0x120		IM12		
73	INTTB10	: 16bitTMRB 10	0x124				
74	INTTB11	: 16bitTMRB 11	0x128				
75	PHCNT0	: Two-phase pulse input counter 0	0x12C			IMCGD	PA0/PA1
76	PHCNT1	: Two-phase pulse input counter 1	0x130		IM13		PA2/PA3
77	PHCNT2	: Two-phase pulse input counter 2	0x134			IMCGE	PA6/PA7
78	PHCNT3	: Two-phase pulse input counter 3	0x138				PB0/PB1
79	PHCNT4	: Two-phase pulse input counter 4	0x13C				PI0/PI1
80	PHCNT5	: Two-phase pulse input counter 5	0x140		IM14		PI2/PI3
81	INTCAP0	: Input capture 0	0x144				
82	INTCAP1	: Input capture 1	0x148				
83	INTCAP2	: Input capture 2	0x14C				
84	INTCAP3	: Input capture 3	0x150		IM15		
85	INTCMP0	: Compare 0	0x154				
86	INTCMP1	: Compare 1	0x158				
87	INTCMP2	: Compare 2	0x15C				

No.	Interrupt Factors		IVR	Activation trigger	Control register		Interrupt pin (Returning from standby)
					Interrupt controller	CG	
88	INTCMP3	: Compare 3	0x160		IM16		
89	INTCMP4	: Compare 4	0x164				
90	INTCMP5	: Compare 5	0x168				
91	INTCMP6	: Compare 6	0x16C				
92	INTCMP7	: Compare 7	0x170		IM17		
93	INTTBT	: Overflow	0x174				
94	INTRTC	: Real time clock timer	0x178			IMCGD	
95	INTDMA0	: DMA transfer completion (channel.0)	0x17C	“L” level			
96	INTDMA1	: DMA transfer completion (channel.1)	0x180		IM18		
97	INTDMA2	: DMA transfer completion (channel.2)	0x184				
98	INTDMA3	: DMA transfer completion (channel.3)	0x188				
99	INTDMA4	: DMA transfer completion (channel.4)	0x18C				
100	INTDMA5	: DMA transfer completion (channel.5)	0x190	“L” level	IM19		
101	INTDMA6	: DMA transfer completion (channel.6)	0x194				
102	INTDMA7	: DMA transfer completion (channel.7)	0x198				
103	Software set		0x19C				
104	Reserved		0x1A0				
105	Reserved		0x1A4				
106	Reserved		0x1A8				
107	Reserved		0x1AC				
108	Reserved		0x1B0				
109	Reserved		0x1B4				
110	Reserved		0x1B8				
111	Reserved		0x1BC				
112	Reserved		0x1C0				
113	Reserved		0x1C4				
114	Reserved		0x1C8				
115	Reserved		0x1CC				
116	Reserved		0x1D0				
117	Reserved		0x1D4				
118	Reserved		0x1D8				
119	Reserved		0x1DC				
120	Reserved		0x1E0				
121	Reserved		0x1E4				
122	Reserved		0x1E8				
123	Reserved		0x1EC				
124	Reserved		0x1F0				
125	Reserved		0x1F4				
126	Reserved		0x1F8				
127	Reserved		0x1FC				

(Note 1) Set “H” level when you use the factor as a standby clear interrupt.

6.5.1.6 Priority

Each of interrupt factors can be individually set to one of the seven interrupt priority levels by INTC. The interrupt level to be applied is set by IMC<IL> of INTC. ”7” is the highest priority level.

If multiple factors are generated simultaneously, an interrupt with the highest priority based on the interrupt level is selected.

If multiple factors with the same interrupt level are generated simultaneously, an interrupt with the smaller interrupt number is prioritized.

In addition to the individual interrupt level, set an interrupt level of the entire INTC in the ILEV<CMASK> bit of the INTC controller register. Interrupts with the lower priority than the specified in the ILEV<CMASK> bit are suspended. Default setting of the ILEV<CMASK>bit is 000”. This setting enables all the interrupt levels.

6.5.1.7 Active State

The active state indicates which change in signal of an interrupt factor triggers an interrupt. It can be set in the IMC<EIM> bit of the interrupt controller register.

The active state is selectable from the “H” level, “L” level, rising or falling edge but some interrupt factors designate a certain level.

As for the interrupt factors with specific conditions, configure as they are.

You can select a condition for the factor of which activation trigger in the Table is “Selectable”. To use an interrupt to clear standby mode, the active state must be set to “H” level. See the next section for details.

6.5.1.8 Factors for clearing standby mode

The interrupts that can be used for clearing standby mode are shown in the Table 6.5 together with the corresponding CG control register name.

To use an interrupt for clearing standby mode, enable the IMCG<INTEN> bit of the CG register and set an active state to the IMCG<EMCG> bit. The active state is selectable from the “H” level, “L” level, rising edge, falling edge or both edges.

When using an interrupt for clearing standby mode, an active state of the IMC<EIM> bit in the INTC controller register must be set to “H”. Setting the active state to “H” level allows INTC to receive a signal in “H” level after CG detects an interrupt.

6.5.1.9 DMAC activation by interrupt

An interrupt factor can be used to activate DAMC. In this case, no interrupt is generated.

By setting the IMC<DM> bit of the INTC register to “1”, a request to start transfer is output to DMAC if the active level condition is satisfied. To use an interrupt as the DMAC activation, the corresponding DMAC channel must be set in IMC<ILC>.

DREQFLG of the DMAC register is used to clear or monitor a transfer request. Reading this register can check whether a transfer is requested or not. Writing “1” to a bit that corresponding to the DMAC channel can clear an transfer request.

See “Chapter 10 DMA Controller“ for details.

6.5.2 Hardware Interrupt Handling

6.5.2.1 Flowchart

The following shows how a hardware interrupt is handled.

▭ indicates hardware handling. □ indicates software handling.

A hardware interrupt factor is generated by hardware or software.

Processing	Details	See
Settings for detection	Set the CPU coprocessor register and the INTC register to detect an interrupt. Set the clock generator as well if the interrupt is made to clear the standby mode. ○Common setting CPU coprocessor register INTC register ○Setting to clear standby mode Clock generator	6.5.2.2 Preparation
Settings for sending interrupt signal	Execute an appropriate setting to send the interrupt signal depending on the interrupt type. ○ Setting for interrupt from the external pin Port ○ Setting for interrupt from peripheral IP Peripheral IP (See chapters of relevant IP for details.)	
⇩		
Interrupt factor is generated	The interrupt factor is generated.	
⇩		

Not clearing standby mode

Clearing standby mode

Processing	Details	See
CG detects interrupt (factor to clear standby mode)	The interrupt, which is used for clearing the standby modes, is connected to INTC via the clock generator.	6.5.2.3 Detection by CG
INTC detects interrupt	INTC detects the interrupt according to the setting in the INTC IMC register. The CPU is notified of the interrupt factor with the highest priority according to the priority order.	6.5.2.4 Detection by INTC
CPU detects interrupt	The CPU detects the interrupt. Detection is unavailable in the following conditions. -Status<ERL> or Status<EXL> bit is set -In debug mode -The CPU is stalled.	6.5.2.5 Detection by CPU
CPU Handles interrupt	The CPU handles the interrupt. Update the field related to the CP0register. Switch a shadow register set. Output a signal to accept interrupt to INTC.	6.5.2.6 CPU processing
CPU branches off to exception handler	The CPU branches off to the exception vector address. An interrupt vector address differs depending on combination of the BEV bit in the Status register and the IV bit in the Cause register.	

	BEV=0	BEV=1
IV=0	0x8000_0180	0xBFC0_0380
IV=1	0x8000_0200	0xBFC0_0400

Processing	Details	See
⇩		
Execute exception handler program	Program for the exception handler. Clear the interrupt factor if needed.	6.5.2.7 Exception handler
⇩		
Returning to preceding program	Configure to return to the preceding program from the exception handler.	

6.5.2.2 Preparation

When preparing for an interrupt, you need to pay attention to the order of configuration to avoid any unexpected interrupt on the way.

Initiating an interrupt or changing its configuration must be implemented in the following order basically. Disable the interrupt by the CPU. Configure from the farthest route from the CPU. Then enable the interrupt by the CPU.

To configure the clock generator and INTC, you must follow the order indicated here not to cause any unexpected interrupt. First, configure the precondition. Secondly, clear the data related to the interrupt in the clock generator and INTC, and then enable the interrupt.

The following sections are listed in the order of interrupt handling and describe how to configure them.

(1) Disabling interrupt by CPU

(2) CPU registers setting

(3) Preconfiguration 1 (Interrupt from external pin)

(4) Preconfiguration 2 (interrupt from peripheral IP)

(5) Preconfiguration 3 (software)

(6) Configuring the clock generator (standby clear factor only)

(7) Configuring INTC

(8) Enabling interrupt by CPU

(1) Disabling interrupt by CPU

To make the CPU for not accepting any interrupt, write “0” to the Status<IE> bit of the CP0 register. This bit is set to be interrupt-disabled after reset.

● CPU register		
Status<IE>	←	“0” (interrupt disabled)

The following shows how to make IE bit “0”.

1. Set “0” to the Status<IE> bit of the CP0 register by 32 bit ISA MTC0 instruction.
2. Set “0” to the IER bit of the CP0 register by 32 bit ISA MTC0 instruction.
3. Set “0” to the Status<IE> bit of the CP0 register by 16 bit ISA MTC0 instruction.
4. Execute 16 bit ISA DI instruction.

You can select one of them. We recommend No. 2 and No. 4 that prevent code increase and enable high-speed processing.

(2) CPU registers setting

Configure the CP0 register.

By setting the Cause<IV> bit to “1” the vector addresses of interrupt exceptions and other exceptions can be distinguished.

Write “1” to the Status<IM4:2> bits to enable an interrupt from INTC.

Clear the SSCR<SSD> bit to “0” when using a shadow register.

●CPU register		
Cause<IV>	←	“1”
Status<IM>	←	“111”
SSCR<SSD>	←	“0”

(3) Preconfiguration 1 (Interrupt from external pin)

Set the port of the corresponding pin. Setting PnFCx [m] of the corresponding port function register to “1” allows the pin to be used as the function pin. Clearing PnCR [m] to “0” allows the pin to be used as the input port.

● Port register		
PnFCx<PnmFx>	←	“1”
PnCR<PnmC>	←	“0”

(Note) **n: port number**
m: corresponding bit
x: function register number

(4) Preconfiguration 2 (interrupt from peripheral IP)

The setting varies depending on the IP to be used. See chapters of relevant IP for details.

(5) Preconfiguration 3 (software set)

The software set interrupt is requested by setting the IMC register of INTC.

Set the active state to “L” level. You can set any interrupt level.

This factor is recognized upon setting the register. Set the active state and interrupt level as appropriate.

● Interrupt controller register		
IMC19<EIM671:670>	←	“00” (“L” level)
IMC19<IL672:670>	←	Interrupt level

(6) Configuring the clock generator (standby factor only)

You can omit this step unless the interrupt is used to clear standby.

You need to configure active state and enabling interrupt for an interrupt to clear the standby mode with the IMCG register of the clock generator. The IMCG register is capable of configuring each factor.

Before enabling an interrupt, clear the corresponding interrupt request already held. This can avoid unexpected interrupt. To clear corresponding interrupt request, write value corresponding to the interrupt to be used to the ICRCG register. See “6.6.3.2 INTCG Clear Register” for each value.

● Clock generator register		
IMCGn<EMCGm>	←	Active state
EICRCG<ICRCG>	←	Value corresponding to the interrupt to be used
IMCGn<INTmEN>	←	“1” (interrupt enabled)

(Note) n: register number
m: number assigned to interrupt factor

(7) Configuring INTC

With INTC, mask levels of the entire INTC, a base address of interrupt vector table and an active state and an interrupt level of each factor are configurable.

An interrupt level to be detected by INTC can be set in the ILEV register. An interrupt factor with the lower interrupt level than the level set in ILEV<CMASK> is suspended.

“0” is set as a default level. In this condition, all the interrupt levels can be detected. To change the level, write “1” to ILEV<MLEV> at the same time when writing the level to ILEV<CMASK>.

●INTC register		
ILEV<CMASK>	←	Interrupt mask level
ILEV<MLEV>	←	“1”

Set a base address of an interrupt vector table to IVR [31:9] of the IVR resister. Prepare the table in the address configured.

●INTC register		
IVR[31:9]	←	Interrupt handler base address

Clear an interrupt factor already held in INTC to generate an interrupt in appropriate timing. To clear the factor, write IVR of the interrupt you want to execute to the INTCLR register.

●INTC register		
INTCLR<EICLR>	←	IVR

Set an active state and an interrupt level to IMC register with which the setting for each factor is available.

Some interrupt factors designate a certain active state. See the IMC register section for details.

Set “H” level when using an interrupt to clear standby mode.

Set “L” level when using software set.

●INTC register		
IMCn<EIMm>	←	Active state
IMCn<ILm>	←	Interrupt level

(8) Enabling interrupt by CPU

Enable the interrupt by the CPU.

Set Status<IE> to “1”.

●CPU register		
Status<IE>	←	“1”

The following shows how to make IE bit “1”.

1. Set “1” to the Status<IE> bit of the CP0 register by 32 bit ISA MTC0 instruction.
2. Set a value other than “0” to the IER bit of the CP0 register by 32 bit ISA MTC0 instruction.
3. Set “1” to the Status<IE> bit of the CP0 register by 16 bit ISA MTC0 instruction.
4. Execute 16 bit ISA EI instruction.

You can select one of them. We recommend No. 2 and No. 4 that prevent code increase and enable high-speed processing.

6.5.2.3 Detection by CG

If the interrupt is used for clearing the standby mode, the interrupt factor is detected by a trigger for an active state specified in the clock generator, and notified to INTC.

The interrupt factor that enters active state triggered by a rising or falling edge is held in the clock generator after detection. However, if a signal in “H” or “L” level is specified as the trigger to enter the active state, the CPU considers that the interrupt factor is cleared upon exiting from the active state. Therefore, the active state needs to be kept until the interrupt is detected.

The interrupt detected by the clock generator is notified to INTC with a signal in “H” level. Therefore, you need to set INTC so that the interrupt to clear standby is activated by a “H” signal.

6.5.2.4 Detection by INTC

INTC handles an interrupt in the following procedure.

(1) INTC detects interrupts according to active state specified per factor. An interrupt factor with the highest priority of the interrupt level and interrupt number is selected.

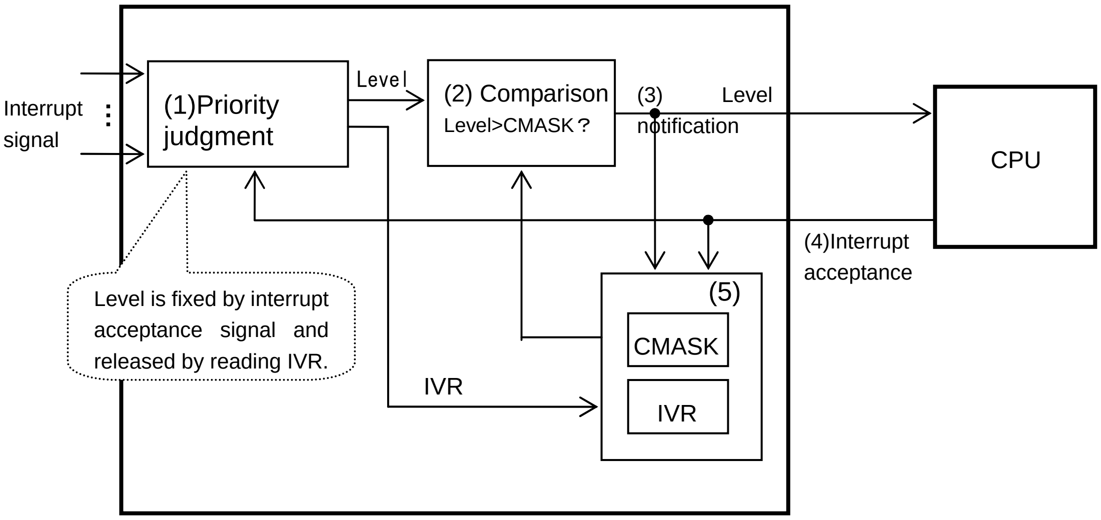
A factor, of which active state is rising or falling edge, is kept in the INTC after its detection. A factor, of which active state is “H” or “L” level, is considered to be cleared if its level is changed from the active state. Keep its active state until an interrupt is detected.

(2) INTC compares the interrupt level of the selected factor with the level specified in the ILEV<CMASK>. If the selected factor has the higher interrupt level, INTC notifies the CPU of the interrupt level.

(3) When the CPU detects the interrupt, it sends a signal, which indicates the interrupt is accepted, to INTC. If an interrupt with the higher priority is detected before the CPU sends the signal, an interrupt level is replaced by it.

(4) After receiving the signal, INTC sets the IVR to the IVR register and the interrupt level to the CMASK of the ILEV register.

(5) INTC holds the information of the interrupt detected until the IVR is read. An interrupt factor with the higher priority is suspended.



6.5.2.5 Detection by CPU

(1) Factors of hardware interrupt

A hardware interrupt factor is recognized by the CPU when the following three settings are configured.

1) Status<IM> (interrupt mask) is “111”.

2) Status<IE> (interrupt enable bit) is “1”.

3) Interrupt level notified from INTC is more than “1”.

(2) Condition of not Generating Interrupt

Even if the three settings shown above are completed, a hardware interrupt cannot be generated under the following conditions. The factor is suspended until the condition is changed as interrupt-acceptable.

1) Status<ERL> or Status<EXL> bit is set

Status<ERL> bit is set to “1” when reset or NMI occurs. Status<EXL> bit is set to “1” when an interrupt or a general exception other than reset / NMI occurs. After an exception or an interrupt is generated, hardware interrupts are prohibited.

Both Status<ERL> and Status<EXL> bits are rewritable. Writing these bits to “0”using exception handler program enables hardware interrupts. These bits automatically return to “0” when returning from exception handler by ERET instruction.

2) In debug mode

In debug mode, which is defined as duration from debug exception generation to returning by DRET instruction, a hardware interrupt is ignored.

3) The CPU is stalled

When the CPU is stalled for any reason, a hardware interrupt is not generated.

6.5.2.6 CPU processing

On detecting the interrupt, the CPU updates appropriate field in the CP0 register and branches off to the exception handler.

(1) Change in CP0 Register

Generating an interrupt changes the following values in the CP0 register.

●CP0 register		
Status<EXL>	←	"1" (Interrupt prohibited)
Cause<BD>	←	Exception/ interrupt occurs in a branch slot: "1" Others: "0"
Cause<ExcCode>	←	Code according to exception/ interrupt Ex) Interrupt: "0y00000"
EPC	←	PC of the instruction being executed when interrupt is generated.
SSCR<CSS>	←	Same value as the interrupt level (note) Interrupt level value is set as the new register set number.
SSCR<PSS>	←	CSS value of the pre-interrupt (*) Shadow register set number of the pre-interrupt

(Note) CSS and PSS change only when SSD of the SSCR register is set to "1" (using shadow register).

(2) Shadow Register Set Switching

By clearing SSCR<SSD> to "0", the shadow registers can be used and the register set is switched to that has the same number as the interrupt level.

Some registers are not switched.

The following illustrates the shadow registers.

Shadow Register Set No.	0	1	2	3	4	5	6	7
	r0							
	r26 (k0)							
	r27 (k1)							
	r28 (gp)							
	r29 (sp)	r29 (sp)						
	r1 (at)	r1 (at)	r1 (at)	r1 (at)	r1 (at)	r1 (at)	r1 (at)	r1 (at)
	r2 (v0)	r2 (v0)	r2 (v0)	r2 (v0)	r2 (v0)	r2 (v0)	r2 (v0)	r2 (v0)
	r3 (v1)	r3 (v1)	r3 (v1)	r3 (v1)	r3 (v1)	r3 (v1)	r3 (v1)	r3 (v1)
	r4 (a0)	r3 (a0)	r4 (a0)	r4 (a0)	r4 (a0)	r4 (a0)	r4 (a0)	r4 (a0)
	r5 (a1)	r5 (a1)	r5 (a1)	r5 (a1)	r5 (a1)	r5 (a1)	r5 (a1)	r5 (a1)
	r6 (a2)	r6 (a2)	r6 (a2)	r6 (a2)	r6 (a2)	r6 (a2)	r6 (a2)	r6 (a2)
	r7 (a3)	r7 (a3)	r7 (a3)	r7 (a3)	r7 (a3)	r7 (a3)	r7 (a3)	r7 (a3)
	r8 (t0)	r8 (t0)	r8 (t0)	r8 (t0)	r8 (t0)	r8 (t0)	r8 (t0)	r8 (t0)
	r9 (t1)	r9 (t1)	r9 (t1)	r9 (t1)	r9 (t1)	r9 (t1)	r9 (t1)	r9 (t1)
	r10 (t2)	r10 (t2)	r10 (t2)	r10 (t2)	r10 (t2)	r10 (t2)	r10 (t2)	r10 (t2)
	r11 (t3)	r11 (t3)	r11 (t3)	r11 (t3)	r11 (t3)	r11 (t3)	r11 (t3)	r11 (t3)
	r12 (t4)	r12 (t4)	r12 (t4)	r12 (t4)	r12 (t4)	r12 (t4)	r12 (t4)	r12 (t4)
	r13 (t5)	r13 (t5)	r13 (t5)	r13 (t5)	r13 (t5)	r13 (t5)	r13 (t5)	r13 (t5)
	r14 (t6)	r14 (t6)	r14 (t6)	r14 (t6)	r14 (t6)	r14 (t6)	r14 (t6)	r14 (t6)
	r15 (t7)	r15 (t7)	r15 (t7)	r15 (t7)	r15 (t7)	r15 (t7)	r15 (t7)	r15 (t7)
	r16 (s0)	r16 (s0)	r16 (s0)	r16 (s0)	r16 (s0)	r16 (s0)	r16 (s0)	r16 (s0)
	r17 (s1)	r17 (s1)	r17 (s1)	r17 (s1)	r17 (s1)	r17 (s1)	r17 (s1)	r17 (s1)
	r18 (s2)	r18 (s2)	r18 (s2)	r18 (s2)	r18 (s2)	r18 (s2)	r18 (s2)	r18 (s2)
	r19 (s3)	r19 (s3)	r19 (s3)	r19 (s3)	r19 (s3)	r19 (s3)	r19 (s3)	r19 (s3)
	r20 (s4)	r20 (s4)	r20 (s4)	r20 (s4)	r20 (s4)	r20 (s4)	r20 (s4)	r20 (s4)
	r21 (s5)	r21 (s5)	r21 (s5)	r21 (s5)	r21 (s5)	r21 (s5)	r21 (s5)	r21 (s5)
	r22 (s6)	r22 (s6)	r22 (s6)	r22 (s6)	r22 (s6)	r22 (s6)	r22 (s6)	r22 (s6)
	r23 (s7)	r23 (s7)	r23 (s7)	r23 (s7)	r23 (s7)	r23 (s7)	r23 (s7)	r23 (s7)
	r24 (t8)	r24 (t8)	r24 (t8)	r24 (t8)	r24 (t8)	r24 (t8)	r24 (t8)	r24 (t8)
	r25 (t9)	r25 (t9)	r25 (t9)	r25 (t9)	r25 (t9)	r25 (t9)	r25 (t9)	r25 (t9)
	r30 (fp)	r30 (fp)	r30 (fp)	r30 (fp)	r30 (fp)	r30 (fp)	r30 (fp)	r30 (fp)
	r31 (ra)	r31 (ra)	r31 (ra)	r31 (ra)	r31 (ra)	r31 (ra)	r31 (ra)	r31 (ra)

(3) Branch to the Exception Handler

An interrupt vector address differs depending on combination of the Status<BEV> bit and the Cause<IV> bit.

Status<BEV> is set to “1” after reset. Keep the setting if you set an exception vector address in the internal ROM.

The Cause<IV> bit is undefined after reset. Setting this bit to “1” is required. By setting this bit to “1” the vector addresses of interrupt exceptions and other exceptions can be distinguished.

	BEV = 0	BEV = 1
IV = 0	0x8000_0180	0xBFC0_0380
IV = 1	0x8000_0200	0xBFC0_0400

(4) Interrupt Acceptance Signal

When the CPU detects the interrupt, it sends a signal, which indicates the interrupt is accepted, to INTC. By the signal, INTC holds the interrupt level and interrupt number that have the highest priority at that time. See “6.5.2.3 Detection by CG” for INTC operation.

6.5.2.7 Exception Handler

The exception handler requires user programming. We describe how to judge and clear a factor here. The procedure is as follows:

(1) Judging factor
(2) Clearing interrupt request (INTC)
(3) Branch to Interrupt Handler
(4) Clearing Interrupt Factor
(5) Interrupt Handler Processing
(6) Returning from Exception Handler

(1) Judging factor

The registers used for judging a factor are shown below.

●CP0 register	
Cause<IP>[4:2]	Hardware interrupt level “000”: No hardware interrupt factor
Cause<IP>[1:0]	Write “1” for using software interrupt. “00”: No software interrupt factor
●INTC	
IVR[8:0]	Value according to factors (See factor list)

The Cause<IP>[4:0] indicates an interrupt factor. If any of the Cause<IP>[1:0] indicates “1”, it is the software interrupt factor. If any of the Cause<IP>[4:2] indicates “1”, it is the hardware interrupt factor. Detecting both interrupts simultaneously enables both factors.

If an interrupt is generated from hardware, the IVR register of INTC indicates the factor. Value is set to IVR[8:0] according to a factor. See factor list for details.

(2) Clearing interrupt request (INTC)

By the signal from the CPU, INTC fixes a request level. INTC clears the request sent to the CPU by setting a value of IVR[8:0] to INTCLR register.

●INTC	
INTCLR[8:0]	Value of IVR[8:0] according to factor (See factor list)

Upon clearing the request, an interrupt with the highest priority that suspended in INTC is requested to the CPU.

When the CPU branches off to the exception handler, “1” is set to the Status<EXL> bit, and next interrupts are prohibited. The setting to use multiple interrupts is described in ”6.6.2.7Multiple Interrupts”.

(3) Branch to Interrupt Handler

If the start address of the interrupt handler table is set in IVR [31:9] of the interrupt controller register, the values specified in IVR [31:0] can be used as the start address. The CPU branches off to the address.

(4) Clearing Interrupt Factor

As for an interrupt detected by a level, an interrupt request exists unless the factor is cleared. Clearing the factor is required.

As for an interrupt detected by an edge, an interrupt factor is cleared by setting the corresponding interrupt IVR to the INTCLR register. When the next valid edge is detected, a new factor is recognized. See (2) Clearing interrupt request (INTC).

(5) Interrupt Handler Processing

Usually an interrupt handler save the required register and handles interrupts. If the shadow register set is enabled (SSCR<SSD> of the CP0 register is “0”), contents in the general purpose register except for r26, r27, r28 and r29（Shadow Register Set Number 1~7） are automatically saved. Save by the user program is not needed.

Save the contents of the Status, EPC, SSCR, HI, LO, Cause and Config registers, if needed.

General exceptions are acceptable if interrupts are prohibited. We recommend saving the contents of the general purpose register and CP0 register, which may be rewritten by general interrupts, even if multiple interrupts are not used.

(6) Returning from Exception Handler

See “6.1.3.6 Returning from Exception/ Interrupts”.

6.5.2.8 Multiple Interrupts

In "multiple interrupts" processing, an interrupt with higher interrupt level is processed while an interrupt is being processed. When an interrupt request is accepted, ILEV <CMASK> of INTC is automatically updated to the interrupt level of the interrupt accepted. It enables an interrupt with the higher interrupt level than the current one.

(1) Preconfiguration

To execute multiple interrupts, the contents in the following registers must be saved before enabling the interrupts. If not, these registers are overwritten by the second and subsequent interrupts.

●CP0 register	
EPC	PC of the interrupt generated
SSCR	Shadow register control
Status	CPU status

In addition to the registers shown above, save the contents of the HI, LO, Cause and Config registers if needed.

(2) Enabling Multiple Interrupts

When an interrupt is accepted, Status <EXL> of the CP0 register is set to "1" disabling further interrupts. Clearing the bit to “0” enables an interrupt.

(3) Returning from Multiple Interrupt

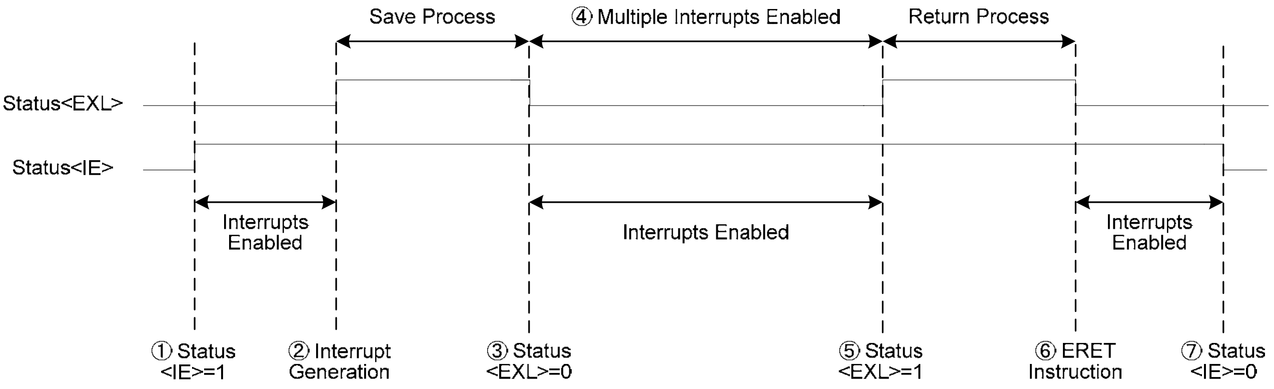
After returning from the multiple interrupts handler, the CPU restarts the suspended interrupt from where the interrupt is discontinued. Set Status<EXL> of the CP0 register to “1” to disable interrupts. It can prevent a new interrupt from occurring before returning from the multiple interrupts. Otherwise, original interrupt data may be destroyed.

(4) Proper use of Status <EXL> and Status <IE>

Status<EXL> and Status<IE> control to enable or disable the multiple interrupts.

Status <EXL> is set to "1" upon interrupt generation and cleared to "0" by the ERET instruction. Interrupts must be prohibited while saving the register contents described in (1) and returning from multiple interrupts described in (3). Usually, interrupts can be prohibited with Status<EXL> controlled by hardware. Status <IE> is used for other general interrupt enable/disable control functions.

The following describes how the multiple interrupts are handled.



①Status<IE>=1

Interrupts can be enabled by setting Status <IE> of the CP0 register to "1" while Status <EXL> is set to "0." This optional setting is made by the software program when it is necessary.

② Interrupt Generation

When an interrupt is generated, Status <EXL> of the CP0 register is set to "1" disabling further interrupts. This process is automatically performed by hardware.

③ Status <EXL> = 0

If multiple interrupts are to be enabled, it is necessary to set Status <EXL> of the CP0 register to "0" to enable interrupts after relevant registers are saved. If interrupts are enabled before saving registers, a higher priority level interrupt could corrupt the register data. This optional setting is made by the software program when it is necessary.

④ Multiple Interrupts Enabled

This is the period multiple interrupts are enabled. Interrupts with a level higher than the present interrupt level (ILEV <CMASK>) are to be accepted. If it is desired to disable interrupts during this period, set Status <IE> of the CP0 register to "0."

⑤ Status <EXL> = 1

If multiple interrupts are enabled, it is necessary to set Status <EXL> of the CP0 register to "1" to disable interrupts before returning relevant register values. If registers are saved before disabling interrupts, a higher priority level interrupt could corrupt the register data. This optional setting is made by the software program when it is necessary.

⑥ ERET Instruction

This instruction returns the system to the state before the interrupt generation. If this instruction is executed while Status <EXL> of the CP0 register is set to "1," the Status <EXL> will be automatically set to "0" and interrupt is enabled (provided that Status <IE> of the CP0 register is set to "1").

⑦ Status<IE>=0

Interrupts can be disabled by setting Status <IE> of the CP0 register to "0." This optional setting is made by the software program when it is necessary.

6.6 Registers

The CP0 register, the clock generator register and the interrupt controller register are used for exceptions/ interrupts.

To access to the CP0 register, you need to specify a register number with a system control co-processor (CP0) instruction.

The clock generator register and the interrupt controller register have own addresses. To access to the registers, you need specify the address with load/ store instruction.

The following shows the addresses of each register.

6.6.1 Register Map

●CP0 register		
BadVAddr	Bad virtual address register	No. 8
Status	Status register	No. 12
Cause	Cause register	No. 13
EPC	Exception program counter register	No. 14
ErrorEPC	Error exception program counter register	No. 30
SSCR	Shadow register set control register	No. 22 (SEL0) No. 9 (SEL6)
IER	Interrupt enable register	No. 9

●Clock generator registers		
IMCGA	CG interrupt mode control register A	0xFF00_1720
IMCGB	CG interrupt mode control register B	0xFF00_1724
IMCGC	CG interrupt mode control register C	0xFF00_1728
IMCGD	CG interrupt mode control register D	0xFF00_172C
IMCGE	CG interrupt mode control register E	0xFF00_1730
ICRCG	CG interrupt request clear register	0xFF00_1714
NMIFLG	NMI flag register	0xFF00_1718
RSTFLG	Reset flag register	0xFF00_171C

●Interrupt controller register		
IVR	Interrupt vector register	0xFF00_1080
ILEV	Interrupt level register	0xFF00_110C
IMC00	Interrupt mode control register 00	0xFF00_1000
IMC01	Interrupt mode control register 01	0xFF00_1004
IMC02	Interrupt mode control register 02	0xFF00_1008
IMC03	Interrupt mode control register 03	0xFF00_100C
IMC04	Interrupt mode control register 04	0xFF00_1010
IMC05	Interrupt mode control register 05	0xFF00_1014
IMC06	Interrupt mode control register 06	0xFF00_1018
IMC07	Interrupt mode control register 07	0xFF00_101C
IMC08	Interrupt mode control register 08	0xFF00_1020
IMC09	Interrupt mode control register 09	0xFF00_1024
IMC0A	Interrupt mode control register 0A	0xFF00_1028
IMC0B	Interrupt mode control register 0B	0xFF00_102C
IMC0C	Interrupt mode control register 0C	0xFF00_1030
IMC0D	Interrupt mode control register 0D	0xFF00_1034
IMC0E	Interrupt mode control register 0E	0xFF00_1038
IMC0F	Interrupt mode control register 0F	0xFF00_103C
IMC10	Interrupt mode control register 10	0xFF00_1040
IMC11	Interrupt mode control register 11	0xFF00_1044
IMC12	Interrupt mode control register 12	0xFF00_1048
IMC13	Interrupt mode control register 13	0xFF00_104C
IMC14	Interrupt mode control register 14	0xFF00_1050
IMC15	Interrupt mode control register 15	0xFF00_1054
IMC16	Interrupt mode control register 16	0xFF00_1058
IMC17	Interrupt mode control register 17	0xFF00_105C
IMC18	Interrupt mode control register 18	0xFF00_1060
IMC19	Interrupt mode control register 19	0xFF00_1064
INTCLR	Interrupt request clear register	0xFF00_10C0
DREQFLG	DMA request clear flag register	0xFF00_10C4

6.6.2 CP0 Register

6.6.2.1 VadVAddr Register

When an address error exception (AdEL or AdES) occurs, the BadVaddr (Bad Virtual Address) register stores its virtual address.

VadVAddr
(No.8)

	7	6	5	4	3	2	1	0
bit Symbol	BadVAddr							
Read/Write	R							
After reset	Undefined							
Function	Virtual address of an address error exception（bit 7~0）							
	15	14	13	12	11	10	9	8
bit Symbol	BadVAddr							
Read/Write	R							
After reset	Undefined							
Function	Virtual address of an address error exception（bit 15~8）							
	23	22	21	20	19	18	17	16
bit Symbol	BadVAddr							
Read/Write	R							
After reset	Undefined							
Function	Virtual address of an address error exception（bit 23~16）							
	31	30	29	28	27	26	25	24
bit Symbol	BadVAddr							
Read/Write	R							
After reset	Undefined							
Function	Virtual address of an address error exception（bit 31~24）							

6.6.2.2 Status Register

Status

(No.12)

	7	6	5	4	3	2	1	0
bit Symbol	KX	SX	UX	UM		ERL	EXL	IE
Read/Write	R	R	R	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	1	0	0
Function	“0” is read.			Operating Mode: 0: Kernel mode 1: User mode Write “0”.	“0” is read.	Error Level: Set when a Reset or NMI exception is taken. (Note 1)	Exception Level: Set when an exception other than Reset and NMI exceptions is taken. (Note 2)	Interrupt Enable: 0: Interrupts are disabled. 1: Interrupts are enabled. (Note 3)

	15	14	13	12	11	10	9	8
bit Symbol	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
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	Write “000”.			Interrupt Mask: (for hardware interrupt) Write ”111”.			Interrupt Mask: (for software interrupt) Write ”1”.	

	23	22	21	20	19	18	17	16
bit Symbol	PX	BEV	TS	SR	NMI		Impl	
Read/Write	R	R/W	R	R	R/W	R	R	
After reset	0	1	0	0	0	0	0	0
Function	“0” is read.	Vector address for Bootstrap Exception 1: Set in internal ROM 0: Set in external memory (Note 4)	“0” is read.		NMI interrupt 1: Generated 0: Not generated (Note 5)	“0” is read.		

	31	30	29	28	27	26	25	24
bit Symbol	CU3	CU2	CU1	CU0	RP	FR	RE	MX
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R
After reset	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Function	CP3 availability 0: Unavailable 1: Available Write “0”.	CP2 availability 0: Unavailable 1: Available Write “0”.	CP1 availability 0: Unavailable 1: Available Write “1” for using FPU.	CP0 availability 0: Unavailable 1: Available (Note 6)	Low-power consumption mode 0: Halt 1: Doze Specifies core mode in IDLE mode.	“0” is read.	“0” is read.	“0” is read.

(Note 1) When this bit is set:

- The processor is running is Kernel mode.
- Interrupts are disabled.
- The ERET instruction will use the return address held in the ErrorEPC register.

(Note 2) When this bit is set:

- The processor is running in Kernel mode.
- Interrupts are disabled.
- The EPC register and the BD bit in the Cause register will not be updated if another exception is taken.

(Note 3) The IE bit is not automatically set or cleared by the interrupt response sequence or the ERET instruction. (This bit is cleared upon reset.)

(Note 4) An interrupt vector address differs depending on combination of the BEV bit in the Status register and the IV bit in the Cause register.

	BEV = 0	BEV = 1
IV = 0	0x8000_0180	0xBFC0_0380
IV = 1	0x8000_0200	0xBFC0_0400

(Note 5) Writing 0 clears this bit. Writing 1 to this bit is ignored.

(Note 6) Write “1” when using CP0 in user mode. In Kernel mode, CP0 is always available, regardless of the value of the CU0 bit.

6.6.2.3 Cause Register

The Cause register indicates the cause of the last exception.

Cause

(No.13)

	7	6	5	4	3	2	1	0
bit Symbol		ExcCode						
Read/Write	R	R					R	
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Exception code 00000: Interrupt (software and hardware) 00100: Address Error (instruction fetch or load) 00101: Address Error (store access) 00110: Bus Error (instruction fetch) 00111: Bus Error (data access: load) 01000: System Call exception 01001: Breakpoint exception 01010: Reserved Instruction exception 01011: Coprocessor Unusable exception 01100: Integer Overflow exception 01101: Trap exception					“0” is read.	

	15	14	13	12	11	10	9	8
bit Symbol	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Function	Interrupt Request (Hardware) Interrupt level is set in IP[4:2] when hardware interrupt is requested.						Interrupt Request (Software) Write “1” for using software interrupt.	

	23	22	21	20	19	18	17	16
bit Symbol	IV	WP						
Read/Write	R/W	R	R					
After reset	Undefined	0	0	0	0	0	0	0
Function	Interrupt vector 1:Different from exception vector 0:Same as exception vector (Note 1)	“0” is read.						

	31	30	29	28	27	26	25	24
bit Symbol	BD		CE1	CE0				
Read/Write	R	R	R		R			
After reset	Undefined	0	Undefined		0	0	0	0
Function	Set when an exception occurred in a jump or branch delay slot. (Note 2)	“0” is read.	Indicates the coprocessor unit number referenced when a Coprocessor Unusable exception was taken. (Note 3)		“0” is read.			

(Note 1) An interrupt vector address differs depending on combination of the BEV bit in the Status register and the IV bit in the Cause register. See Note 4 in the Status register section.

(Note 2) The processor updates the BD bit only if the EXL bit is 0 when an interrupt or exception occurred.

(Note 3) The value in this field is undefined for any other exception.

6.6.2.4 EPC Register

Reading/ writing is available with the EPC register.

When an exception/ instruction other than RESET/ NMI occurs, the EPC register stores its address. (If the instruction in the branch slot causes an exception/ interrupt, the address of the precedent branch instruction is stored.)

When an exception/ instruction other than RESET/ NMI is generated, Status<EXL> bit is set to "1". If "1" is set, the normal operation restarts from the address set in the EPC register in executing ERET instruction.

The processor does not write to the EPC register when the EXL bit in the Status register is set to "1".

EPC
(No.14)

	7	6	5	4	3	2	1	0
bit Symbol	EPC							
Read/Write	R/W							
After reset	Undefined							
Function	Exception Program Counter（bit 7~0）							

	15	14	13	12	11	10	9	8
bit Symbol	EPC							
Read/Write	R/W							
After reset	Undefined							
Function	Exception Program Counter（bit 15~8）							

	23	22	21	20	19	18	17	16
bit Symbol	EPC							
Read/Write	R/W							
After reset	Undefined							
Function	Exception Program Counter（bit 23~16）							

	31	30	29	28	27	26	25	24
bit Symbol	EPC							
Read/Write	R/W							
After reset	Undefined							
Function	Exception Program Counter（bit 31~24）							

6.6.2.5 ErrorEPC Register

Reading/ writing is available with the EPC register.

When an exception/ instruction other than RESET/ NMI occurs, the EPC register stores its address. (If the instruction in the branch slot causes an exception/ interrupt, the address of the precedent branch instruction is stored).

When an exception/ instruction other than RESET/ NMI is generated, Status<EXL> bit is set to "1". If "1" is set, the normal operation restarts from the address set in the EPC register in executing ERET instruction.

ErrorEPC
(No.30)

	7	6	5	4	3	2	1	0
bit Symbol	ErrorEPC							
Read/Write	R/W							
After reset	Undefined							
Function	Error Exception Program Counter（bit 7~0）							
	15	14	13	12	11	10	9	8
bit Symbol	ErrorEPC							
Read/Write	R/W							
After reset	Undefined							
Function	Error Exception Program Counter（bit 15~8）							
	23	22	21	20	19	18	17	16
bit Symbol	ErrorEPC							
Read/Write	R/W							
After reset	Undefined							
Function	Error Exception Program Counter（bit 23~16）							
	31	30	29	28	27	26	25	24
bit Symbol	ErrorEPC							
Read/Write	R/W							
After reset	Undefined							
Function	Error Exception Program Counter（bit 31~24）							

6.6.2.6 Shadow Register Set Control Register: SSCR Register

SSCR
(No.22
or 9:SEL6)

	7	6	5	4	3	2	1	0
bit Symbol					CSS3	CSS2	CSS1	CSS0
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.				Current Shadow Register Set x000: Main GPRs x001: Shadow Register 1 x010: Shadow Register 2 x011: Shadow Register 3 x100: Shadow Register 4 x101: Shadow Register 5 x110: Shadow Register 6 x111: Shadow Register 7			
	15	14	13	12	11	10	9	8
bit Symbol					PSS3	PSS2	PSS1	PSS0
Read/Write	R				R/W			
After reset	0	0	0	0	Undefined			
Function	“0” is read.				Previous Shadow Register Set x000: Main GPRs x001: Shadow Register 1 x010: Shadow Register 2 x011: Shadow Register 3 x100: Shadow Register 4 x101: Shadow Register 5 x110: Shadow Register 6 x111: Shadow Register 7			
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							
	31	30	29	28	27	26	25	24
bit Symbol	SSD							
Read/Write	R/W	R						
After reset	1	0						
Function	Shadow Register Set 0: Enabled 1: Disabled	“0” is read.						

(Note 1) When the processor accepts an interrupt request from the interrupt controller, the value of the CSS field is copied to the PSS field, and the CSS field is updated with the value of the new interrupt request level.

(Note 2) On an ERET, the value of the PSS field is restored to the CSS field.

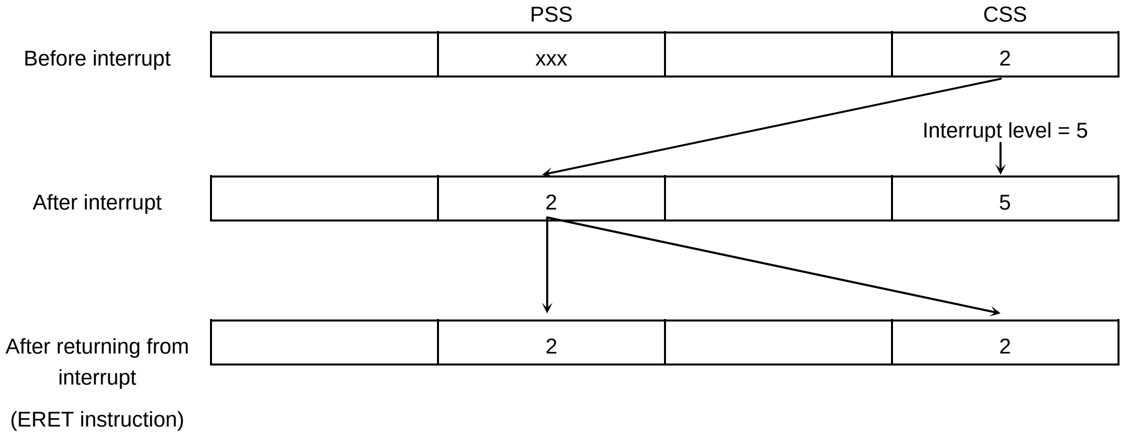
(Note 3) The instruction that modifies the contents of the SSCR register must be followed by two NOPs to avoid pipeline hazards.

```
Ex.)    MTC0    r18, SSCR
        NOP
        NOP
        ADD     r19, r12, r13
```

(Note 4) When the SSD bit is set, the Shadow Register Set is not updated by any interruptions.

(Note 5) When the SSD bit is set, only shadow set 0 is accessible, and the value of the CSS field is ignored.

(Note 6) The following shows how the SSCR register operates in generating an interrupt or in returning from an interrupt.



6.6.2.7 IER Register

The IER register is used to set or clear the IE bit in the Status register. Writing “0” to the IER register causes the IE bit in the Status register to be cleared. Writing a value other than “0” to the IER register causes the IE bit to be set.

IER
(No.9:SEL7)

	7	6	5	4	3	2	1	0
bit Symbol	IER							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt Enable Register（bit 7~0）							
	15	14	13	12	11	10	9	8
bit Symbol	IER							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt Enable Register（bit 15~8）							
	23	22	21	20	19	18	17	16
bit Symbol	IER							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt Enable Register（bit 23~16）							
	31	30	29	28	27	26	25	24
bit Symbol	IER							
Read/Write	R/W							
After reset	Undefined							
Function	Interrupt Enable Register（bit 31~24）							

6.6.3 Clock Generator Register

6.6.3.1 INTCG register (STOP/SLEEP/IDLE clear interrupt)

IMCGA

	7	6	5	4	3	2	1	0
bit Symbol		EMCG02	EMCG01	EMCG00	EMST01	EMST00		INT0EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT0 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT0 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT0 clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol		EMCG12	EMCG11	EMCG10	EMST11	EMST10		INT1EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT1 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT1 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT1 clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol		EMCG22	EMCG21	EMCG20	EMST21	EMST20		INT2EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT2 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT2 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT2 clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol		EMCG32	EMCG31	EMCG30	EMST31	EMST30		INT3EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT3 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT3 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT3 clear input 0: Disable 1: Enable

(Note 1) **Refer to EMSTxx bit to know the active condition which is used for clearing standby.**

(Note 2) **Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.**

IMCGB

	7	6	5	4	3	2	1	0
bit Symbol		EMCG42	EMCG41	EMCG40	EMST41	EMST40		INT4EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT4 standby clear request (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT4 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT4 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG52	EMCG51	EMCG50	EMST51	EMST50		INT5EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT5 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT5 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT5 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG62	EMCG61	EMCG60	EMST61	EMST60		INT6EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT6 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT6 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT6 Clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG72	EMCG71	EMCG70	EMST71	EMST70		INT7EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT7 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT7 standby clear request 00: — 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT7 Clear input 0: Disable 1: Enable

(Note 1) **Refer to EMSTxx bit to know the active condition which is used for clearing standby.**

(Note 2) **Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.**

IMCGC

	7	6	5	4	3	2	1	0
bit Symbol		EMCG82	EMCG81	EMCG80	EMST81	EMST80		INT8EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INT8 standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT8 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INT8 clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol		EMCG92	EMCG91	EMCG90	EMST91	EMST90		INT9EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INT9 standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT9 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INT9 clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol		EMCGA2	EMCGA1	EMCGA0	EMSTA1	EMSTA0		INTAEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INTA standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTA standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INTA clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol		EMCGB2	EMCGB1	EMCGB0	EMSTB1	EMSTB0		INTBEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INTB standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INTB standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INTB clear input 0: Disable 1: Enable

(Note 1) **Refer to EMSTxx bit to know the active condition which is used for clearing standby.**

(Note 2) **Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.**

IMCGD

	7	6	5	4	3	2	1	0
bit Symbol		EMCGC2	EMCGC1	EMCGC0				KWUPEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of KWUP standby clear request. **Set it as shown below.** 001: "H" level			Undefined value is read.		"0" is read.	KWUP clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol		EMCGD2	EMCGD1	EMCGD0				INTRTCEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of INTRTC standby clear request. **Set it as shown below.** 010: Falling edge			Undefined value is read.		"0" is read.	INTRTC clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol		EMCGE2	EMCGE1	EMCGE0				PHCNT0EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of PHCNT0 standby clear request. **Set it as shown below.** 011: Rising edge			Undefined value is read.		"0" is read.	PHCNT0 clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol		EMCGF2	EMCGF1	EMCGF0				PHCNT1EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of PHCNT1 standby clear request. **Set it as shown below.** 011: Rising edge			Undefined value is read.		"0" is read.	PHCNT1 clear input 0: Disable 1: Enable

(Note 1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

(Note 2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

IMCGE

	7	6	5	4	3	2	1	0
bit Symbol		EMCG102	EMCG101	EMCG100				PHCNT2EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of PHCNT2 standby clear request. **Set it as shown below.** 011: Rising edge			Undefined value is read.		"0" is read.	PHCNT2 clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol		EMCG112	EMCG111	EMCG110				PHCNT3EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of PHCNT3 standby clear request. **Set it as shown below.** 011: Rising edge			Undefined value is read.		"0" is read.	PHCNT3 clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol		EMCG122	EMCG121	EMCG120				PHCNT4EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of PHCNT4 standby clear request. **Set it as shown below.** 011: Rising edge			Undefined value is read.		"0" is read.	PHCNT4 clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol		EMCG132	EMCG131	EMCG130				PHCNT5EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	-	-	0	0
Function	"0" is read.	Active state setting of PHCNT5 standby clear request. **Set it as shown below.** 011: Rising edge			Undefined value is read.		"0" is read.	PHCNT5 clear input 0: Disable 1: Enable

(Note 1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

(Note 2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

IMCGF

	7	6	5	4	3	2	1	0
bit Symbol		EMCG142	EMCG141	EMCG140	EMST141	EMST140		INT10EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INT10 standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT10 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INT10 clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
bit Symbol		EMCG152	EMCG151	EMCG150	EMST151	EMST150		INT11EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INT11 standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT11 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INT11 clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
bit Symbol		EMCG162	EMCG161	EMCG160	EMST161	EMST160		INT12EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INT12 standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT12 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INT12 clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
bit Symbol		EMCG172	EMCG171	EMCG170	EMST171	EMST170		INT13EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	“0” is read.	Active state setting of INT13 standby clear request. (101~111: setting prohibited) 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT13 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		“0” is read.	INT13 clear input 0: Disable 1: Enable

(Note 1) **Refer to EMSTxx bit to know the active condition which is used for clearing standby.**

(Note 2) **Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.**

IMCG10

	7	6	5	4	3	2	1	0
bit Symbol		EMCG182	EMCG181	EMCG180	EMST181	EMST180		INT14EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT14 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT14 standby clear request 00: ─ 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT14 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG192	EMCG191	EMCG190	EMST191	EMST190		INT15EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT15 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT15 standby clear request 00: ─ 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT15 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG1A2	EMCG1A1	EMCG1A0	EMST1A1	EMST1A0		INT16EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT16 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT16 standby clear request 00: ─ 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT16 clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG1B2	EMCG1B1	EMCG1B0	EMST1B1	EMST1B0		INT17EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT17 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT17 standby clear request 00: ─ 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT17 clear input 0: Disable 1: Enable

(Note 1) Refer to EMSTxx bit to know the active condition which is used for clearing standby.

(Note 2) Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.

IMCG11

	7	6	5	4	3	2	1	0
bit Symbol		EMCG1 C2	EMCG1 C1	EMCG1 C0	EMST1 C1	EMST1 C0		INT18EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT18 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT18 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT18 clear input 0: Disable 1: Enable

	15	14	13	12	11	10	9	8
bit Symbol		EMCG1 D2	EMCG1 D1	EMCG1 D0	EMST1 D1	EMST1 D0		INT19EN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT19 standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT19 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT19 clear input 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol		EMCG1 E2	EMCG1 E1	EMCG1 E0	EMST1 E1	EMST1 E0		INT1AEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT1A standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT1A standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT1A clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol		EMCG1 F2	EMCG1 F1	EMCG1 F0	EMST1 F1	EMST1 F0		INT1BEN
Read/Write	R	R/W			R		R	R/W
After reset	0	0	1	0	0	0	0	0
Function	"0" is read.	Active state setting of INT1B standby clear request. (101~111: setting prohibited) 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active state of INT1B standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		"0" is read.	INT1B clear input 0: Disable 1: Enable

(Note 1) **Refer to EMSTxx bit to know the active condition which is used for clearing standby.**

(Note 2) **Please specify the bit for the edge first and then specify the bit for the <INTxEN>. Setting them simultaneously is prohibited.**

Be sure to set active state of the clear request if interrupt is enabled for clearing the STOP/ IDLE / SLEEP standby modes.

(Note 1) When using interrupts, be sure to follow the sequence of actions shown below:
1) If shared with other general ports, enable the target interrupt input.
2) Set conditions such as active state 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 modes with TMP19A44, 32 interrupt factors (INT0 through INTB, INT10 through INT1B, PHCNT0 through PHCNT5 and KWUP) are available. You can use CG for selecting edge/level of active state and judging whether the aforementioned factors are to be used for clearing the STOP/ SLEEP/ IDLE modes.

(Note 4) Among the above 12 factors to be assigned as the STOP/IDLE mode clear request interrupts, INT0 to INTF and INT10 to INT1F can be used as a normal interrupt without setting CG. Select level or edge with INTC. Setting CG is required when using an interrupt of both edges. The interrupt factors excluding other than the assigned as the STOP/ IDLE clear request factors need to be set to the INTC block.

6.6.3.2 INTCG Clear Register

ICRCG

	7	6	5	4	3	2	1	0
bit Symbol				ICRCG4	ICRCG3	ICRCG2	ICRCG1	ICRCG0
Read/Write	R			W				
After reset	0			0	0	0	0	0
Function	"0" is read.			Clear interrupt request. ("0" is read.) 0_0000: INT0 0_1000: INT8 1_0000: PHCNT2 1_1000: INT14 0_0001: INT1 0_1001: INT9 1_0001: PHCNT3 1_1001: INT15 0_0010: INT2 0_1010: INTA 1_0010: PHCNT4 1_1010: INT16 0_0011: INT3 0_1011: INTB 1_0011: PHCNT5 1_1011: INT17 0_0100: INT4 0_1100: KWUP 1_0100: INT10 1_1100: INT18 0_0101: INT5 0_1101: INTRTC 1_0101: INT11 1_1101: INT19 0_0110: INT6 0_1110: PHCNT0 1_0110: INT12 1_1110: INT1A 0_0111: INT7 0_1111: PHCNT1 1_0111: INT13 1_1111: INT1B				
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write								
After reset								
Function	"0" is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is read.							

(Note) The following describes how to clear an interrupt request of the above 32 factors assigned to STOP/SLEEP/IDLE clear request.

a) If the factor is used to clear an interrupt.

KWUP: use KWUPCLR.

INT0 through INTB, INT10 through INT1B, INTRTC and PHCNT0 through PHCNT5: use the ICRCG register in the CGblockshown above.

b) If the factor is used to clear an interrupt.

Clear the interrupt factor by INTCLR.

6.6.3.3 NMI Flag Register

NMIFLG

	7	6	5	4	3	2	1	0
bit Symbol								NMIFLG0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							NMI factor generation flag 0: not applicable 1: generated from WDT

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

(Note) <NMIFLG0> are cleared to “0” when they are read.

6.6.4 Interrupt Controller Register

6.6.4.1 Interrupt Vector Registers (IVR)

For an interrupt generated, the IVR register indicates the interrupt vector address of the corresponding interrupt factor. When an interrupt request is accepted, the corresponding value as listed in Table 6.5 is set to IVR [8:0]. By setting the base address of interrupt vectors to IVR [31:7], a read/write register, simply reading the IVR value can provide the corresponding interrupt vector address.

Interrupt Vector Register

IVR

	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.						Always reads “0”.	
	15	14	13	12	11	10	9	8
bit Symbol	IVR15	IVR14	IVR13	IVR12	IVR11	IVR10	IVR9	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	IVR23	IVR22	IVR21	IVR20	IVR19	IVR18	IVR17	IVR16
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	IVR31	IVR30	IVR29	IVR28	IVR27	IVR26	IVR25	IVR24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function								

6.6.4.2 Interrupt Level Register (ILEV)

ILEV is the register to control the interrupt level to be used by INTC in notifying interrupt requests to the TX19A/ H1 processor core.

Interrupts with interrupt levels not higher than ILEV <CMASK> are suspended. The interrupt priority level “7” is the highest priority and “1” the lowest. Note that any interrupt with interrupt level 0 is not suspended.

When a new interrupt is generated, the corresponding interrupt level is stored in <CMASK> and any previously stored values are incremented in mask levels such that the previous CMASK is saved in PMASK0 and PMASK0 is saved in PMASK1 and so on. For writing a new value to <CMASK>, set “1” to <MLEV> and write <CMASK> simultaneously. Writing a new value to <PMASKx> cannot be made.

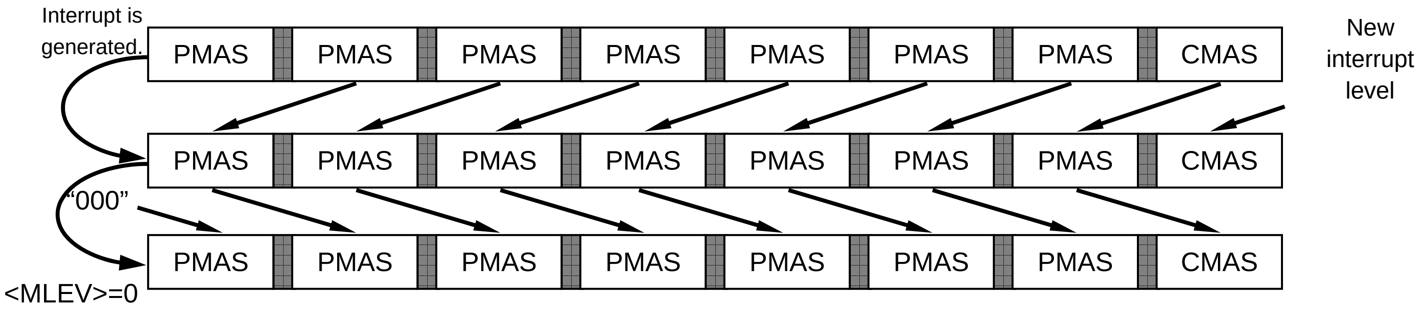
When <MLEV> is set to “0,” the interrupt mask levels in the register shift back to the previous state such that PMASK0 is moved to CMASK and PMASK1 is moved to PMASK0, and so on. The last <PMASK6> is set to “000.” If it is used in returning from an interrupt process, be sure to set <MLEV> to “0” before executing the ERET instruction. <MLEV> always reads “0.”

Interrupt Level Register

ILEV

	7	6	5	4	3	2	1	0
bit Symbol	—	PMASK0			—	CMASK		
Read/Write	R					R/W		
After reset	0	000			0	000		
Function		Interrupt mask level (previous) 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		Interrupt mask level (previous) 2				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		Interrupt mask level (previous) 4				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	0: Return mask level 1: Change CMASK	Interrupt mask level (previous) 6				Interrupt mask level (previous) 5		

(Note 1) This register must be 32-bit accessed.
(Note 2) Please set the mask level and <MLEV> individually.
(Note 3) Be sure to read the IVR value before changing the ILEV value. If the ILEV value is changed before reading IVR, an unexpected interrupt request may be generated.
(Note 4) Bit manipulation instructions cannot be used to access this register.



6.6.4.3 Interrupt mode control register (IMCxx)

IMCxx is comprised of <Ilxxx>, which determines the interrupt levels of individual interrupt factors, <DMxx>, which is used to set activation factors of DMA transfer, and <EIMXX>, which determines active state of interrupt requests.

IMC00

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.							

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

	23	22	21	20	19	18	17	16
bit Symbol		EIM021	EIM020	DM02		IL022	IL021	IL020
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 2 is set as the activation factor.	"0" is read.	If DM02 = 0, select the interrupt level for interrupt number 2 (INT0). 000: Disable Interrupt 001-111: 1-7 If DM02 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM031	EIM030	DM03		IL032	IL031	IL030
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 3 is set as the activation factor.	"0" is read.	If DM03 = 0, select the interrupt level for interrupt number 3 (INT1). 000: Disable Interrupt 001-111: 1-7 If DM03 = 1, select the DMAC channel. 000 to 111: 0 to 7		

IMC01

	7	6	5	4	3	2	1	0
bit Symbol		EIM041	EIM040	DM04		IL042	IL041	IL040
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 3 is set as the activation factor.	"0" is read.	If DM4 = 0, select the interrupt level for interrupt number 4 (INT2). 000: Disable Interrupt 001-111: 1-7 If DM4 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM051	EIM050	DM05		IL052	IL051	IL050
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 3 is set as the activation factor.	"0" is read.	If DM5 = 0, select the interrupt level for interrupt number 5 (INT3). 000: Disable Interrupt 001-111: 1-7 If DM5 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM061	EIM060	DM06		IL062	IL061	IL060
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 3 is set as the activation factor.	"0" is read.	If DM6 = 0, select the interrupt level for interrupt number 6 (INT4). 000: Disable Interrupt 001-111: 1-7 If DM6 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM071	EIM070	DM07		IL072	IL071	IL070
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 3 is set as the activation factor.	"0" is read.	If DM7 = 0, select the interrupt level for interrupt number 7 (INT5). 000: Disable Interrupt 001-111: 1-7 If DM7 = 1, select the DMAC channel. 000 to 111: 0 to 7		

IMC02

	7	6	5	4	3	2	1	0
bit Symbol		EIM081	EIM080	DM08		IL082	IL081	IL080
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 8 is set as the activation factor.	“0” is read.	If DM8 = 0, select the interrupt level for interrupt number 8 (INT6). 000: Disable Interrupt 001-111: 1-7 If DM8 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM091	EIM090	DM09		IL092	IL091	IL090
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 9 is set as the activation factor.	“0” is read.	If DM9 = 0, select the interrupt level for interrupt number 9 (INT7). 000: Disable Interrupt 001-111: 1-7 If DM9 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM0A1	EIM0A0	DM0A		IL0A2	IL0A1	IL0A0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 10 is set as the activation factor.	“0” is read.	If DMA = 0, select the interrupt level for interrupt number 10 (INT8). 000: Disable Interrupt 001-111: 1-7 If DMA = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM0B1	EIM0B0	DM0B		IL0B2	IL0B1	IL0B0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 11 is set as the activation factor.	“0” is read.	If DMB = 0, select the interrupt level for interrupt number 11 (INT9). 000: Disable Interrupt 001-111: 1-7 If DMB = 1, select the DMAC channel. 000 to 111: 0 to 7		

IMC03

	7	6	5	4	3	2	1	0
bit Symbol		EIM0C1	EIM0C0	DM0C		IL0C2	IL0C1	IL0C0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 12 is set as the activation factor.	"0" is read.	If DMC = 0, select the interrupt level for interrupt number 12 (INTA). 000: Disable Interrupt 001-111: 1-7 If DMC = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM0D1	EIM0D0	DM0D		IL0D2	IL0D1	IL0D0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 13 is set as the activation factor.	"0" is read.	If DMD = 0, select the interrupt level for interrupt number 13 (INTB). 000: Disable Interrupt 001-111: 1-7 If DMD = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM0E1	EIM0E0	DM0E		IL0E2	IL0E1	IL0E0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 14 is set as the activation factor.	"0" is read.	If DME = 0, select the interrupt level for interrupt number 14 (INTC). 000: Disable Interrupt 001-111: 1-7 If DME = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM0F1	EIM0F0	DM0F		IL0F2	IL0F1	IL0F0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 15 is set as the activation factor.	"0" is read.	If DMF = 0, select the interrupt level for interrupt number 15 (INTD). 000: Disable Interrupt 001-111: 1-7 If DMF = 1, select the DMAC channel. 000 to 111: 0 to 7		

IMC04

	7	6	5	4	3	2	1	0
bit Symbol		EIM101	EIM100	DM10		IL102	IL101	IL100
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 16 is set as the activation factor.	"0" is read.	If DM10 = 0, select the interrupt level for interrupt number 16 (INTE). 000: Disable Interrupt 001-111: 1-7 If DM10 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM111	EIM110	DM11		IL112	IL111	IL110
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 17 is set as the activation factor.	"0" is read.	If DM11 = 0, select the interrupt level for interrupt number 17 (INTF). 000: Disable Interrupt 001-111: 1-7 If DM11 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM121	EIM120	DM12		IL122	IL121	IL120
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 18 is set as the activation factor.	"0" is read.	If DM12 = 0, select the interrupt level for interrupt number 18 (KWUP). 000: Disable Interrupt 001-111: 1-7 If DM12 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM131	EIM130	DM13		IL132	IL131	IL130
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 19 is set as the activation factor.	"0" is read.	If DM13= 0, select the interrupt level for interrupt number 19 (INT10). 000: Disable Interrupt 001-111: 1-7 If DM13 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC05

	7	6	5	4	3	2	1	0
bit Symbol		EIM141	EIM140	DM14		IL142	IL141	IL140
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 20 is set as the activation factor.	“0” is read.	If DM14 = 0, select the interrupt level for interrupt number 20 (INT11). 000: Disable Interrupt 001-111: 1-7 If DM14 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM151	EIM150	DM15		IL152	IL151	IL150
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 21 is set as the activation factor.	“0” is read.	If DM15 = 0, select the interrupt level for interrupt number 21 (INT12). 000: Disable Interrupt 001-111: 1-7 If DM15 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM161	EIM160	DM16		IL162	IL161	IL160
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 22 is set as the activation factor.	“0” is read.	If DM16 = 0, select the interrupt level for interrupt number 22 (INT13). 000: Disable Interrupt 001-111: 1-7 If DM16 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM171	EIM170	DM17		IL172	IL171	IL170
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 23 is set as the activation factor.	“0” is read.	If DM17 = 0, select the interrupt level for interrupt number 23 (INT14). 000: Disable Interrupt 001-111: 1-7 If DM17 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC06

	7	6	5	4	3	2	1	0
bit Symbol		EIM181	EIM180	DM18		IL182	IL181	IL180
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 24 is set as the activation factor.	“0” is read.	If DM18 = 0, select the interrupt level for interrupt number 24 (INT15). 000: Disable Interrupt 001-111: 1-7 If DM18 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM191	EIM190	DM19		IL192	IL191	IL190
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 25 is set as the activation factor.	“0” is read.	If DM19 = 0, select the interrupt level for interrupt number 25 (INT16). 000: Disable Interrupt 001-111: 1-7 If DM19 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM1A1	EIM1A0	DM1A		IL1A2	IL1A1	IL1A0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 26 is set as the activation factor.	“0” is read.	If DM1A = 0, select the interrupt level for interrupt number 26 (INT17). 000: Disable Interrupt 001-111: 1-7 If DM1A = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM1B1	EIM1B0	DM1B		IL1B2	IL1B1	IL1B0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **CG setting “01”.**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 27 is set as the activation factor.	“0” is read.	If DM1B = 0, select the interrupt level for interrupt number 27 (INT18). 000: Disable Interrupt 001-111: 1-7 If DM1B = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC07

	7	6	5	4	3	2	1	0
bit Symbol		EIM1C1	EIM1C0	DM1C		IL1C2	IL1C1	IL1C0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 28 is set as the activation factor.	"0" is read.	If DM1C = 0, select the interrupt level for interrupt number 28 (INT19). 000: Disable Interrupt 001-111: 1-7 If DM1C = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM1D1	EIM1D0	DM1D		IL1D2	IL1D1	IL1D0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 29 is set as the activation factor.	"0" is read.	If DM1D = 0, select the interrupt level for interrupt number 29 (INT1A). 000: Disable Interrupt 001-111: 1-7 If DM1D = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM1E1	EIM1E0	DM1E		IL1E2	IL1E1	IL1E0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **CG setting "01".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 30 is set as the activation factor.	"0" is read.	If DM1E = 0, select the interrupt level for interrupt number 30 (INT1B). 000: Disable Interrupt 001-111: 1-7 If DM1E = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM1F1	EIM1F0	DM1F		IL1F2	IL1F1	IL1F0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 31 is set as the activation factor.	"0" is read.	If DM1F = 0, select the interrupt level for interrupt number 31 (INT1C). 000: Disable Interrupt 001-111: 1-7 If DM1F = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC08

	7	6	5	4	3	2	1	0
bit Symbol		EIM201	EIM200	DM20		IL202	IL201	IL200
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 32 is set as the activation factor.	“0” is read.	If DM20= 0, select the interrupt level for interrupt number 32 (INT1D). 000: Disable Interrupt 001-111: 1-7 If DM20 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM211	EIM210	DM21		IL212	IL211	IL210
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 33 is set as the activation factor.	“0” is read.	If DM21 = 0, select the interrupt level for interrupt number 33 (INT1E). 000: Disable Interrupt 001-111: 1-7 If DM21 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM221	EIM220	DM26		IL222	IL221	IL220
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 34 is set as the activation factor.	“0” is read.	If DM22 = 0, select the interrupt level for interrupt number 34 (INT1F). 000: Disable Interrupt 001-111: 1-7 If DM22 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM231	EIM230	DM23		IL232	IL231	IL230
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM23 = 0, select the interrupt level for interrupt number 35 (INTRX0). 000: Disable Interrupt 001-111: 1-7 If DM23 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC09

	7	6	5	4	3	2	1	0
bit Symbol		EIM241	EIM240	DM24		IL242	IL241	IL240
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM24 = 0, select the interrupt level for interrupt number 36 (INTTX0). 000: Disable Interrupt 001-111: 1-7 If DM24 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM251	EIM250	DM25		IL252	IL251	IL250
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM25 = 0, select the interrupt level for interrupt number 37 (INTRX1). 000: Disable Interrupt 001-111: 1-7 If DM25 = 1, select the DMAC channel.		

	23	22	21	20	19	18	17	16
bit Symbol		EIM261	EIM260	DM26		IL262	IL261	IL260
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM26 = 0, select the interrupt level for interrupt number 38 (INTTX1). 000: Disable Interrupt 001-111: 1-7 If DM26 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM271	EIM270	DM27		IL272	IL271	IL270
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM27 = 0, select the interrupt level for interrupt number 39 (INTRX2). 000: Disable Interrupt 001-111: 1-7 If DM27 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC0A

	7	6	5	4	3	2	1	0
bit Symbol		EIM281	EIM280	DM28		IL282	IL281	IL280
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM28 = 0, select the interrupt level for interrupt number 40 (INTTX2). 000: Disable Interrupt 001-111: 1-7 If DM28 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM291	EIM290	DM29		IL292	IL291	IL290
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM29 = 0, select the interrupt level for interrupt number 41 (HINTRX0). 000: Disable Interrupt 001-111: 1-7 If DM29 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM2A1	EIM2A0	DM2A		IL2A2	IL2A1	IL2A0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM2A = 0, select the interrupt level for interrupt number 42 (HINTTX0). 000: Disable Interrupt 001-111: 1-7 If DM2A = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM2B1	EIM2B0	DM2B		IL2B2	IL2B1	IL2B0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM2B = 0, select the interrupt level for interrupt number 43 (HINTRX1). 000: Disable Interrupt 001-111: 1-7 If DM2B = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC0B

bit Symbol		EIM2C1	EIM2C0	DM2C		IL2C2	IL2C1	IL2C0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM2C = 0, select the interrupt level for interrupt number 44 (HINTTX1). 000: Disable Interrupt 001-111: 1-7 If DM2C = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM2D1	EIM2D0	DM2D		IL2D2	IL2D1	IL2D0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM2D = 0, select the interrupt level for interrupt number 45 (HINTRX2). 000: Disable Interrupt 001-111: 1-7 If DM2D = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM2E1	EIM2E0	DM2E		IL2E2	IL2E1	IL2E0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM2E = 0, select the interrupt level for interrupt number 46 (HINTTX2). 000: Disable Interrupt 001-111: 1-7 If DM2E = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM2F1	EIM2F0	DM2F		IL2F2	IL2F1	IL2F0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM2F = 0, select the interrupt level for interrupt number 47 (INTS0). 000: Disable Interrupt 001-111: 1-7 If DM2F = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC0C

	7	6	5	4	3	2	1	0
bit Symbol		EIM301	EIM300	DM30		IL302	IL301	IL300
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 48 is set as the activation factor.	“0” is read.	If DM30 = 0, select the interrupt level for interrupt number 48 (INTADHPA). 000: Disable Interrupt 001-111: 1-7 If DM30 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM311	EIM310	DM31		IL312	IL311	IL310
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 49 is set as the activation factor.	“0” is read.	If DM31 = 0, select the interrupt level for interrupt number 49 (INTADMA). 000: Disable Interrupt 001-111: 1-7 If DM31 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM321	EIM320	DM32		IL322	IL321	IL320
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 50 is set as the activation factor.	“0” is read.	If DM32 = 0, select the interrupt level for interrupt number 50 (INTADHPB). 000: Disable Interrupt 001-111: 1-7 If DM32 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM331	EIM330	DM33		IL332	IL331	IL330
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 is set as the activation factor.	“0” is read.	If DM33 = 0, select the interrupt level for interrupt number 51 (INTADMB). 000: Disable Interrupt 001-111: 1-7 If DM33 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC0D

	7	6	5	4	3	2	1	0
bit Symbol		EIM341	EIM340	DM34		IL342	IL341	IL340
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	"0" is read.	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 is set as the activation factor.	"0" is read.	If DM34 = 0, select the interrupt level for interrupt number 52 (INTADHPC). 000: Disable Interrupt 001-111: 1-7 If DM34 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM351	EIM350	DM35		IL352	IL351	IL350
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 is set as the activation factor.	"0" is read.	If DM35 = 0, select the interrupt level for interrupt number 53 (INTADMC). 000: Disable Interrupt 001-111: 1-7 If DM35 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM361	EIM360	DM36		IL362	IL361	IL360
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 54 is set as the activation factor.	"0" is read.	If DM36 = 0, select the interrupt level for interrupt number 54 (INTTB0). 000: Disable Interrupt 001-111: 1-7 If DM36 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM371	EIM370	DM37		IL372	IL371	IL370
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 is set as the activation factor.	"0" is read.	If DM37 = 0, select the interrupt level for interrupt number 55 (INTTB1). 000: Disable Interrupt 001-111: 1-7 If DM37 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC0E

	7	6	5	4	3	2	1	0
bit Symbol		EIM381	EIM380	DM38		IL382	IL381	IL380
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	“0” is read.	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 56 is set as the activation factor.	“0” is read.	If DM38 = 0, select the interrupt level for interrupt number 56 (INTTB2). 000: Disable Interrupt 001-111: 1-7 If DM38 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM391	EIM390	DM39		IL392	IL391	IL390
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 57 is set as the activation factor.	“0” is read.	If DM39 = 0, select the interrupt level for interrupt number 57 (INTTB3). 000: Disable Interrupt 001-111: 1-7 If DM39 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM3A1	EIM3A0	DM3A		IL3A2	IL3A1	IL3A0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 58 is set as the activation factor.	“0” is read.	If DM3A = 0, select the interrupt level for interrupt number 58 (INTTB4). 000: Disable Interrupt 001-111: 1-7 If DM3A = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM3B1	EIM3B0	DM3B		IL3B2	IL3B1	IL3B0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 59 is set as the activation factor.	“0” is read.	If DM3B = 0, select the interrupt level for interrupt number 59 (INTTB5). 000: Disable Interrupt 001-111: 1-7 If DM3B = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC0F

	7	6	5	4	3	2	1	0
bit Symbol		EIM3C1	EIM3C0	DM3C		IL3C2	IL3C1	IL3C0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	“0” is read.	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 60 is set as the activation factor.	“0” is read.	If DM3C = 0, select the interrupt level for interrupt number 60 (INTTB6). 000: Disable Interrupt 001-111: 1-7 If DM3C = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM3D1	EIM3D0	DM3D		IL3D2	IL3D1	IL3D0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 61 is set as the activation factor.	“0” is read.	If DM3D = 0, select the interrupt level for interrupt number 61 (INTTB7). 000: Disable Interrupt 001-111: 1-7 If DM3D = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM3E1	EIM3E0	DM3E		IL3E2	IL3E1	IL3E0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 62 is set as the activation factor.	“0” is read.	If DM3E = 0, select the interrupt level for interrupt number 62 (INTTB8). 000: Disable Interrupt 001-111: 1-7 If DM3E = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM3F1	EIM3F0	DM3F		IL3F2	IL3F1	IL3F0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 63 is set as the activation factor.	“0” is read.	If DM3F = 0, select the interrupt level for interrupt number 63 (INTTB9). 000: Disable Interrupt 001-111: 1-7 If DM3F = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC10

	7	6	5	4	3	2	1	0
bit Symbol		EIM401	EIM400	DM40		IL402	IL401	IL400
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 64 is set as the activation factor.	“0” is read.	If DM40 = 0, select the interrupt level for interrupt number 64 (INTTBA). 000: Disable Interrupt 001-111: 1-7 If DM40 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM411	EIM410	DM41		IL412	IL411	IL410
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 65 is set as the activation factor.	“0” is read.	If DM41 = 0, select the interrupt level for interrupt number 65 (INTTBB). 000: Disable Interrupt 001-111: 1-7 If DM41 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM421	EIM420	DM42		IL422	IL421	IL420
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 66 is set as the activation factor.	“0” is read.	If DM42 = 0, select the interrupt level for interrupt number 66 (INTTBC). 000: Disable Interrupt 001-111: 1-7 If DM42 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM431	EIM430	DM43		IL432	IL431	IL430
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 67 is set as the activation factor.	“0” is read.	If DM43 = 0, select the interrupt level for interrupt number 67 (INTTBD). 000: Disable Interrupt 001-111: 1-7 If DM43 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC11

	7	6	5	4	3	2	1	0
bit Symbol		EIM441	EIM440	DM44		IL442	IL441	IL440
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	“0” is read.	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 68 is set as the activation factor.	“0” is read.	If DM44 = 0, select the interrupt level for interrupt number 68 (INTTBE). 000: Disable Interrupt 001-111: 1-7 If DM44 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM451	EIM450	DM45		IL452	IL451	IL450
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 69 is set as the activation factor.	“0” is read.	If DM45 = 0, select the interrupt level for interrupt number 69 (INTTBF). 000: Disable Interrupt 001-111: 1-7 If DM45 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM461	EIM460	DM46		IL462	IL461	IL460
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 70 is set as the activation factor.	“0” is read.	If DM46 = 0, select the interrupt level for interrupt number 70 (INTADA). 000: Disable Interrupt 001-111: 1-7 If DM46 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM471	EIM470	DM47		IL472	IL471	IL470
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 71 is set as the activation factor.	“0” is read.	If DM47 = 0, select the interrupt level for interrupt number 71 (INTADB). 000: Disable Interrupt 001-111: 1-7 If DM47 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC12

	7	6	5	4	3	2	1	0
bit Symbol		EIM481	EIM480	DM48		IL482	IL481	IL480
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	"0" is read.	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 72 is set as the activation factor.	"0" is read.	If DM48 = 0, select the interrupt level for interrupt number 72 (INTADC). 000: Disable Interrupt 001-111: 1-7 If DM48 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM491	EIM490	DM49		IL492	IL491	IL490
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 73 is set as the activation factor.	"0" is read.	If DM49 = 0, select the interrupt level for interrupt number 73 (INTTB10). 000: Disable Interrupt 001-111: 1-7 If DM49 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM4A1	EIM4A0	DM4A		IL4A2	IL4A1	IL4A0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 74 is set as the activation factor.	"0" is read.	If DM4A = 0, select the interrupt level for interrupt number 74 (INTTB11). 000: Disable Interrupt 001-111: 1-7 If DM4A = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM4B1	EIM4B0	DM4B		IL4B2	IL4B1	IL4B0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 75 is set as the activation factor.	"0" is read.	If DM4B = 0, select the interrupt level for interrupt number 75 (PHCNT0). 000: Disable Interrupt 001-111: 1-7 If DM4B = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC13

	7	6	5	4	3	2	1	0
bit Symbol		EIM4C1	EIM4C0	DM4C		IL4C2	IL4C1	IL4C0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	"0" is read.	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 76 is set as the activation factor.	"0" is read.	If DM4C = 0, select the interrupt level for interrupt number 76 (PHCNT1). 000: Disable Interrupt 001-111: 1-7 If DM4C = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM4D1	EIM4D0	DM4D		IL4D2	IL4D1	IL4D0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 77 is set as the activation factor.	"0" is read.	If DM4D = 0, select the interrupt level for interrupt number 77 (PHCNT2). 000: Disable Interrupt 001-111: 1-7 If DM4D = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM4E1	EIM4E0	DM4E		IL4E2	IL4E1	IL4E0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 78 is set as the activation factor.	"0" is read.	If DM4E = 0, select the interrupt level for interrupt number 78 (PHCNT3). 000: Disable Interrupt 001-111: 1-7 If DM4E = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM4F1	EIM4F0	DM4F		IL4F2	IL4F1	IL4F0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 79 is set as the activation factor.	"0" is read.	If DM4F = 0, select the interrupt level for interrupt number 79 (PHCNT4). 000: Disable Interrupt 001-111: 1-7 If DM4F = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC14

	7	6	5	4	3	2	1	0
bit Symbol		EIM501	EIM500	DM50		IL502	IL501	IL500
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 80 is set as the activation factor.	"0" is read.	If DM50 = 0, select the interrupt level for interrupt number 80 (PHCNT5). 000: Disable Interrupt 001-111: 1-7 If DM51 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	15	14	13	12	11	10	9	8
bit Symbol		EIM511	EIM510	DM51		IL512	IL511	IL510
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 81 is set as the activation factor.	"0" is read.	If DM51 = 0, select the interrupt level for interrupt number 81 (INTCAP0). 000: Disable Interrupt 001-111: 1-7 If DM51 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	23	22	21	20	19	18	17	16
bit Symbol		EIM521	EIM520	DM52		IL522	IL521	IL520
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 82 is set as the activation factor.	"0" is read.	If DM52 = 0, select the interrupt level for interrupt number 82 (INTCAP1). 000: Disable Interrupt 001-111: 1-7 If DM52 = 1, select the DMAC channel. 000 to 111: 0 to 7		
	31	30	29	28	27	26	25	24
bit Symbol		EIM531	EIM530	DM53		IL532	IL531	IL530
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 83 is set as the activation factor.	"0" is read.	If DM53 = 0, select the interrupt level for interrupt number 83 (INTCAP2). 000: Disable Interrupt 001-111: 1-7 If DM53 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC15

	7	6	5	4	3	2	1	0
bit Symbol		EIM541	EIM540	DM54		IL542	IL541	IL540
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	“0” is read.	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 84 is set as the activation factor.	“0” is read.	If DM54 = 0, select the interrupt level for interrupt number 84 (INTCAP3). 000: Disable Interrupt 001-111: 1-7 If DM54 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM551	EIM550	DM55		IL552	IL551	IL550
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 85 is set as the activation factor.	“0” is read.	If DM55 = 0, select the interrupt level for interrupt number 85 (INTCMP0). 000: Disable Interrupt 001-111: 1-7 If DM55 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM561	EIM560	DM56		IL562	IL561	IL560
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 88 is set as the activation factor.	“0” is read.	If DM56 = 0, select the interrupt level for interrupt number 86 (INTCMP1). 000: Disable Interrupt 001-111: 1-7 If DM56 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM571	EIM570	DM57		IL572	IL571	IL570
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 87 is set as the activation factor.	“0” is read.	If DM57 = 0, select the interrupt level for interrupt number 87 (INTCMP2). 000: Disable Interrupt 001-111: 1-7 If DM57 = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC16

	7	6	5	4	3	2	1	0
bit Symbol		EIM581	EIM580	DM58		IL582	IL581	IL580
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	"0" is read.	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 88 is set as the activation factor.	"0" is read.	If DM58 = 0, select the interrupt level for interrupt number 88 (INTCMP3). 000: Disable Interrupt 001-111: 1-7 If DM58 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM591	EIM590	DM59		IL592	IL591	IL590
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 89 is set as the activation factor.	"0" is read.	If DM59 = 0, select the interrupt level for interrupt number 89 (INTCMP4). 000: Disable Interrupt 001-111: 1-7 If DM59 = 1, select the DMAC channel. 000 to 111: 0 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM5A1	EIM5A0	DM5A		IL5A2	IL5A1	IL5A0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 90 is set as the activation factor.	"0" is read.	If DM5A = 0, select the interrupt level for interrupt number 90 (INTCMP5). 000: Disable Interrupt 001-111: 1-7 If DM5A = 1, select the DMAC channel. 000 to 111: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM5B1	EIM5B0	DM5B		IL5B2	IL5B1	IL5B0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	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 91 is set as the activation factor.	"0" is read.	If DM5B = 0, select the interrupt level for interrupt number 91 (INTCMP6). 000: Disable Interrupt 001-111: 1-7 If DM5B = 1, select the DMAC channel. 000 to 111: 0 to 7		

(Note) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

IMC17

	7	6	5	4	3	2	1	0
bit Symbol		EIM5C1	EIM5C0	DM5C		IL5C2	IL5C1	IL5C0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	“0” is read.	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 92 is set as the activation factor.	“0” is read.	If DM5C = 0, select the interrupt level for interrupt number 92 (INTCMP7). 000: Disable Interrupt 001-111: 1-7 If DM5C = 1, select the DMAC channel. 000 to 111: 0 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM5D1	EIM5D0	DM5D		IL5D2	IL5D1	IL5D0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 93 is set as the activation factor.	“0” is read.	If DM5D = 0, select the interrupt level for interrupt number 93 (INTTBT). 000: Disable Interrupt 001-111: 1-7 If DM5D = 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		EIM5E1	EIM5E0	DM5E		IL5E2	IL5E1	IL5E0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	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 94 is set as the activation factor.	“0” is read.	If DM5E = 0, select the interrupt level for interrupt number 94 (INTRTC). 000: Disable Interrupt 001-111: 1-7 If DM5E = 1, select the DMAC channel. 000 to 011: 0 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM5F1	EIM5F0			IL5F2	IL5F1	IL5F0
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level **Be sure to set “00”.**		Be sure to set “0”.	“0” is read.	If DM5F = 0, select the interrupt level for interrupt number 95 (INTDMA0). 000: Disable Interrupt 001-111: 1-7		

(Note 1) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

(Note 2) The access to the DMAC register by DMAC is prohibited.

IMC18

	7	6	5	4	3	2	1	0
bit Symbol		EIM601	EIM600			IL602	IL601	IL600
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level **Be sure to set “00”.**		Be sure to set “0”.	“0” is read.	If DM60 = 0, select the interrupt level for interrupt number 96 (INTDMA1). 000: Disable Interrupt 001-111: 1-7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM611	EIM610			IL612	IL611	IL610
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level **Be sure to set “00”.**		Be sure to set “0”.	“0” is read.	If DM61 = 0, select the interrupt level for interrupt number 97 (INTDMA2). 000: Disable Interrupt 001-111: 1-7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM621	EIM620			IL622	IL621	IL620
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level **Be sure to set “00”.**		Be sure to set “0”.	“0” is read.	If DM62 = 0, select the interrupt level for interrupt number 98 (INTDMA3). 000: Disable Interrupt 001-111: 1-7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM631	EIM630			IL632	IL631	IL630
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.	Selects active state of interrupt request. 00: “L” level **Be sure to set “00”.**		Be sure to set “0”.	“0” is read.	If DM63 = 0, select the interrupt level for interrupt number 99 (INTDMA4). 000: Disable Interrupt 001-111: 1-7		

(Note 1) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

(Note 2) The access to the DMAC register by DMAC is prohibited.

IMC19

	7	6	5	4	3	2	1	0
bit Symbol		EIM641	EIM640			IL642	IL641	IL640
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0		
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level **Be sure to set "00".**		Be sure to set "0".	"0" is read.	If DM64 = 0, select the interrupt level for interrupt number 100 (INTDMA5). 000: Disable Interrupt 001-111: 1-7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM651	EIM650			IL652	IL651	IL650
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level **Be sure to set "00".**		Be sure to set "0".	"0" is read.	If DM65 = 0, select the interrupt level for interrupt number 101 (INTDMA6). 000: Disable Interrupt 001-111: 1-7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM661	EIM660			IL662	IL661	IL660
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level **Be sure to set "00".**		Be sure to set "0".	"0" is read.	If DM66 = 0, select the interrupt level for interrupt number 102 (INTDMA7). 000: Disable Interrupt 001-111: 1-7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM671	EIM670	DM67		IL672	IL671	IL670
Read/Write	R	R/W		R/W	R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	Selects active state of interrupt request. 00: "L" level **Be sure to set "00".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 103 is set as the activation factor.	"0" is read.	If DM67 = 0, select the interrupt level for interrupt number 103 (software set). 000: Disable Interrupt 001-111: 1-7 If DM67 = 1, select the DMAC channel. 000 to 011: 0 to 7		

(Note 1) Default values of <EIMxx0:1> are different from the values to be used. Properly set them to the specified values before use.

(Note 2) The access to the DMAC register by DMAC is prohibited.

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

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

(Note 3) An active condition must be changed after an interrupt output of the corresponding device becomes negate especially when you change it to the level detection.

(1) IL	**Set to “0” if it is set to “Excluding 0”**
(2) Change	**Detection condition (EIM)**
(3) INTCLR	**Clear pertinent interrupt.**
(4) IL	**Set to” Excluding 0”.**

6.6.4.4 Interrupt Request Clear Registers (INTCLR)

Setting the IVR [8:0] for the corresponding interrupt factor into the INTCLR register enables to clear any interrupt request being suspended. Do not clear an interrupt request before reading the IVR value. When an interrupt request is cleared, the IVR value is also cleared and the interrupt factor cannot be determined anymore.

IVR <IVR8:0> value setting to clear the interrupt request

INTCLR

	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	“0” is read.							Set the IVR <IVR8:0> value that corresponds to the interrupt request that you would like to clear.
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							

(Note 1) This register must be 16-bit accessed.

(Note 2) In order to maintain interrupt factors regardless of the active state setting of INTC IMCx <EIMxx>, clear the interrupt request in any case whether in “H” level, “L” level, rising edge, or falling edge.

(Note 3) Bit manipulation instructions cannot be used to access this register.

(Note 4) External transfer requests due to DMAC interrupt factors are not cleared. Once an external transfer request is accepted, it will not be canceled until the DMA transfer is executed. Therefore, any unnecessary external transfer request should be cleared by executing DMA transfer, by disabling interrupts using IMCx <ILxxx> or by canceling the corresponding DMAC activation factors using IMCx<DMxx> before accepting such external transfer requests.

6.6.4.5 DMAC Transfer Request Clear Registers (DREQFLG)

Setting the DREQ [7:0] for the corresponding factor into the DREQFLG register enables to clear any DMAC transfer request.

DREQ [7:0] value setting to clear the DMAC transfer request

DREQFLG
(0xFF00_10C4)

	7	6	5	4	3	2	1	0
bit Symbol	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Corresponding DMAC transfer request is cleared.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							

Reading 0: DMAC transfer is requested.

1: DMAC transfer is not requested.

Writing 0: Invalid

1 :DMAC transfer request is cleared.

7. Input/Output Ports

Port Registers

Px : Port register
To read/ write port data.

PxCR : Control register
To control input/output
* Need to enable the input with PxIE register even when input is set.

PxFRn : Function register
To set functions. An assigned function can be activated by setting "1".

PxOD : Open drain control register
To switch the input of a register that can be set as programmable open drain.

PxPUP : Pull up control register
To control program pull ups.

PxIE : Input control enable register

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.

If the BUSMD pin (port P45) 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).

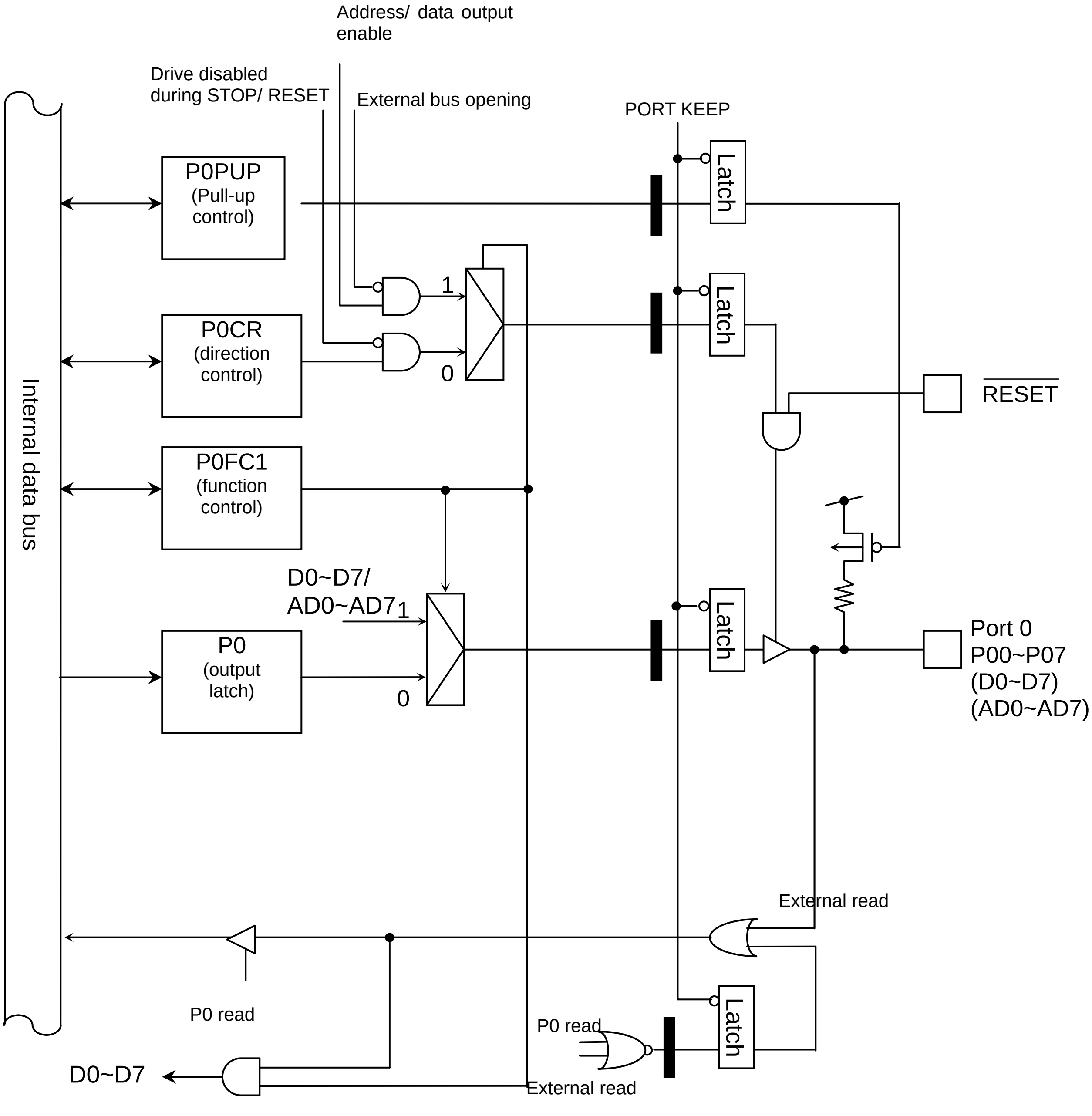


Fig. 7.1 Port 0 (P00 through P07)

Port 0 register

P0
(0xFF00_4000)

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

Port 0 control register

P0CR
(0xFF00_4004)

	7	6	5	4	3	2	1	0
Bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
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.")							

Port 0 function register 1

P0FC1
(0xFF00_4008
)

	7	6	5	4	3	2	1	0
Bit Symbol	P07F	P06F	P05F	P04F	P03F	P02F	P01F	P00F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1:External bus							

Port 0 Pull-up control register

P0PE
(0xFF00_402C)

	7	6	5	4	3	2	1	0
Bit Symbol	PE07	PE06	PE05	PE04	PE03	PE02	PE01	PE00
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

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 registers P1FC1 and P1FC2. 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, registers P1CR and P1FC must be provisioned to allow the port 1 to function as either an address bus or an address data bus.

If the BUSMD pin (port 45) 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).

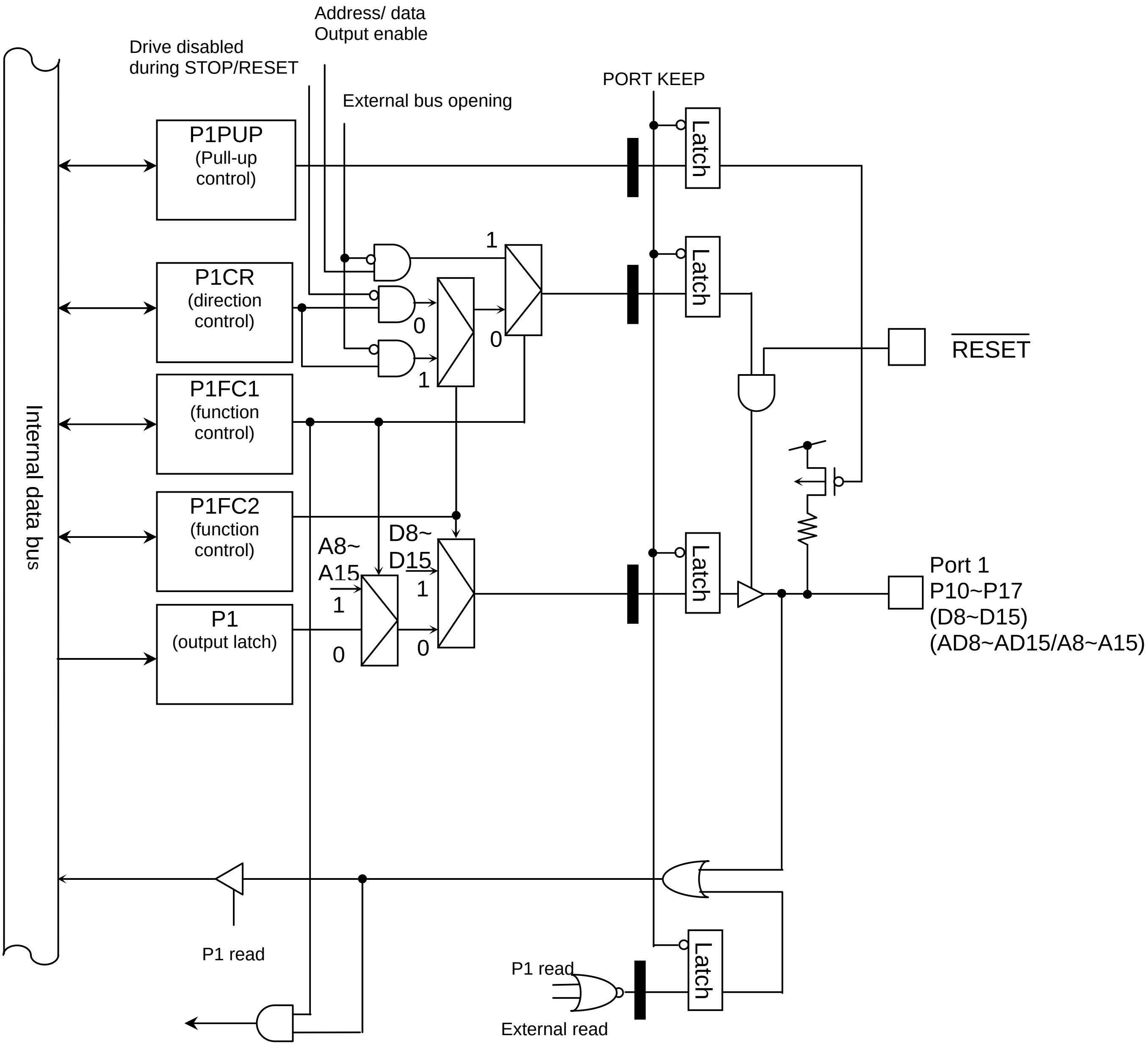


Fig. 7.2 Port 1 (P10 through P17)

Port 1 register

		7	6	5	4	3	2	1	0
P1	Bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
(0xFF00_4040)	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
(0xFF00_4004)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Output disabled 1: Output enabled							

Port 1 function register 1

		7	6	5	4	3	2	1	0
P1FC	Bit Symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
(0xFF00_4048)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: External bus							

Port 1 function register 2

		7	6	5	4	3	2	1	0
P1FC2	Bit Symbol	P17F2	P16F2	P15F2	P14F2	P13F2	P12F2	P11F2	P10F2
(0xFF00_404C)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: External bus							

Port 1 Pull-up control register

		7	6	5	4	3	2	1	0
P1PUP	Bit Symbol	PE17	PE16	PE15	PE14	PE13	PE12	PE11	PE10
(0xFF00_406C)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

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 P2FC1, P2FC2 and P2FC3. A reset allows all bits of the output latch P2 to be set to "1," all bits of P2CR, P2FC1, P2FC2 and P2FC3 to be cleared to "0," and the port 2 to be put in output disabled mode.

The port 2 also performs a 16-bit timer input function. This function is enabled by setting the corresponding bit of P2FC3 to "1" and the corresponding bits of P2CR, P2FC1 and P2FC2 to "0." A reset allows P2CR, P2FC1, P2FC2 and P2FC3 to be cleared to "0" and the port 2 to function as an input port.

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 (port P45) 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).

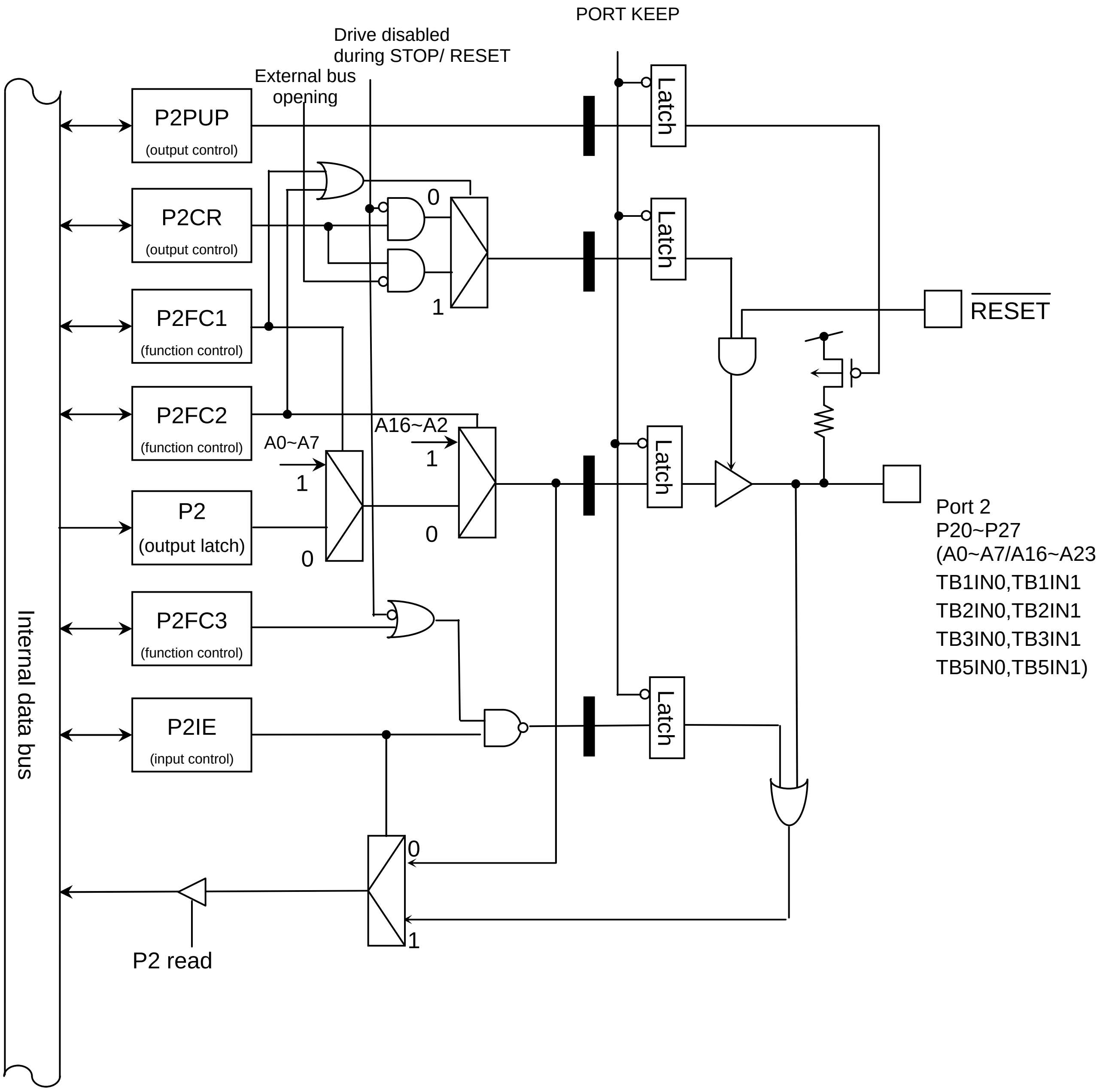


Fig. 7.3 Port 2 (P20 through P27)

Port 2 register

P2
(0xFF00_4080)

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

Port 2 control register

P2CR
(0xFF00_4084)

	7	6	5	4	3	2	1	0
Bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Output disabled 1: Output enabled							

Port 2 function register 1

P2FC1
(0xFF00_4088)

	7	6	5	4	3	2	1	0
Bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Port 1: External bus	0: Port 1: External bus	0: Port 1: External bus	0: Port 1: External bus	0: Port 1: External bus	0: Port 1: External bus	0: Port 1: External bus	0: Port 1: External bus

Port 2 function register 2

P2FC2
(0xFF00_408C)

	7	6	5	4	3	2	1	0
Bit Symbol	P27F2	P26F2	P25F2	P24F2	P23F2	P22F2	P21F2	P20F2
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Port 1: External bus							

Port 2 function register 3

P2FC3
(0xFF00_4090)

	7	6	5	4	3	2	1	0
Bit Symbol	P27F3	P26F3	P25F3	P24F3	P23F3	P22F3	P21F3	P20F3
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:TB5IN1	0:PORT 1:TB5IN0	0:PORT 1:TB3IN1	0:PORT 1:TB3IN0	0:PORT 1:TB2IN1	0:PORT 1:TB2IN0	0:PORT 1:TB1IN1	0:PORT 1:TB1IN0

Port 2 pull-up control register

P2PUP
(0xFF00_40AC)

	7	6	5	4	3	2	1	0
Bit Symbol	PE27	PE26	PE25	PE24	PE23	PE22	PE21	PE20
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 2 input enable control register

P2IE
(0xFF00_40B8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE27	PIE26	PIE25	PIE24	PIE23	PIE22	PIE21	PIE20
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.4 Port 3 (P30 through P37)

The port 3 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 P3CR and the function registers P3FC1 and P3FC2.

Besides the input/output port function, the port 3 performs other functions: P34 outputs a 16-bit timer, and P32, P35 through P37 perform the 32-bit capture and trigger input function. These functions are enabled by setting the corresponding bit of P3FC2 to "1." A reset allows P3CR, P3FC1 and P3FC2 to be cleared to "0" and the port 3 to function as an input port. Input is disabled right after reset. To enable input, set the corresponding bit of P3IE to “1”.

In addition to above functions, a function of inputting and outputting the control and status signals of CPU is provided. If the P30 pin is set to $\overline{\text{RD}}$ signal output mode (<P30F>="1"), the $\overline{\text{RD}}$ strobe is output only when an external address area is accessed. Likewise, if the P31 pin is set to $\overline{\text{WR}}$ signal output mode (<P31F>="1"), the $\overline{\text{WR}}$ strobe is output only when an external address area is accessed.

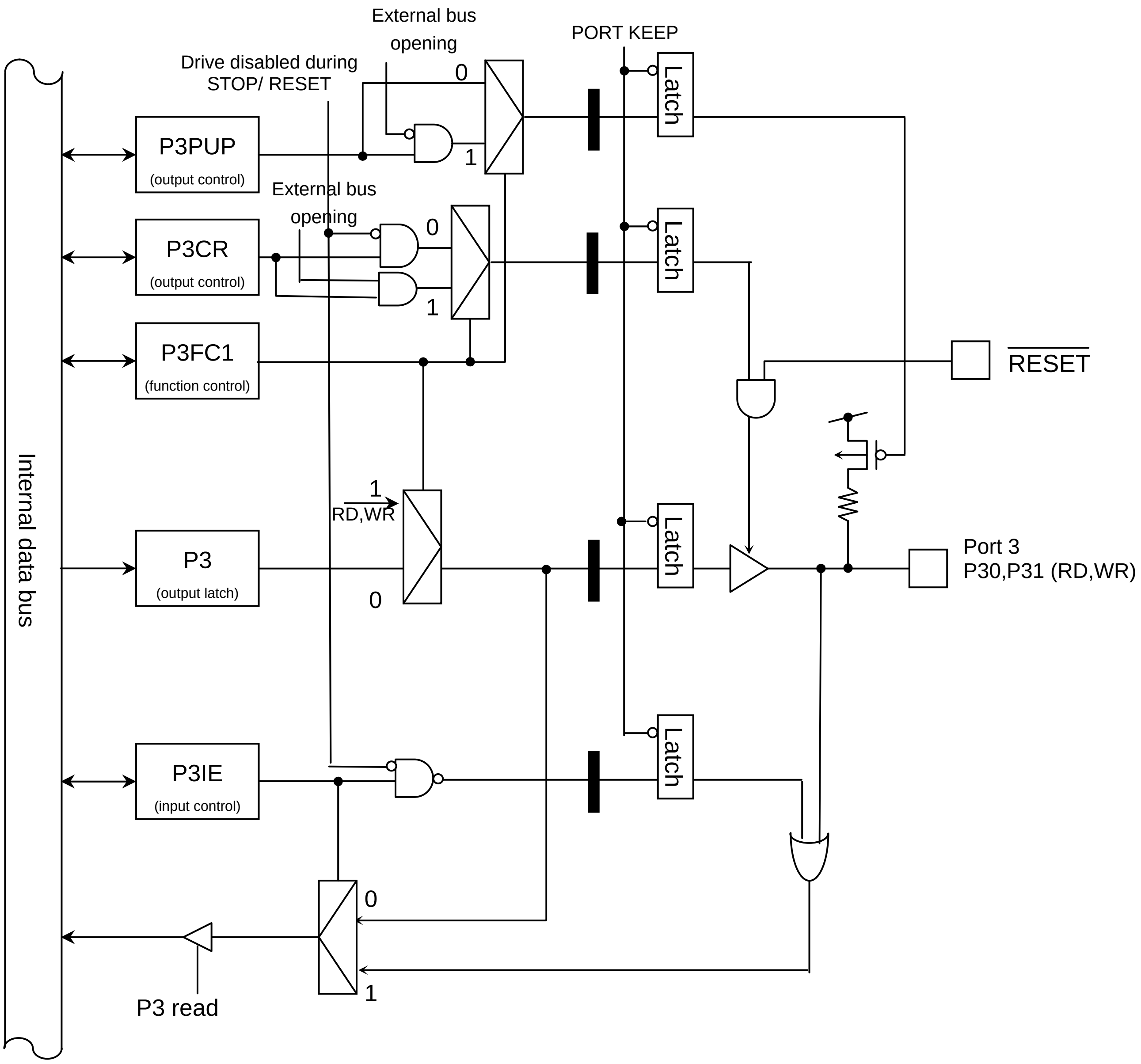


Fig. 7.4 Port 3 (P30, P31)

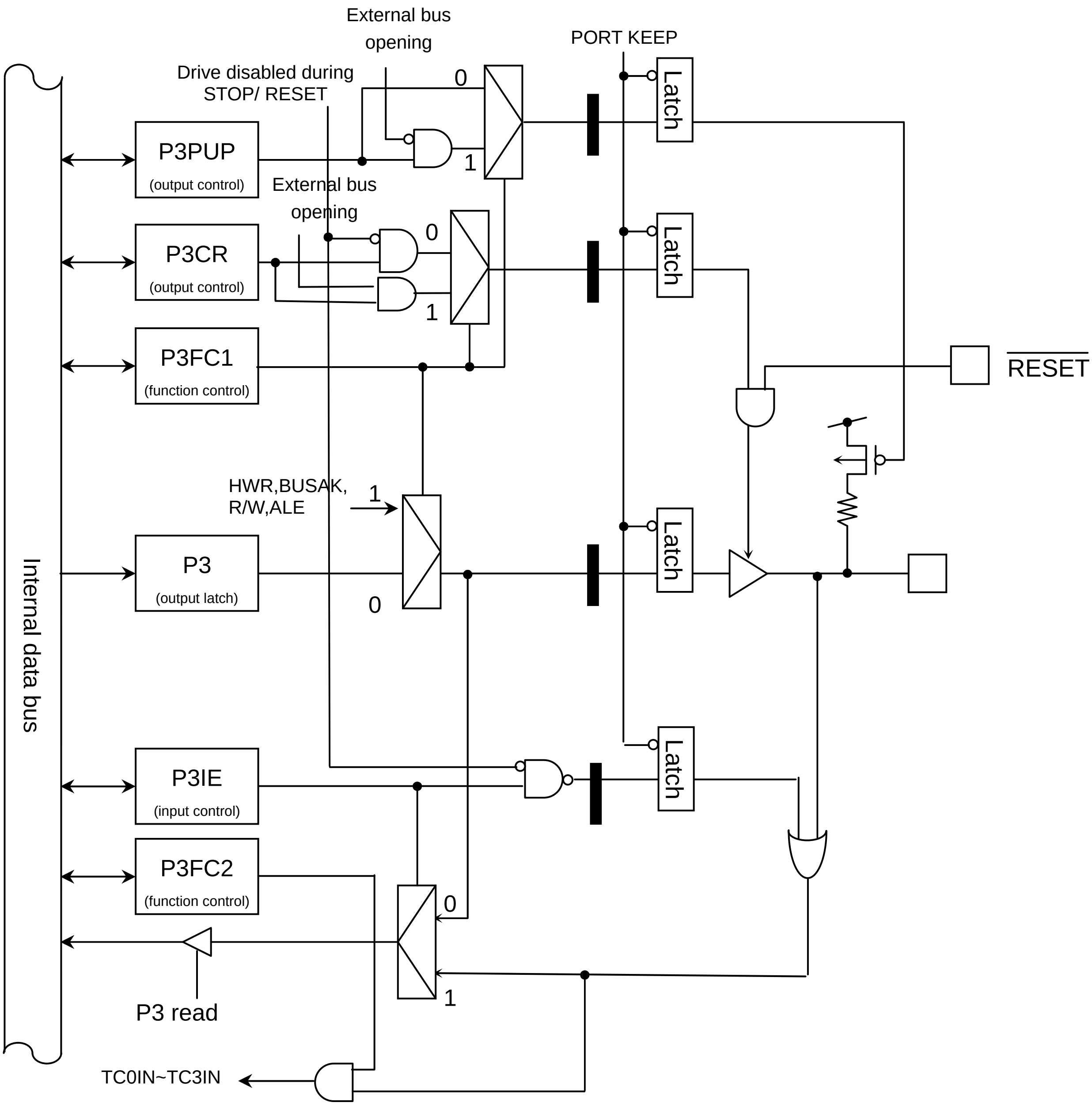


Fig. 7.5 Port 3 (P32, P35 through P37)

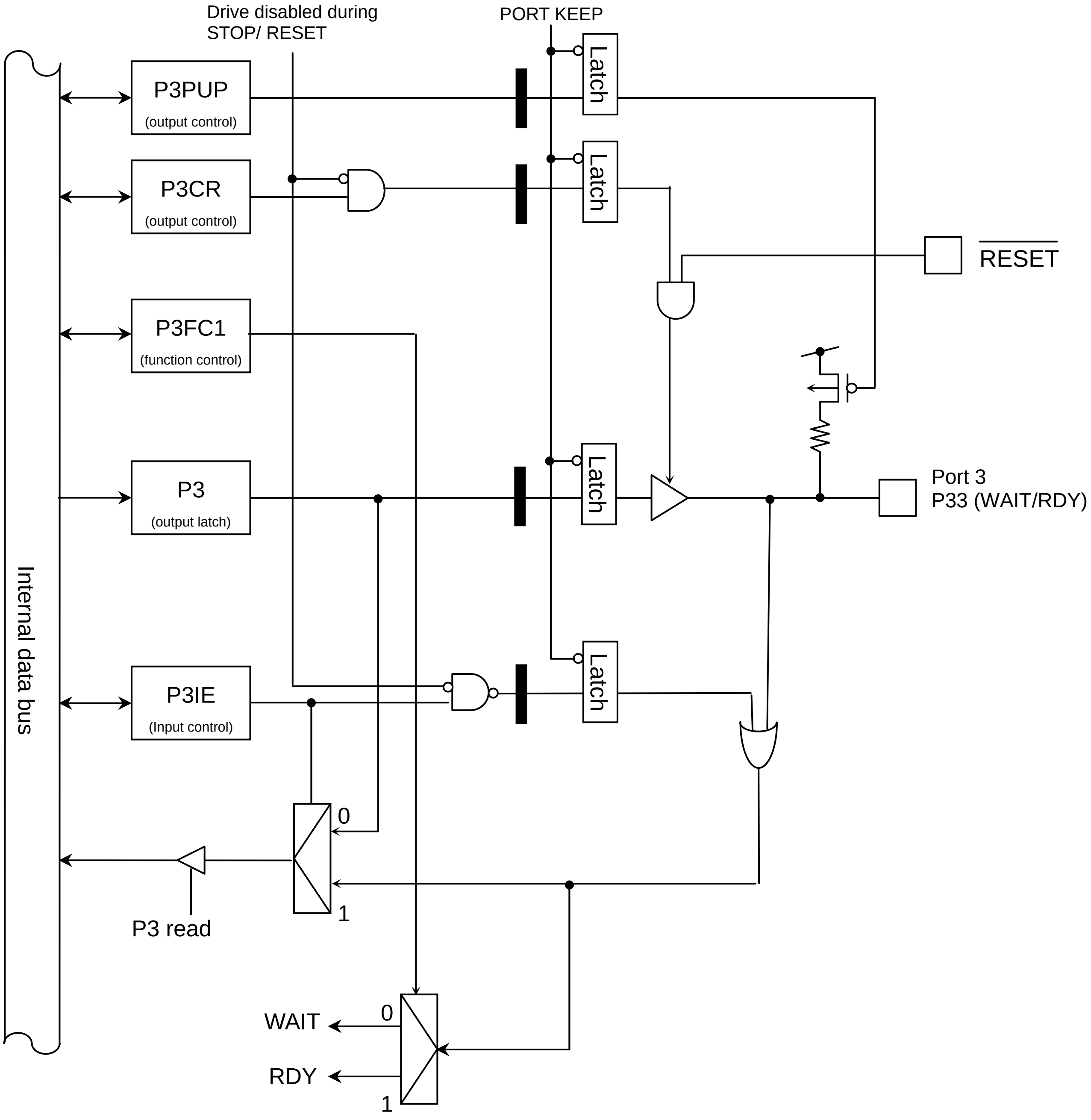


Fig. 7.6 Port 3 (P33)

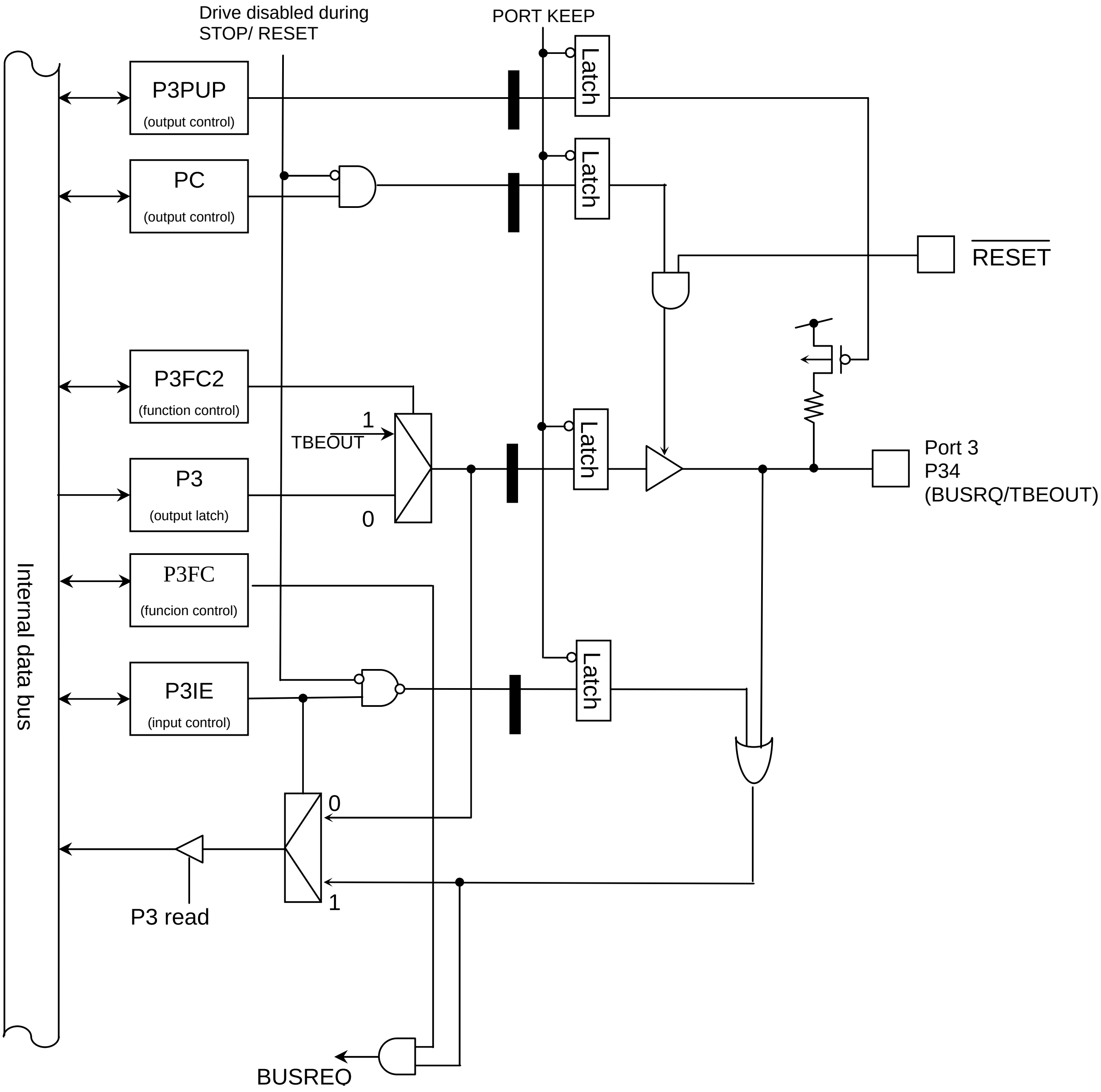


Fig. 7.7 Port 3 (P34)

Port 3 register

P3
(0xFF00_40C0)

	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)	Input mode						
		1						

(*1) Default setting of the bit is determined according to the bus mode P45(BUSMD).
L (separate bus): 1   H (multiplex bus): 0

Port 3 control register

P3CR
(0xFF00_40C4)

	7	6	5	4	3	2	1	0
Bit Symbol	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
Read/Write	R/W							
After reset	To be determined according to the bus mode (*2)	0	0	0	0	0	0	0
Function		0: Input 1: Output						

(*1) Default setting of the bit is determined according to the bus mode P45(BUSMD).
L (separate bus): 0   H (multiplex bus): 1

Port 3 function register 1

P3FC
(0xFF00_4008)

	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: R/$\overline{W}$	0: PORT 1: BUSAK	0: PORT 1: BUSRQ	0: PORT/ $\overline{WAIT}$ 1:RDY	0: PORT 1: $\overline{HWR}$	0: PORT 1: $\overline{WR}$	0: PORT 1: $\overline{RD}$

Port 3 function register 2

P3FC2
(0xFF00_40CC)

	7	6	5	4	3	2	1	0
Bit Symbol	P37F	P36F	P35F	P34F	–	P32F	–	–
Read/Write	R/W				R	R/W	R	
After reset	0	0	0	0	0	0	0	
Function	0: PORT 1: TC3IN	0: PORT 1: TC2IN	0: PORT 1: TC1IN	0: PORT 1: TBEOUT	"0" is read.	0: PORT 1: TC0IN	"0" is read.	

Port 3 pull-up control register

P3PUP
(0xFF00_40EC)

	7	6	5	4	3	2	1	0
Bit Symbol	PE37	PE36	PE35	PE34	PE33	PE32	PE31	PE30
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 3 input enable control register

P3IE
(0xFF00_40F8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE37	PIE36	PIE35	PIE34	PIE33	PIE32	PIE31	PIE30
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

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.

Besides the general-purpose input/output port function, the port 4 performs other functions: P40 through P43 output the chip select signal ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$) and input the key-on wake-up, P44 functions as the SCOUT output pin for outputting internal clocks, and P47 outputs a 16-bit timer. By making necessary settings during a reset, P45 functions as a BUSMD pin for setting external bus modes, and P46 as an ENDIAN setting pin.

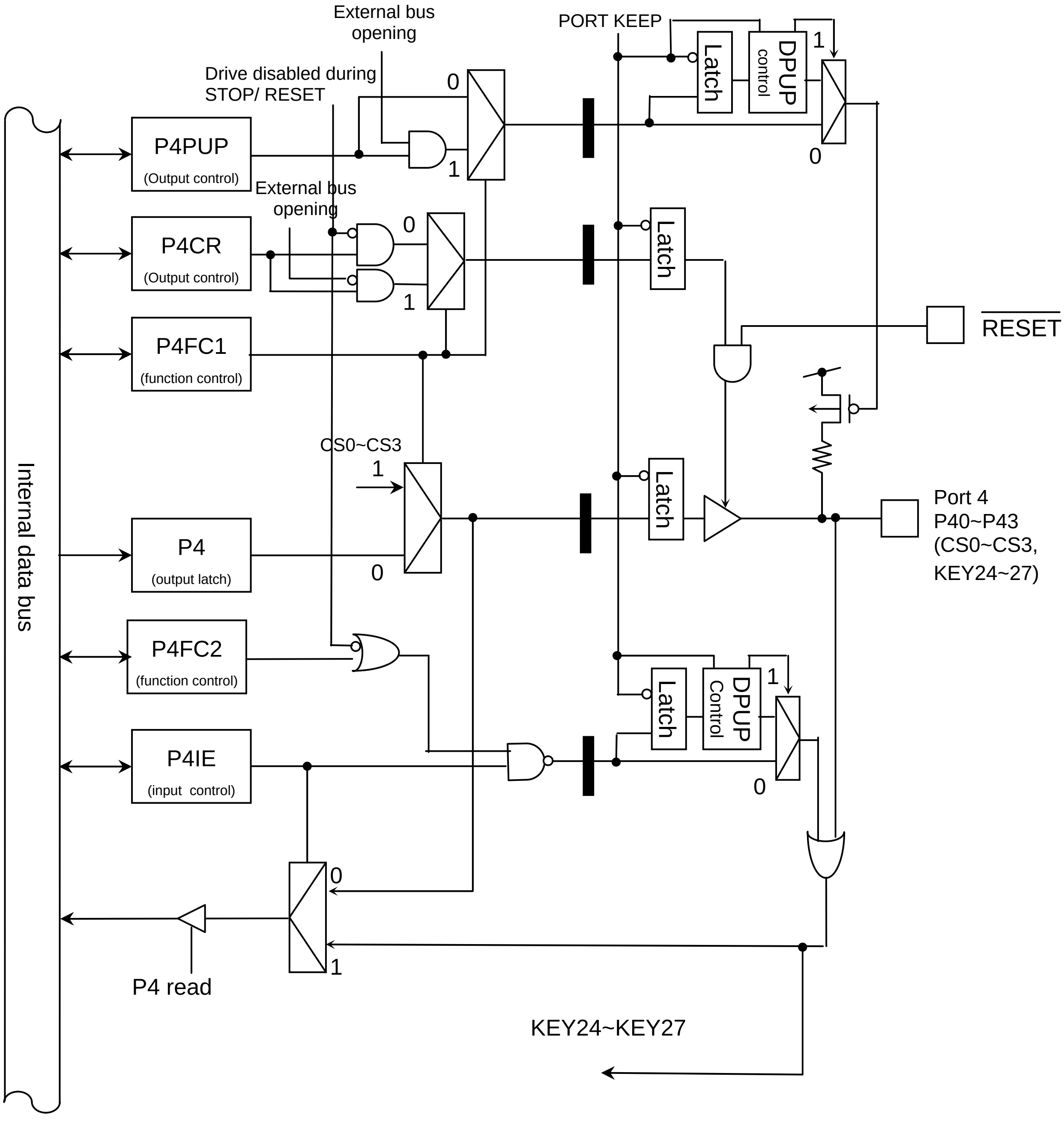


Fig. 7.8 Port 4(P40 through P43)

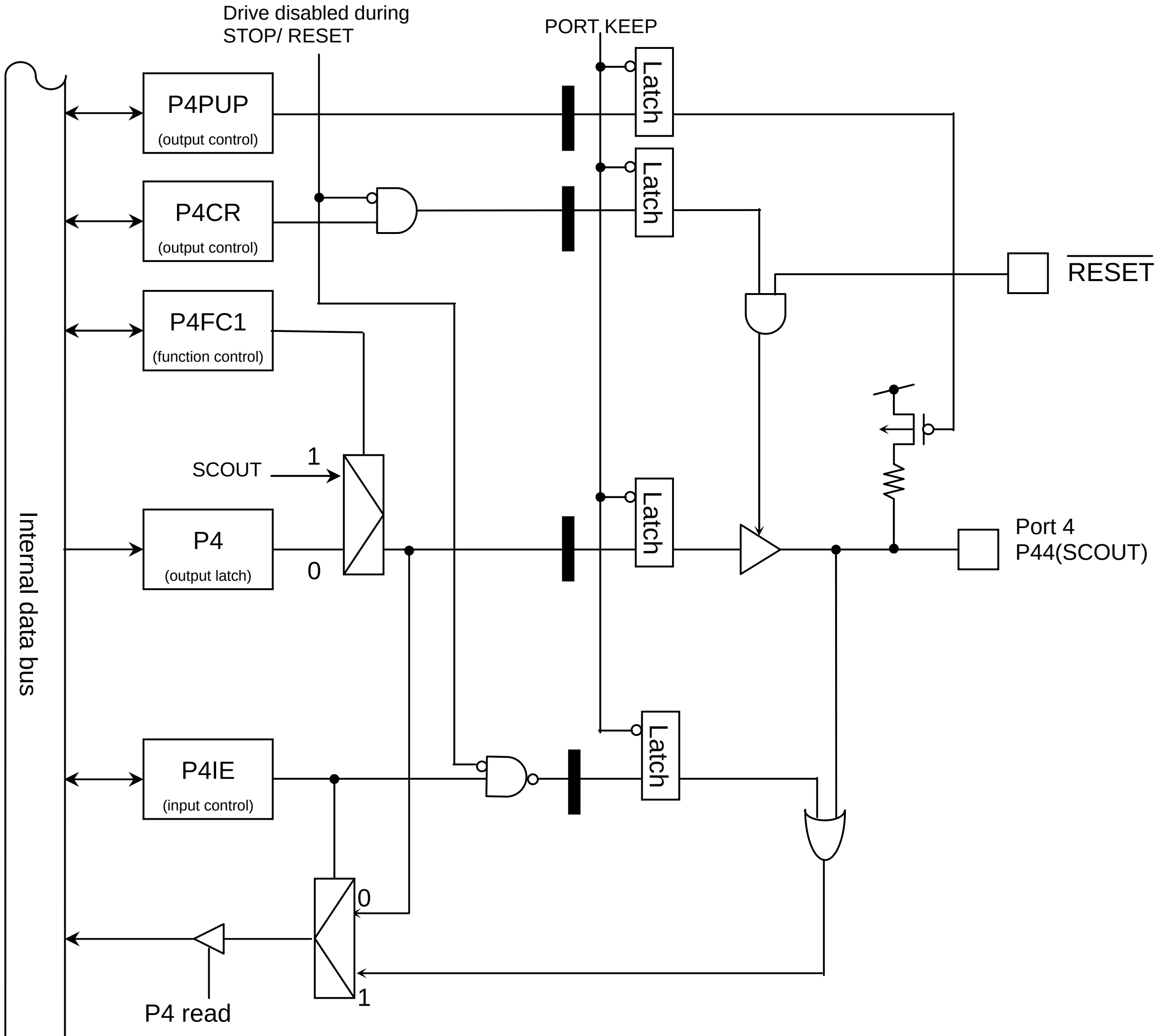


Fig. 7.9 Port 4 (P44)

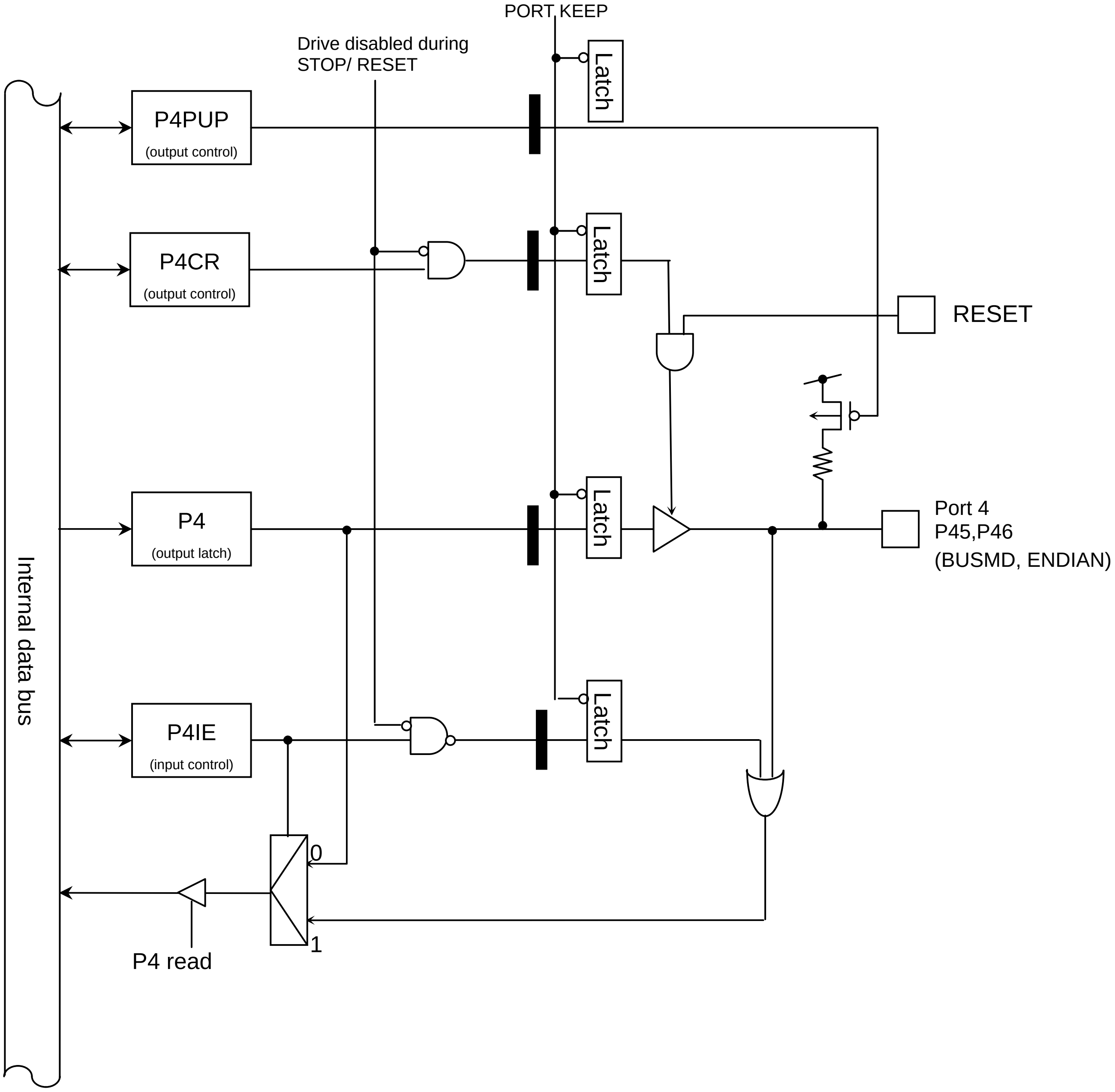


Fig. 7.10 Port 4 (P45,P46)

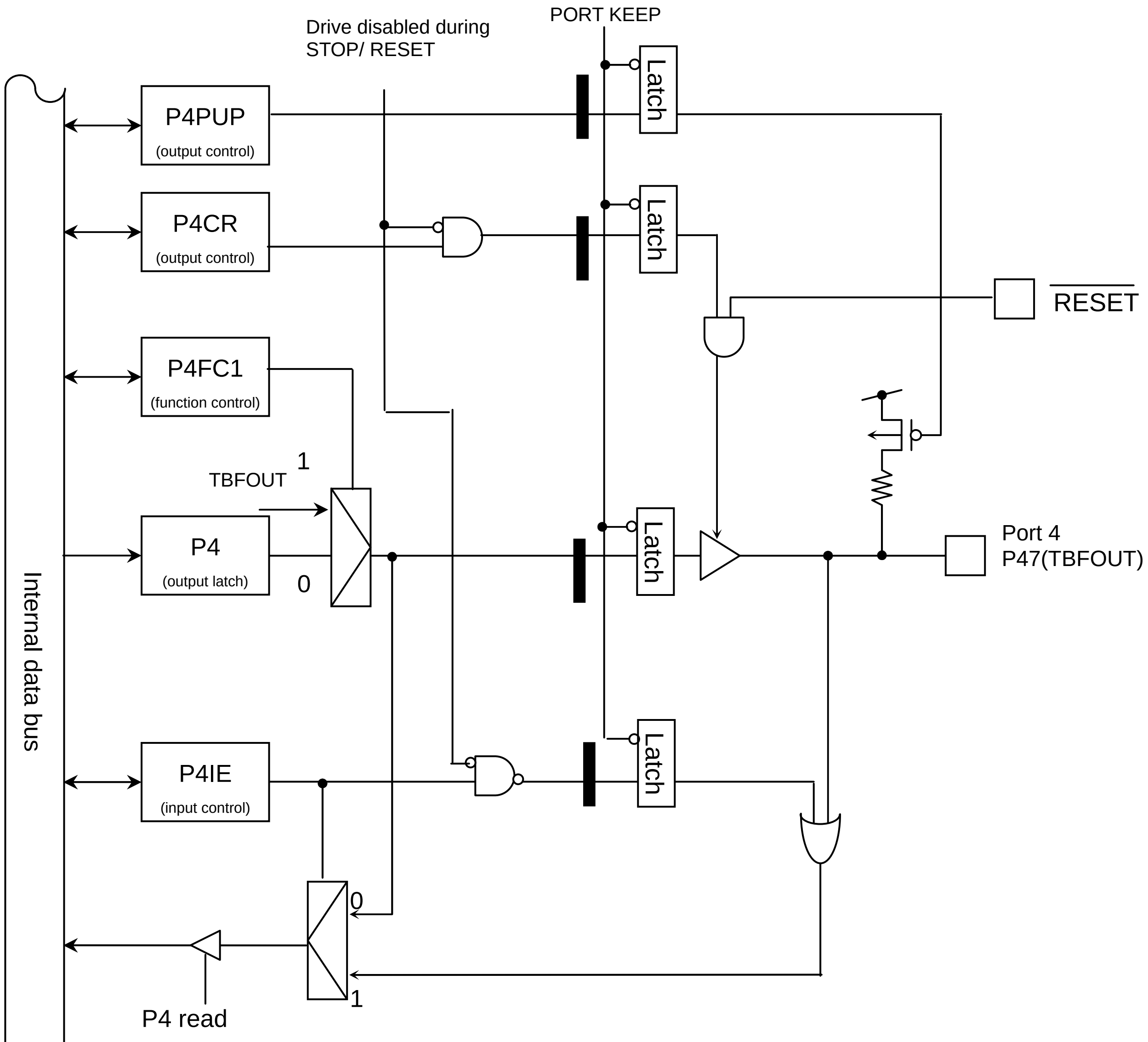


Fig. 7.11 Port 4 (P47)

Port 4 register

P4
(0xFF00_4100)

	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 (output latch register is set to "1.")							

Port 4 control register

P4CR
(0xFF00_4104)

	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 1

P4FC1
(0xFF00_4108)

	7	6	5	4	3	2	1	0
Bit Symbol	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
Read/Write	R/W	R		R/W				
After reset	0	0		0	0	0	0	0
Function	0: PORT 1: TBFOUT	“0” is read.		0: PORT 1: SCOUT	0: PORT 1: CS3	0: PORT 1: CS2	0: PORT 1: CS1	0: PORT 1: CS0

Port 4 function register

P4FC2
(0xFF00_410C)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	–	P43F	P42F	P41F	P40F
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.				0: PORT 1: KEY27	0: PORT 1: KEY26	0: PORT 1: KEY25	0: PORT 1: KEY24

Port 4 pull-up control register

P4PUP
(0xFF00_412C)

	7	6	5	4	3	2	1	0
Bit Symbol	PE47	PE46	PE45	PE44	PE43	PE42	PE41	PE40
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 4 input enable control register

P4IE
(0xFF00_4138)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE47	PIE46	PIE45	PIE44	PIE43	PIE42	PIE41	PIE40
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

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 registers P5FC1, P5FC2 and P5FC3. A reset allows all bits of the output latch P5 to be set to "1," all bits of P5CR, P5FC1, P5FC2 and P5FC3 to be cleared to "0," and the port 5 to be put in input mode.

Besides the input/output port function, the port 5 performs other functions: P50 through P53 input external interrupts, P54 through P57 output a 16-bit timer, and P56 and P57 input the key-on wake-up. These functions are enabled by setting the corresponding bit of P5FC to "1."

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 (port 45) must be set to "L" level during a reset.)

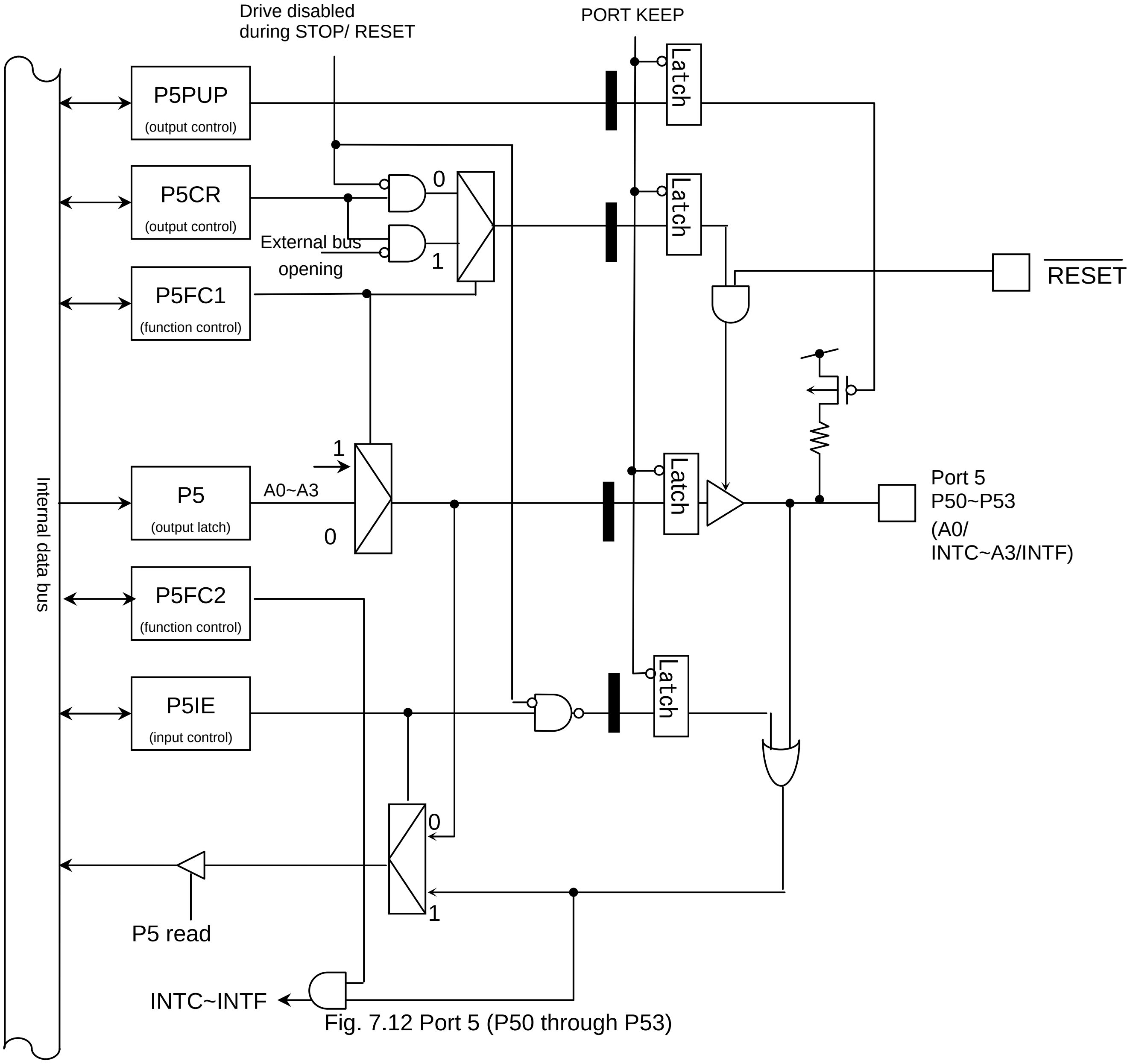


Fig. 7.12 Port 5 (P50 through P53)

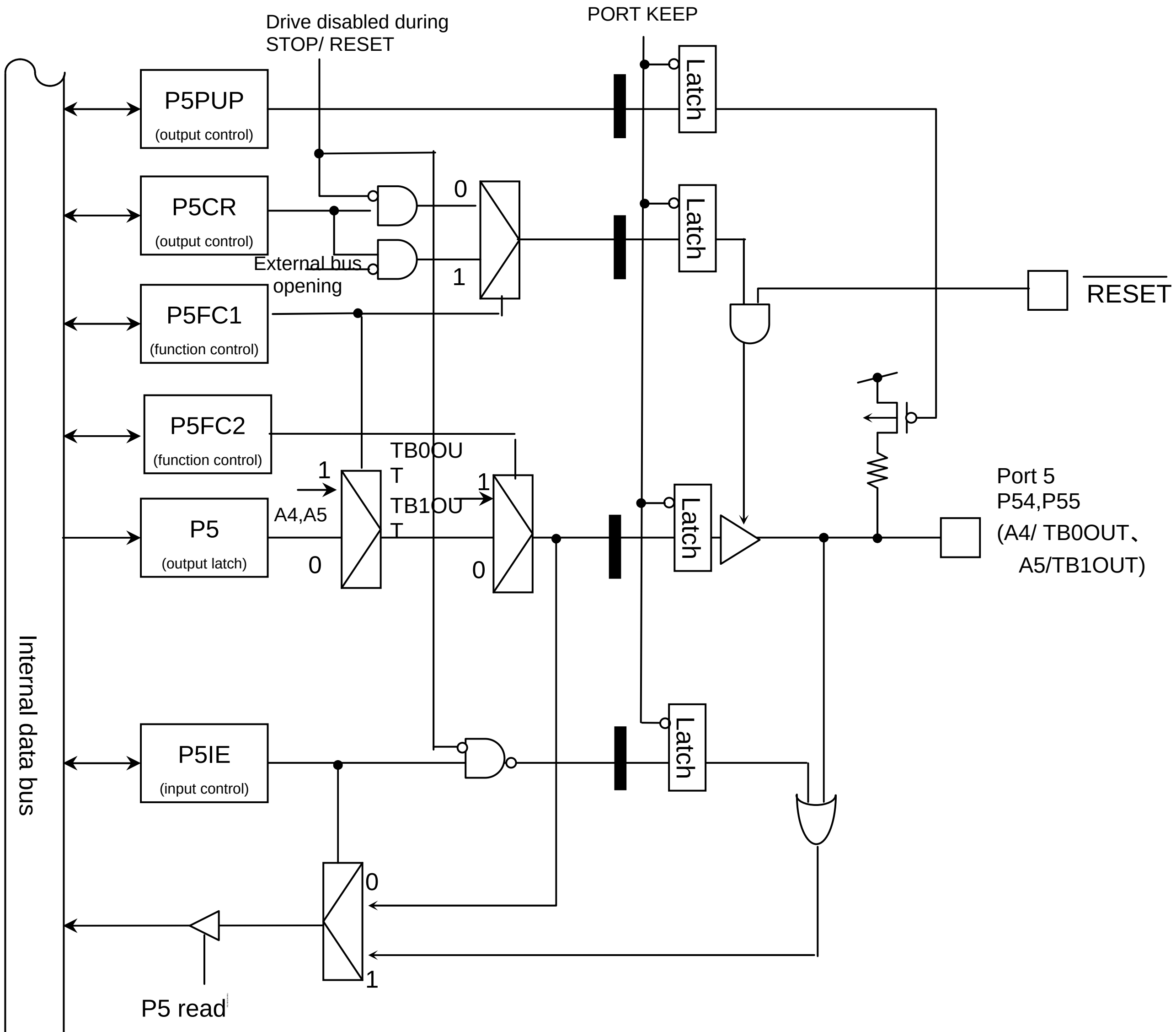


Fig. 7.13 Port 5 (P54, P55)

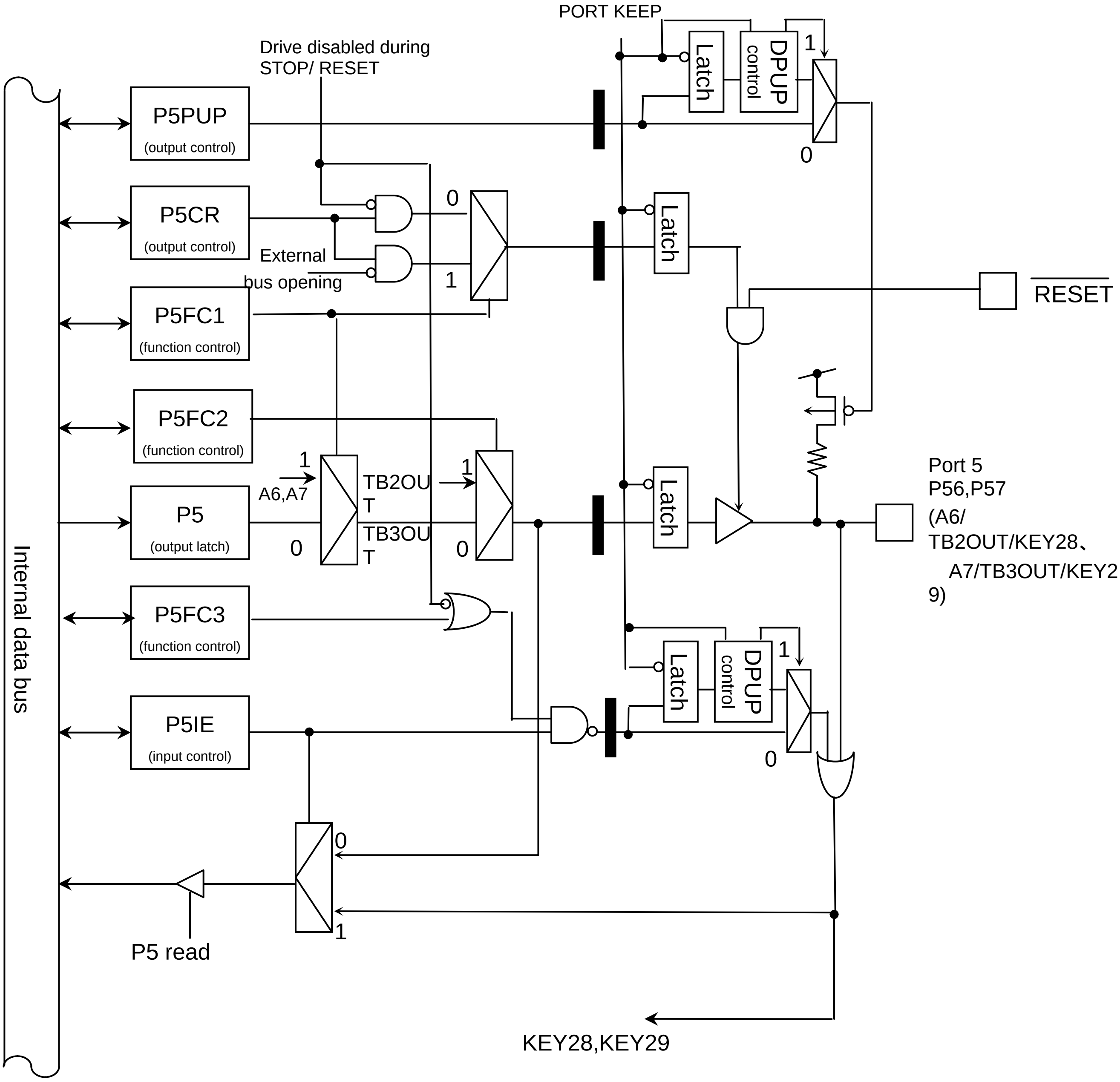


Fig. 7.14 Port 5 (P56, P57)

Port 5 register

P5
(0xFF00_4140)

	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
(0xFF00_4144)

	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 1

P5FC1
(0xFF00_4148)

	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: external bus							

Port 5 function register 2

P5FC2
(0xFF00_414C)

	7	6	5	4	3	2	1	0
Bit Symbol	P57F2	P56F2	P55F2	P54F2	P53F2	P52F2	P51F2	P50F2
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:TB0OUT	0:PORT 1:INTF	0:PORT 1:INTE	0:PORT 1:INTD	0:PORT 1:INTC

Port 5 function register 3

P5FC3
(0xFF00_4150)

	7	6	5	4	3	2	1	0
Bit Symbol	P57F3	P56F3	–					
Read/Write	R/W		R					
After reset	0	0	0					
Function	0:PORT 1:KEY29	0:PORT 1:KEY28	“0” is read.					

Port 5 pull-up control register

P5PE
(0xFF00_416C)

	7	6	5	4	3	2	1	0
Bit Symbol	PE57	PE56	PE55	PE54	PE53	PE52	PE51	PE50
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 5 Input enable control register

P5IE
(0xFF00_4178)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE57	PIE56	PIE55	PIE54	PIE53	PIE52	PIE51	PIE50
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled

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 registers P6FC1, P6FC2 and P6FC3. A reset allows all bits of the output latch P6 to be set to "1," all bits of P6CR, P6FC1, P6FC2 andP6FC3 to be cleared to "0," and the port 6 to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of P6IE to “1”.

Besides the input/output port function, the port 6 performs other functions: P60 and P64 output SIO data, P61 and P65 input SIO data, P62 and P66 input and output SIO CLK or input CTS, P61 and P65 input external interrupts, and P63 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 P6FC1 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 (port 45) must be set to "L" level during a reset.)

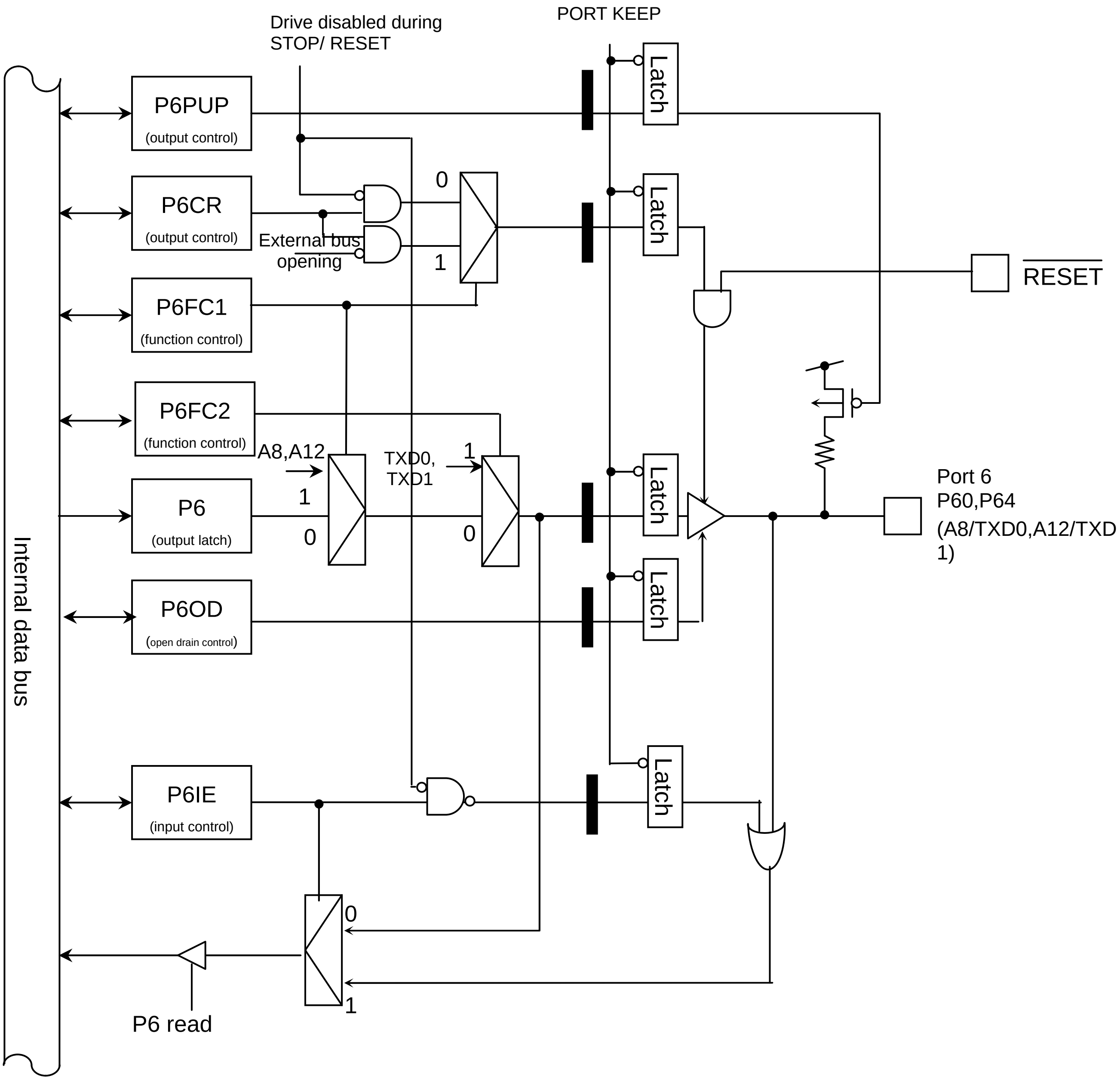


Fig. 7.15 Port 6 (P60, P64)

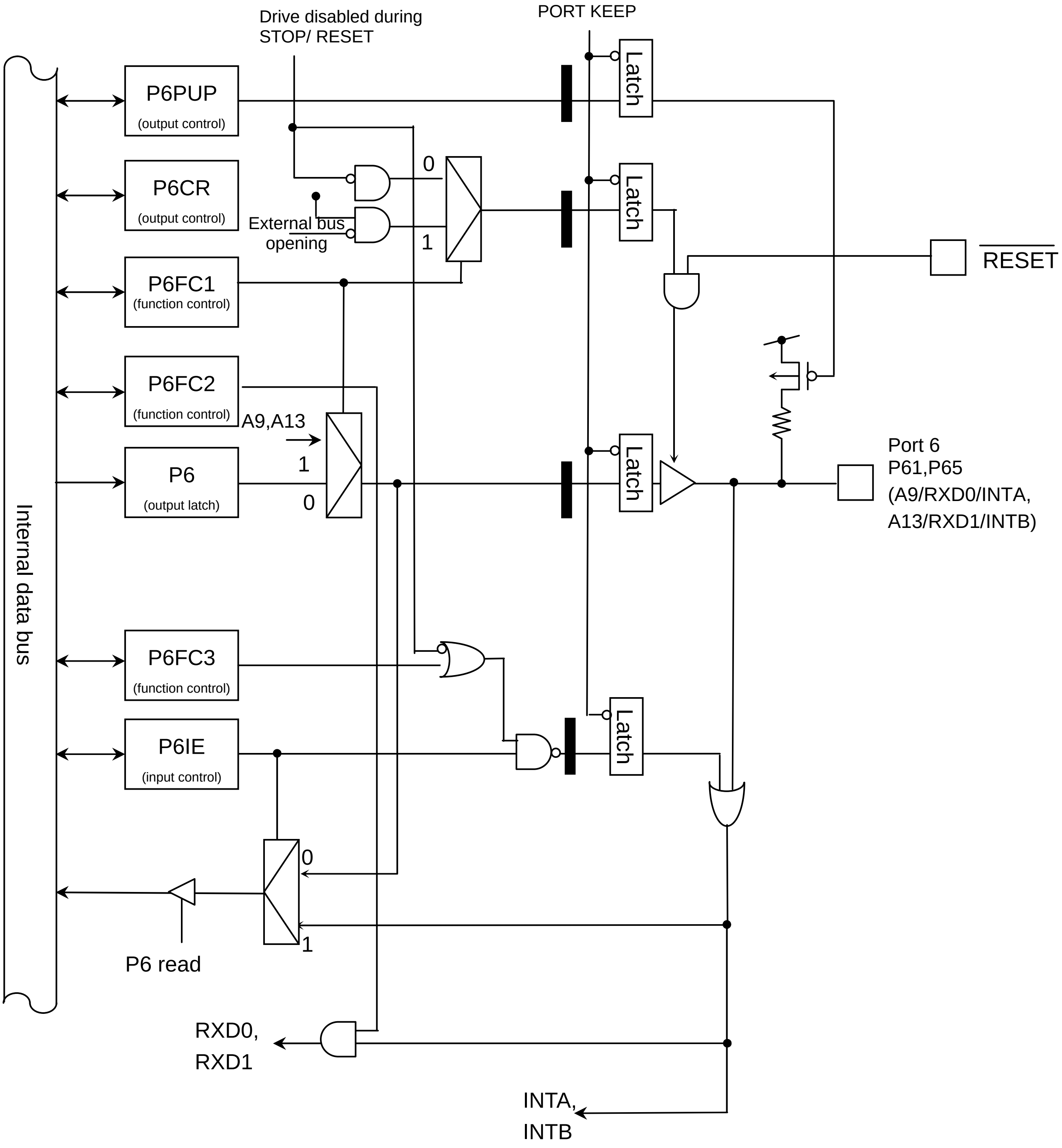


Fig. 7.16 Port 6 (P61, P65)

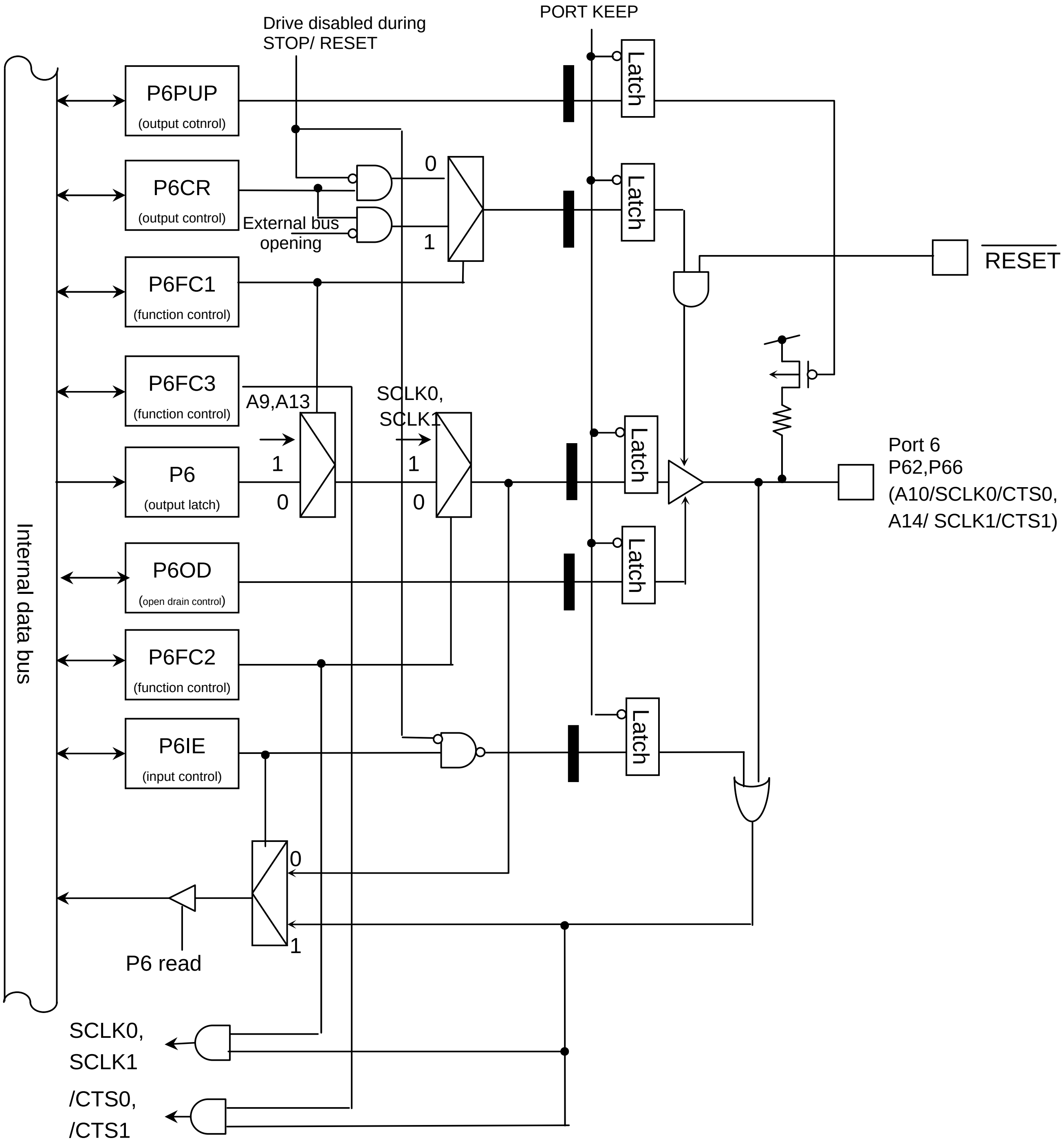


Fig. 7.17 Port 6(P62, P66)

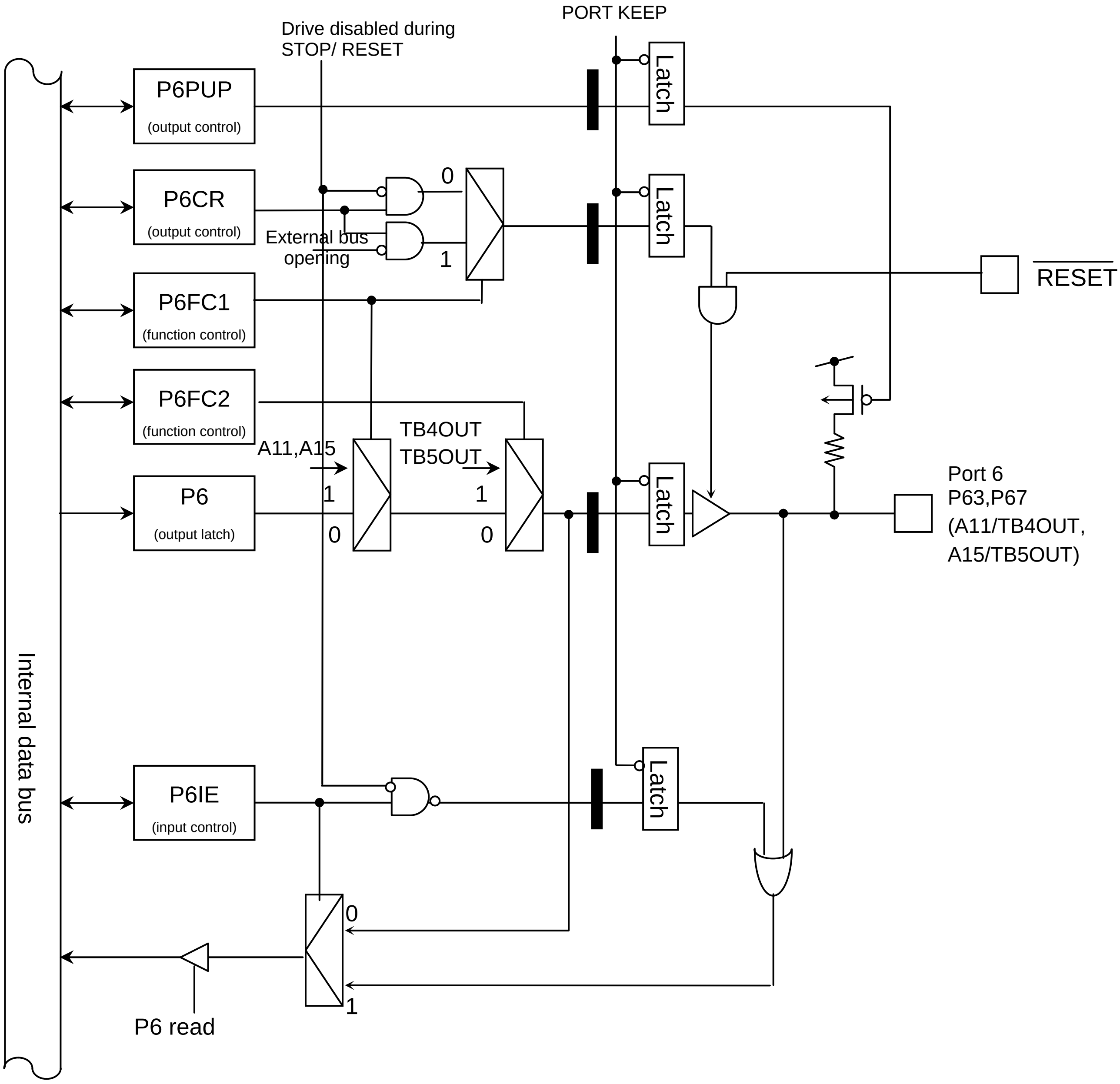


Fig. 7.18 Port 6 (P63, P67)

Port 6 register

P6
(0xFF00_4180)

	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
(0xFF00_4184)

	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 1

P6FC1
(0xFF00_4188)

	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:External bus							

Port 6 function register 2

P6FC2
(0xFF00_418C)

	7	6	5	4	3	2	1	0
Bit Symbol	P67F2	P66F2	P65F2	P64F2	P63F2	P62F2	P61F2	P60F2
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:TB5OUT	0:PORT 1:SCLK1	0:PORT 1:RXD1	0:PORT 1: TXD1	0:PORT 1:TB4OUT	0:PORT 1:SCLK0	0:PORT 1:RXD0	0:PORT 1: TXD0

Port 6 function register 3

P6FC3
(0xFF00_4190)

	7	6	5	4	3	2	1	0
Bit Symbol	—	P66F3	P65F3	—	—	P62F3	P61F3	—
Read/Write	R	R/W		R		R/W		R
After reset	0	0	0	0		0	0	
Function	“0” is read.	“0” is read.	0:PORT 1:INTB	“0” is read.		0:PORT 1:CTS0	0:PORT 1:INTA	“0” is read.

Port 6 open drain control register

P6ODE
(0xFF00_ 41A8)

	7	6	5	4	3	2	1	0
Bit Symbol	—	P66ODE	—	P64ODE	—	P62ODE	—	P60ODE
Read/Write	R	R/W	R	R/W	R	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	0: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain

Port 6 pull-up control register

P6PUP
(0xFF00_41AC)

	7	6	5	4	3	2	1	0
Bit Symbol	PE67	PE66	PE65	PE64	PE63	PE62	PE61	PE60
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

7.8 Port 7 (P70 through P77)

The port 7 is an 8-bit, analog input port for the A/D converter (unit A and B). Although P72 P73, P76 and P77 form part of the analog input port, they also perform another function of inputting external interrupt.

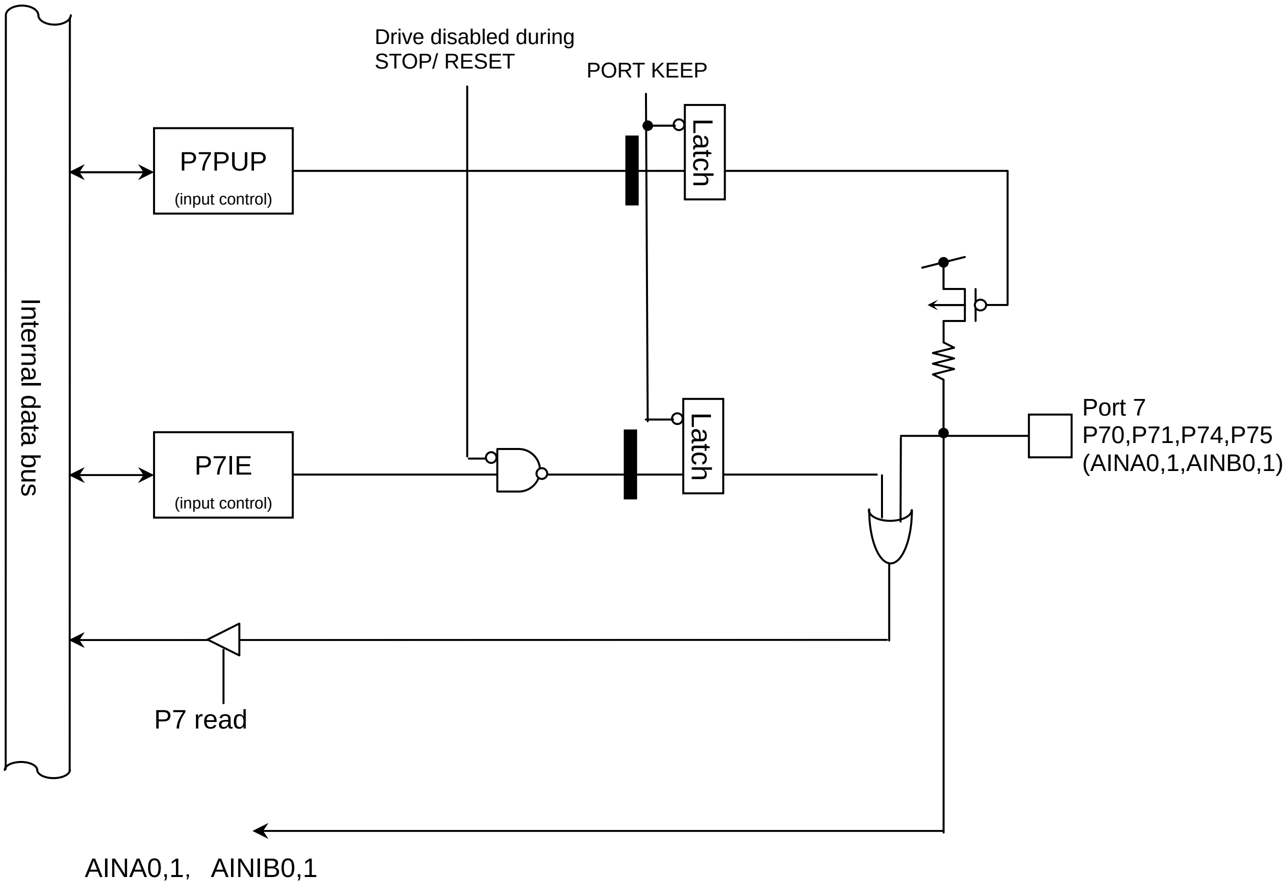


Fig. 7.19 Port 7 (P70, P71, P74 and P75)

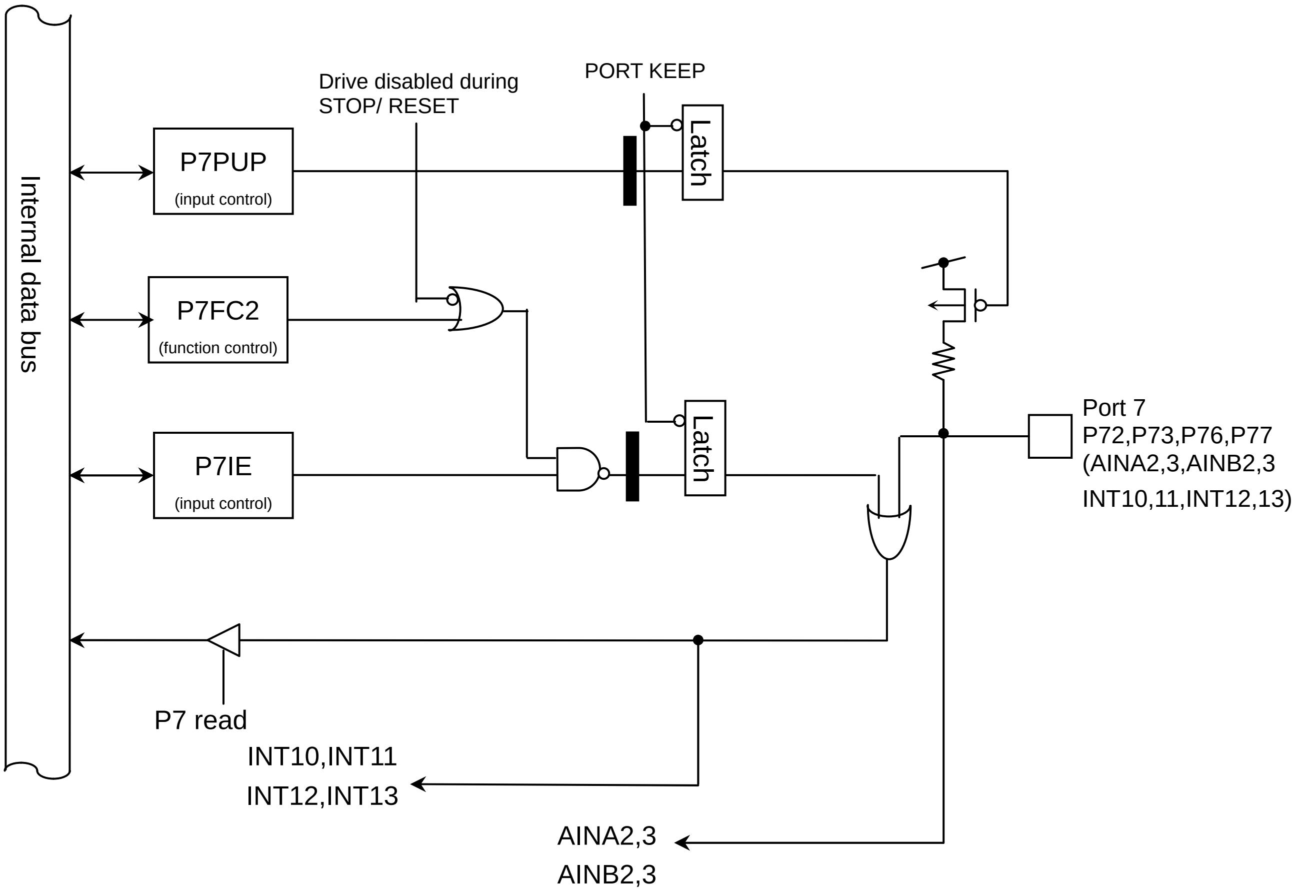


Fig. 7.20 Port 7 (P72, P73, P76 and P77)

Port 7 register

P7
(0xFF00_41C0)

	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 2

P7FC2
(0xFF00_41CC)

	7	6	5	4	3	2	1	0
Bit Symbol	P77F2	P76F2	–	–	P77F2	P76F2	–	–
Read/Write	R/W		R		R/W		R	
After reset	0	0	0		0	0	0	
Function	0:PORT 1: INTC13	0:PORT 1: INTC12	"0" is read.		0:PORT 1: INTC11	0:PORT 1: INTC10	"0" is read.	

Port 7 pull-up control register

P7PUP
(0xFF00_41EC)

	7	6	5	4	3	2	1	0
Bit Symbol	PE77	PE76	PE75	PE74	PE73	PE72	PE71	PE70
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 7 Input enable control register

P7IE
(0xFF00_41F8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE77	PIE76	PIE75	PIE74	PIE73	PIE72	PIE71	PIE70
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled

7.9 Port 8 (P80 through P87)

The port 8 is an 8-bit, analog input port for the A/D converter (unit C). Besides this analog input port function, P86 and P87 input external interrupts.

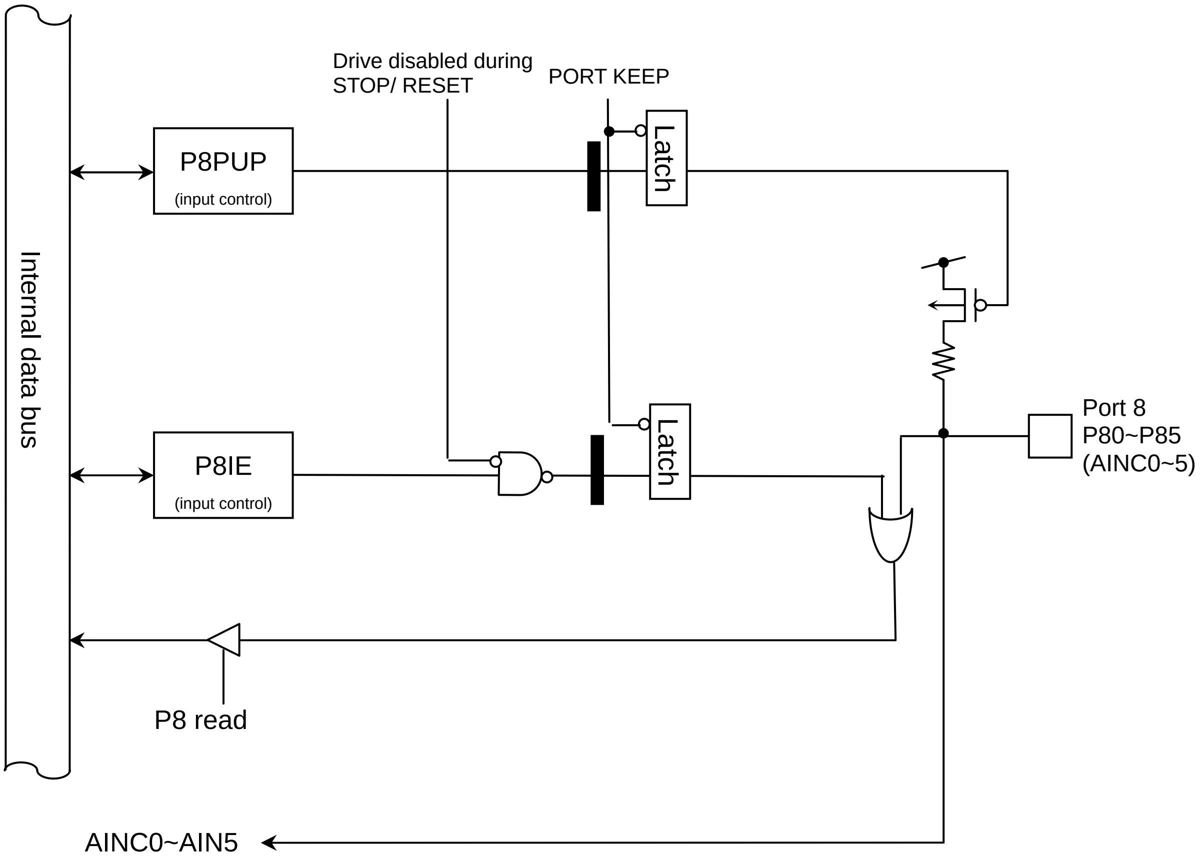


Fig. 7.21 Port 8 (P80 through P85)

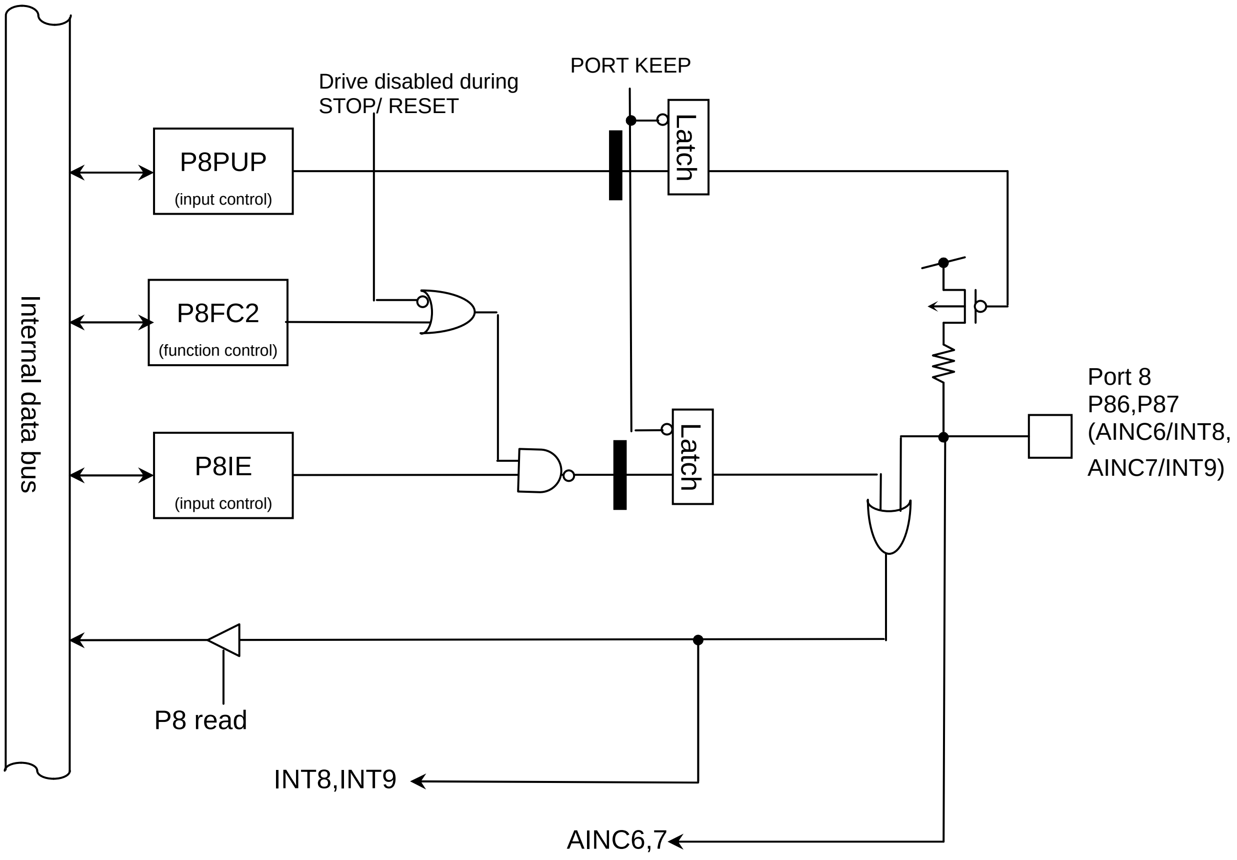


Fig. 7.22 Port 8 (P86, P87)

Port 8 register

P8
(0xFF00_4200)

	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 2

P8FC2
(0xFF00_420C)

	7	6	5	4	3	2	1	0
Bit Symbol	P87F2	P86F2	—	—	—	—	—	—
Read/Write	R/W		R					
After reset	0	0	0					
Function	0:PORT 1: INTC9	0:PORT 1: INTC8	“0” is read.					

Port 8 pull-up control register

P8PUP
(0xFF00_422C)

	7	6	5	4	3	2	1	0
Bit Symbol	PE87	PE86	PE85	PE84	PE83	PE82	PE81	PE80
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 8 Input enable control register

P8IE
(0xFF00_4238)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE87	PIE86	PIE85	PIE84	PIE83	PIE82	PIE81	PIE80
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled

7.10 Port 9 (P90 through P97)

The port 9 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 P9CR and the function registers P9FC1 and P9FC2. A reset allows all bits of the output latch P9 to be set to "1," all bits of P9CR, P9FC1 and P9FC2 to be cleared to "0," and the port 9 to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of P9IE to “1”.

Besides the input/output port function, the port 9 performs other functions: P90 and P94 output HSIO/SI0 data, P92 and P96 inputs and outputs SI0 CLK or inputs CTS, and P93 and P97 output a 16-bit timer.

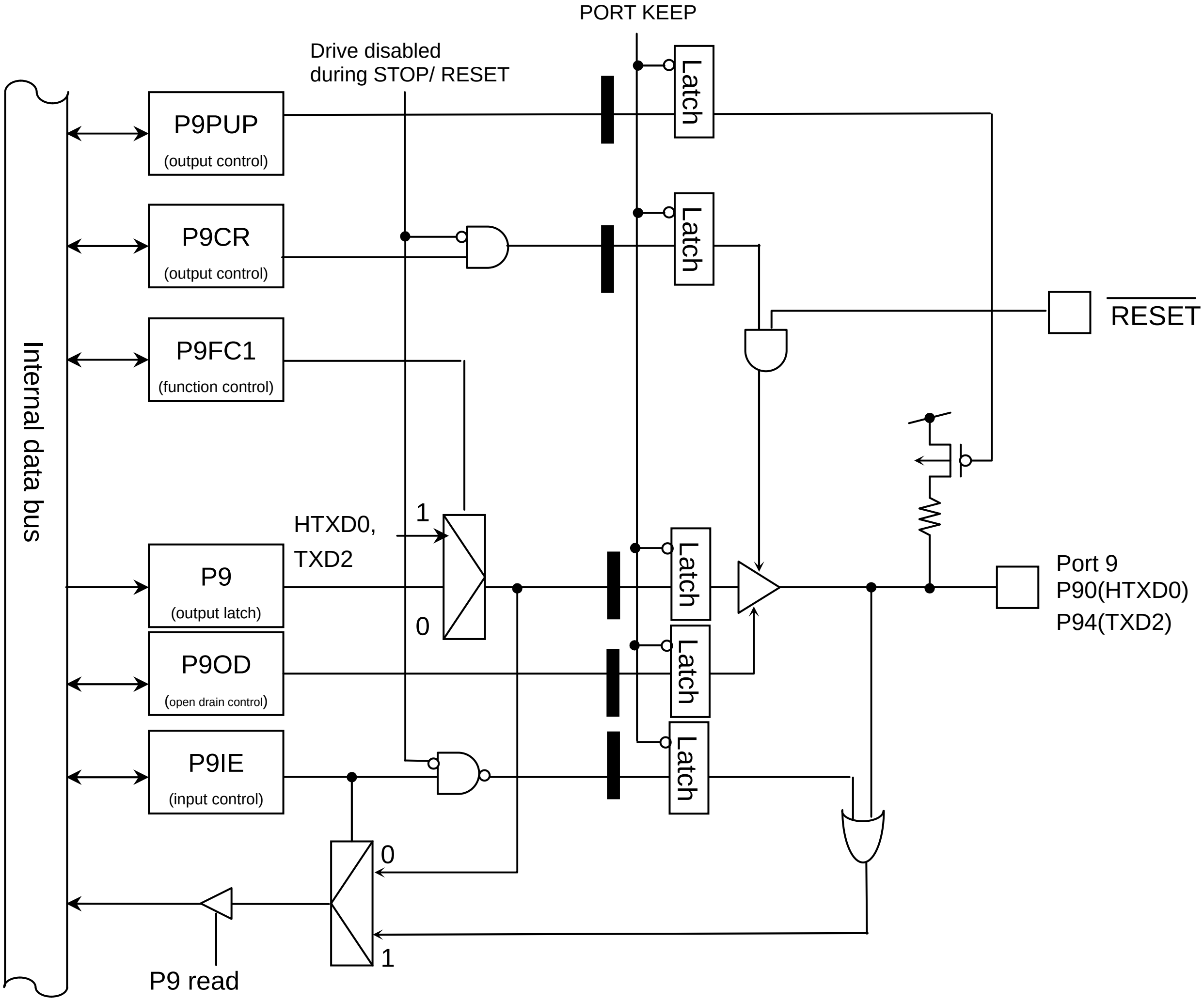


Fig. 7.23 Port 9 (P90, P94)

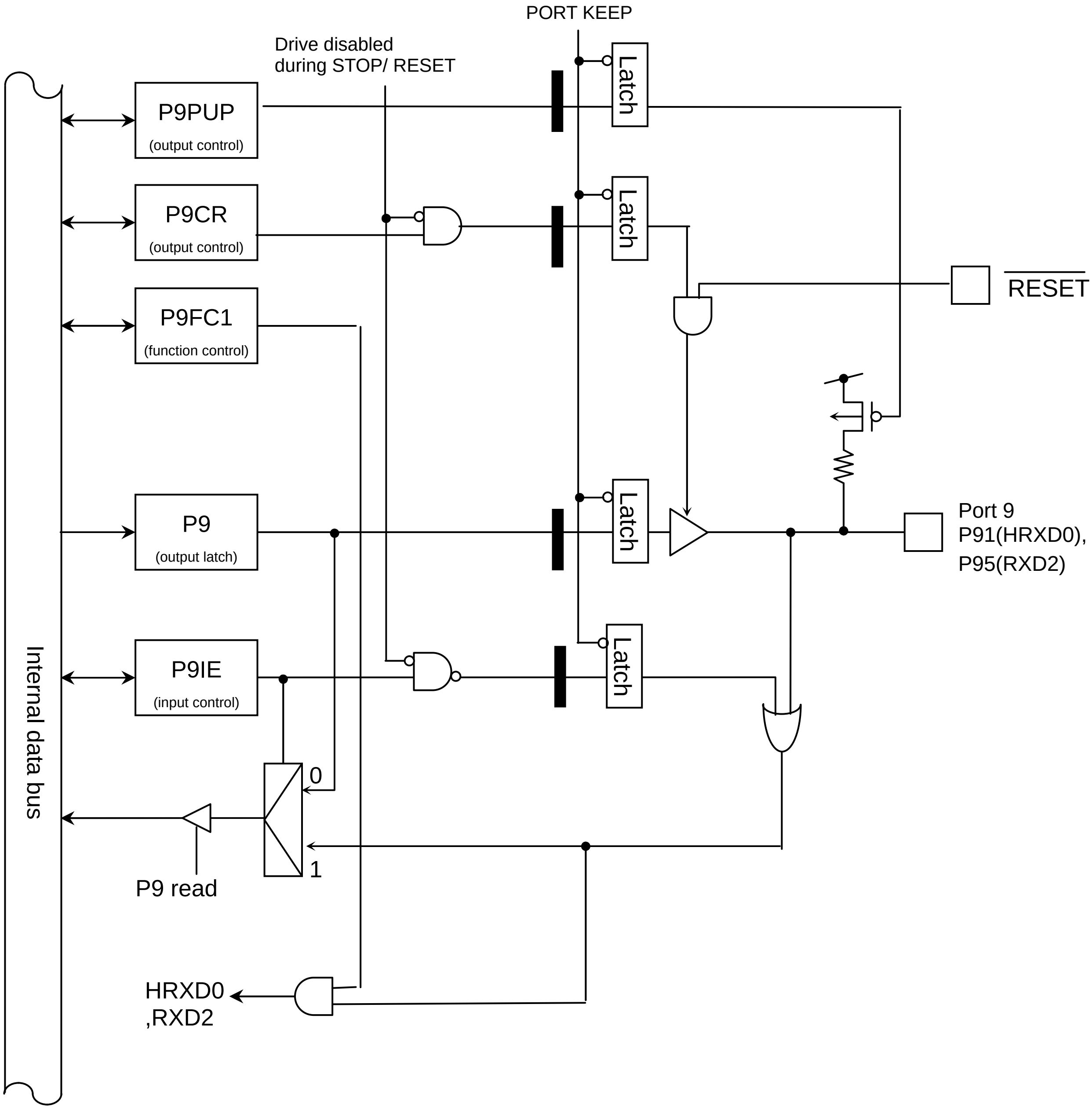


Fig. 7.24 Port 9 (P91, P95)

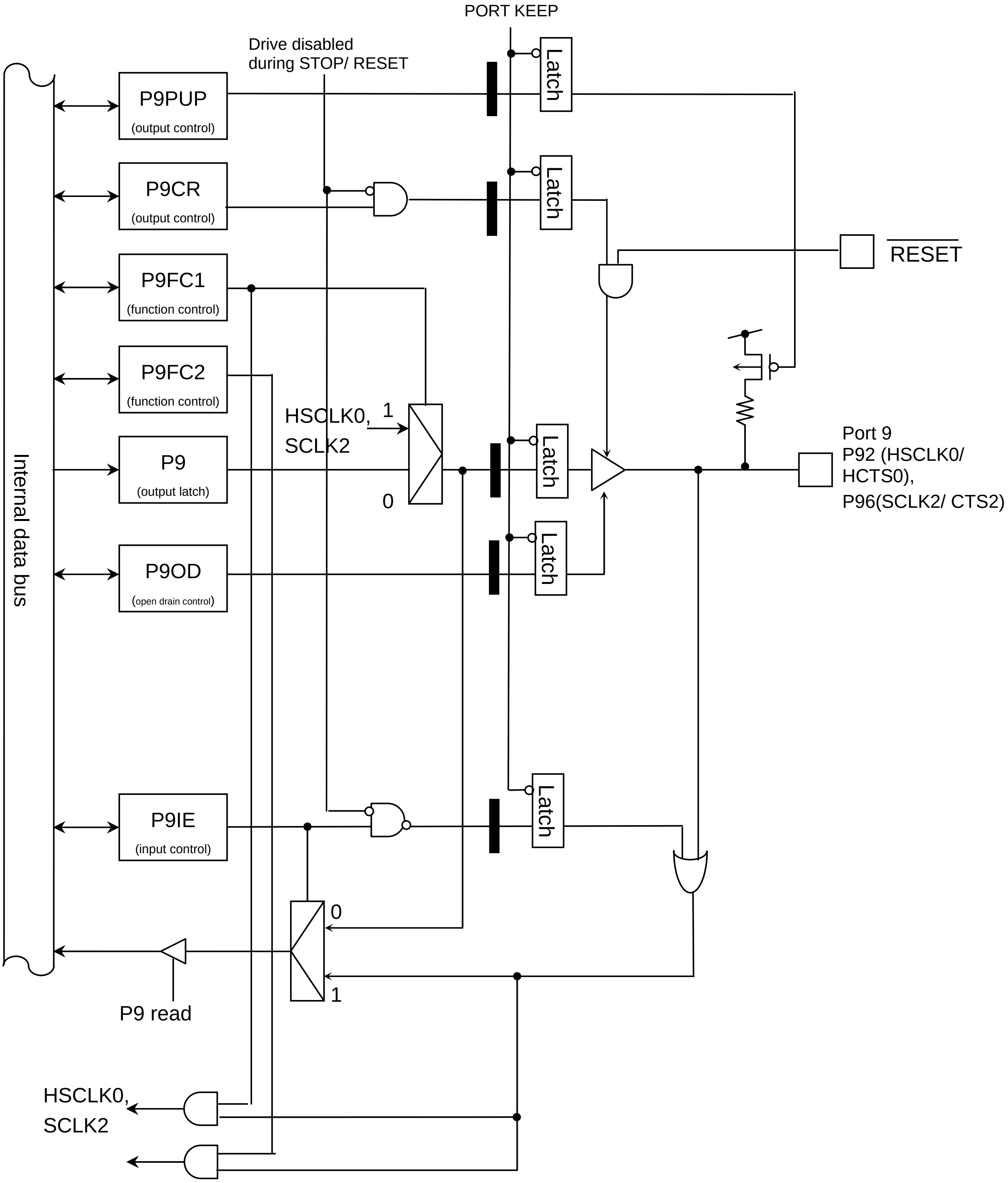


Fig. 7.25 Port 9 (P92, P96)

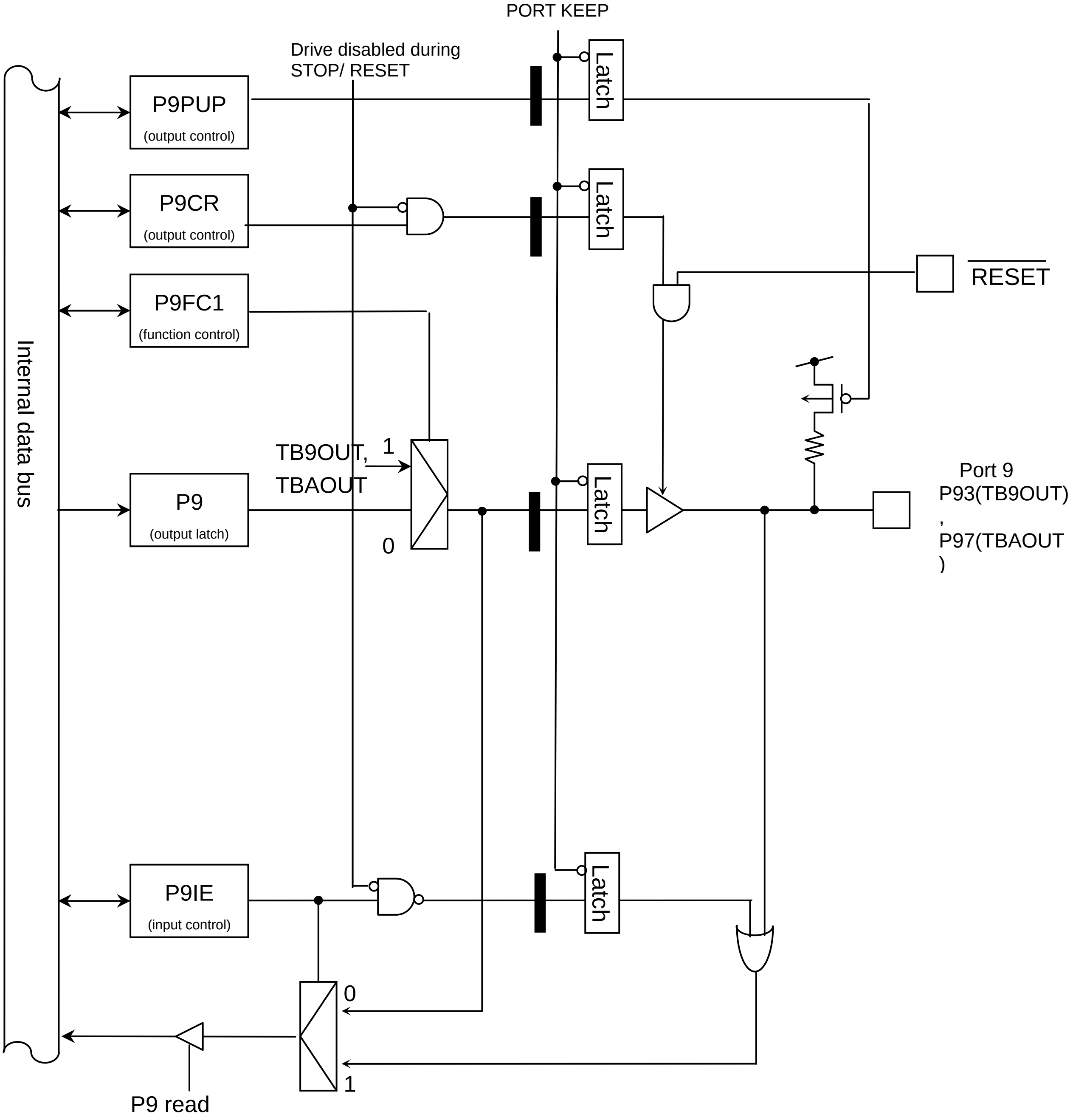


Fig. 7.26 Port 9 (P93, P97)

Port 9 register

P9
(0xFF00_4240)

	7	6	5	4	3	2	1	0
Bit Symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port 9 control register

P9CR
(0xFF00_4244)

	7	6	5	4	3	2	1	0
Bit Symbol	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port 9 function register 1

P9FC1
(0xFF00_4248)

	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:TBAOUT	0:PORT 1:SCLK2	0:PORT 1:RXD2	0:PORT 1:TXD2	0:PORT 1:TB9OUT	0:PORT 1:HSCLK0	0:PORT 1:HRXD0	0:PORT 1:HTXD0

Port 9 function register 2

P9FC2
(0xFF00_424C)

	7	6	5	4	3	2	1	0
Bit Symbol	–	P96F	–	–	–	P92F	–	–
Read/Write	R	R/W	R			R/W	R	
After reset	0	0	0			0	0	
Function	“0” is read.	0:PORT 1:CTS2	“0” is read.			0:PORT 1:HCTS0	“0” is read.	

Port 9 open drain control register

P9ODE
(0xFF00_4268)

	7	6	5	4	3	2	1	0
Bit Symbol	–	P96ODE	–	P94ODE	–	P92ODE	–	P90ODE
Read/Write	R	R/W	R	R/W	R	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	"0" Is read.	0: CMOS 1: Open drain	"0" Is read.	0: CMOS 1: Open drain	"0" Is read.	0: CMOS 1: Open drain	"0" Is read.	0: CMOS 1: Open drain

Port 9 pull-up control register

P9PUP
(0xFF00_426C)

	7	6	5	4	3	2	1	0
Bit Symbol	PE97	PE96	PE95	PE94	PE93	PE92	PE91	PE90
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port 9 Input enable control register

P9IE
(0xFF00_4278)

	7	6	5	4	3	2	1	0
Bit Symbol	PIE97	PIE96	PIE95	PIE94	PIE93	PIE92	PIE91	PIE90
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled

7.11 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 and the function registers PAFC1 and PAFC2. A reset allows all bits of the output latch PA to be set to "1," all bits of PACR, PAFC1 and PAFC2 to be cleared to "0," and the port A to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PAIE to “1”.

Besides the input/output port function, the port A performs other functions: PA0 through PA5 input external interrupts, PA0 through PA3, PA6 and PA7 count two-phase pulse input, and PA6 and PA7 input 16-bit timer.

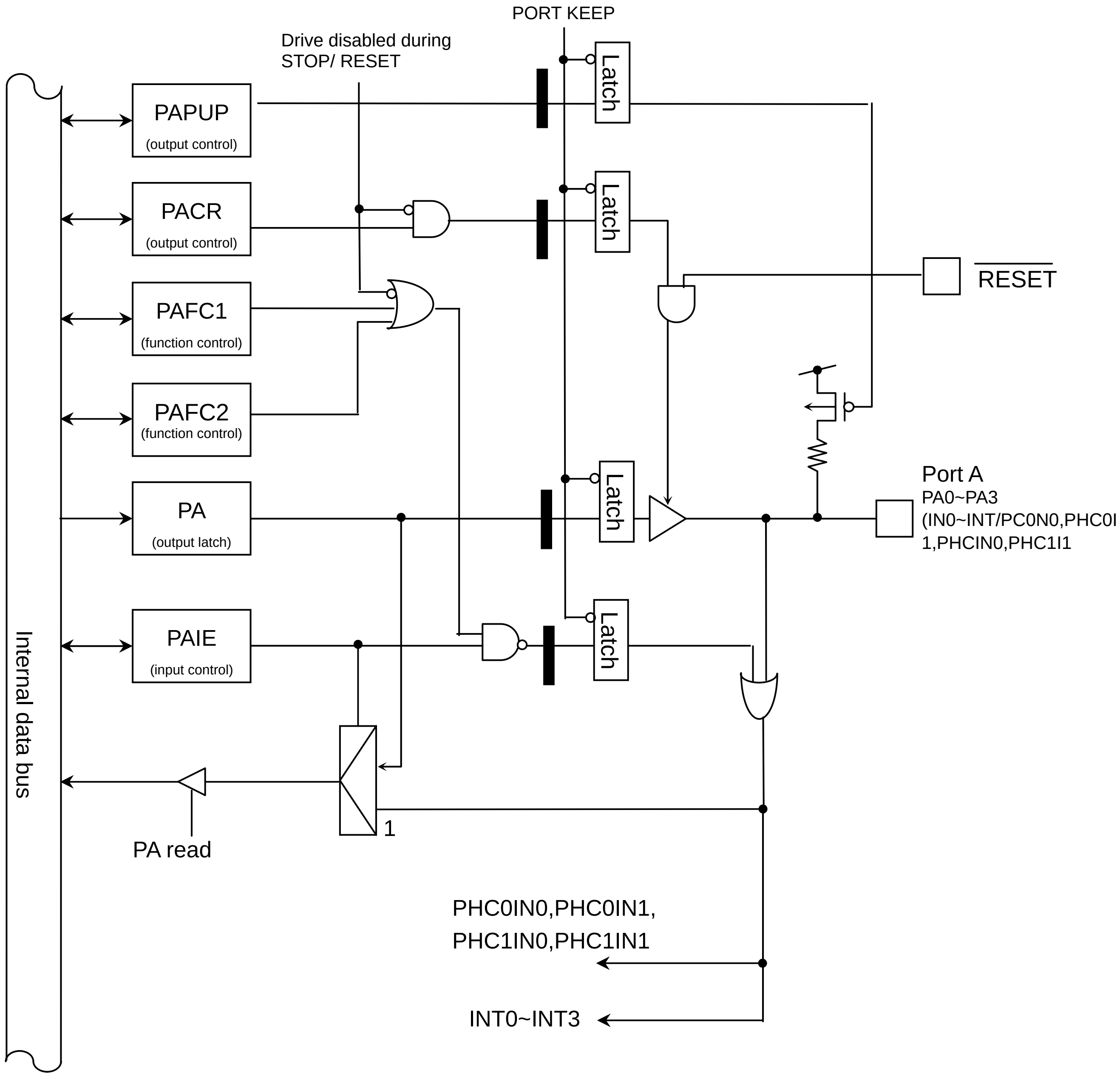


Fig. 7.27 Port A (PA0 through PA3)

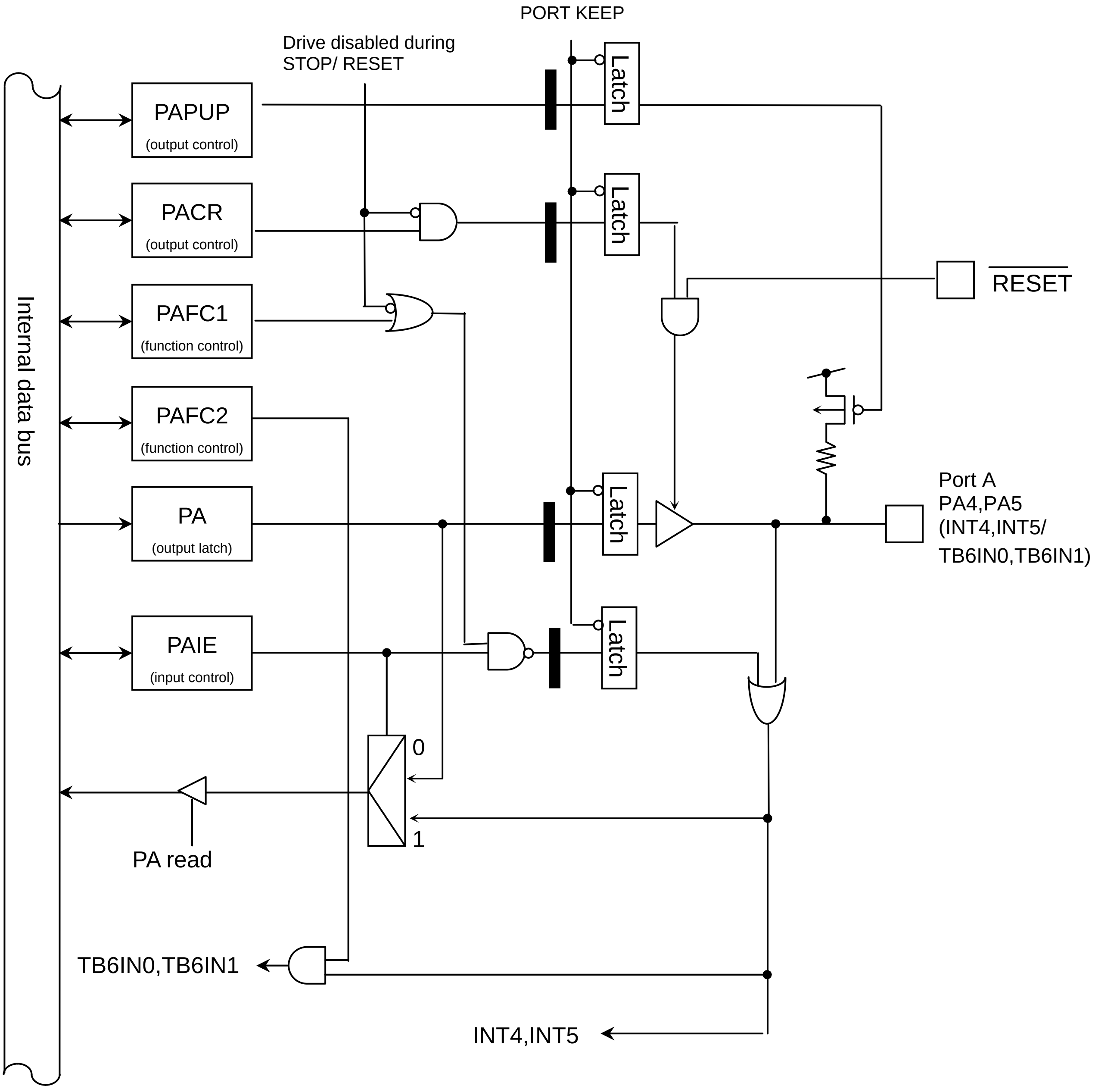


Fig. 7.28 Port A (PA4, PA5)

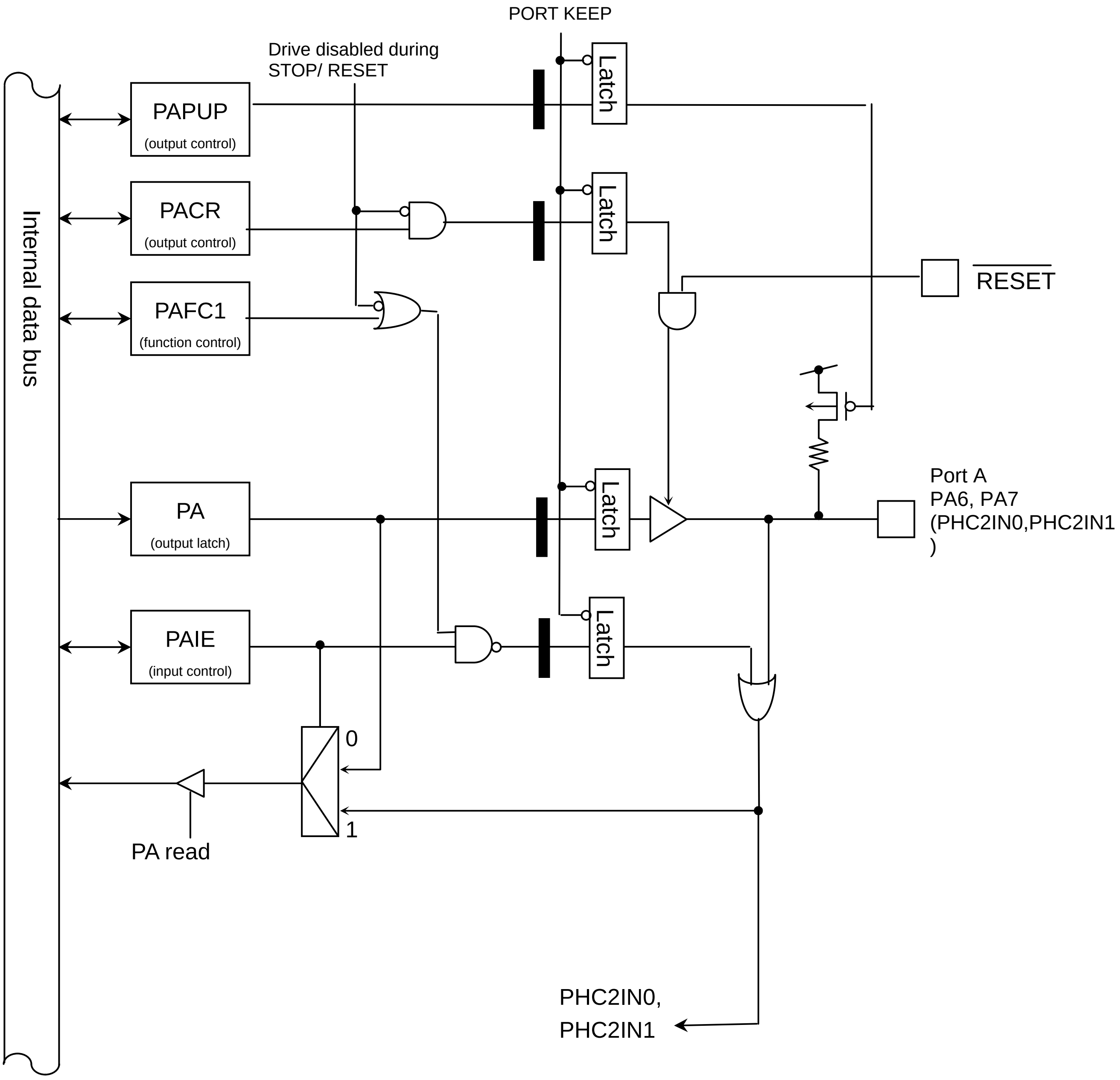


Fig. 7.29 Port A (PA6,PA7)

Port A register

PA
(0xFF00_4280)

	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
(0xFF00_4284)

	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 1

PAFC1
(0xFF00_4288)

	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:PHC2IN0	0:PORT 1:PHC2IN0	0:PORT 1:INT5	0:PORT 1:INT4	0:PORT 1:INT3	0:PORT 1:INT2	0:PORT 1:INT1	0:PORT 1:INT0

Port A function register 2

PAFC2
(0xFF00_428C)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	PA5F2	PA4F2	PA3F2	PA2F2	PA1F2	PA0F2
Read/Write	R/W							
After reset	0		0	0	0	0	0	0
Function	“0” is read.		0:PORT 1:TB6IN1	0:PORT 1:TB6IN0	0:PORT 1:PHC1IN1	0:PORT 1:PHC1IN0	0:PORT 1:PHC0IN1	0:PORT 1:PHC0IN0

Port A pull-up control register

PAPE
(0xFF00_42AC)

	7	6	5	4	3	2	1	0
Bit Symbol	PEA7	PEA6	PEA5	PEA4	PEA3	PEA2	PEA1	PEA0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port A Input enable control register

PAIE
(0xFF00_42B8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEA7	PIEA6	PIEA5	PIEA4	PIEA3	PIEA2	PIEA1	PIEA0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled	Input 0: disabled 1: Enabled

7.12 Port B (PB0 to 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 and the function registers PBFC1 and PBFC2. A reset allows all bits of the output latch PB to be set to "1," all bits of PBCR, PBFC1 and PBFC2 to be cleared to "0," and the port B to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PBIE to “1”.

Besides the input/output port function, the port B performs other functions: PB2, PB3 and PB7 output 16-bit timer, PB4 outputs HSIO data, PB5 inputs HSIO data, PB6 inputs and outputs HSIO HCLK or input HCTS, and PB0 and PB1 perform a two-phase pulse input counter function with a dial input function.

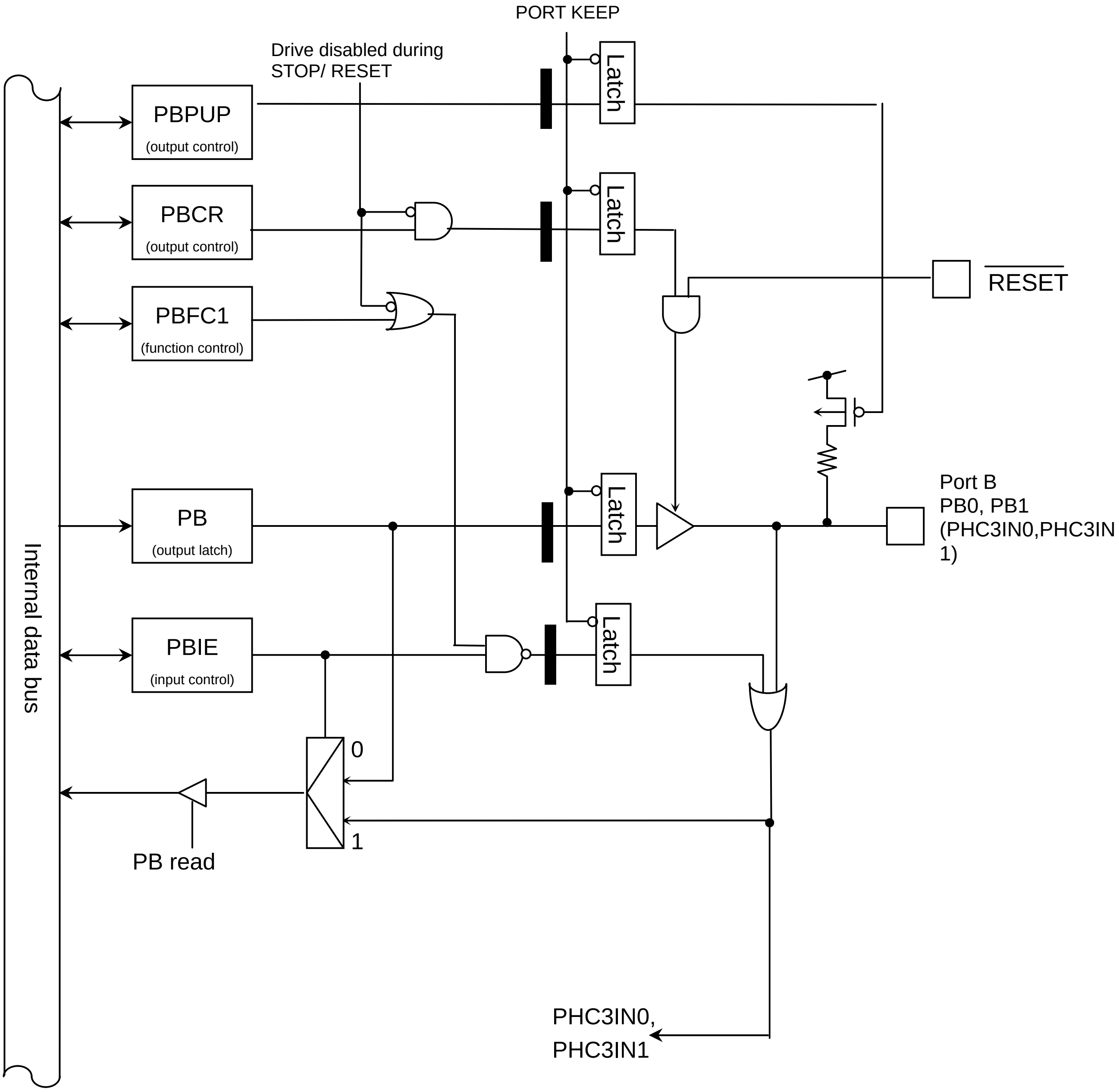


Fig. 7.30 Port B (PB0, PB1)

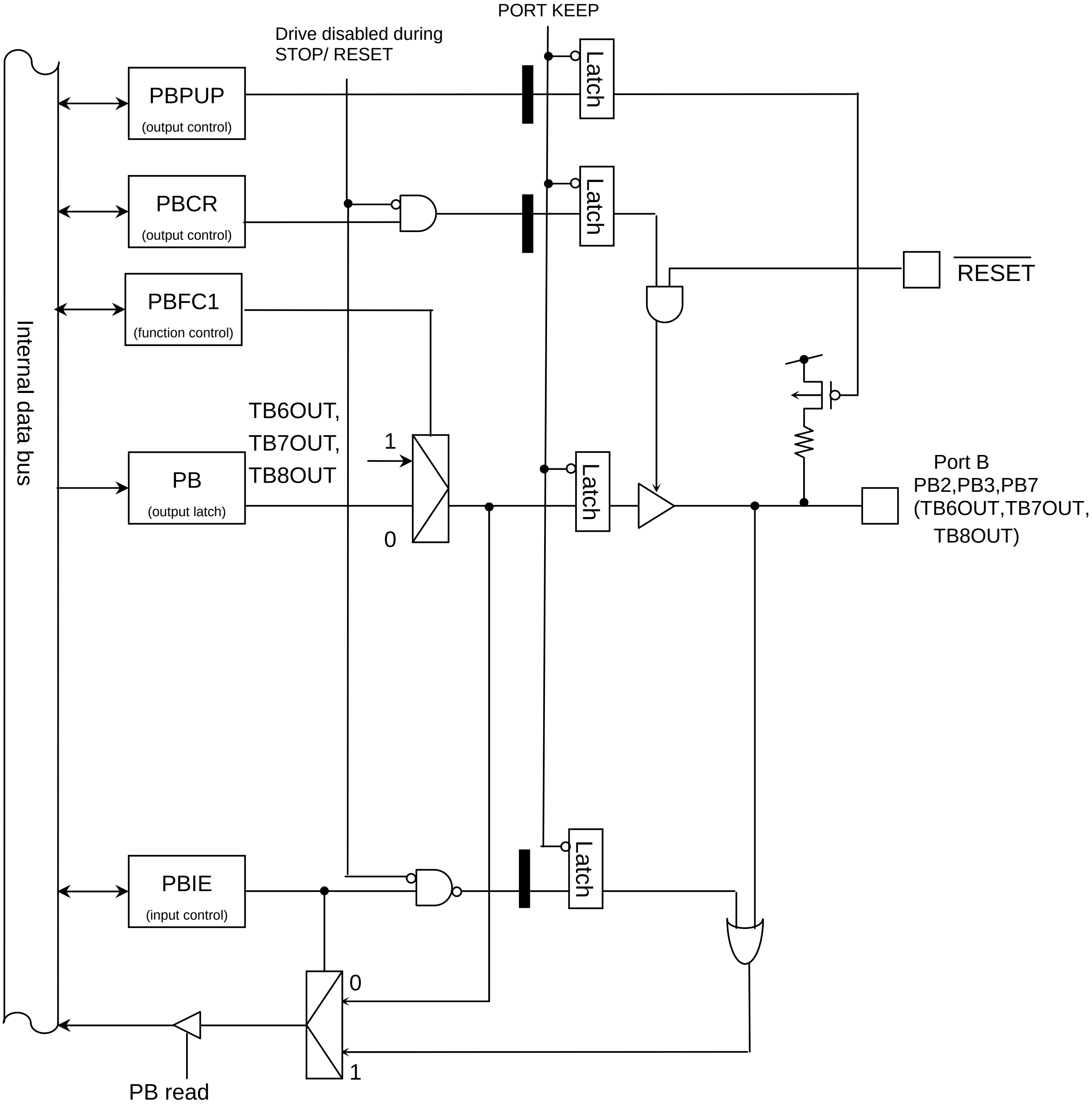


Fig. 7.31 Port B (PB2, PB3, PB7)

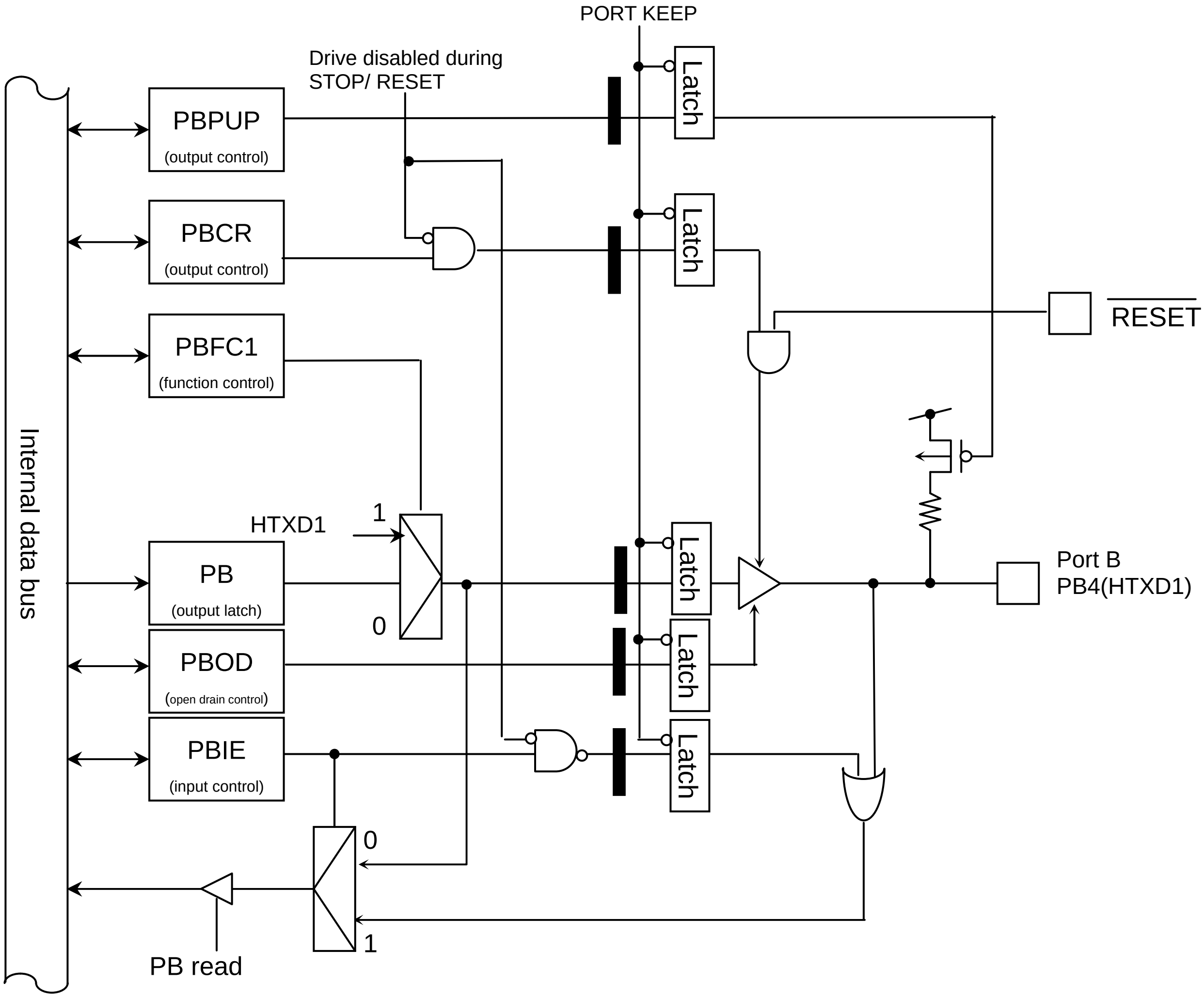


Fig. 7.32 Port B (PB4)

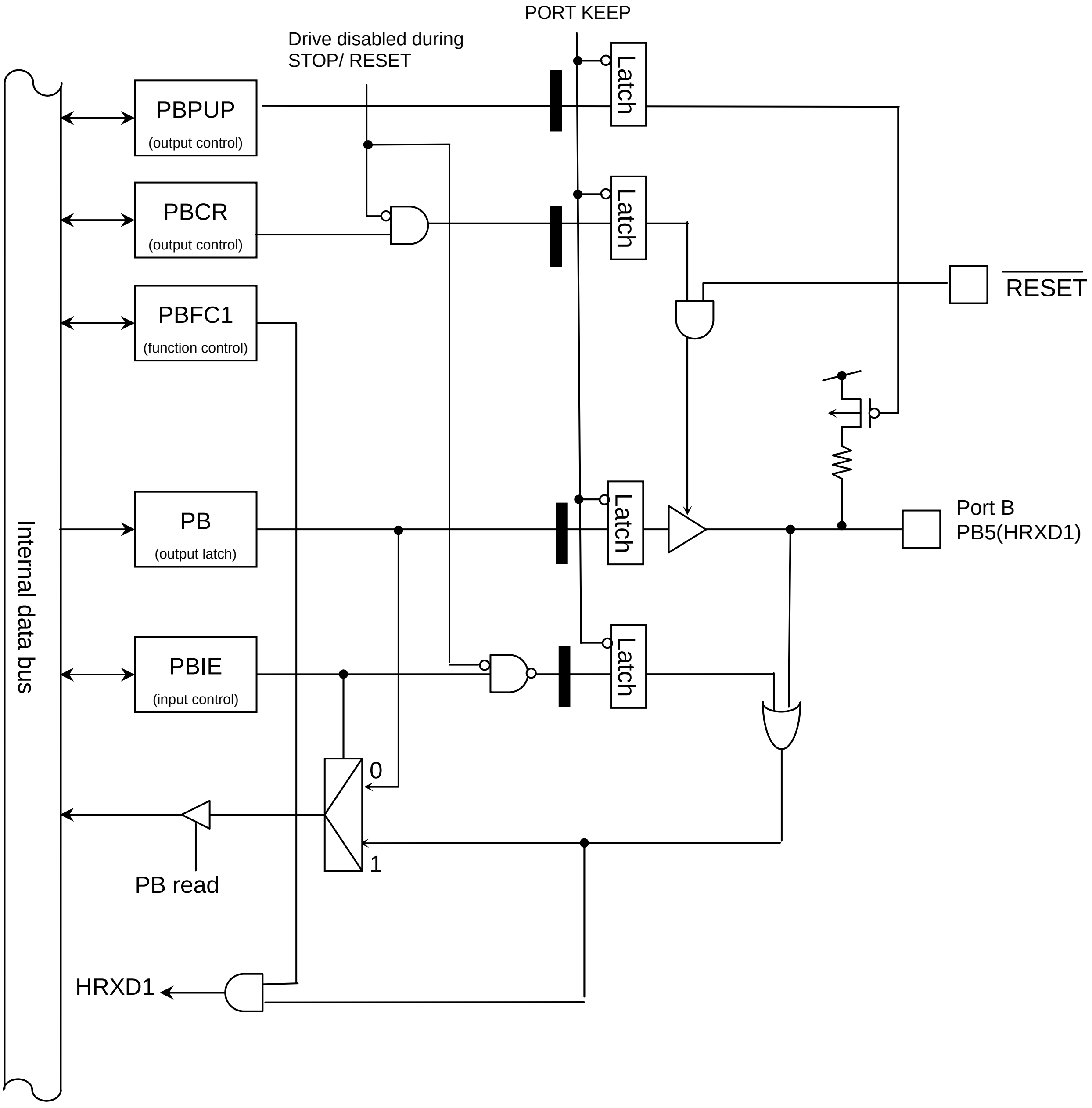


Fig. 7.33 Port B (PB5)

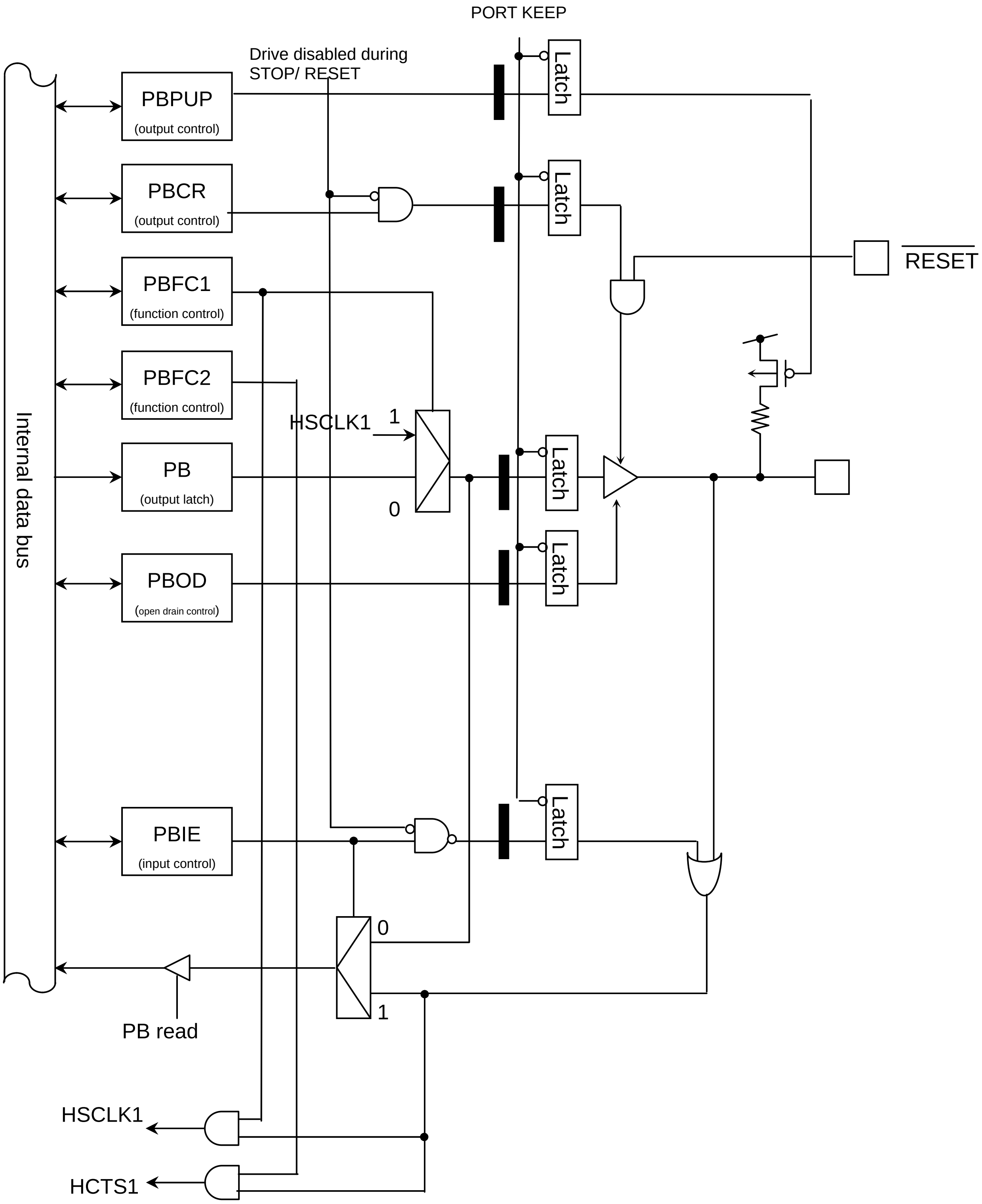


Fig. 7.34 Port B (PB6)

Port B register

PB
(0xFF00_42C0)

	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
(0xFF00_42C4)

	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 1

PBFC1
(0xFF00_42C8)

	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:TB8OUT	0:PORT 1: HSCLK0	0:PORT 1:HRXD1	0:PORT 1: HTXD1	0:PORT 1:TB7OUT	0:PORT 1:TB6OUT	0:PORT 1PHC3IN1	0:PORT 1: PHC3IN0

Port B function register 2

PBFC2
(0xFF00_42CC)

	7	6	5	4	3	2	1	0
Bit Symbol	–	PB6F2	–	–	–	–	–	–
Read/Write	R	R/W	R					
After reset	0	0	0					
Function	“0” is read.	0:PORT 1:/HCTS0	“0” is read.					

Port B open drain control register

PBODE
(0xFF00_42E8)

	7	6	5	4	3	2	1	0
Bit Symbol	–	PB6ODE	–	PB4ODE	–	–	–	–
Read/Write	R	R/W	R	R/W	R			
After reset	0	0	0	0	0			
Function	“0” is read.	0: CMOS 1: Open drain	“0” is read.	0: CMOS 1: Open drain	“0” is read.			

Port B pull-up control register

PBPUP
(0xFF00_42EC)

	7	6	5	4	3	2	1	0
Bit Symbol	PEB7	PEB6	PEB5	PEB4	PEB3	PEB2	PEB1	PEB0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

7.13 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 and the function registers PCFC1 and PCFC2. A reset allows all bits of the output latch PC to be set to "1," all bits of PCCR, PCFC1 and PCFC2 to be cleared to "0," and the port C to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PCIE to “1”.

Besides the input/output port function, the port C performs other functions: PC0 inputs external clock sources into a 32-bit time base timer and inputs the key-on wake-up, PC1 through PC3 and PC7 perform the 32-bit compare output function, and PC4 through PC6 input and output SBI.

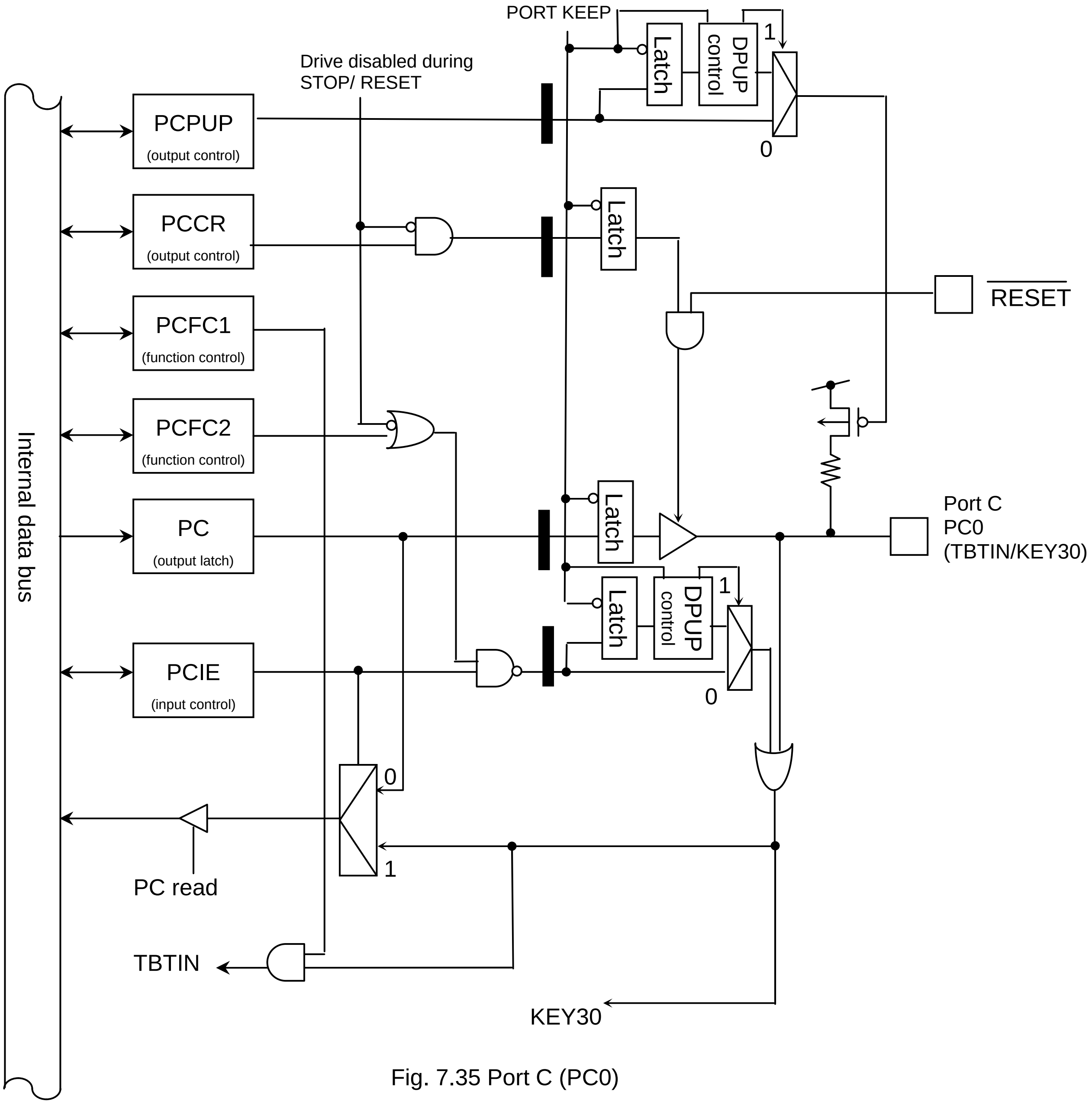


Fig. 7.35 Port C (PC0)

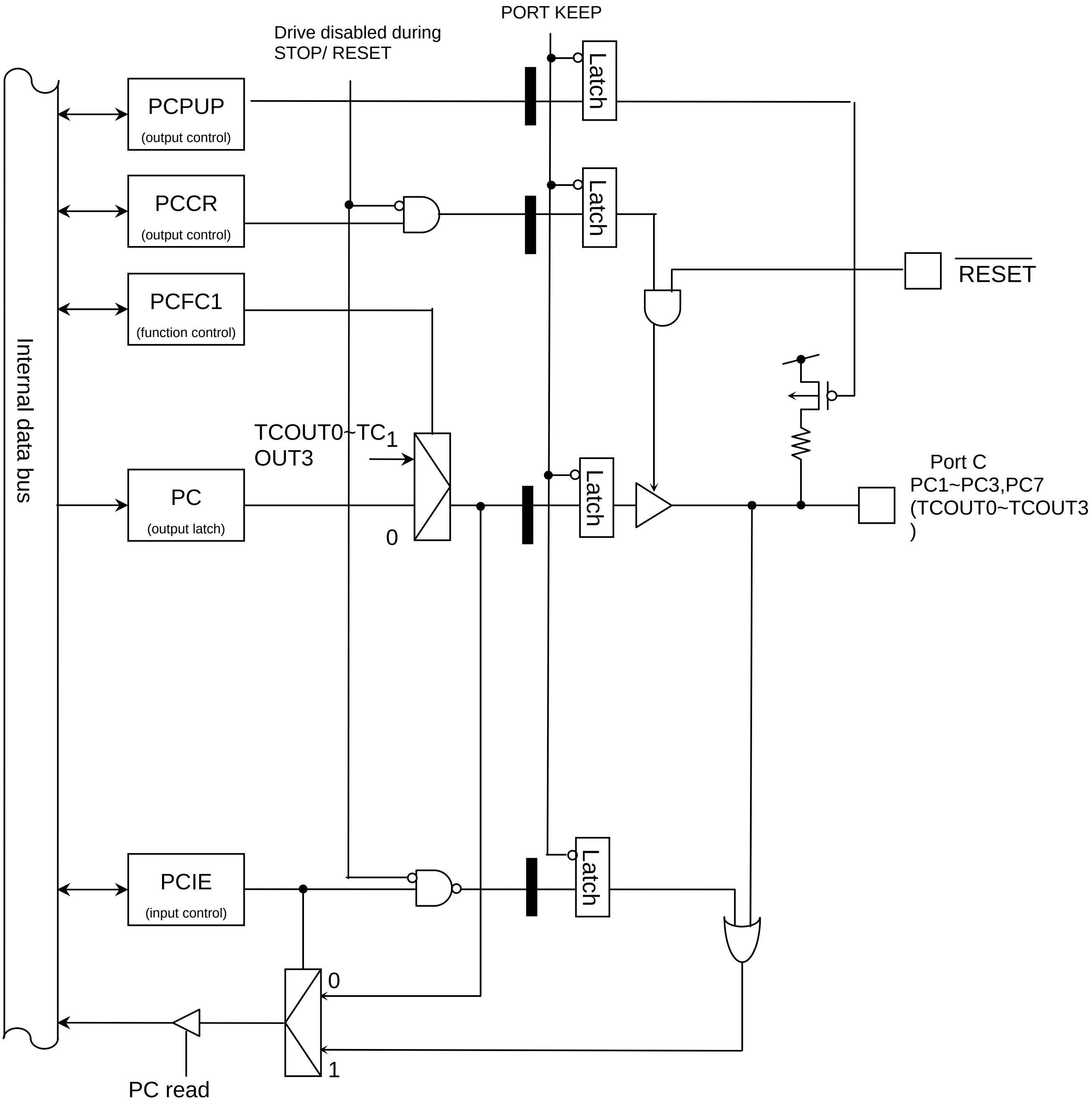


Fig. 7.36 Port C (PC1 through PC3, PC7)

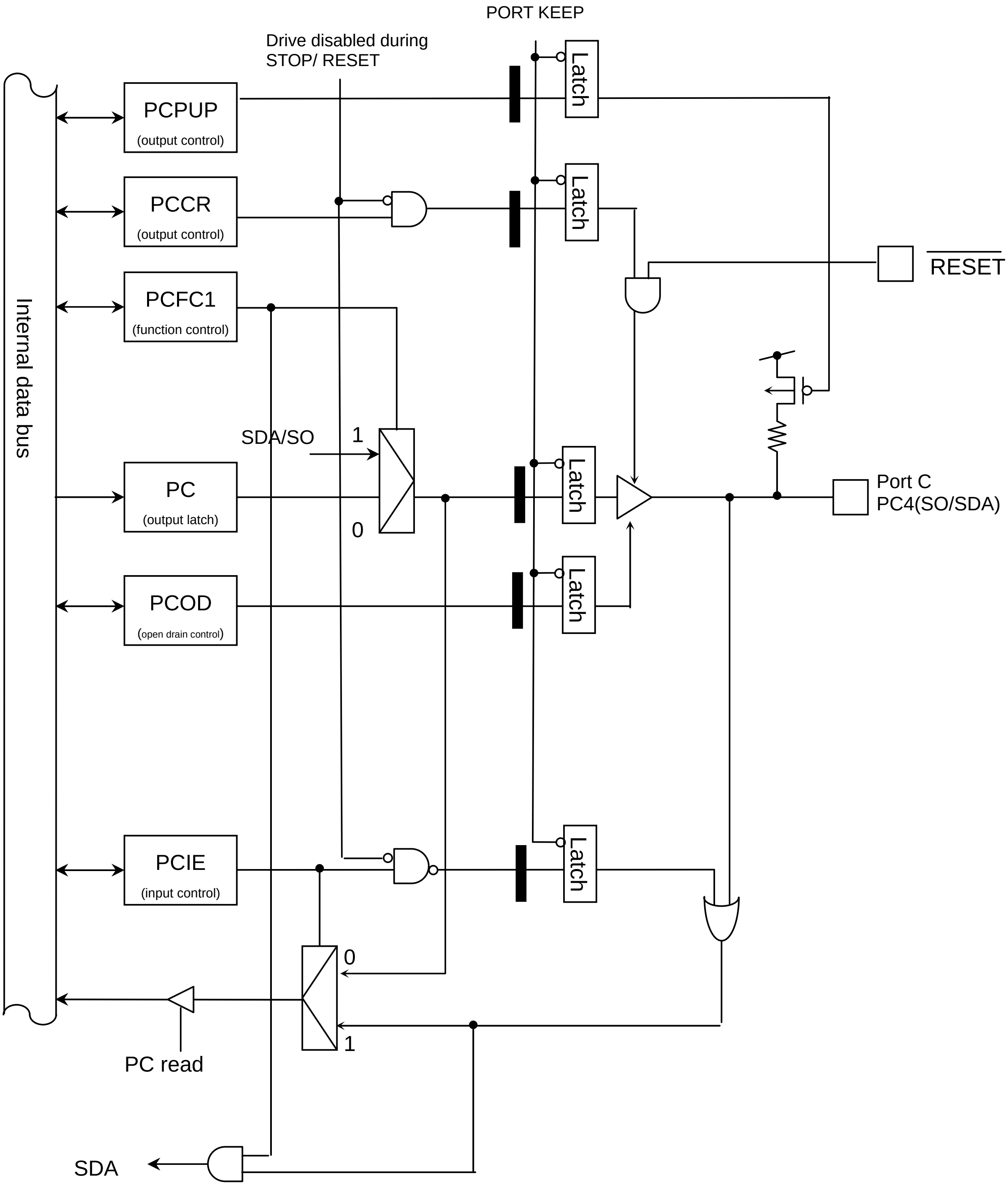


Fig. 7.37 Port C (PC4)

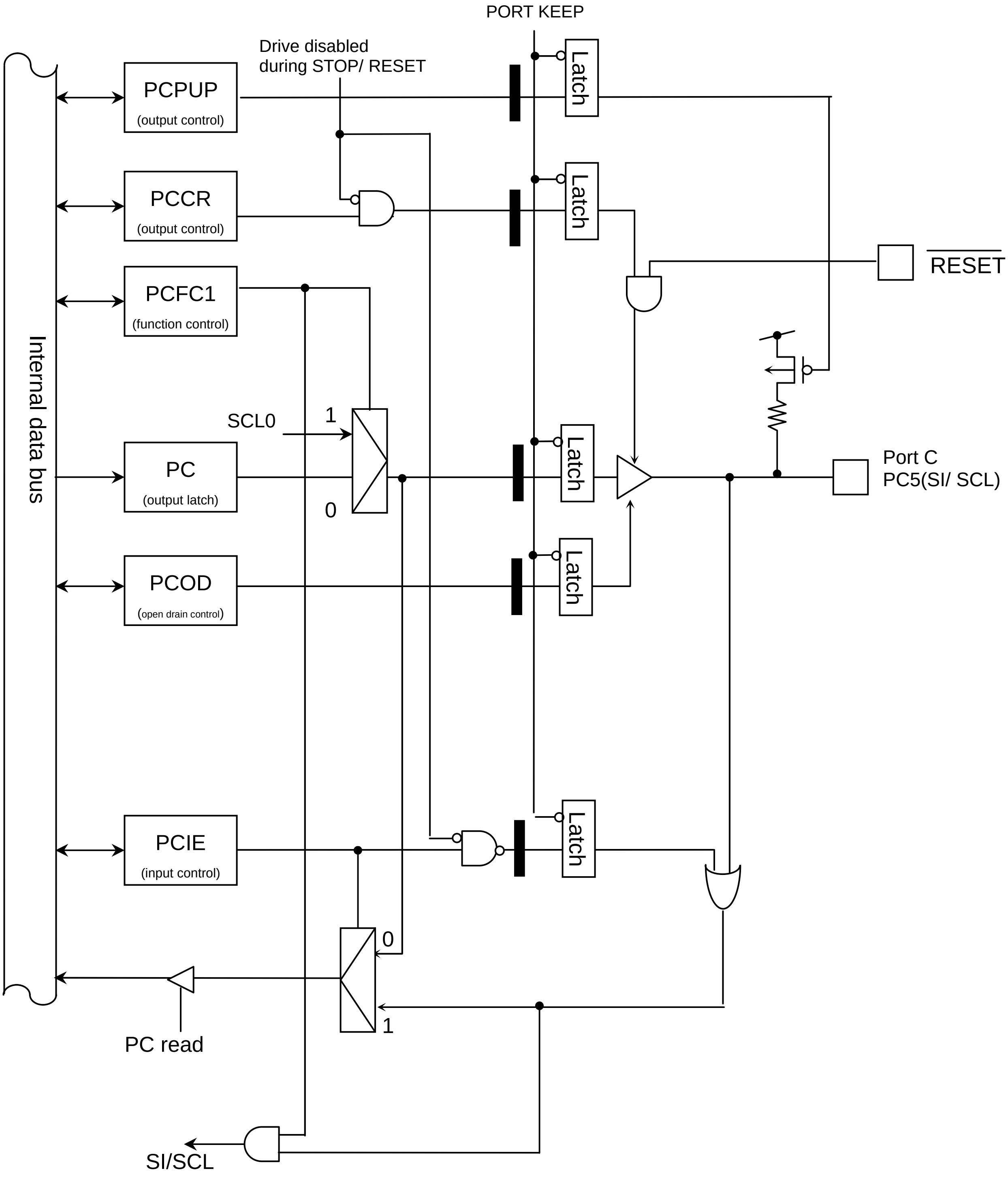


Fig. 7.38 Port C(PC5)

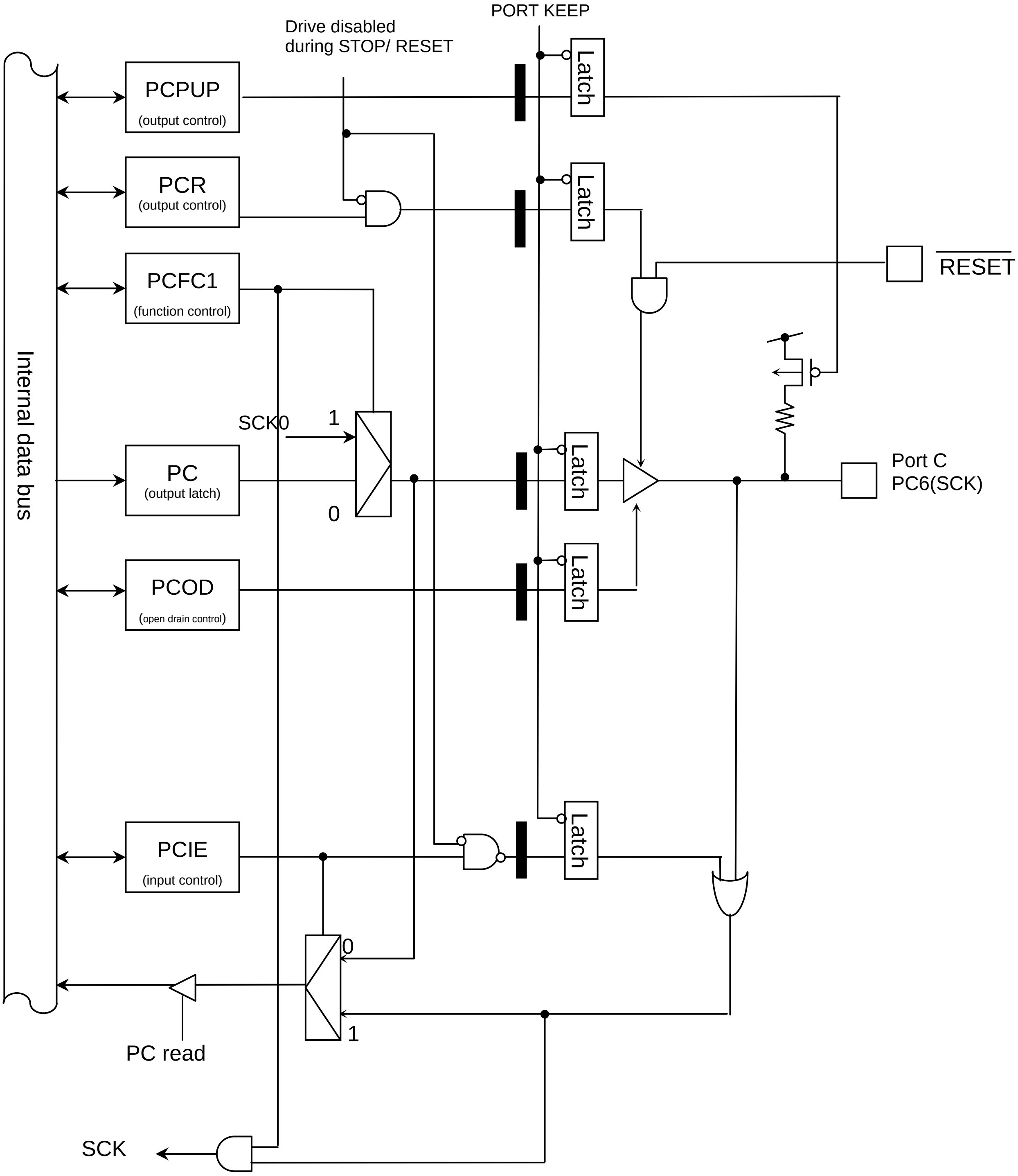


Fig. 7.39 Port C (PC6)

Port C register

PC
(0xFF00_4300)

	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
(0xFF00_4304)

	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 1

PCFC1
(0xFF00_4308)

	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:TCOUT3	0:PORT 1:SCK	0:PORT 1:SI /SCL	0:PORT 1:SO /SDA	0:PORT 1:TCOUT2	0:PORT 1:TCOUT1	0:PORT 1:TCOUT0	0:PORT 1: TBTIN

Port C function register 2

PCFC2
(0xFF00_430C)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	–	–	–	–	PC0F2
Read/Write	R							R/W
After reset	0							0
Function	"0" is read.							0:PORT 1: /KEY30

Port C open drain control register

PCODE
(0xFF00_4328)

	7	6	5	4	3	2	1	0
Bit Symbol	–	PC6ODE	PC5ODE	PC4ODE	–	–	–	–
Read/Write	R	R/W			R			
After reset	0	0	0	0	0			
Function	"0" is read.	0: CMOS 1: Open drain	0: CMOS 1: Open drain	0: CMOS 1: Open drain	"0" is read.			

Port C pull-up control register

PCPE
(0xFF00_432C)

	7	6	5	4	3	2	1	0
Bit Symbol	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port C input enable control register

PCIE
(0xFF00_4338)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEC7	PIEC6	PIEC5	PIEC4	PIEC3	PIEC2	PIEC1	PIEC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.14 Port D (PD0 to PD7)

The port D is a general-purpose, 7-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PDCR and the function registers PDFC1 and PDFC2. A reset allows all bits of the output latch PD to be set to "1," all bits of PDCR, PDFC1 and PDFC2 to be cleared to "0," and the port D to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PDIE to “1”.

Besides the input port function, the port D performs other functions: PD0 outputs HSIO data, PD1 inputs HSIO data, PD2 inputs and outputs HSIO HCLK or inputs HCTS, PD3, PD4 and PD5 output a 16-bit timer, and PD6 inputs the key-on wake-up, PD6 and PD7 input A/D triggers into the A/D converter.

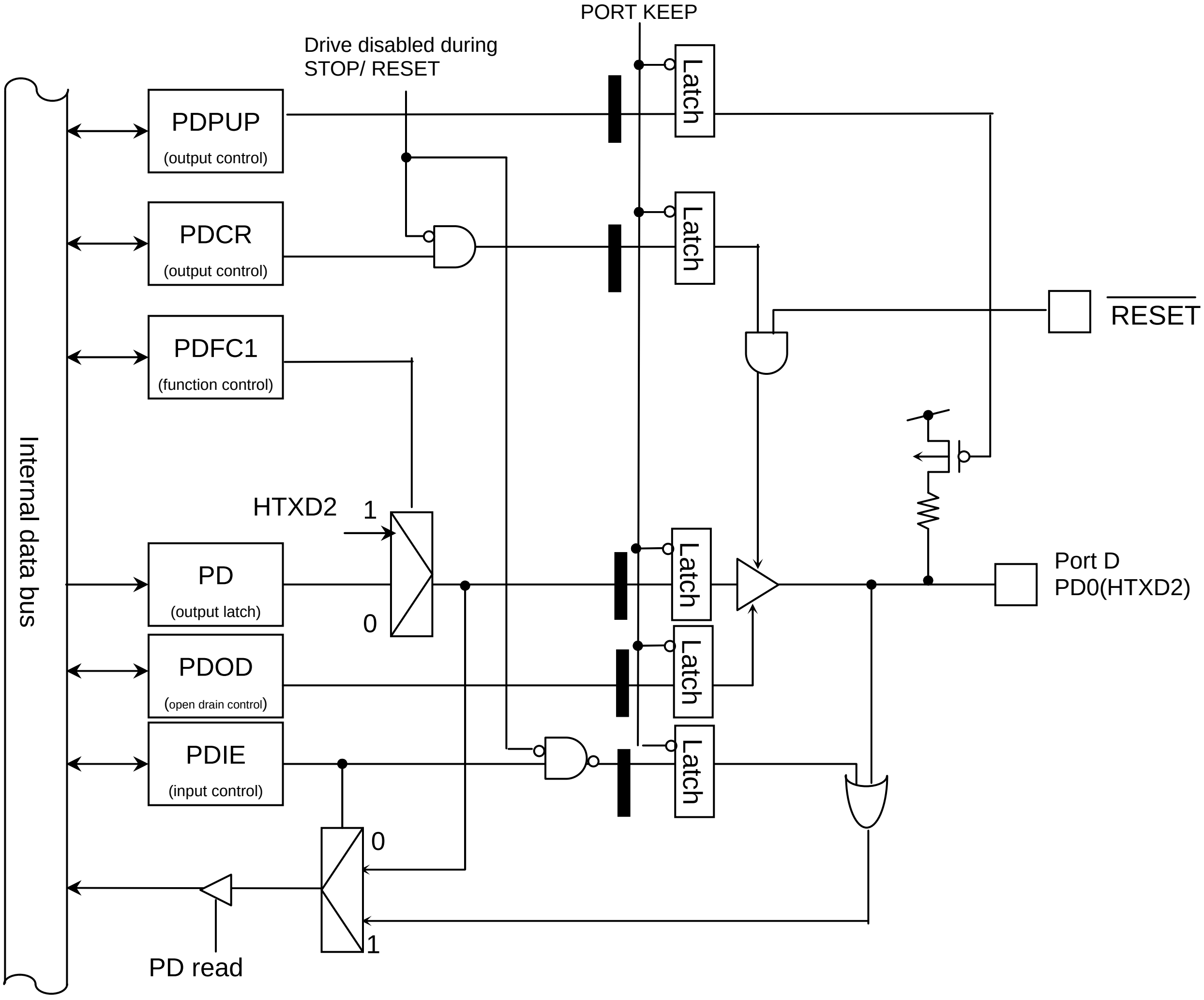


Fig. 7.40 Port D (PD0)

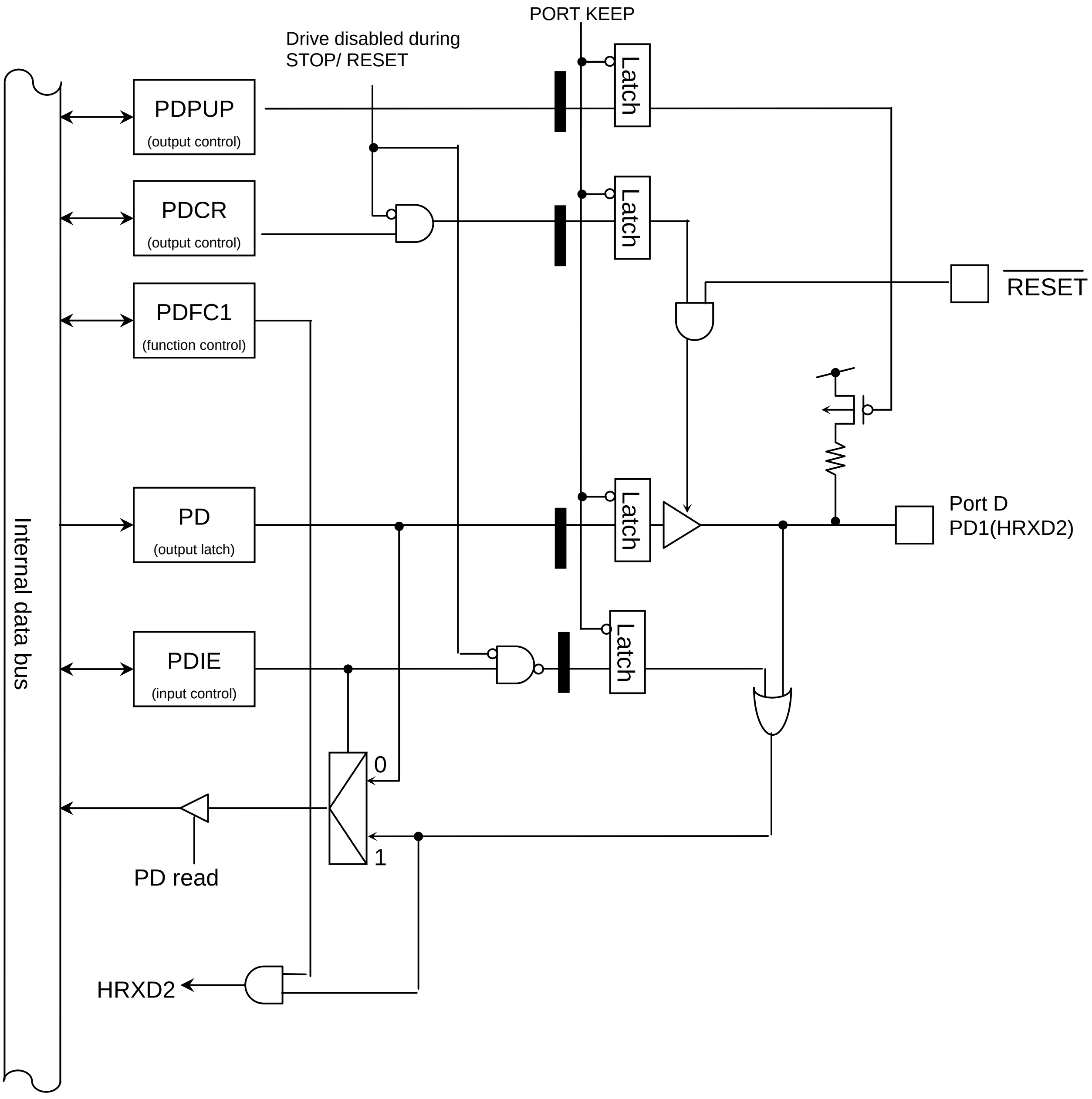


Fig. 7.41 Port D (PD1)

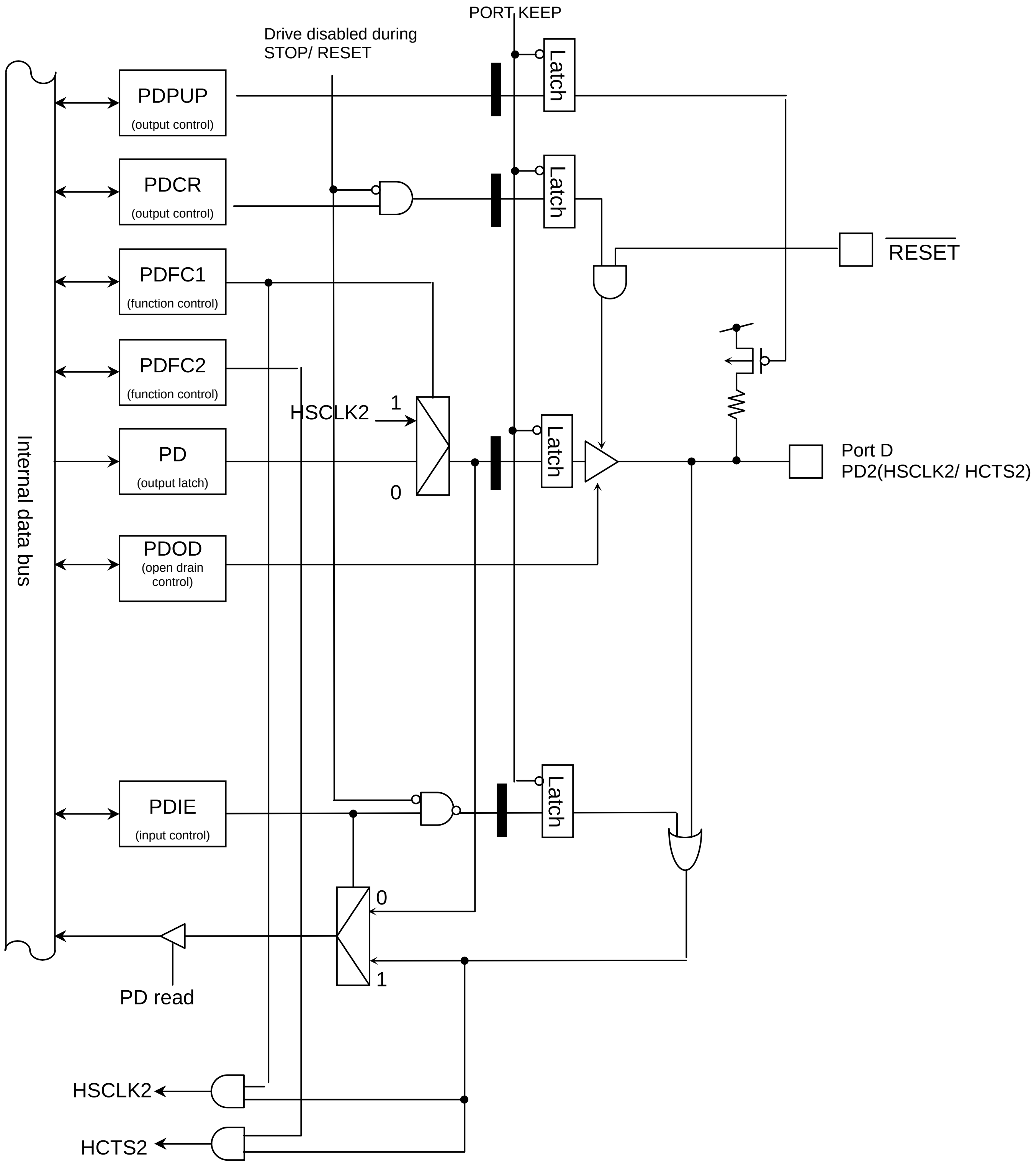


Fig. 7.42 Port D (PD2)

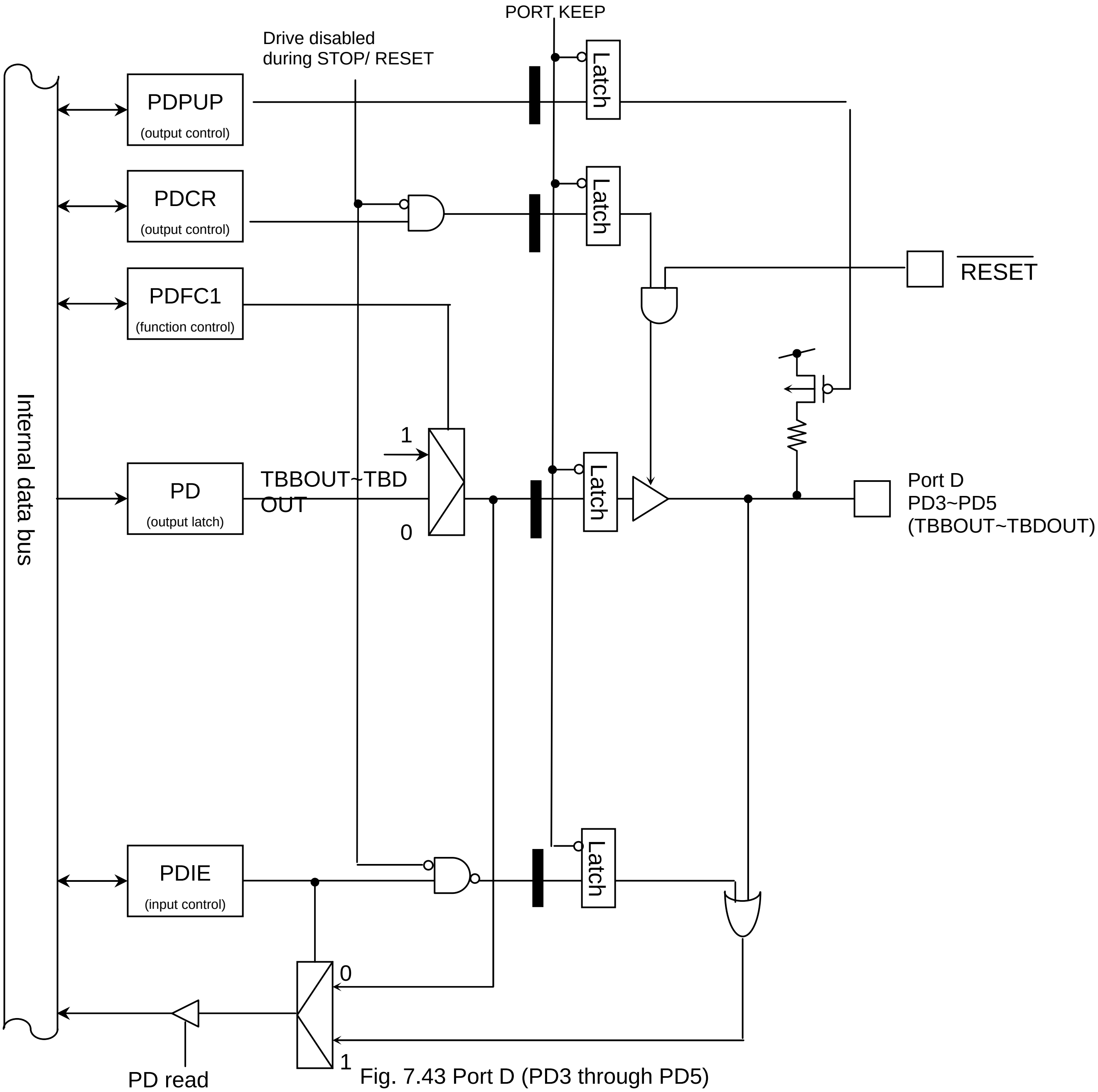


Fig. 7.43 Port D (PD3 through PD5)

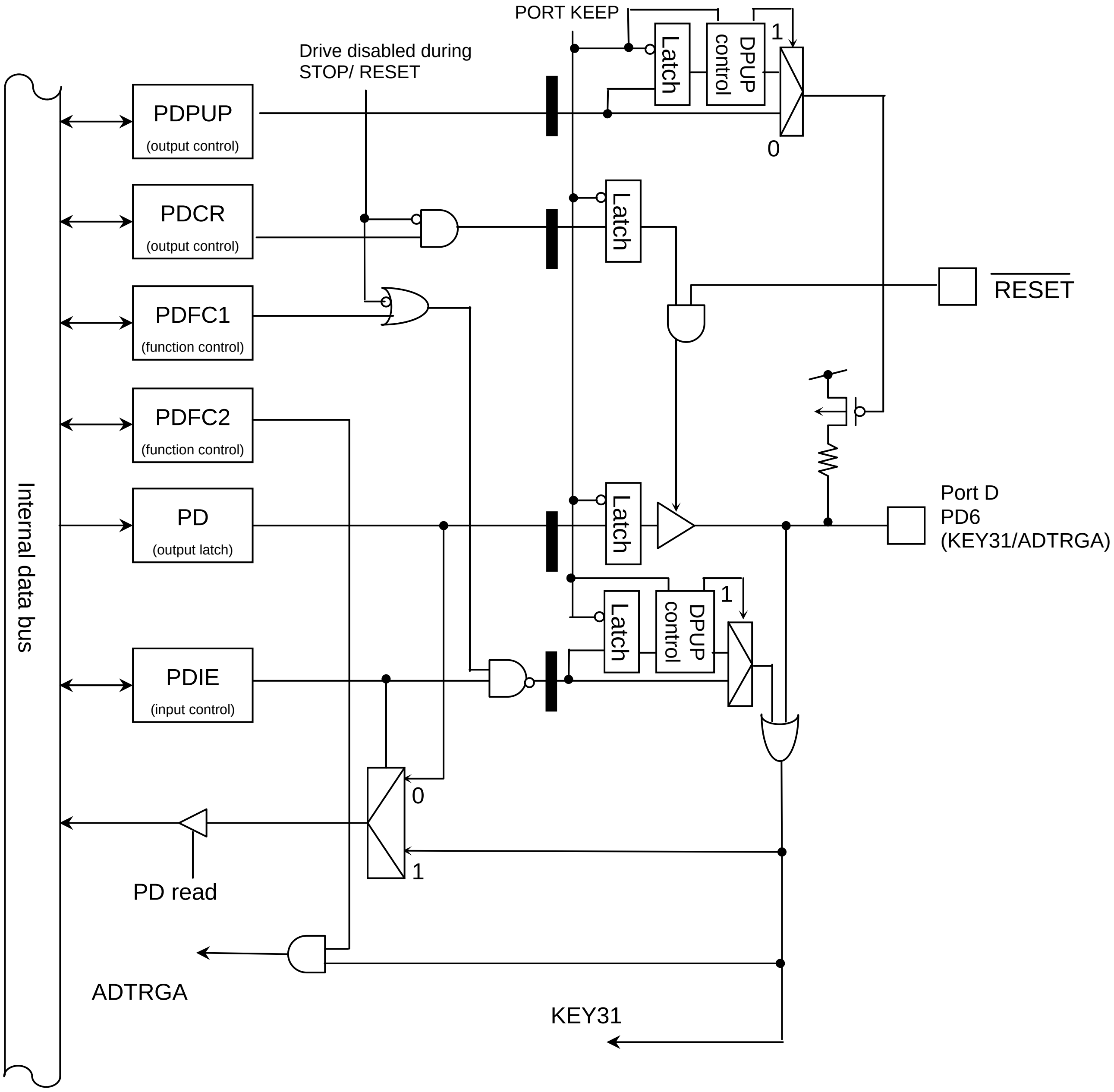


Fig. 7.44 Port D (PD6)

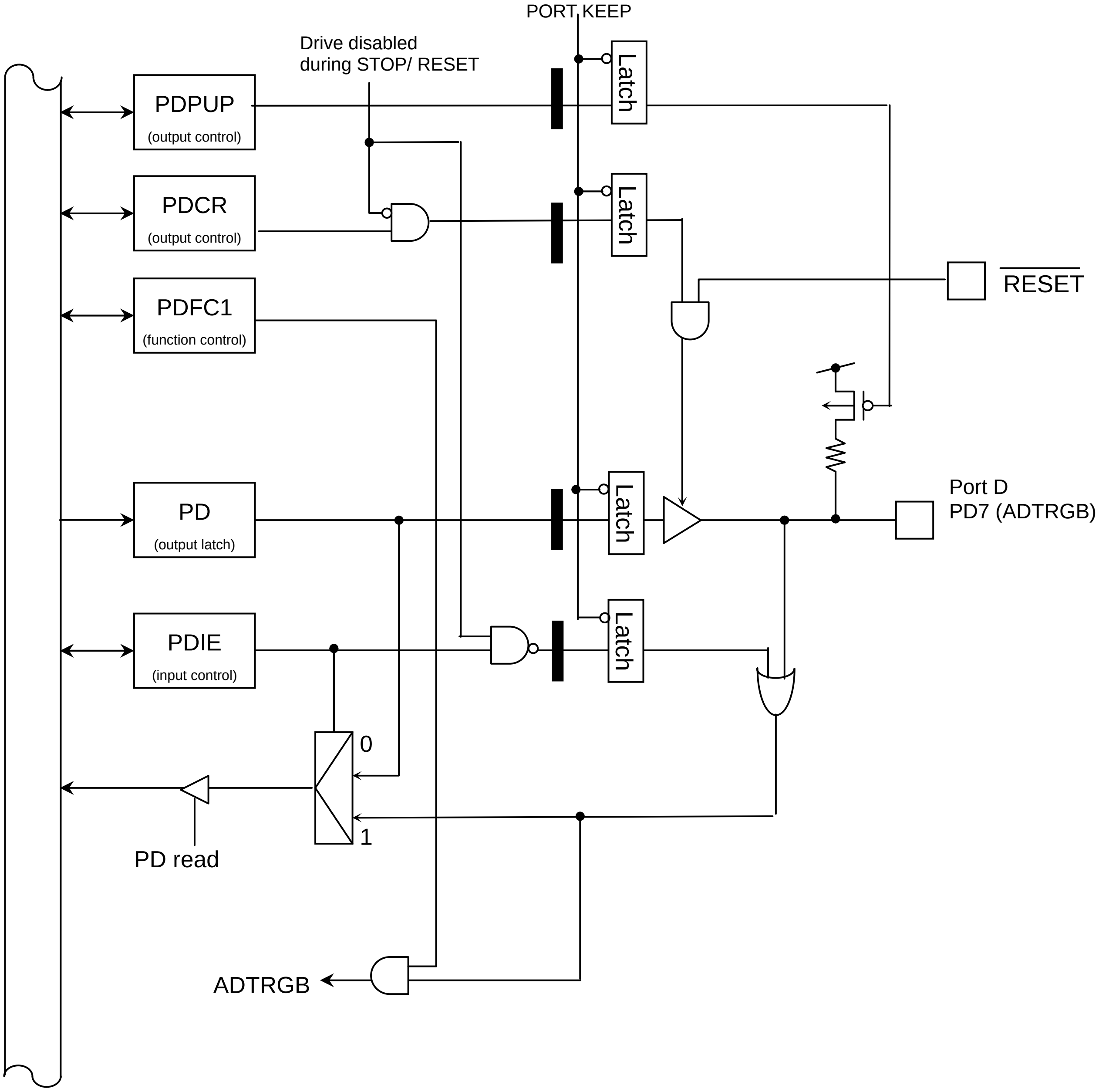


Fig. 7.45 Port D (PD7)

Port D register

PD (0xFF00_4340)

	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 (0xFF00_4344)

	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 1

PDFC1 (0xFF00_4348)

	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:ADTRG	0:PORT 1:/KEY31	0:PORT 1:TBDOUT	0:PORT 1:TBCOUT	0:PORT 1:TBBOUT	0:PORT 1:HSCLK2	0:PORT 1:HRXD2	0:PORT 1:HTXD2

Port D function register 2

PDFC2 (0xFF00_434C)

	7	6	5	4	3	2	1	0
Bit Symbol	–	PD6F2	–	–	–	PD2F2	–	–
Read/Write	R	R/W	R			R/W	R	
After reset	0	0	0			0	0	
Function	"0" is read.	0:PORT 1:ADTRG	"0" is read.			0:PORT 1: /HCTS2	"0" is read.	

Port D open drain control register

PDODE (0xFF00_4368)

	7	6	5	4	3	2	1	0
Bit Symbol	–	–	–	–	–	PD2ODE	–	PD0ODE
Read/Write	R					R/W	R	R/W
After reset	0					0	0	0
Function	"0" is read.					0: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain

Port D pull-up control register

PDPUP (0xFF00_436C)

	7	6	5	4	3	2	1	0
Bit Symbol	PED7	PED6	PED5	PED4	PED3	PED2	PED1	PED0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port D Input enable control register

PDIE (0xFF00_4378)

	7	6	5	4	3	2	1	0
Bit Symbol	PIED7	PIED6	PIED5	PIED4	PIED3	PIED2	PIED1	PIED0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.15 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 and the function register PEFC1. A reset allows all bits of the output latch PE to be set to "1," all bits of PECR and PEFC to be cleared to "0," and the port E to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PEIE to “1”.

Besides the input/output port function, the port E performs the key-on wake-up input function.

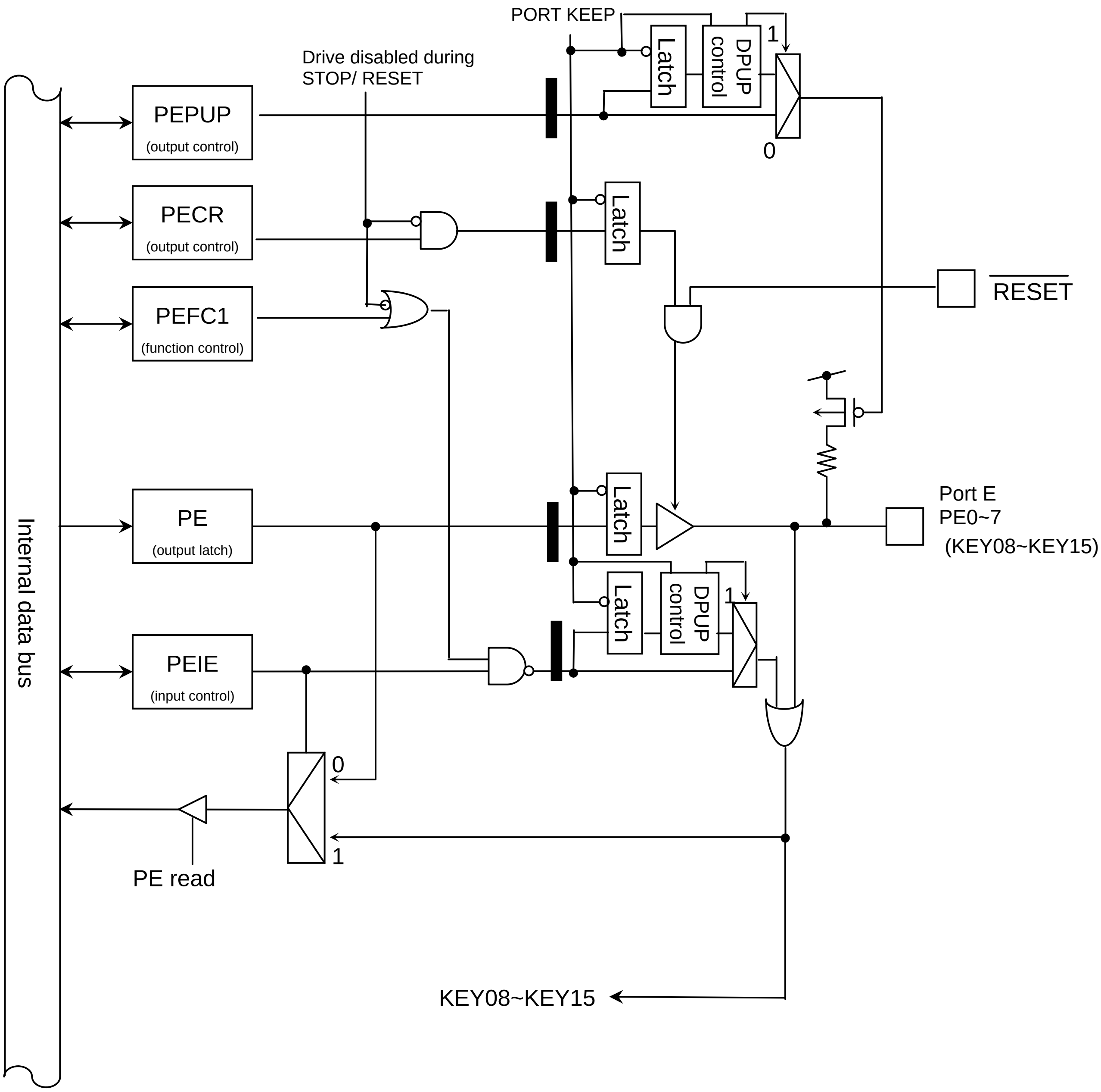


Fig. 7.46 Port E (PE0 through PE7)

Port E register

PE (0xFF00_4380)

	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 (0xFF00_4384)

	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 1

PEFC1 (0xFF00_4388)

	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:KEY15	0:PORT 1:KEY14	0:PORT 1:KEY13	0:PORT 1:KEY12	0:PORT 1:KEY11	0:PORT 1:KEY10	0:PORT 1:KEY09	0:PORT 1:KEY08

Port E pull-up control register

PEPUP (0xFF00_43AC)

	7	6	5	4	3	2	1	0
Bit Symbol	PEE7	PEE6	PEE5	PEE4	PEE3	PEE2	PEE1	PEE0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port E Input enable control register

PEIE (0xFF00_43B8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEE7	PIEE6	PIEE5	PIEE4	PIEE3	PIEE2	PIEE1	PIEE0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.16 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 and the function registers PFFC1 and PFFC2. A reset allows all bits of the output latch PF to be set to "1," all bits of PFCR, PFFC1 and PFFC2 to be cleared to "0," and the port F to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PFIE to “1”.

Besides the input/output port function, the port F inputs external clock source of 32-bit time base timer and performs the key-on wake-up input function, PF1 through PF3 and PC7 perform a 32-bit timer compare output function and PC4 through PC6 perform SBI input/ output function..

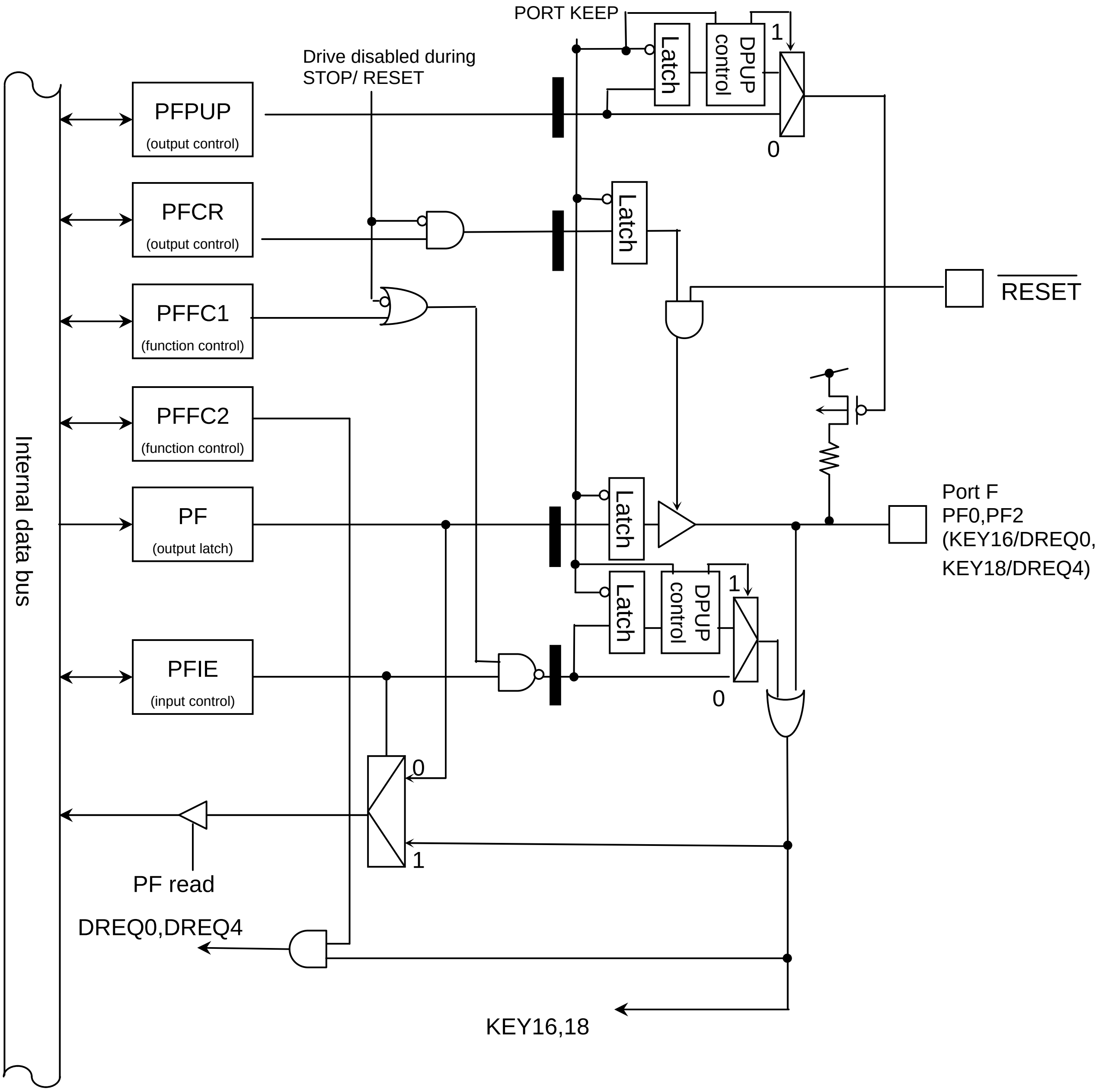


Fig. 7.47 Port F (PF0, PF2)

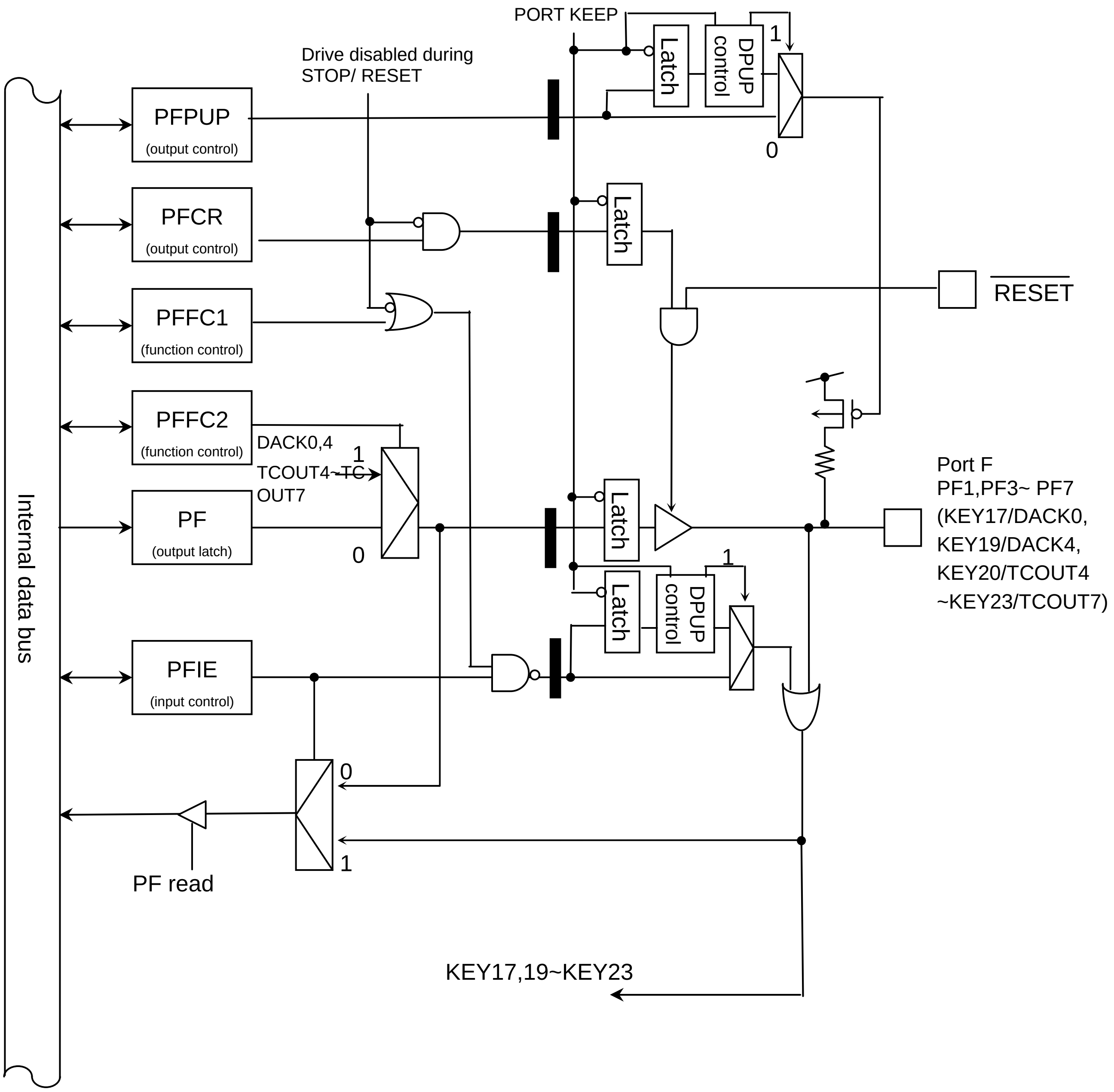


Fig. 7.48 Port F (PF1, PF3 through PF7)

Port F register

PF
(0xFF00_43C0)

	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
(0xFF00_43C4)

	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 1

PFFC1
(0xFF00_43C8)

	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:/ KEY23	0:PORT 1:/ KEY22	0:PORT 1:/KEY21	0:PORT 1:/KEY20	0:PORT 1:/KEY19	0:PORT 1:/KEY18	0:PORT 1:/KEY17	0:PORT 1:/KEY16

Port F function register 2

PFFC2
(0xFF00_43CC)

	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:TCOUT7	0:PORT 1:TCOUT6	0:PORT 1:TCOUT5	0:PORT 1:TCOUT4	0:PORT 1:DACK4	0:PORT 1:DREQ4	0:PORT 1:DACK0	0:PORT 1:DREQ0

Port F pull-up control register

PFPE
(0xFF00_43EC)

	7	6	5	4	3	2	1	0
Bit Symbol	PEF7	PEF6	PEF5	PEF4	PEF3	PEF2	PEF1	PEF0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port F input enable control register

PFIE
(0xFF00_43F8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEF7	PIEF6	PIEF5	PIEF4	PIEF3	PIEF2	PIEF1	PIEF0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.17 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 and the function register PGFC1. A reset allows all bits of the output latch PG to set to "1," all bits of PGCR and PGFC1 to be cleared to "0," and the port G to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PGIE to “1”.

Besides the input/output port function, the port G performs the key-on wake-up input function.

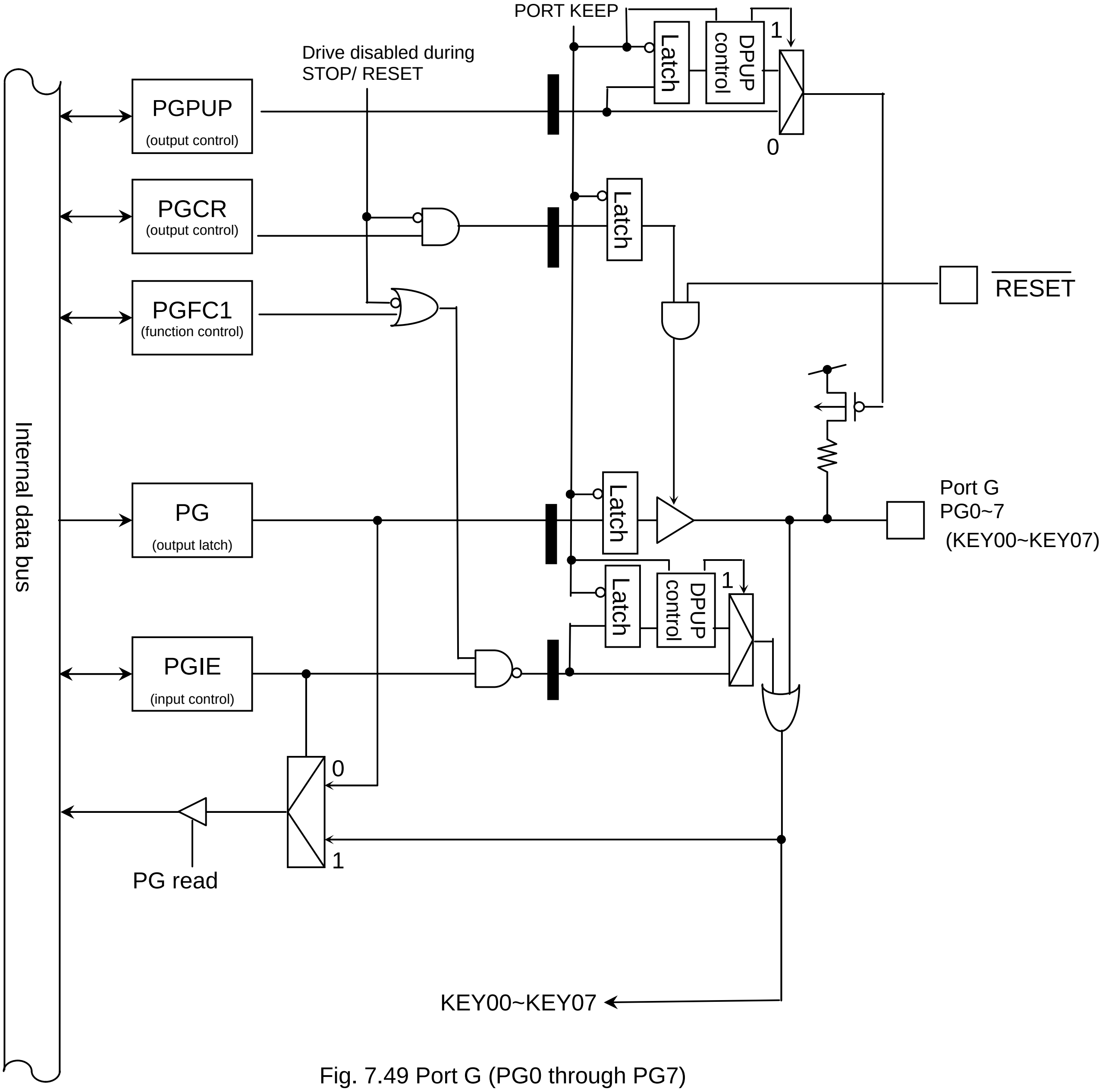


Fig. 7.49 Port G (PG0 through PG7)

Port G register

PG
(0xFF00_4400)

	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
(0xFF00_4404)

	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 1

PGFC1
(0xFF00_4408)

	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:KEY07	0:PORT 1:KEY06	0:PORT 1:KEY05	0:PORT 1:KEY04	0:PORT 1:KEY03	0:PORT 1:KEY02	0:PORT 1:KEY01	0:PORT 1:KEY00

Port G pull-up control register

PGPUP
(0xFF00_442C)

	7	6	5	4	3	2	1	0
Bit Symbol	PEG7	PEG6	PEG5	PEG4	PEG3	PEG2	PEG1	PEG0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port G input enable control register

PGIE
(0xFF00_4438)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEG7	PIEG6	PIEG5	PIEG4	PIEG3	PIEG2	PIEG1	PIEG0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.18 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 and the function registers PHFC1 and PHFC2. A reset allows all bits of the output latch PH to be set to "1," all bits of PHCR to be cleared to "0," and the port H to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PHIE to “1”.

Besides the port function, the port H inputs external interrupts and a 16-bit timer.

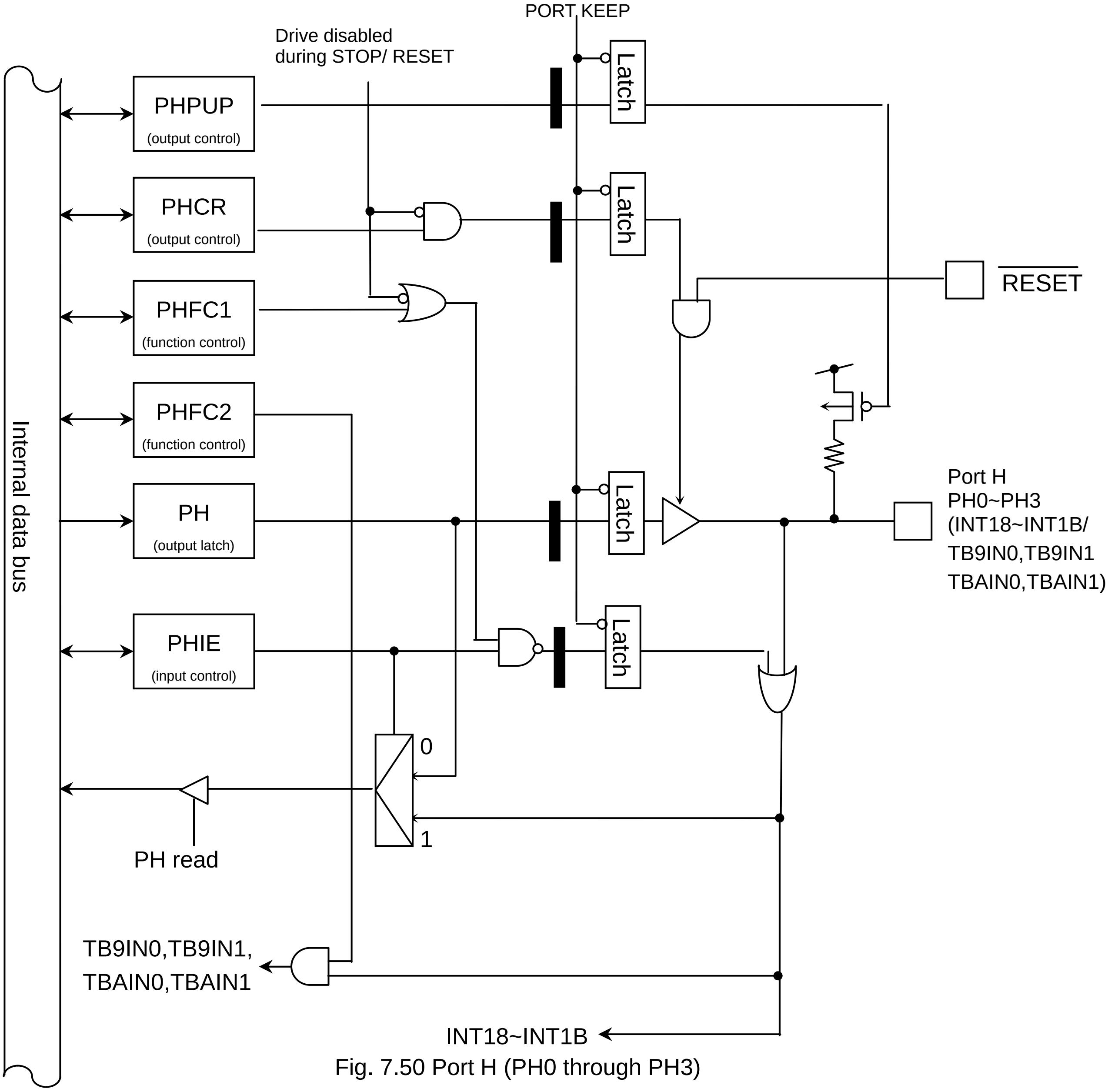


Fig. 7.50 Port H (PH0 through PH3)

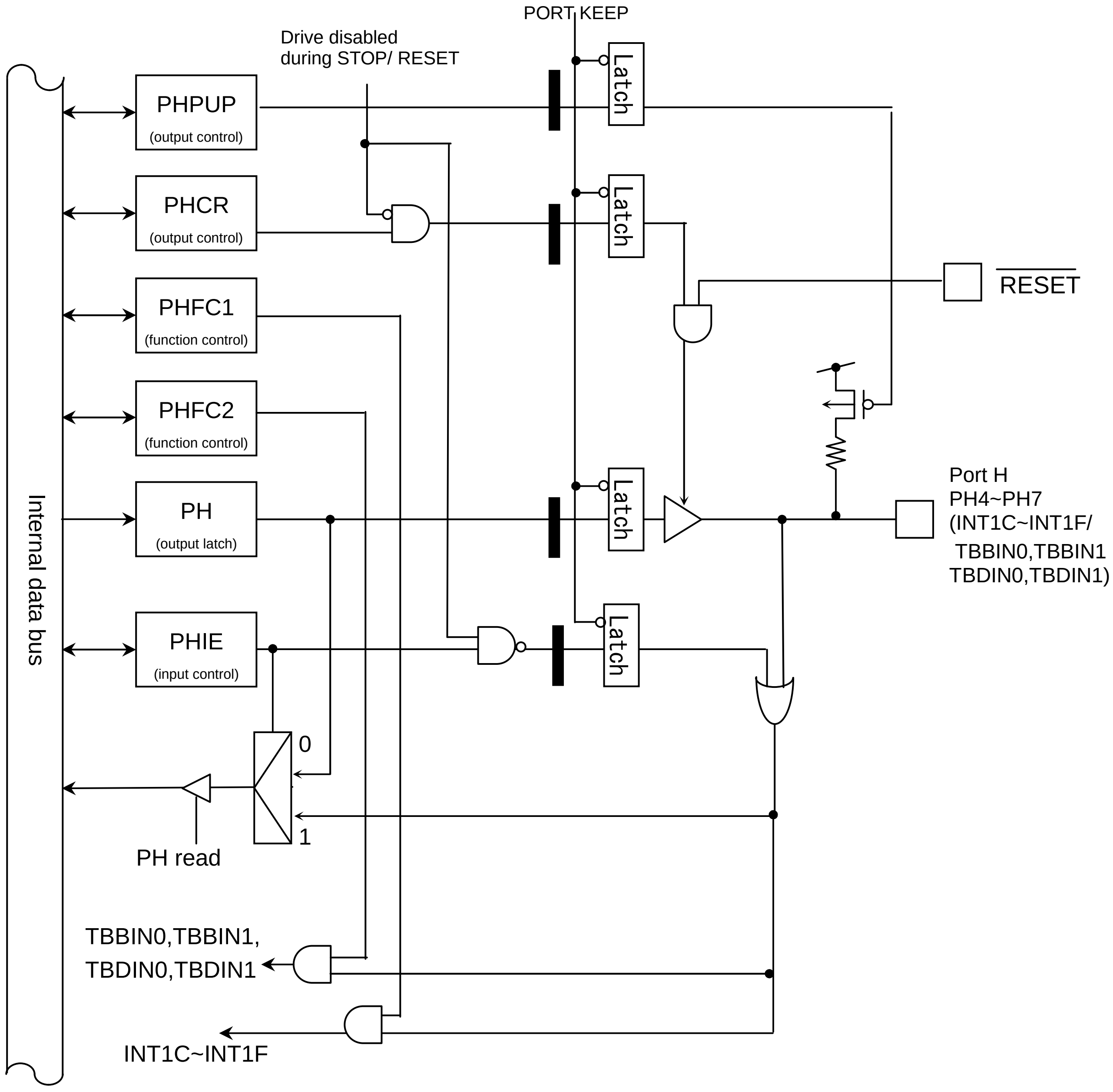


Fig. 7.51 Port H (PH4 thorough PH7)

Port H register

PH (0xFF00_4440)

	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 (0xFF00_4444)

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

Port H function register 1

PHFC1 (0xFF00_4448)

	7	6	5	4	3	2	1	0
Bit Symbol	PH7F	PH6F	PH5F	PH4F	PH3F	PH2F	PH1F	PH0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:INT1F	0:PORT 1:INT1E	0:PORT 1:INT1D	0:PORT 1:INT1C	0:PORT 1:INT1B	0:PORT 1:INT1A	0:PORT 1:INT19	0:PORT 1:INT18

Port H function register 2

PHFC2 (0xFF00_444C)

	7	6	5	4	3	2	1	0
Bit Symbol	PH7F2	PH6F2	PH5F2	PH4F2	PH3F2	PH2F2	PH1F2	PH0F2
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:TBDIN1	0:PORT 1:TBDIN0	0:PORT 1:TBBIN1	0:PORT 1:TBBIN0	0:PORT 1:TBAIN1	0:PORT 1:TBAIN0	0:PORT 1:TB9IN1	0:PORT 1:TB9IN0

Port H pull-up control register

PHPUP (0xFF00_446C)

	7	6	5	4	3	2	1	0
Bit Symbol	PEH7	PEH6	PEH5	PEH4	PEH3	PEH2	PEH1	PEH0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port H input enable control register

PHIE (0xFF00_4478)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.19 Port I (PI0 through PI7)

The port I 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 PICR and the function register PIFC1. A reset allows all bits of the output latch PI to be set to "1," all bits of PICR to be cleared to "0," and the port I to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PIIE to “1”.

Besides the port function, the port I performs other functions: PI0 through PI3 input two-phase pulse, PI4 and PI7 input A/D triggers into the A/D converter., PI5 and PI6 output a 16-bit timer.

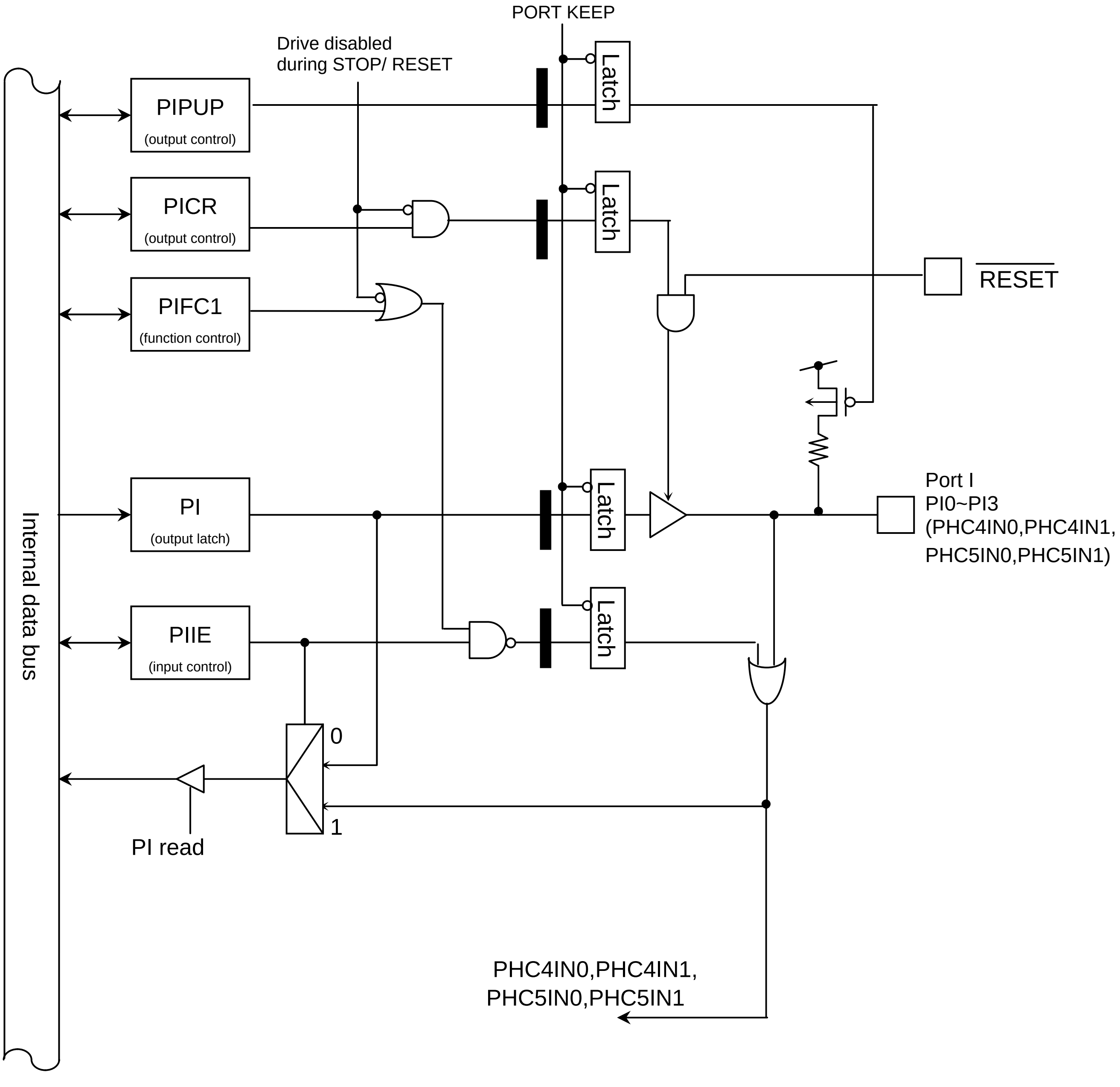


Fig. 7.52 Port I (PI0 through PI3)

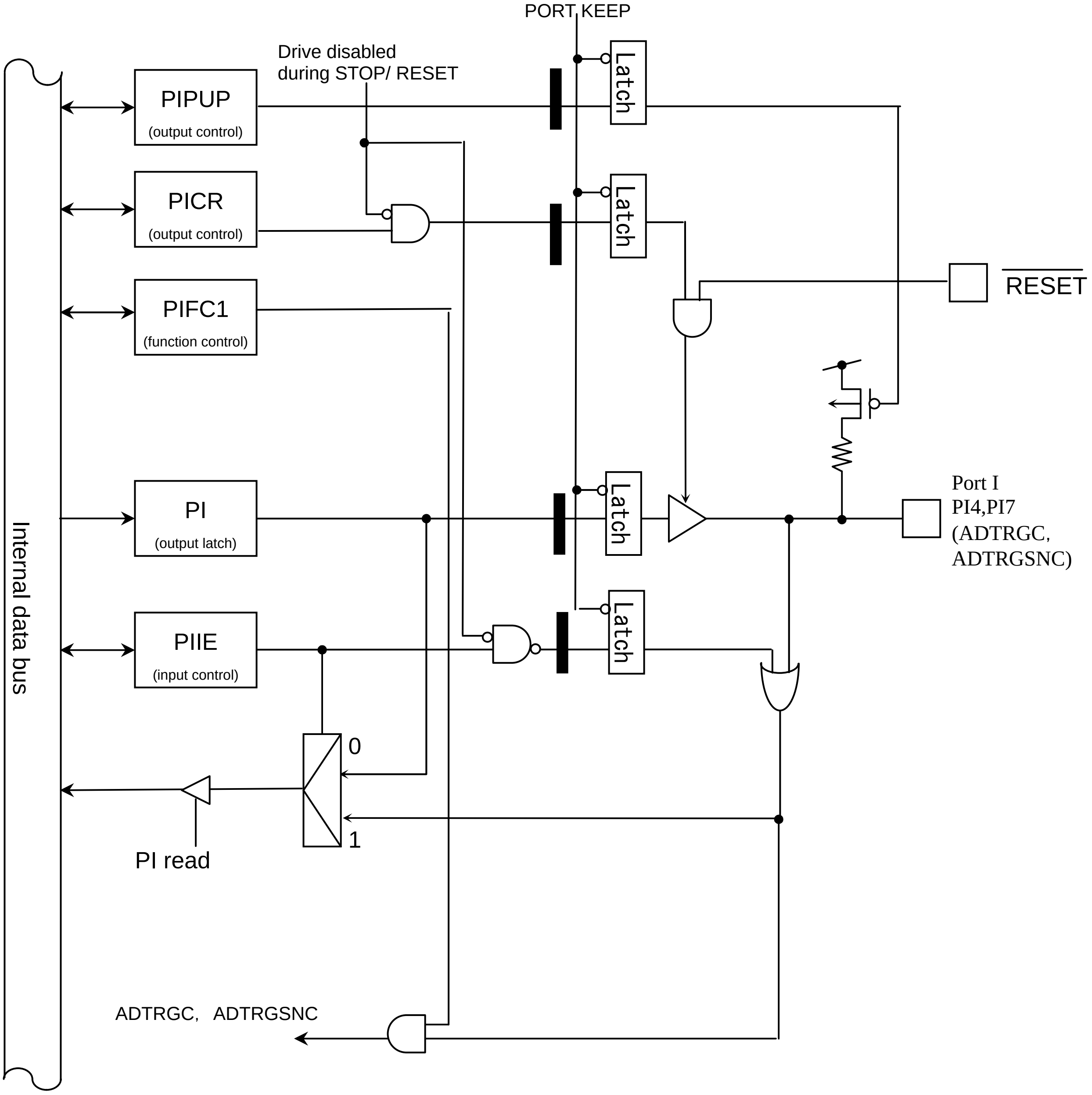


Fig. 7.53 Port I (PI4, PI7)

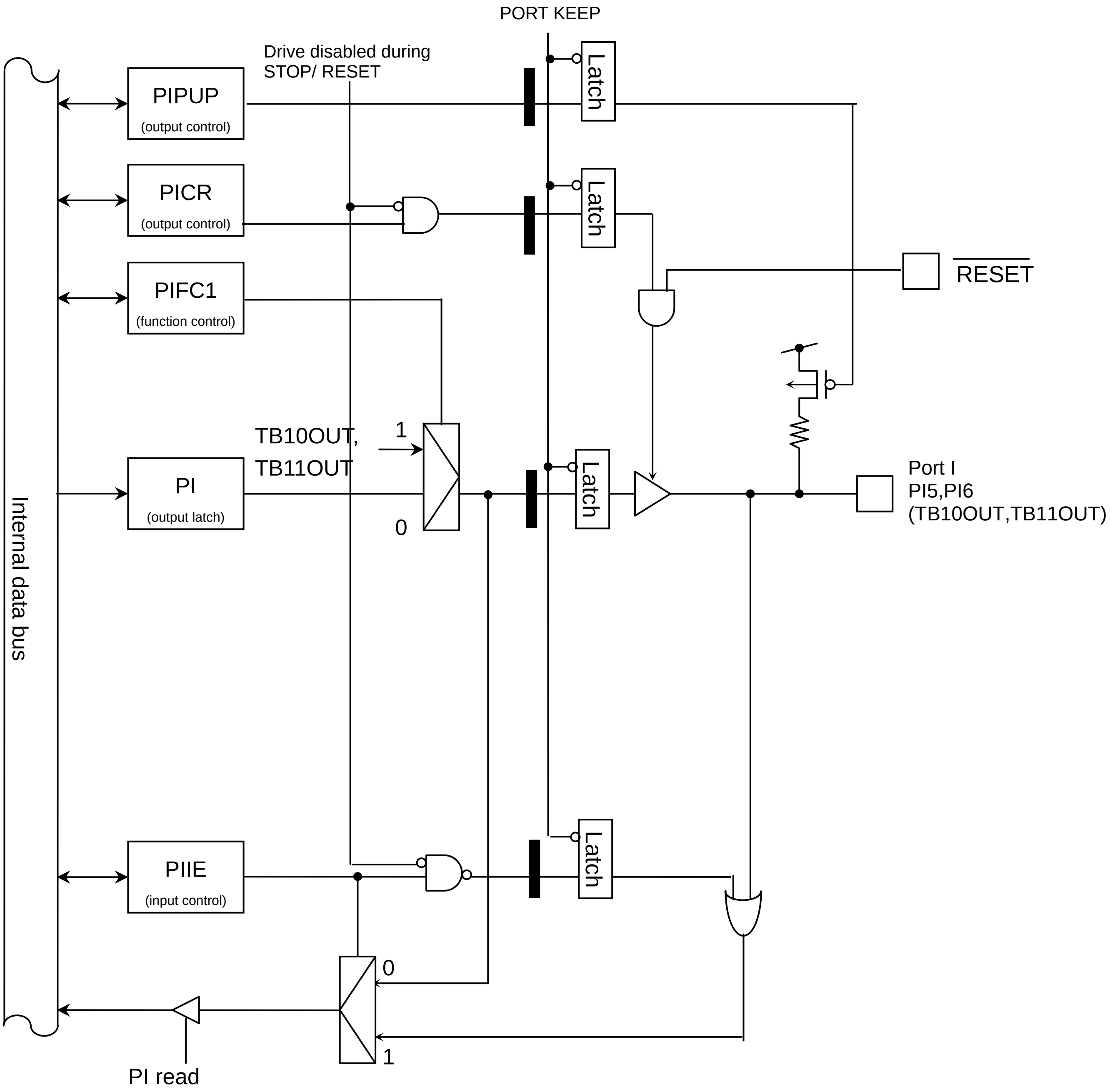


Fig. 7.54 Port I (PI5, PI6)

Port I register

PI
(0xFF00_4480)

	7	6	5	4	3	2	1	0
Bit Symbol	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port I control register

PICR
(0xFF00_4484)

	7	6	5	4	3	2	1	0
Bit Symbol	PI7C	PI6C	PI5C	PI4C	PI3C	PI2C	PI1C	PI0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port I function register 1

PIFC1
(0xFF00_4488)

	7	6	5	4	3	2	1	0
Bit Symbol	PI7F	PI6F	PI5F	PI4F	PI3F	PI2F	PI1F	PI0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:ADTRGSNC	0:PORT 1:TB11OUT	0:PORT 1:TB10OUT	0:PORT 1:ADTRGC	0:PORT 1:PHC5IN1	0:PORT 1:PHC5IN0	0:PORT 1:PHC4IN1	0:PORT 1:PHC4IN0

Port I pull-up control register

PIPUP
(0xFF00_44AC)

	7	6	5	4	3	2	1	0
Bit Symbol	PEI7	PEI6	PEI5	PEI4	PEI3	PEI2	PEI1	PEI0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port I input enable control register

PIIE
(0xFF00_44B8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEI7	PIEI6	PIEI5	PIEI4	PIEI3	PIEI2	PIEI1	PIEI0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

7.20 Port J (PJ0 through PJ7)

The port J 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 PJCR and the function register PJFC1. A reset allows all bits of the output latch PJ to be set to "1," all bits of PJCR to be cleared to "0," and the port J to be put in input mode.

Input is disabled right after reset. To enable input, set the corresponding bit of PJIE to “1”.

Besides the port function, the port J performs other functions: PJ0 and PJ1 input a 16-bit timer PJ2 through PJ7 input external interrupts.

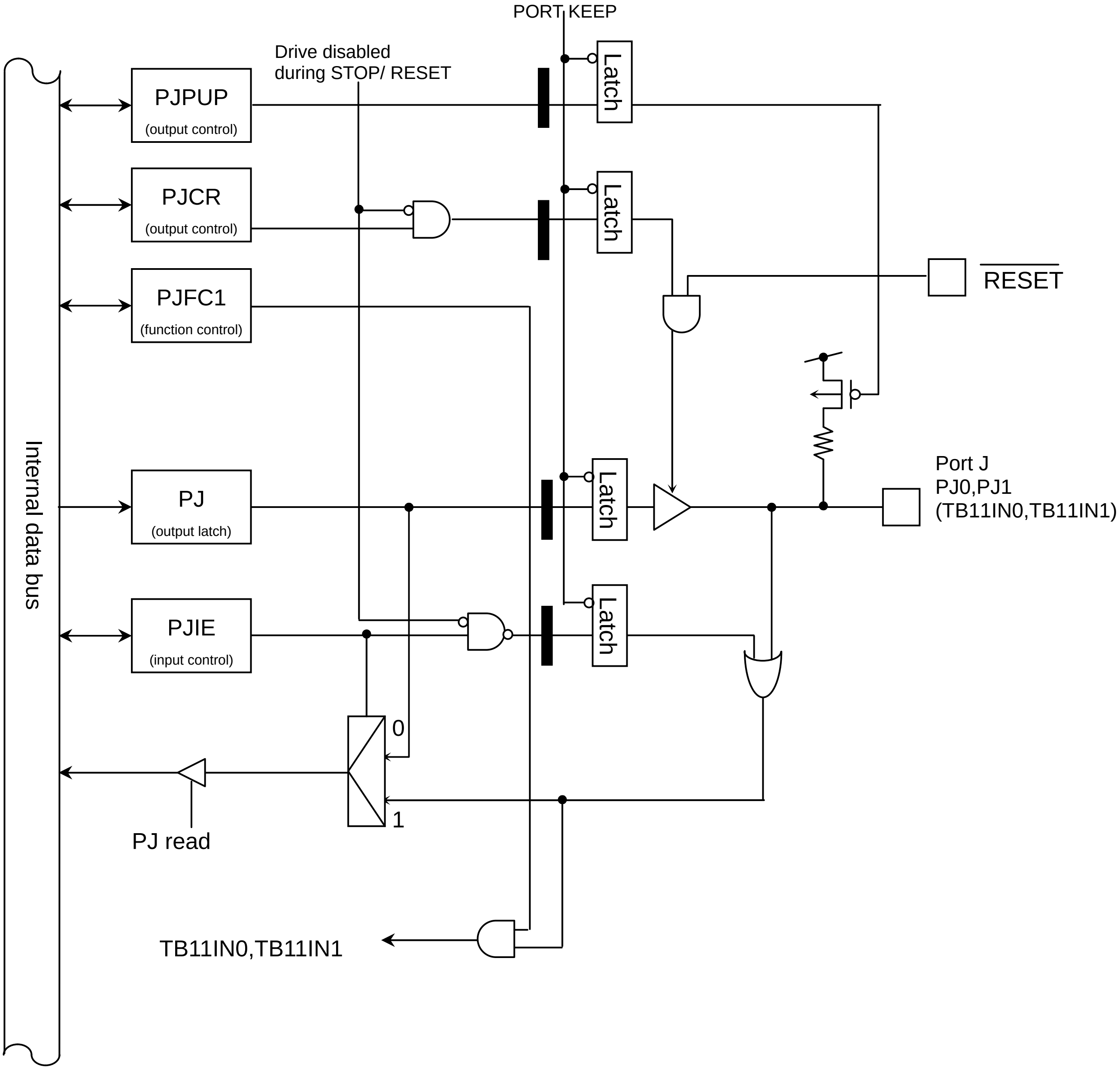


Fig. 7.55 Port J (PJ0, PJ1)

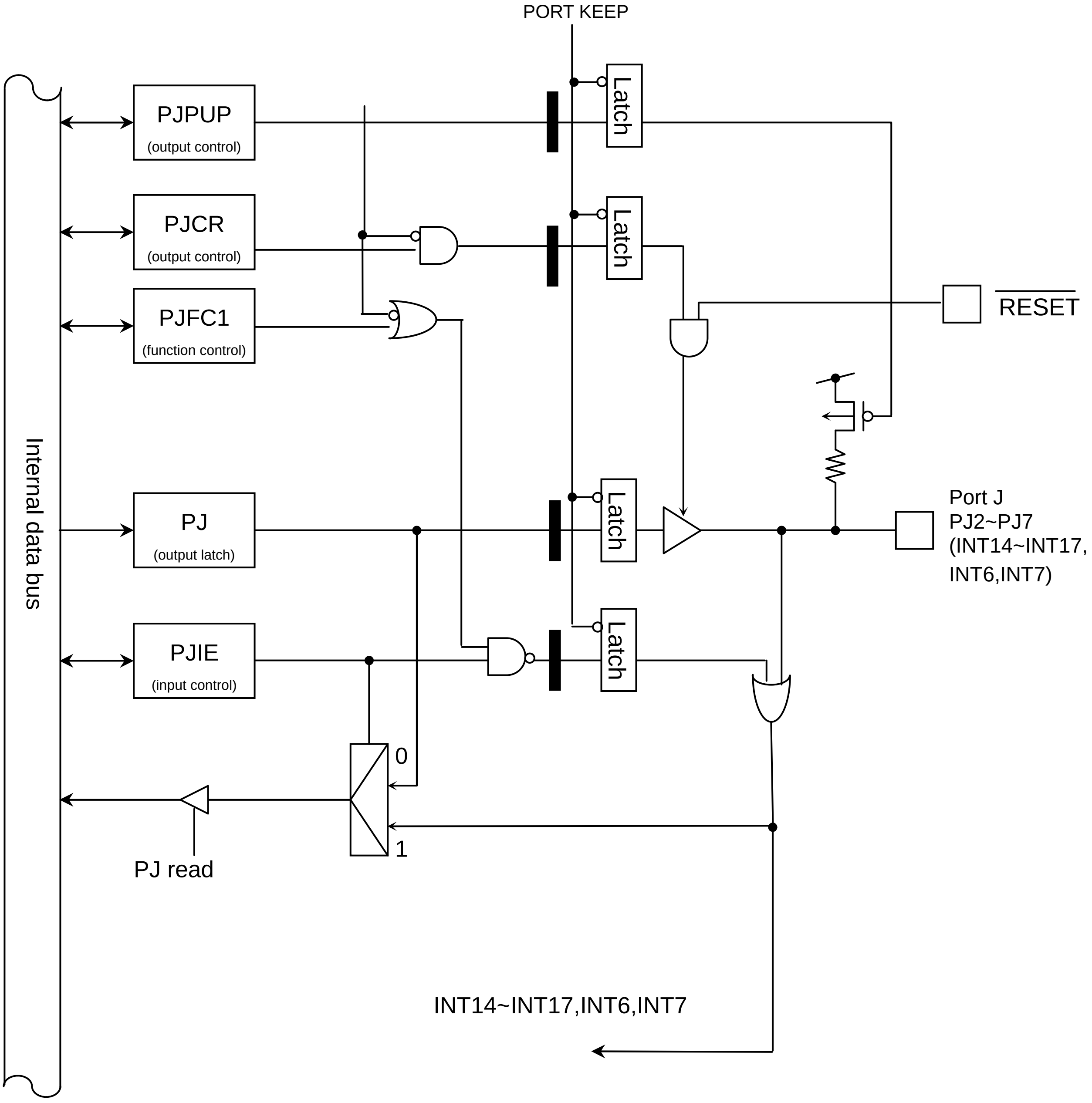


Fig. 7.56 Port J (PJ2 through PJ7)

Port J register

PJ
(0xFF00_44C0)

	7	6	5	4	3	2	1	0
Bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
Read/Write	R/W							
After reset	Input mode (output latch register is set to "1.")							

Port J control register

PJCR
(0xFF00_44C4)

	7	6	5	4	3	2	1	0
Bit Symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port J function register 1

PJFC1
(0xFF00_44C8)

	7	6	5	4	3	2	1	0
Bit Symbol	PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:PORT 1:INT7	0:PORT 1:INT6	0:PORT 1:INT17	0:PORT 1:INT16	0:PORT 1:INT15	0:PORT 1:INT14	0:PORT 1:TB11IN1	0:PORT 1:TB11IN0

Port J pull-up control register

PJPUP
(0xFF00_44EC)

	7	6	5	4	3	2	1	0
Bit Symbol	PEJ7	PEJ6	PEJ5	PEJ4	PEJ3	PEJ2	PEJ1	PEJ0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up	Pull-up 0: Off 1: Pull-Up

Port J input enable control register

PJIE
(0xFF00_44F8)

	7	6	5	4	3	2	1	0
Bit Symbol	PIEJ7	PIEJ6	PIEJ5	PIEJ4	PIEJ3	PIEJ2	PIEJ1	PIEJ0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled	Input 0: Disabled 1: Enabled

8. External Bus Interface

The TMP19A44 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 4-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 15 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 4 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 8 clocks 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 TMP19A44 can be set to either separate bus or multiplexed bus mode. Setting the BUSMD pin (port P45) to the "L" level (DVSS) at a reset activates the separate bus mode, and setting the pin to the "H" level (DVCC3) 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 shows these.

Table 8.1 Bus Mode, Address and Data Pins

	Separate **BUSMD (P45) ="L"**	**Multiplex** **BUSMD (P45) ="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 3 (P37)	General-purpose port	ALE
Port 5 (P50 to P57)	A0-A7	General-purpose port
Port 6 (P60 to P67)	A8-A15	General-purpose port

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 (PnFCm), and set the input enable register (PnIE).

8.2 Data Format

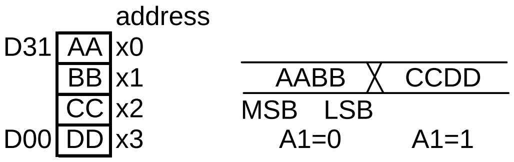
Internal registers and external bus interfaces of the TMP19A44 are configured as described below.

(1) Big-endian mode

① Word access

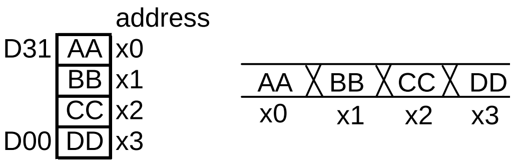
- 16-bit bus width

Internal registers External buses



- 8-bit bus width

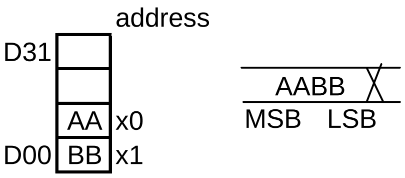
Internal registers External buses

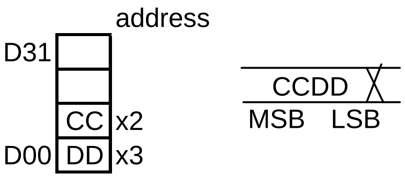


② Half word access

- 16-bit bus width

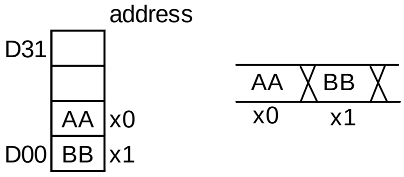
Internal registers External buses



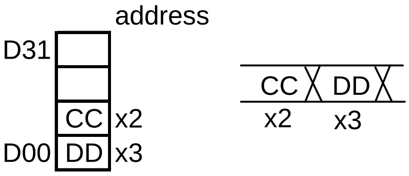


- 8-bit bus width

Internal registers External buses



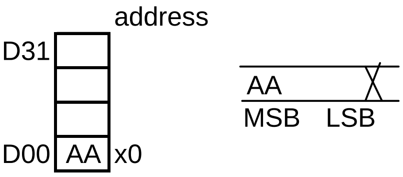
Internal registers External buses

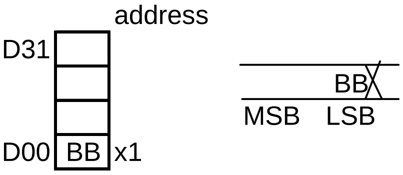


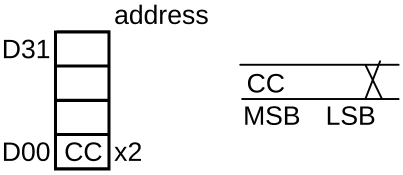
③ Byte access

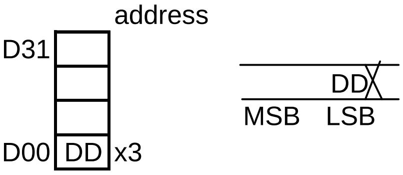
- 16-bit bus width

Internal registers External buses



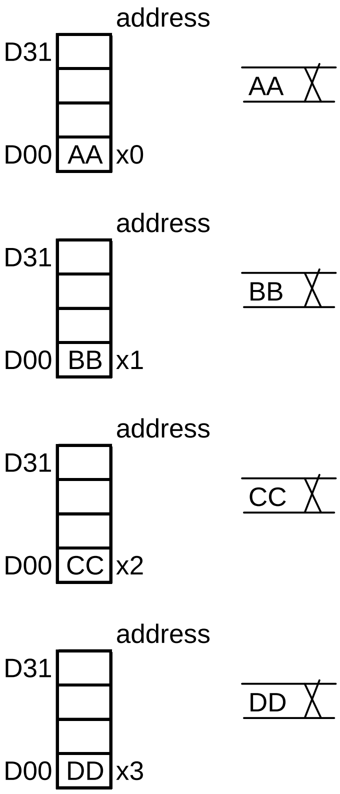






- 8-bit bus width

Internal registers External buses

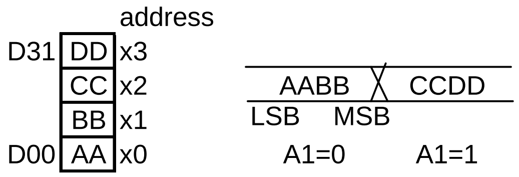


(2) Little-endian mode

① Word access

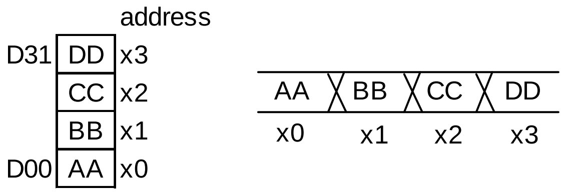
- 16-bit bus width

Internal registers External buses



- 8-bit bus width

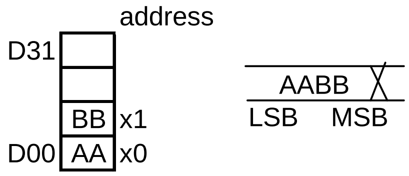
Internal registers External buses

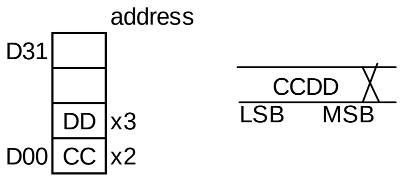


② Half word access

- 16-bit bus width

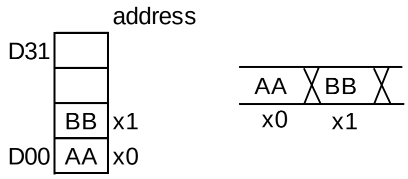
Internal registers External buses



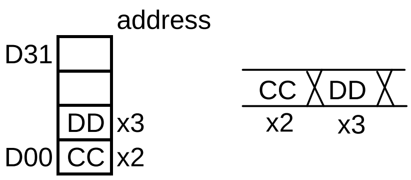


- 8-bit bus width

Internal registers External buses



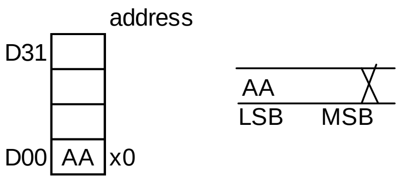
Internal registers External buses

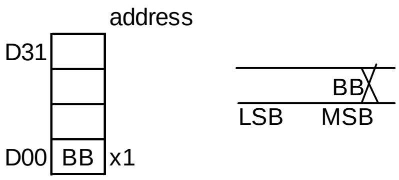


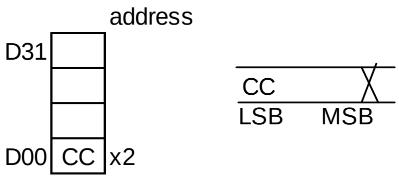
③ Byte access

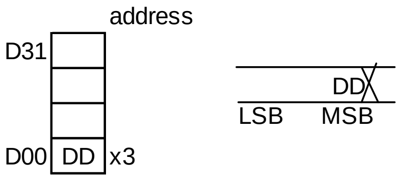
- 16-bit bus width

Internal registers External buses

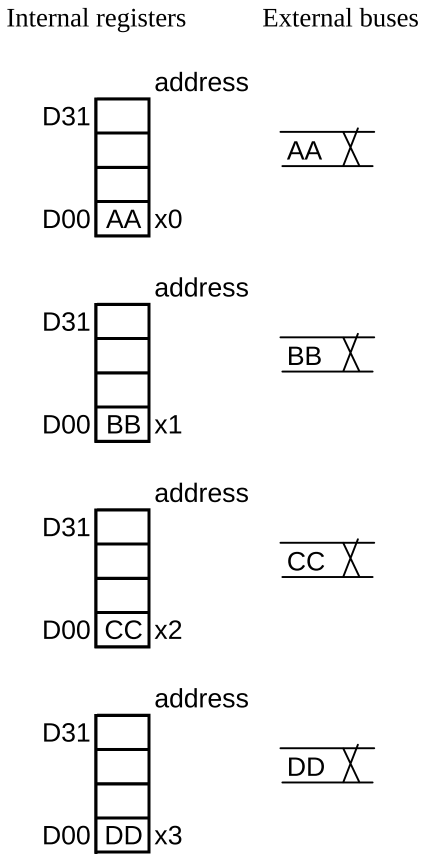








- 8-bit bus width



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 TMP19A44 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. 8.1 shows read bus timing and Fig. 8.2 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 $\overline{\text{RD}}$ and $\overline{\text{WR}}$ do not become active.

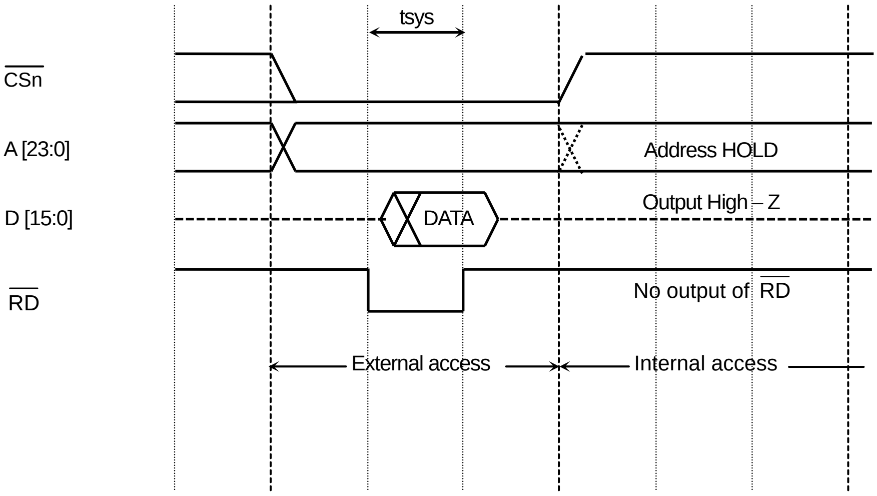


Fig. 8.1 Read Operation Timing Diagram

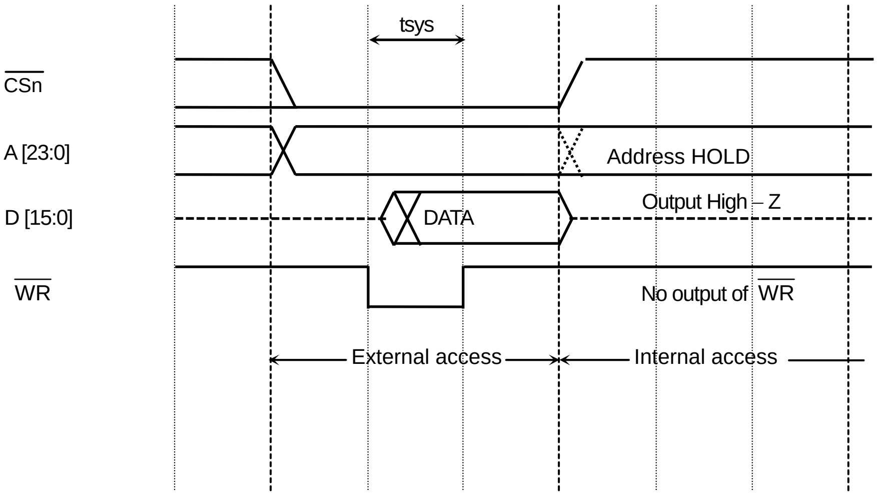


Fig. 8.2 Write Operation Timing Diagram

(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 15 clocks can be automatically inserted.

② A wait can be inserted via the $\overline{\text{WAIT}}$ pin (from 2+2N through 15+2N). Note: 2N/4N is the number of external waits that can be inserted.

③ A wait can be inserted via the $\overline{\text{RDY}}$ pin (from 2+2N through 15+2N). Note: 2N/4N 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 through Fig. 8.10 show the timing diagrams in which waits have been inserted.

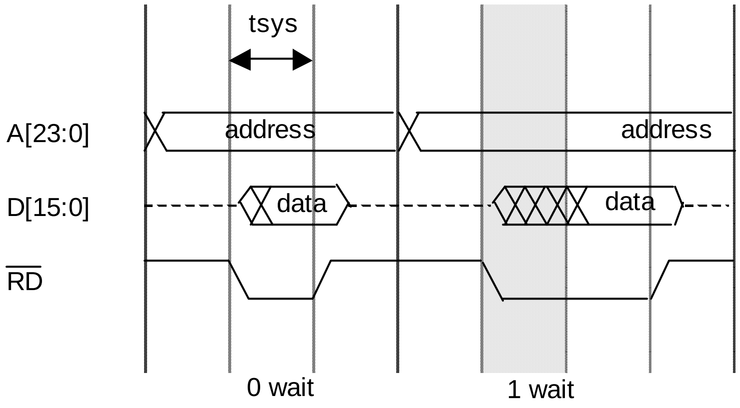


Fig. 8.3 Read Operation Timing Diagram (0 Wait and 1 Wait Automatically Inserted)

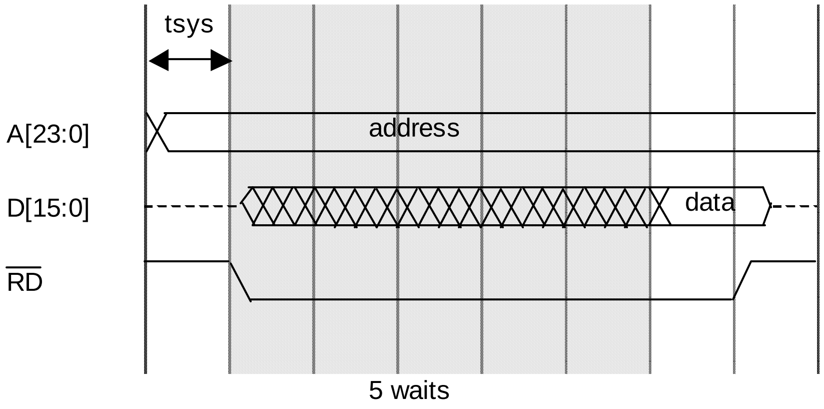
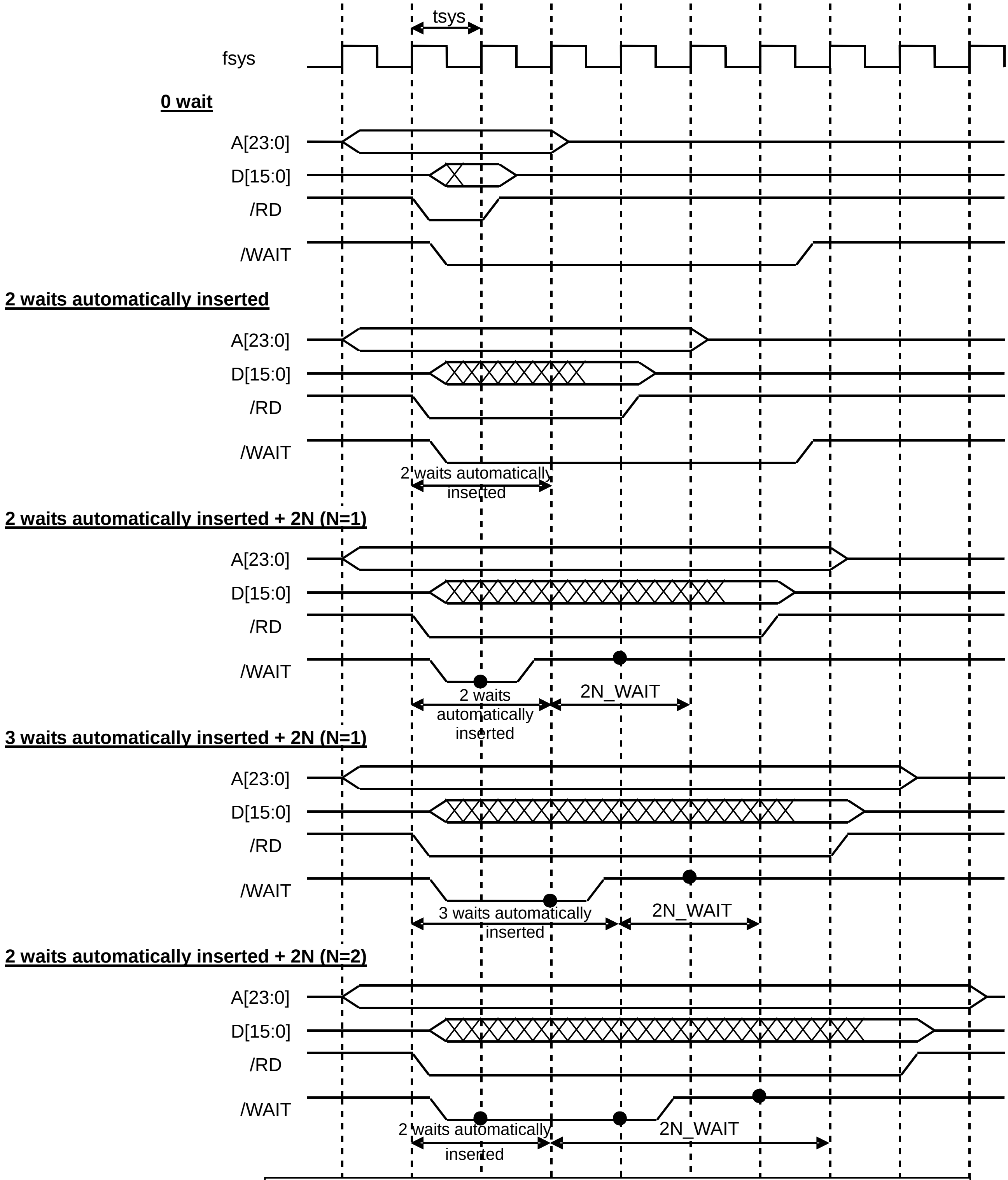


Fig. 8.4 Read Operation Timing Diagram (5 Waits Automatically Inserted)

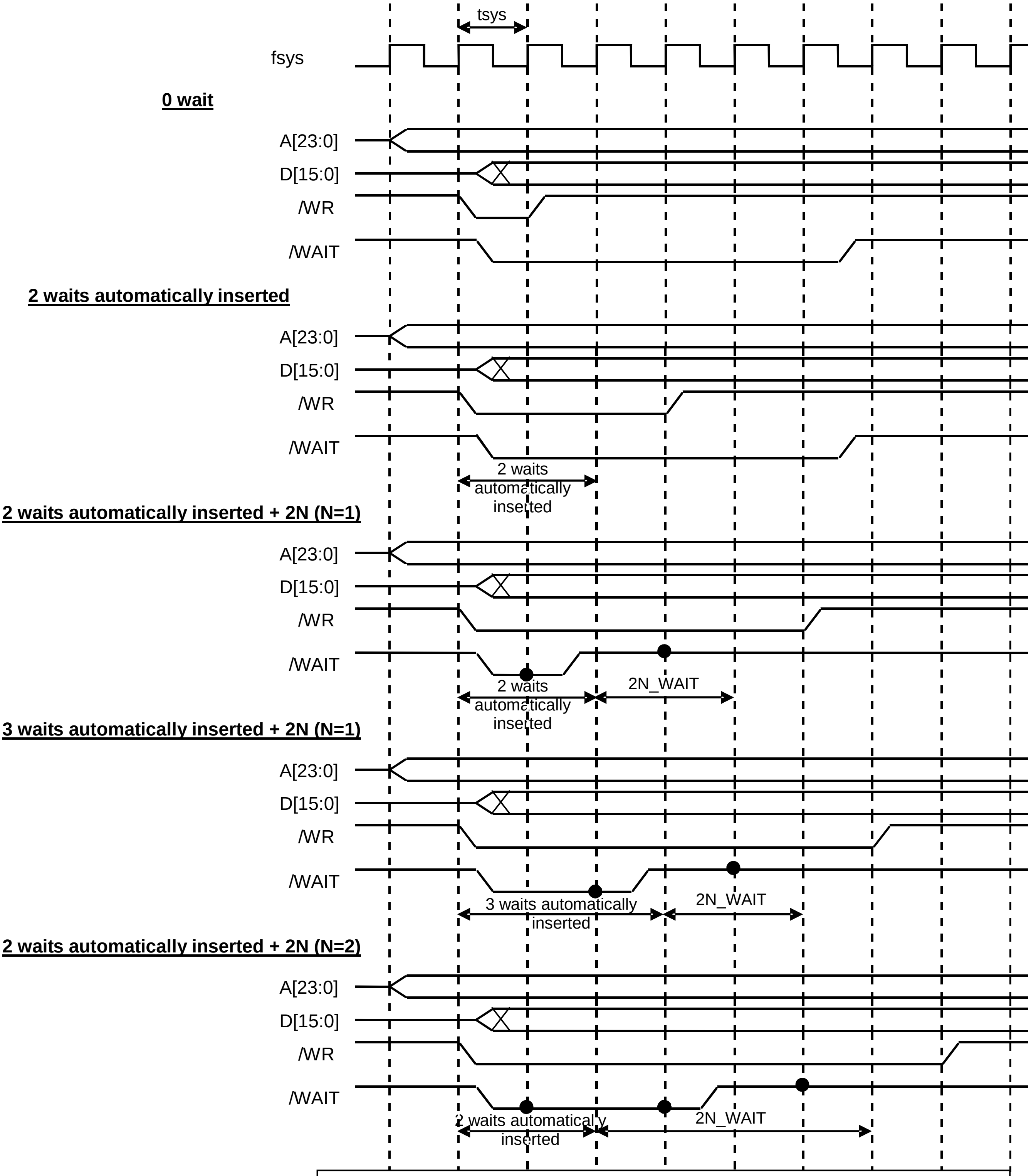
Fig. 8.5 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the separate bus mode.


tsys
fsys
0 wait
A[23:0]
D[15:0]
/RD
/WAIT
2 waits automatically inserted
A[23:0]
D[15:0]
/RD
/WAIT
2 waits automatically inserted
2 waits automatically inserted + 2N (N=1)
A[23:0]
D[15:0]
/RD
/WAIT
2 waits automatically inserted
2N_WAIT
3 waits automatically inserted + 2N (N=1)
A[23:0]
D[15:0]
/RD
/WAIT
3 waits automatically inserted
2N_WAIT
2 waits automatically inserted + 2N (N=2)
A[23:0]
D[15:0]
/RD
/WAIT
2 waits automatically inserted
2N_WAIT

● --- 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. 8.5 Read Operation Timing Diagram

Fig. 8.6 shows the write operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the separate bus mode.


tsys
fsys
0 wait
A[23:0]
D[15:0]
/WR
/WAIT
2 waits automatically inserted
A[23:0]
D[15:0]
/WR
/WAIT
2 waits automatically inserted
2 waits automatically inserted + 2N (N=1)
A[23:0]
D[15:0]
/WR
/WAIT
2 waits automatically inserted
2N_WAIT
3 waits automatically inserted + 2N (N=1)
A[23:0]
D[15:0]
/WR
/WAIT
3 waits automatically inserted
2N_WAIT
2 waits automatically inserted + 2N (N=2)
A[23:0]
D[15:0]
/WR
/WAIT
2 waits automatically inserted
2N_WAIT

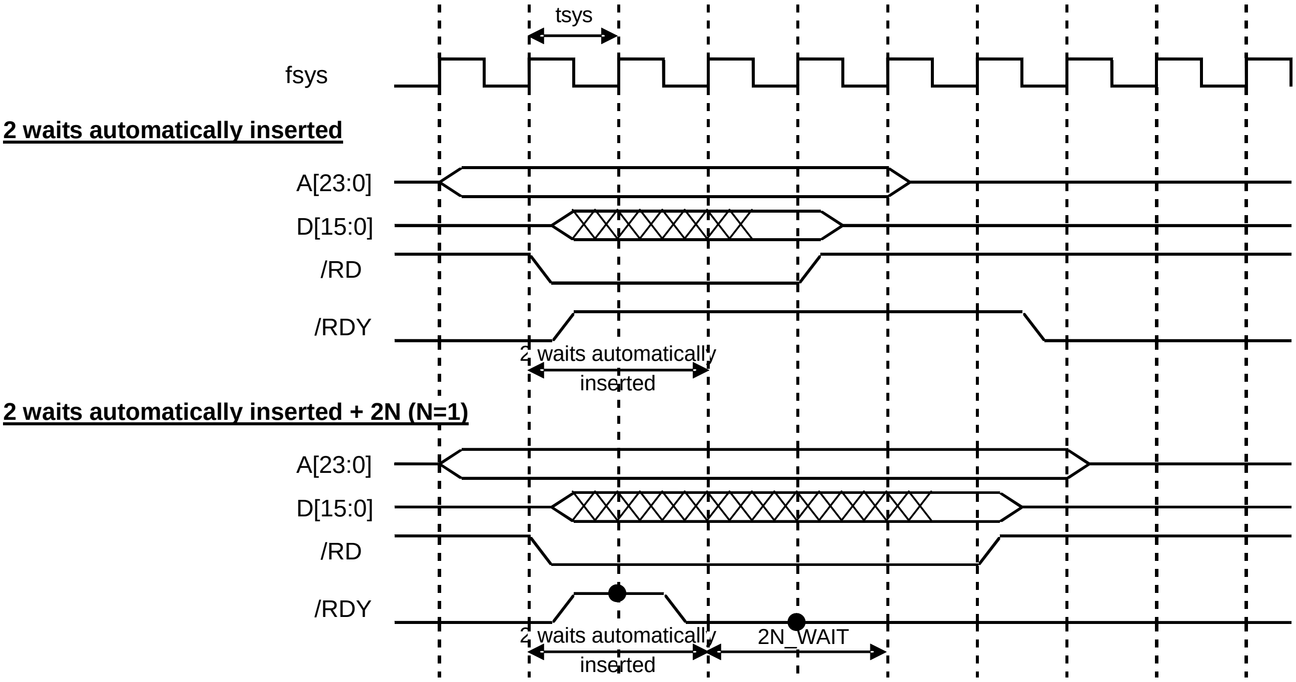
● --- External wait sampling point
External wait sampling points take place before a cycle of waits automatically inserted is finished and before a 2N_wait cycle is finished as shown above.
The same applies to combinations of other numbers of waits.

Fig. 8.6 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. 8.7 shows the $\overline{\text{RDY}}$ inputs and the number of waits.


tsys
fsys
2 waits automatically inserted
A[23:0]
D[15:0]
/RD
/RDY
2 waits automatically inserted
2 waits automatically inserted + 2N (N=1)
A[23:0]
D[15:0]
/RD
/RDY
2 waits automatically inserted
2N_WAIT

● --- External RDY sampling point
External RDY 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. 8.7 $\overline{\text{RDY}}$ Input and Wait Operation Timing Diagram

(3) Time that it takes before ALE is asserted

When the external bus of the TMP19A44 is used as a multiplexed bus, the ALE width (assert time) can be specified by using the system control register BUSCR<ALESEL1:0> 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 $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal is different depending on the BUSCR<ALESEL1:0>.

During a reset, <ALESEL 1:0> = "1" is set and the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal is asserted as a point of two system (internal) clocks after an address is established. If <ALESEL 1:0> is cleared to "0," the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ 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.

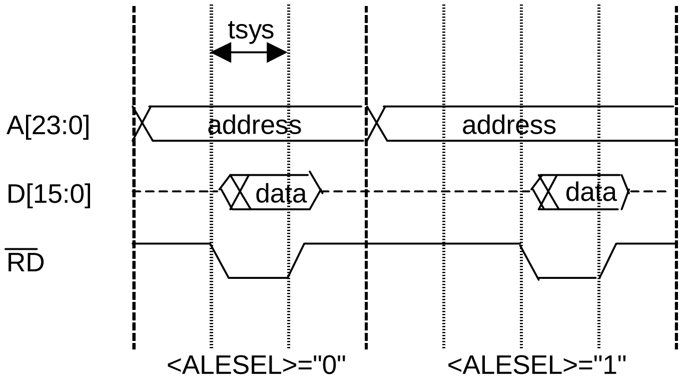


Fig. 8.8 ALE Assert Ttiming in Separate Bus Mode

(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 dummy cycle, none, one, two or four system clocks (internal) can be specified for each block. Fig. 8.9 shows the timing of recovery time insertion.

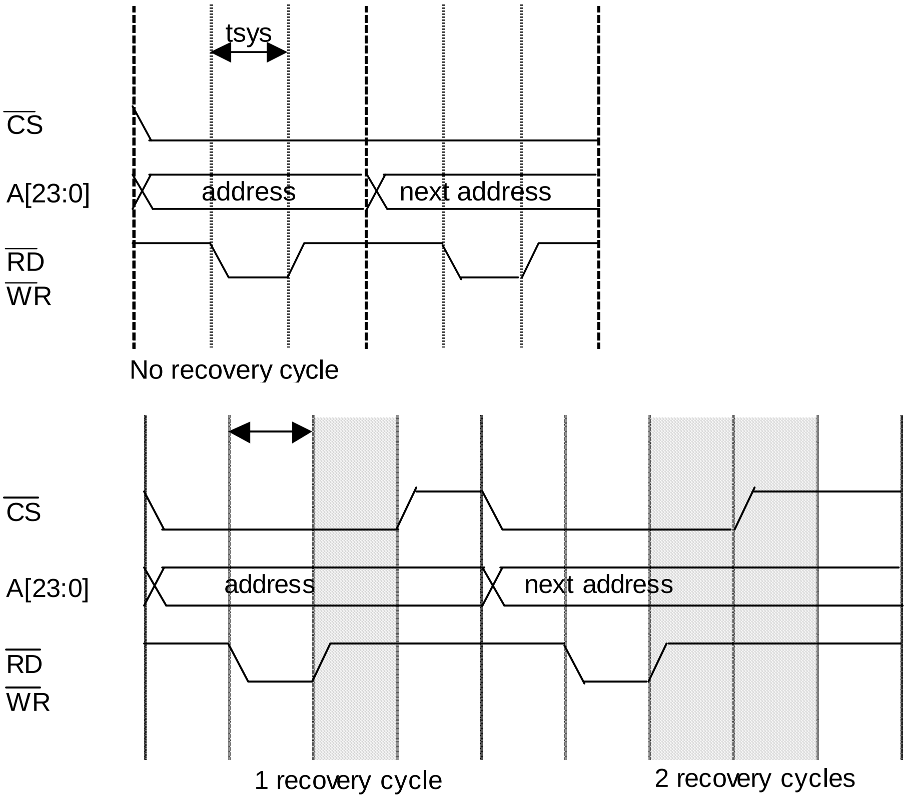


Fig. 8.9 Timing of Recovery Time Insertion in Separate Bus Mode

(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, none, one, two three, four, six and eight system clocks (internal) can be specified for each block. Fig. 8.10 shows the timing of recovery time insertion.

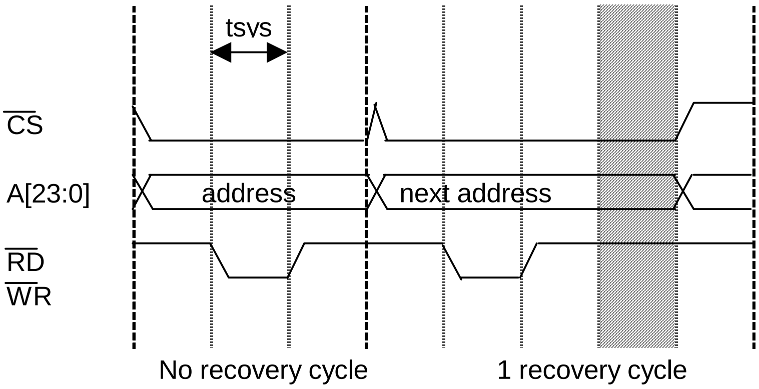


Fig. 8.10 Timing of Chip Selector Recovery Time Insertion

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 TMP19A44 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. 8.11 shows read bus timing and Fig. 8.12 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 $\overline{\text{RD}}$ and $\overline{\text{WR}}$ do not become active.

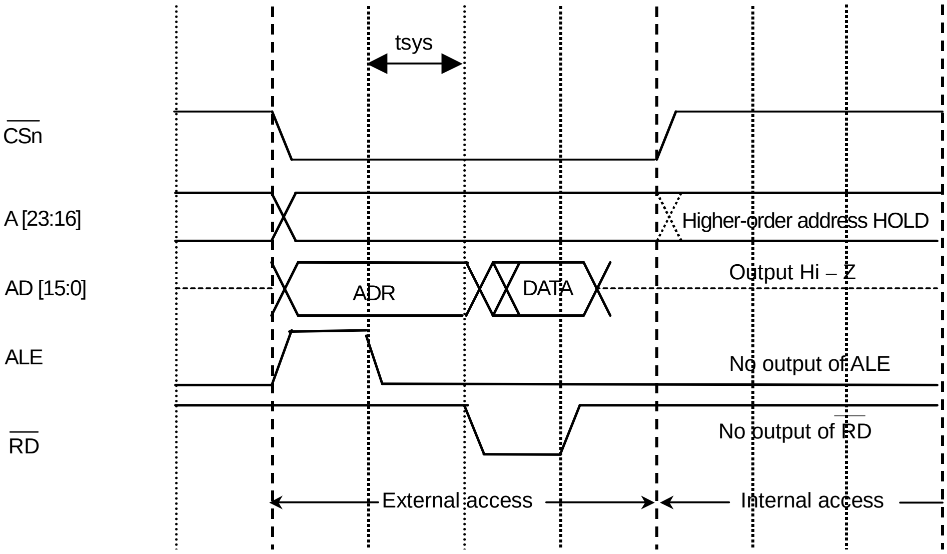


Fig. 8.11 Read Operation Timing Diagram

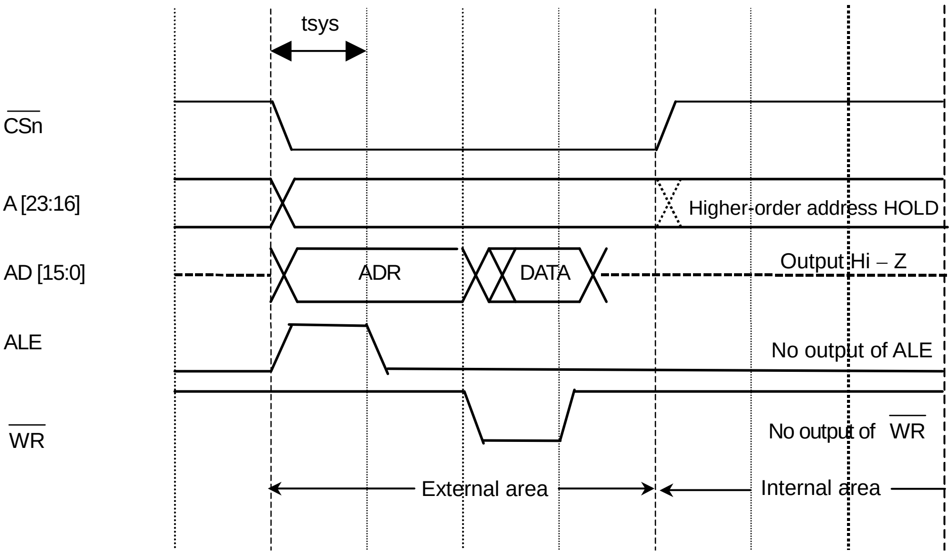


Fig. 8.12 Write Operation Timing Diagram

(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:

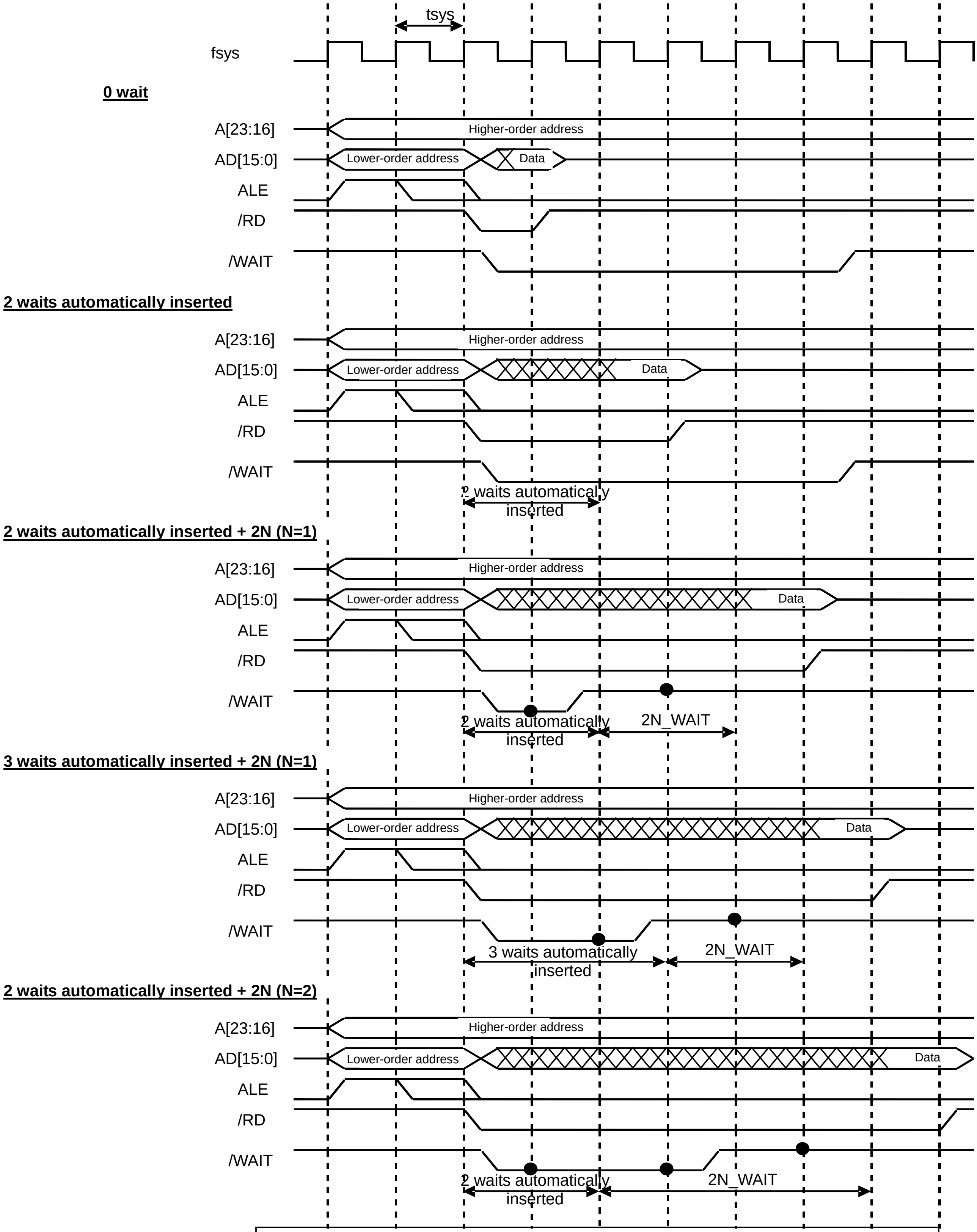
① A wait of up to 15 clocks can be automatically inserted.

② A wait can be inserted via the $\overline{\text{WAIT}}$ pin (from 2+2N through 15+2N).
Note: 2N/4N is the number of external waits that can be inserted.

③ A wait can be inserted via the $\overline{\text{RDY}}$ pin (from 2+2N through 15+2N).
Note: 2N/4N is the number of external waits that can be inserted.
BmnCS<BnW>,BUSCR<WAITSMP>

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>.

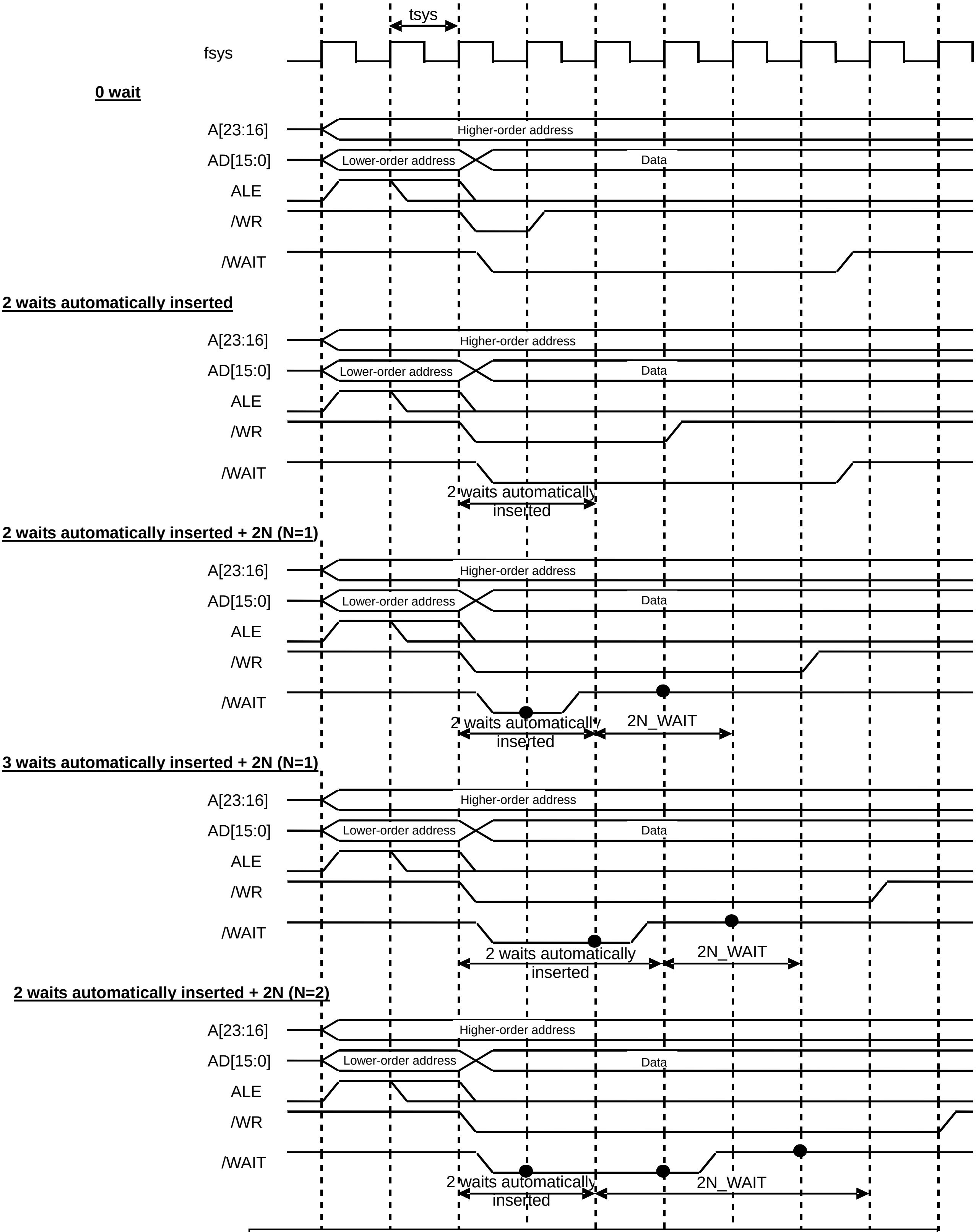
Fig. 8.13 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the multiplexed bus mode.



● --- 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. 8.13 Read Operation Timing Diagram

Fig. 8.14 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. 8.14 Write Operation Timing Diagram

(3) Time that it takes before ALE is asserted

One of system clocks of 1 to 4 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.

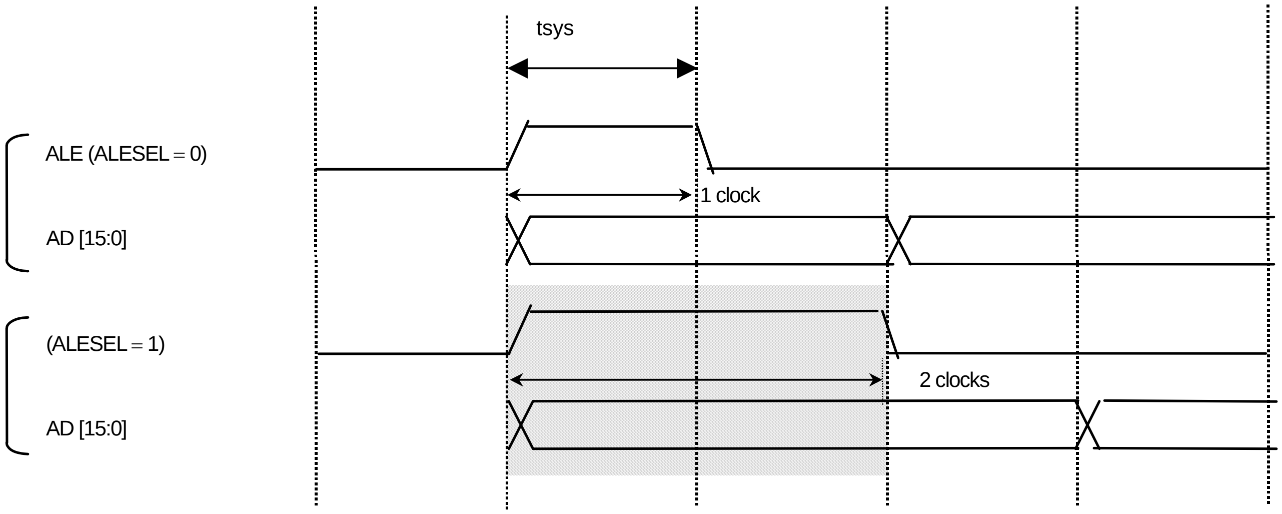


Fig. 8.15 Time That It Takes Before ALE Is Asserted

Fig. 8.16 shows the timing when the ALE is 1 clock or 2 clocks.

When the ALE is 1 clock or 2 clocks

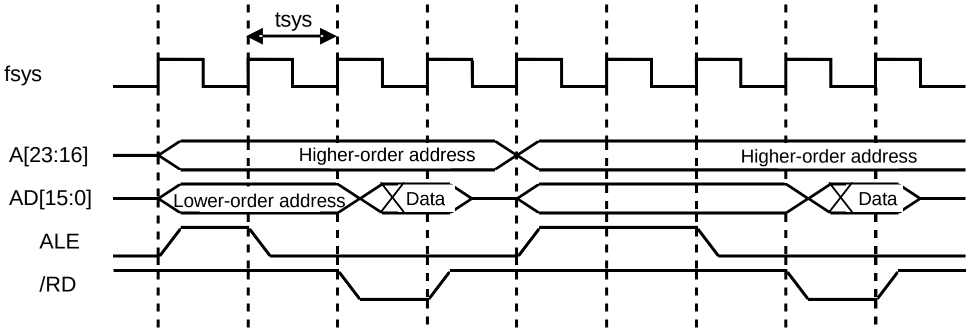


Fig. 8.16 Read Operation Timing Diagram (When the ALE is 1 Clock or 2 Clocks)

(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, none, one, two or four system clocks (internal) can be specified for each block. Fig. 8.17 shows the timing of recovery time insertion.

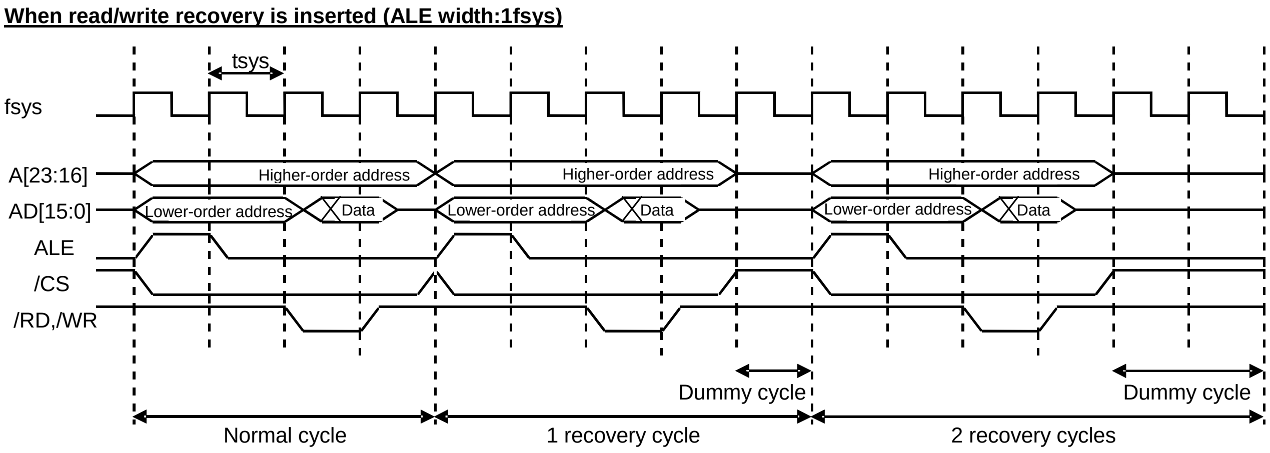


Fig. 8.17 Timing of Recovery Time Insertion

(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. 8.18 shows the timing of recovery time insertion.

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

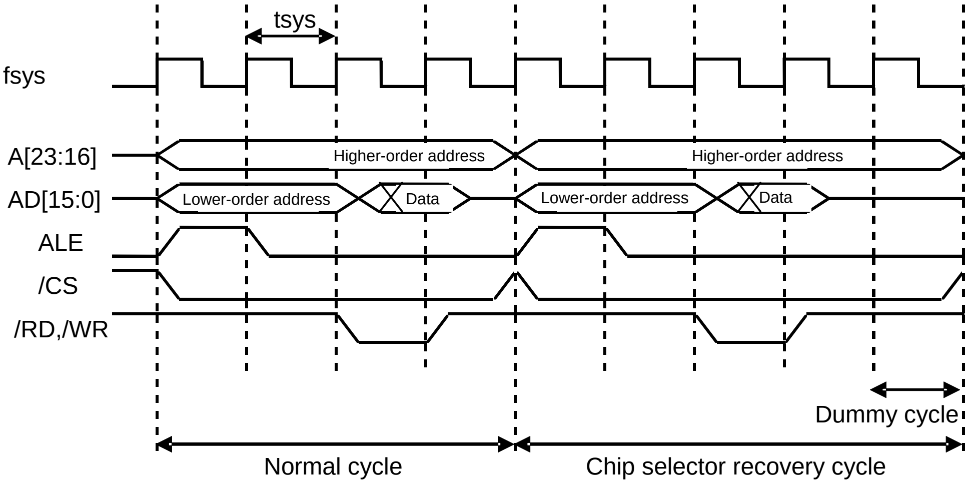


Fig. 8.18 Timing of Recovery Time Insertion

8.5 Bus Arbitration

The TMP19A44 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 TMP19A44 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 TMP19A44 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 TMP19A44 can lock.

(2) Acquisition of bus control authority

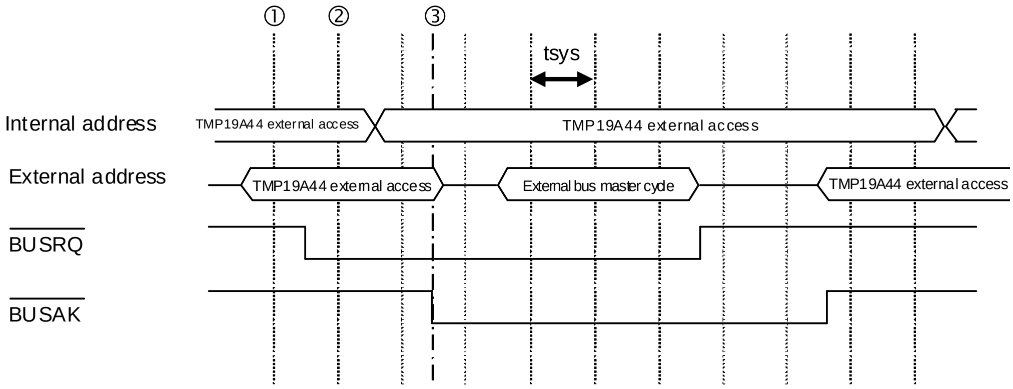
The external bus master requests the TMP19A44 for bus control authority by asserting the $\overline{\text{BUSRQ}}$ signal. The TMP19A44 samples the $\overline{\text{BUSRQ}}$ signal at the break of external bus cycles on the internal buses (G-BUS) and determines whether or not to give the bus control 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 $\text{R}/\overline{\text{W}}$, $\overline{\text{HWR}}$ 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 TMP19A44, 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. 8.19 shows the timing of acquiring bus control authority by the external bus master.



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

② The TMP19A44 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 TMP19A44 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.

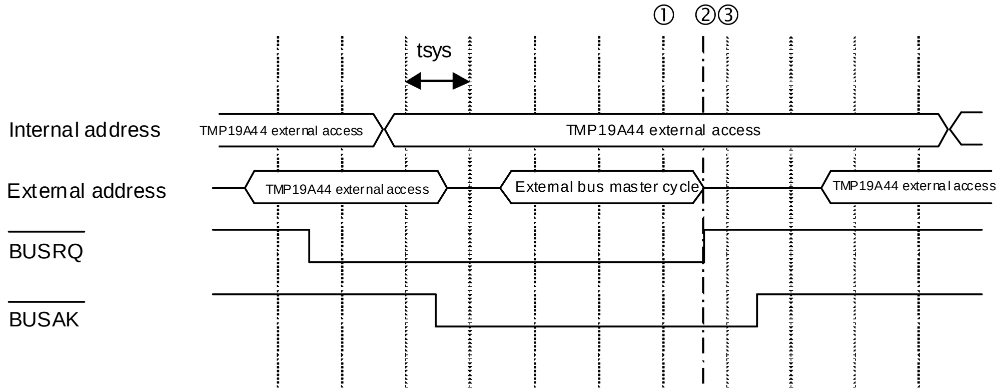
Fig. 8.19 Bus Control Authority Acquisition Timing

(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 negates the $\overline{\text{BUSRQ}}$ signal and returns the bus control authority to the TMP19A44.

Fig. 8.20 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 TMP19A44 recognizes that the $\overline{\text{BUSRQ}}$ is at the "H" level, and deasserts the $\overline{\text{BUSAK}}$.

Fig. 8.20 Timing of Releasing Bus Control Authority

9. The Chip Selector and Wait Controller

The TMP19A44 can be connected to external devices (I/O devices, ROM and SRAM).

4-block address spaces (CS0 through CS3) can be established in the TMP19A44 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.

$\overline{\text{CS0}}$ through $\overline{\text{CS3}}$ (also used as P40 through P43) are the output pins corresponding to spaces CS0 through CS3. These pins generate chip selector signals (for ROM and SRAM) to each space when the CPU designates an address in which spaces CS0 through CS3 are selected. For chip selector signals to be generated, however, the port 4 controller register (P4CR) and the port 4 function registers (P4FC1 and P4FC2) must be set appropriately.

The specification of the spaces CS0 through CS3 is to be performed with a combination of base addresses (BAn, n=0 to 3) and mask addresses (MAn, n=0 to 3) using the base and mask address setting registers (BMA0 through BMA3).

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).

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

9.1 Specifying Address Spaces

Spaces CS0 through CS3 are specified using the base and mask address setting registers (BMA0 through BMA3).

In each bus cycle, a comparison is made to see if each address on the bus is located in the space CS0 through CS3. 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 $\overline{\text{CS0}}$ through $\overline{\text{CS3}}$ and the operations specified by the chip selector and wait controller registers (B01CS and B23CS) are executed. (Refer to "9.2 The Chip Selector and Wait Controller.")

9.1.1 Base and Mask Address Setting Registers

Fig. 9.1 show base and mask address setting registers. For base addresses (BA0 through BA3), a start address in the space CS0 through CS3 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 MA3) 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.

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 CS3 spaces. The bit for which "1" is written is not included in address comparison.

CS0 to CS3 spaces have different address bits that can be masked by MA0 to MA3.

CS0 space and CS1 space: A29 through A14

CS2 space and CS3 space: A30 through A15

(Note 1) Address settings must be made using physical addresses.

(Note 2) CS areas must not be set in the internal area (0xFF00_0000~0xFFFF_FFFF).

Base and mask address setting registers BMA0 (0xFFFF_E400) to BMA3 (0xFFFF_E40C)

		7	6	5	4	3	2	1	0
BMA0	Bit symbol	MA0							
(0xFF00_1400)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
	Function	CS0 space size setting 0: Address for comparison							
		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."						CS0 space size setting 0: Address for comparison	
		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							
		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							

		7	6	5	4	3	2	1	0
BMA1	Bit symbol	MA1							
(0xFF00_1404)	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
	Function	CS1 space size setting 0: Address for comparison							
		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."						CS1 space size setting 0: Address for comparison	
		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							
		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							

(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 TMP19A44 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 Base and Mask Address Setting Registers (BMA0, BMA1)

BMA2
(0xFF00_1408)

	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	CS0 space size setting 0: Address for comparison							
	15	14	13	12	11	10	9	8
Bit symbol	MA2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	1	1
Function	Make sure that you write "0."							
	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							
	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							

BMA3
(0xFF00_140C)

	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	CS1 space size setting 0: Address for comparison							
	15	14	13	12	11	10	9	8
Bit symbol	MA3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	1	1
Function	Make sure that you write "0."							
	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							
	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							

(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 TMP19A44 is 16 MB and so bits 9 through 15 must be set to "0" as addresses A24 through A30 are not masked.

Fig. 9.2 Base and Mask Address Setting Registers (BMA2, BMA3)

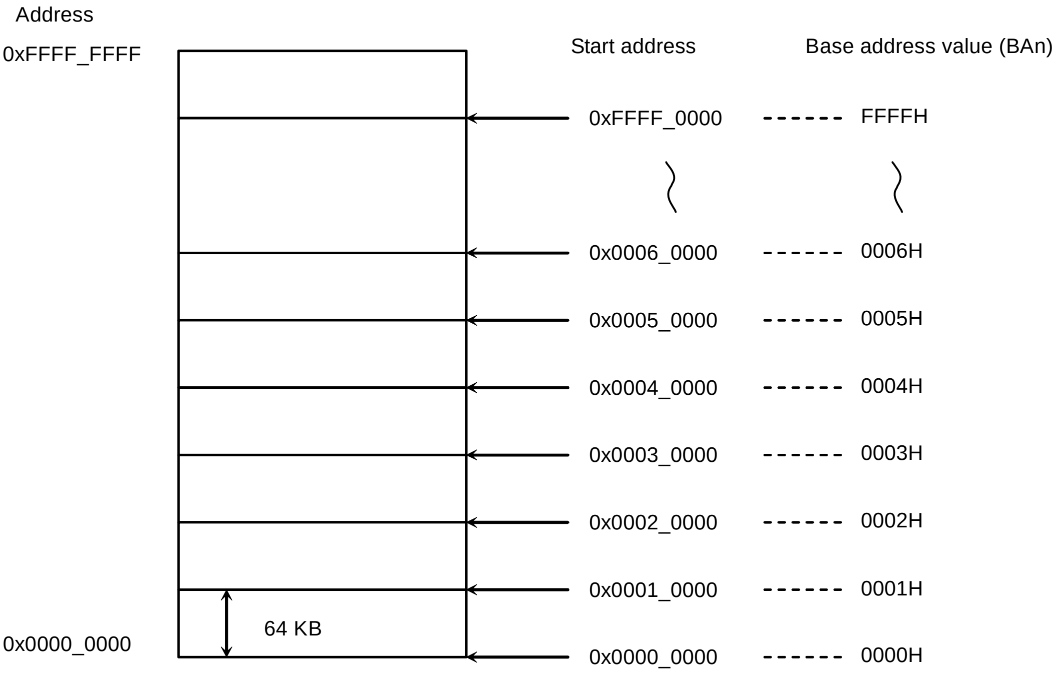


Fig. 9.3 Start and Base Address Register Values

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.

31															16	15															0
BA0																MA0															
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
C				0				0				0				0				0				0				3			

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 A14 is to be made or not. A comparison of A31 and A30 will definitely be made and to ensure a comparison of A29 through A24, set bits 15 to 10 of the mask address (MA0) to "0."

This setting allows A31 through A16 to be compared with the value specified as a start address. Therefore, a space of 64 KB from 0xC000_0000 to 0xC000_FFFF is designated as a CS0 space and the $\overline{\text{CS0}}$ 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.

31															16	15															0
BA2																MA2															
0	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1				F				D				0				0				0				1				F			

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

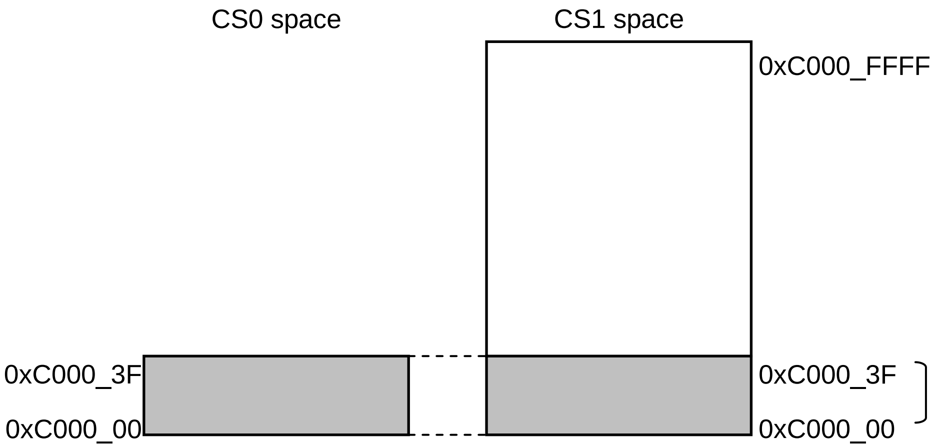
In the base address (BA2), specify "0x1FD0" that corresponds to higher 16 bits of a start address, while in the mask address (MA2), specify whether a comparison of addresses in the space A30 through A15 is to be made or not. A comparison of A31 will definitely be made and to ensure a comparison of A30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

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.

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

Table 9.1 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



If a space of 0xC000_0000 to 0xC000_3FFF is accessed, the CS0 space is selected.

Table 9.1 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		○	○	○	○	○	○	○	○	○	○

9.2 The Chip Selector and Wait Controller

The chip selector and wait controller registers are shown from the next page. For each address space (spaces CS0 through CS3 and other address spaces), each chip selector and wait controller register (B01CS through B23CS) 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>EXCS).

		7	6	5	4	3	2	1	0
B01CS	bit Symbol			B0BUS	B0W				
(0xFF00_1480)	Read/Write	R	R/W	R/W	R/W				
	After reset	0	0	1	0	0	0	0	1
	Function	“0” is read.	Write “0”.	Select data bus width 0: 8bit 1: 16bit	Specify the number of waits. (automatic WAIT insertion) 0_0000:0WAIT 0_0100:4WAIT 0_1000:8WAIT 0_1100:12WAIT 0_0001:1WAIT 0_0101:5WAIT 0_1001:9WAIT 0_1101:13WAIT 0_0010:2WAIT 0_0110:6WAIT 0_1010:10WAIT 0_1110:14WAIT 0_0011:3WAIT 0_0111:7WAIT 0_1011:11WAIT 0_1111:15WAIT (External wait input) x=2,4 1_0010: (2+ xN) WAIT 1_1001: (9+ xN) WAIT 1_0011: (3+ xN) WAIT 1_1010: (10+ xN) WAIT 1_0100: (4+ xN) WAIT 1_1011: (11+ xN) WAIT 1_0101: (5+ xN) WAIT 1_1100: (12+ xN) WAIT 1_0110: (6+ xN) WAIT 1_1101: (13+ xN) WAIT 1_0111: (7+ xN) WAIT 1_1110: (14+ xN) WAIT 1_1000: (8+ xN) WAIT 1_1111: (15+ xN) WAIT				
		15	14	13	12	11	10	9	8
	bit Symbol	B0CSCV			B0WCV		B0RCV		B0E
	Read/Write	R/W			R/W		R/W		R/W
	After reset	0	0	0	1	0	1	0	0
	Function	Specify the number of dummy cycles to be inserted. (CS0 recovery time) 000: Setting prohibited 001: 1 cycle 010~ 111: Setting prohibited			Specify the number of dummy cycles to be inserted. (write, recovery time) 00: Setting prohibited 01: 1 cycle 1x: Setting prohibited		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: Setting prohibited 01: 1 cycle 1x: Setting prohibited		Enable or disable CS0. 0: Disable 1: Enable
		23	22	21	20	19	18	17	16
	bit Symbol			B1BUS	B1W				
	Read/Write	R	R/W	R/W	R/W				
	After reset	0	0	1	0	0	0	0	1
	Function	“0” is read .	Write “0”.	Select data bus width. 0: 8bit 1: 16bit	Specify the number of waits. (automatic WAIT insertion) 0_0000:0WAIT 0_0100:4WAIT 0_1000:8WAIT 0_1100:12WAIT 0_0001:1WAIT 0_0101:5WAIT 0_1001:9WAIT 0_1101:13WAIT 0_0010:2WAIT 0_0110:6WAIT 0_1010:10WAIT 0_1110:14WAIT 0_0011:3WAIT 0_0111:7WAIT 0_1011:11WAIT 0_1111:15WAIT (external WAIT input) x=2,4 1_0010: (2+xN) WAIT 1_1001: (9+xN) WAIT 1_0011: (3+xN) WAIT 1_1010: (10+xN) WAIT 1_0100: (4+xN) WAIT 1_1011: (11+xN) WAIT 1_0101: (5+xN) WAIT 1_1100: (12+xN) WAIT 1_0110: (6+xN) WAIT 1_1101: (13+xN) WAIT 1_0111: (7+xN) WAIT 1_1110: (14+xN) WAIT 1_1000: (8+xN) WAIT 1_1111: (15+xN) WAIT				
		31	30	29	28	27	26	25	24
	bit Symbol	B1CSCV			B1WCV		B1RCV		B1E
	Read/Write	R/W			R/W		R/W		R/W
	After reset	0	0	0	1	0	1	0	0
	Function	Specify the number of dummy cycles to be inserted. (CS1 recovery time) 000: Setting prohibited 100: 4 cycles 001: 1 cycle 101: 6 cycles 010: 2 cycles 110: 8 cycles 011: 3 cycles 111: Setting prohibited			Specify the number of dummy cycles to be inserted. (write, recovery time) 00: Setting prohibited 01: 1 cycle 10: 2 cycles 11: 4 cycles		Specify the number of dummy cycles to be inserted. 00: Setting prohibited 01: 1 cycle 10: 2 cycles 11: 4 cycles		Enable or disable CS1 0: Disable 1: Enable

Fig. 9.4 Chip Selector and Wait Controller Registers 0, 1

B23CS
(0xFF00_1484)

	7	6	5	4	3	2	1	0
bit Symbol			B2BUS	B2W				
Read/Write	R	R/W	R/W	R/W				
After reset	0	0	1	0	0	0	0	1
Function	“0” is read.	Write “0”.	Select data bus width 0: 8bit 1: 16bit	Specify the number of waits. (automatic WAIT insertion) 0_0000:0WAIT 0_0100:4WAIT 0_1000:8WAIT 0_1100:12WAIT 0_0001:1WAIT 0_0101:5WAIT 0_1001:9WAIT 0_1101:13WAIT 0_0010:2WAIT 0_0110:6WAIT 0_1010:10WAIT 0_1110:14WAIT 0_0011:3WAIT 0_0111:7WAIT 0_1011:11WAIT 0_1111:15WAIT (External wait input) x=2,4 1_0010: (2+xN) WAIT 1_1001: (9+ xN) WAIT 1_0011: (3+ xN) WAIT 1_1010: (10+ xN) WAIT 1_0100: (4+ xN) WAIT 1_1011: (11+ xN) WAIT 1_0101: (5+ xN) WAIT 1_1100: (12+ xN) WAIT 1_0110: (6+ xN) WAIT 1_1101: (13+ xN) WAIT 1_0111: (7+ xN) WAIT 1_1110: (14+ xN) WAIT 1_1000: (8+ xN) WAIT 1_1111: (15+ xN) WAIT				

	15	14	13	12	11	10	9	8
bit Symbol	B2CSCV			B2WCV		B2RCV		B2E
Read/Write	R/W			R/W		R/W		R/W
After reset	0	0	0	1	0	1	0	0
Function	Specify the number of dummy cycles to be inserted. (CS2 recovery time) 000: Setting prohibited 100: 4 cycles 001: 1 cycle 101: 6 cycles 010: 2 cycles 110: 8 cycles 011: 3 cycles 111: Setting prohibited			Specify the number of dummy cycles to be inserted. (write, recovery time) 00: Setting prohibited 01: 1 cycle 10: 2 cycles 11: 4 cycles		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: Setting prohibited 01: 1 cycle 10: 2 cycles 11: 4 cycles		Enable or disable CS2 0: Disable 1: Enable

	23	22	21	20	19	18	17	16
bit Symbol			B3BUS	B3W				
Read/Write	R	R/W	R/W	R/W				
After reset	0	0	1	0	0	0	0	1
Function	“0” is read.	Write “0”.	Select data bus width 0: 8bit 1: 16bit	Specify the number of waits. (automatic WAIT insertion) 0_0000:0WAIT 0_0100:4WAIT 0_1000:8WAIT 0_1100:12WAIT 0_0001:1WAIT 0_0101:5WAIT 0_1001:9WAIT 0_1101:13WAIT 0_0010:2WAIT 0_0110:6WAIT 0_1010:10WAIT 0_1110:14WAIT 0_0011:3WAIT 0_0111:7WAIT 0_1011:11WAIT 0_1111:15WAIT (External wait input) x=2,4 1_0010: (2+ xN) WAIT 1_1001: (9+ xN) WAIT 1_0011: (3+ xN) WAIT 1_1010: (10+ xN) WAIT 1_0100: (4+ xN) WAIT 1_1011: (11+ xN) WAIT 1_0101: (5+ xN) WAIT 1_1100: (12+ xN) WAIT 1_0110: (6+ xN) WAIT 1_1101: (13+ xN) WAIT 1_0111: (7+ xN) WAIT 1_1110: (14+ xN) WAIT 1_1000: (8+ xN) WAIT 1_1111: (15+ xN) WAIT				

	31	30	29	28	27	26	25	24
bit Symbol	B3CSCV			B3WCV		B3RCV		B3E
Read/Write	R/W			R/W		R/W		R/W
After reset	0	0	0	1	0	1	0	0
Function	Specify the number of dummy cycles to be inserted. (CS3 recovery time) 000: Setting prohibited 100: 4 cycles 001: 1 cycle 101: 6 cycles 010: 2 cycles 110: 8 cycles 011: 3 cycles 111: Setting prohibited			Specify the number of dummy cycles to be inserted. (write, recovery time) 00: Setting prohibited 01: 1 cycle 10: 2 cycles 11: 4 cycles		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: Setting prohibited 01: 1 cycle 10: 2 cycles 11: 4 cycles		Enable or disable CS3. 0: Disable 1: Enable

Fig. 9.5 Chip Selector and Wait Controller Registers 2, 3

9.3 Bus Control Register

Fig. 9.6 shows the bus control register BUSCR that is capable of setting ALE width and the number of WAIT sampling.

BUSCR
(0xFF00_14C0)

	7	6	5	4	3	2	1	0
bit Symbol						WAITSMP	ALESEL	
Read/Write	R					R/W	R/W	
After reset	0	0	0	0	0	0	0	1
Function	“0” is read.					Specify the number of waits to be sampled 0: 2N 1: 4N	Multiplex bus 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles Separate bus 00: 0 cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles	

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

Fig. 9.6 Bus Control Register

<ALESEL1:0>: Setting for ALE width cycle differs depending on a bus to be used: separate bus or multiplex bus.

<WAITSMP>: Sampling point of WAIT input can be changed according to operating frequency.

Recommended value

2N: fsys = 4MHz~40MHz

4N: fsys = 40MHz~80MHz

10. DMA Controller (DMAC)

The TMP19A44 has a built-in 8-channel DMA Controller (DMAC).

10.1 Features

The DMAC of the TMP19A44 has the following features:

(1) DMA with 8 independent channels

(eight interrupt factors, INTDMA0 through INTDMA7)

(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: Level mode

(4) Transfer mode: Dual address mode

(5) Transfer devices: Memory space transfer

(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

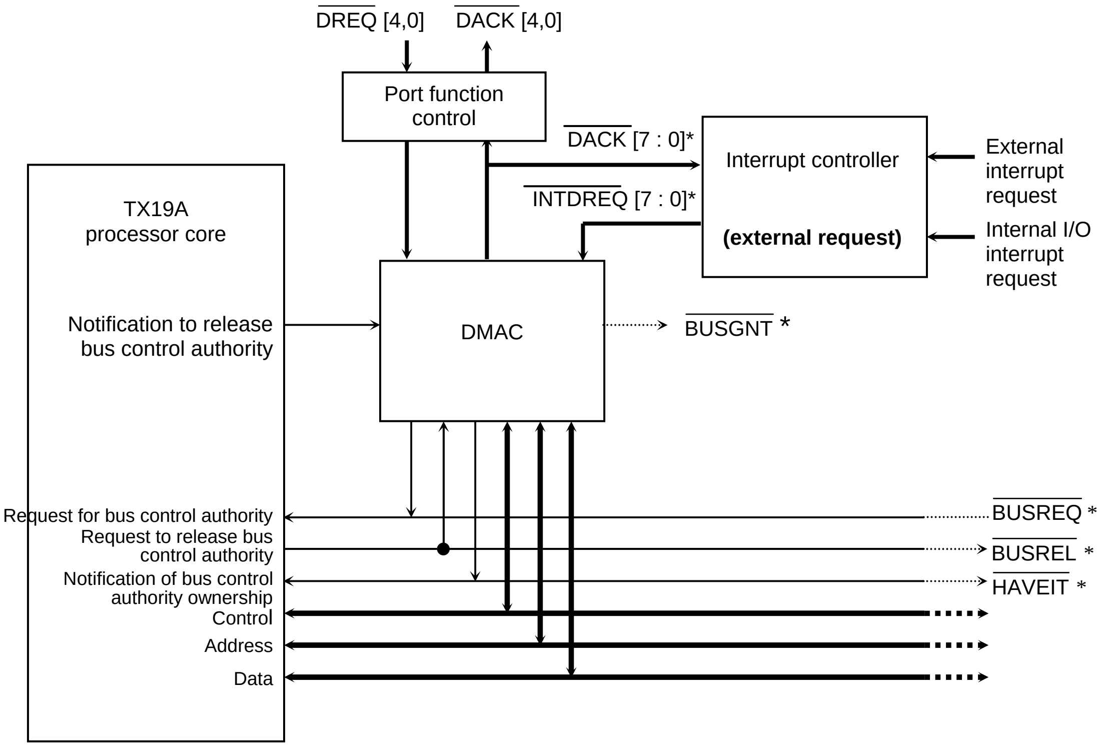
(8) Channel priority: Fixed (in ascending order of channel numbers)

(9) Endian switchover function

10.2 Configuration

10.2.1 Internal Connections of the TMP19A44

Fig. 10.1 shows the internal connections with the DMAC in the TMP19A44.


DREQ [4,0]
DACK [4,0]
Port function control
DACK [7 : 0]*
INTDREQ [7 : 0]*
Interrupt controller
(external request)
External interrupt request
Internal I/O interrupt request
TX19A processor core
Notification to release bus control authority
DMAC
BUSGNT *
Request for bus control authority
Request to release bus control authority
Notification of bus control authority ownership
Control
Address
Data
BUSREQ *
BUSREL *
HAVEIT *

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

Fig. 10.1 DMAC Connections in the TMP19A44

The DMAC has eight DMA channels. Each of these channels handles the data transfer request signal (INTDREOn) from the interrupt controller and the acknowledgment signal (DACKn) generated in response to INTDREOn, where "n" is a channel number from 0 to 7. External pins (DREQ0 and DREQ4) are internally wired to allow them to function as pins of the port F. To use them as pins of the port F, they must be selected by setting the function control register PFFC to an appropriate setting.

Pins, DACK0 and DACK4, handle the data transfer request and acknowledge signal output supplied through external pins, DREQ0 and DREQ4. 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/ H1 processor core has a snoop function. Using the snoop function, the TX19A/ H1 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 shows the internal blocks of the DMAC.

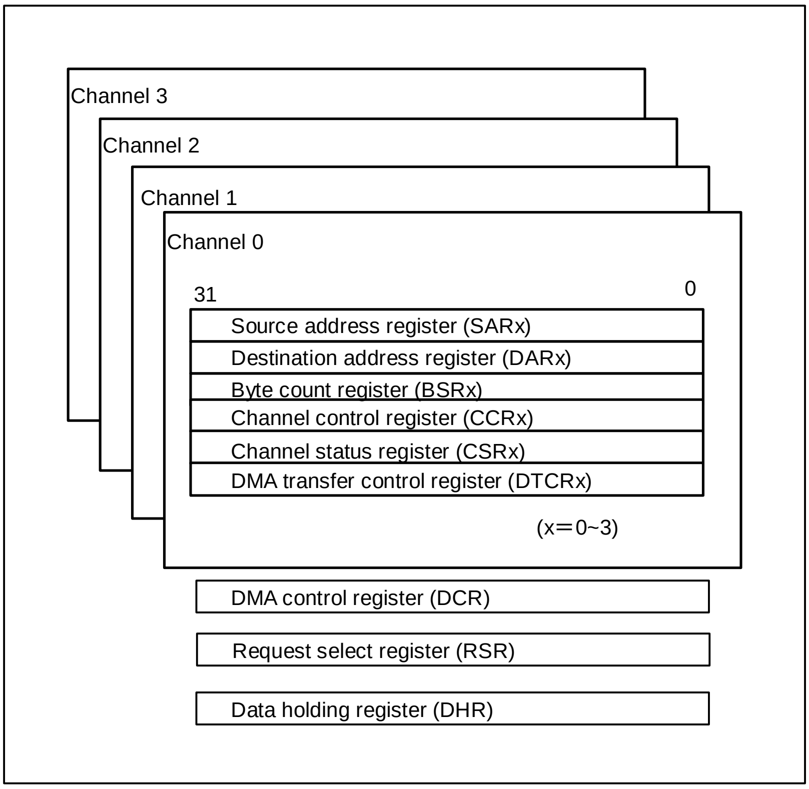

Channel 3
Channel 2
Channel 1
Channel 0
31
0
Source address register (SARx)
Destination address register (DARx)
Byte count register (BSRx)
Channel control register (CCRx)
Channel status register (CSRx)
DMA transfer control register (DTCRx)
(x＝0~3)
DMA control register (DCR)
Request select register (RSR)
Data holding register (DHR)

Fig. 10.2 DMAC Internal Blocks

10.2.3 Snoop Function

The TX19A/ H1 processor core has a snoop function. If the snoop function is activated, the TX19A/ H1 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/ H1 processor core attempts to access memory space 1by 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/ H1 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 10.1 shows the register map of the DMAC.

Table 10.1 DMAC Registers (1 of 2)

Address	Register symbol	Register name
0xFF00_1200	CCR0	Channel control register (ch. 0)
0xFF00_1204	CSR0	Channel status register (ch. 0)
0xFF00_1208	SAR0	Source address register (ch. 0)
0xFF00_120C	DAR0	Destination address register (ch. 0)
0xFF00_1210	BCR0	Byte count register (ch. 0)
0xFF00_1218	DTCR0	DMA transfer control register (ch. 0)
0xFF00_1220	CCR1	Channel control register (ch. 1)
0xFF00_1224	CSR1	Channel status register (ch. 1)
0xFF00_1228	SAR1	Source address register (ch. 1)
0xFF00_122C	DAR1	Destination address register (ch. 1)
0xFF00_1230	BCR1	Byte count register (ch. 1)
0xFF00_1238	DTCR1	DMA transfer control register (ch. 1)
0xFF00_1240	CCR2	Channel control register (ch. 2)
0xFF00_1244	CSR2	Channel status register (ch. 2)
0xFF00_1248	SAR2	Source address register (ch. 2)
0xFF00_124C	DAR2	Destination address register (ch. 2)
0xFF00_1250	BCR2	Byte count register (ch. 2)
0xFF00_1258	DTCR2	DMA transfer control register (ch. 2)
0xFF00_1260	CCR3	Channel control register (ch. 3)
0xFF00_1264	CSR3	Channel status register (ch. 3)
0xFF00_1268	SAR3	Source address register (ch. 3)
0xFF00_126C	DAR3	Destination address register (ch. 3)
0xFF00_1270	BCR3	Byte count register (ch. 3)
0xFF00_1278	DTCR3	DMA transfer control register (ch. 3)
0xFF00_1280	CCR4	Channel control register (ch. 4)
0xFF00_1284	CSR4	Channel status register (ch. 4)
0xFF00_1288	SAR4	Source address register (ch. 4)
0xFF00_128C	DAR4	Destination address register (ch. 4)
0xFF00_1290	BCR4	Byte count register (ch. 4)
0xFF00_1298	DTCR4	DMA transfer control register (ch. 4)
0xFF00_12A0	CCR5	Channel control register (ch. 5)
0xFF00_12A4	CSR5	Channel status register (ch. 5)
0xFF00_12A8	SAR5	Source address register (ch. 5)
0xFF00_12AC	DAR5	Destination address register (ch. 5)
0xFF00_12B0	BCR5	Byte count register (ch. 5)
0xFF00_12B8	DTCR5	DMA transfer control register (ch. 5)
0xFF00_12C0	CCR6	Channel control register (ch. 6)
0xFF00_12C4	CSR6	Channel status register (ch. 6)
0xFF00_12C8	SAR6	Source address register (ch. 6)
0xFF00_12CC	DAR6	Destination address register (ch. 6)
0xFF00_12D0	BCR6	Byte count register (ch. 6)
0xFF00_12D8	DTCR6	DMA transfer control register (ch. 6)

Table 10.2 DMAC Registers (2 of 2)

0xFF00_12E0	CCR7	Channel control register (ch. 7)
0xFF00_12E	CSR7	Channel status register (ch. 7)
0xFF00_12E8	SAR7	Source address register (ch. 7)
0xFF00_12EC	DAR7	Destination address register (ch. 7)
0xFF00_12F0	BCR7	Byte count register (ch. 7)
0xFF00_12F8	DTCR7	DMA transfer control register (ch. 7)
0xFF00_1300	DCR	DMA control register (DMAC)
0xFF00_1304	RSR	Request select register (DMAC)
0xFF00_130C	DHR	Data holding register (DMAC)

10.3.1 DMA Control Register (DCR)

DCR
(0xFF00_1300)

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

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 DMA Control Register (DCR)

When software is reset, the CCRn,CSRn,SARn,DARn,DTCRn,DCR,RSR register is initialized.
The BCRn,DHR register is not initialized.

(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)

CCRn

	7	6	5	4	3	2	1	0
bit Symbol	SAC	DIO	DAC		TrSiz		DPS	
Read/Write	R/W	R/W	R/W		R/W		R/W	
After reset	0	0	0					
Function	See detailed description.	Always set this bit to "0."	See detailed description.					

	15	14	13	12	11	10	9	8
bit Symbol		ExR	PosE	Lev	SReq	RelEn	SIO	SAC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0							
Function	Always set this bit to "0."	See detailed description.						

	23	22	21	20	19	18	17	16
bit Symbol	NIEn	AblEn					Big	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1		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							
Function	See detailed description.							Always set this bit to "0."

Fig. 10.4 Channel Control Registers (CCRn) (1 of 2)

Bit	Mnemonic	Field name	Description
31	Str	Channel start	Start (initial value:–) 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 it's 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."
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. (only for 0ch and 4ch) 1: External transfer request (interrupt request or external $\overline{\text{DREQn}}$ request) 0: Internal transfer request (software initiated)
13	PosE	Positive edge	Positive Edge (initial value: 0) The effective level of the transfer request signal $\overline{\text{INTDREQn}}$ or $\overline{\text{DREQn}}$ 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 $\overline{\text{INTDREQn}}$ and $\overline{\text{DREQn}}$ 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 $\overline{\text{INTDREQn}}$ or $\overline{\text{DREQn}}$ signal or the "L" level is effective. The $\overline{\text{DACKn}}$ 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 $\overline{\text{INTDREQn}}$ signal is active at "L" level, make sure that you set the Lev bit to "1." The state of active $\overline{\text{DREQn}}$ is determined by the Lev bit setting. 1: Level mode The level of the $\overline{\text{DREQn}}$ 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 $\overline{\text{DREQn}}$ 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/ H1 processor core is enabled and the DMAC can use the data bus of the TX19A /H1 processor core. If the snoop function is not used, the snoop function of the TX19A/ H1 processor core does not work. 1: Use snoop function (SREQ) 0: Do not use snoop function (GREQ)

Bit	Mnemonic	Field name	Description
10	RelEn	Bus control release request enable	Release Request Enable (initial value: 0) Acknowledgment of the bus control release request made by the TX19A/ H1 processor core is specified. This function is valid only if GREQ is generated. If SREQ is generated, the TX19A/ H1 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/ H1 processor core issues a bus control release request, the DMAC relinquishes control of the bus to the TX19A/ H1 processor core during a pause in bus operation. 0: The bus control release request is not acknowledged.
9	SIO	Transfer type selection	Transfer type selection: (initial value: 0) 1 Single transfer 0: Continuous transfer (Data is transferred successively until BCRx becomes "0")
8 : 7	SAC	Source address count	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	(Reserved)	This is a reserved bit. Always set this bit to “0”.
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) *This needs to be set to the same size as that of the device port size (DPS).
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) *This needs to be set to the same size as that of the transfer unit (TrSiz).

Fig. 10.4 Channel Control Registers (CCRn) (2 of 2)

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

(Note 2) In executing continuous data transfer, a value set in DPS becomes invalid.

(Note 3) Set the mode first and then set the <Str> bit.

10.3.3 Request Select Register (RSR)

RSR
(0xFF00_1304)

	7	6	5	4	3	2	1	0
bit Symbol				ReqS4				ReqS0
Read/Write				R/W				R/W
After reset	0							
Function	Always set this bit to "0."			See detailed description.	Always set this bit to "0."			See detailed description.
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write								
After reset	0							
Function								
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write								
After reset	0							
Function								
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write								
After reset	0							
Function								

Bit	Mnemonic	Field name	Description
4	ReqS4	Request select (ch.4)	Request Select (initial value: 0) Selects a source of the external transfer request for the DMA channel 4. 1: Request made by $\overline{\text{DREQ4}}$ 0: Request made by the interrupt controller (INTC)
0	ReqS0	Request select (ch.0)	Request Select (initial value: 0) Selects a source of the external transfer request for the DMA channel 0. 1: Request made by $\overline{\text{DREQ0}}$ 0: Request made by the interrupt controller (INTC)

(Note) Make sure that you write "0" to bits 1 through 3 and 5 through 7 of the RSR register.

Fig. 10.5 DMA Control Register (RSR)

10.3.4 Channel Status Registers (CSRn)

CSRn

	7	6	5	4	3	2	1	0
bit Symbol								
Read/Write						R/W		
After reset	0							
Function						Always set this bit to "0."		
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write								
After reset	0							
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		
After reset	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							
After reset	0							
Function	See detailed description.							

Fig. 10.6 Channel Status Register (CSRn) (1 of 2)

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.
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. 10.6 Channel Status Register (CSRn) (2 of 2)

10.3.5 Source Address Registers (SARn)

SARn

	7	6	5	4	3	2	1	0
bit Symbol	SAddr7	SAddr6	SAddr5	SAddr4	SAddr3	SAddr2	SAddr1	SAddr0
Read/Write	R/W							
After reset	Indeterminate							
Function	See detailed description.							
	15	14	13	12	11	10	9	8
bit Symbol	SAddr15	SAddr14	SAddr13	SAddr12	SAddr11	SAddr10	SAddr9	SAddr8
Read/Write	R/W							
After reset	Indeterminate							
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	Indeterminate							
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	Indeterminate							
Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31 : 0	SAddr	Source address	Source Address (initial value: –) 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. 10.7 Source Address Register (SARn)

10.3.6 Destination Address Register (DARn)

DARn

	7	6	5	4	3	2	1	0
bit Symbol	DAddr7	DAddr6	DAddr5	DAddr4	DAddr3	DAddr2	DAddr1	DAddr0
Read/Write	R/W							
After reset	Indeterminate							
Function	See detailed description.							
	15	14	13	12	11	10	9	8
bit Symbol	DAddr15	DAddr14	DAddr13	DAddr12	DAddr11	DAddr10	DAddr9	DAddr8
Read/Write	R/W							
After reset	Indeterminate							
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	Indeterminate							
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	Indeterminate							
Function	See detailed description.							

Bit	Mnemonic	Field name	Description
31 : 0	DAddr	Destination address	Destination Address (initial value: –) 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. 10.8 Destination Address Register (DARn)

10.3.7 Byte Count Registers (BCRn)

BCRn

	7	6	5	4	3	2	1	0
bit Symbol	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
Read/Write	R/W							
After reset	Indeterminate							
Function	See detailed description.							
	15	14	13	12	11	10	9	8
bit Symbol	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
Read/Write	R/W							
After reset	Indeterminate							
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	Indeterminate							
Function	See detailed description.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write								
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. 10.9 Byte Count Register (BCRn)

10.3.8 DMA Transfer Control Register (DTCRn)

DTCRn

	7	6	5	4	3	2	1	0
bit Symbol			DACM			SACM		
Read/Write			R/W			R/W		
After reset	0							
Function			See detailed description.			See detailed description.		
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write								
After reset	0							
Function								
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write								
After reset	0							
Function								
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write								
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. 10.10 DMA Transfer Control Register (DTCRn)

10.3.9 Data Holding Register (DHR)

DHR
(0xFF00_130C)

	7	6	5	4	3	2	1	0
bit Symbol	DOT7	DOT6	DOT5	DOT4	DOT3	DOT2	DOT1	DOT0
Read/Write	R/W							
After reset	Indeterminate							
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	Indeterminate							
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	Indeterminate							
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	Indeterminate							
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. 10.11 Data Holding Register (DHR)

10.4 Functions

The DMAC is a 32-bit DMA controller capable of transferring data in a system using the TX19A/ H1 processor core at high speeds without routing data via the core.

10.4.1 Overview

(1) Source and destination

The DMAC handles data transfers from memory to memory. 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. Memory can be designated as a source or destination device.

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/ H1 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, a request made to the DMAC is cleared after completion of each data transfer (transfer of the amount of data specified by TrSiz). On the other hand, during a continuous transfer, 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 TMP19A44, 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/ H1 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/ H1 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/ H1 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/ H1 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) Do not bring the TX19A to a halt when the DMAC is in operation.

(Note 2) 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 $\overline{\text{INTDREQn}}$ output by the INTC is input, or when a transfer request signal $\overline{\text{DREQn}}$ 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 $\overline{\text{INTDREQn}}$ signal is detected and a mode in which a transfer request is generated when the falling edge or "L" level of the $\overline{\text{DREQn}}$ signal is detected.

(4) Address mode

For the DMAC of the TMP19A44, only one address mode is provided: a dual address mode. A single address mode is not available.

In the dual address mode, both single and continuous transfers are available. Source and destination device addresses are output by the DMAC. To access an I/O device, the DMAC asserts the DACKn 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. 10.12 shows the state transitions of channel operation.

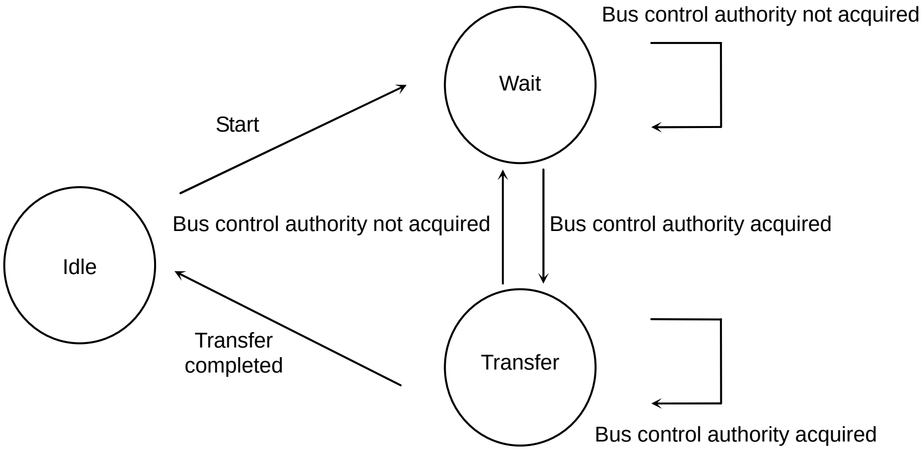

Bus control authority not acquired
Wait
Start
Idle
Bus control authority not acquired
Bus control authority acquired
Transfer
completed
Transfer
Bus control authority acquired

Fig. 10.12 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 method
Internal	—	Dual	Continuous
External	"L" level ($\overline{\text{INTDREQn}}$)		Single
External	"L" level ($\overline{\text{DREQn}}$)		Continuous
	Falling edge ($\overline{\text{DREQn}}$)		Single

(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. If a single transfer 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_1004	0xB000_0010
3rd	0xA000_1008	0xB000_0020
4th	0xA000_100C	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

- 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. The external transfer request is used for a single and a continuous transfer.

  The TMP19A44 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 DACKn signal. This DACKn signal is asserted only if a bus cycle for single transfer or the number of bytes (value set in the BCRn register) transferred at continuous transfer becomes "0." Therefore, at the single transfer, the amount of data specified by TrSiz is transferred only once because INTDREQn is cleared upon completion of one data transfer from one transfer request. On the other hand, at the continuous transfer, it can be transferred successively in response to a transfer request because 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 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 (DREQ0 and DREQ4) are internally wired to allow them to function as pins of the port 5 and the port A. These pins can be selected by setting the function control register PFFC to an appropriate setting.

In the edge mode, the DREQn signal must be negated 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. At the continuous transfer, only the "L" level mode can be used. At the single transfer, only the falling edge mode can be used.

– Level mode

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

If an external circuit asserts the DREQn signal, the DREQn signal must be maintained at the "L" level until the DACKn signal is asserted. If the DREQn signal is negated before the DACKn signal is asserted, a transfer request may not be recognized.

If the 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.

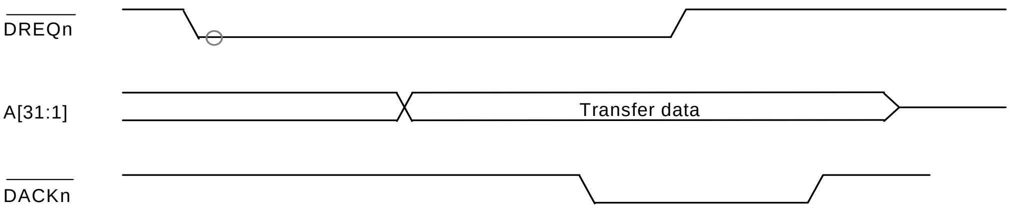


Fig. 10.13 Transfer Request Timing (Level Mode)

– 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.

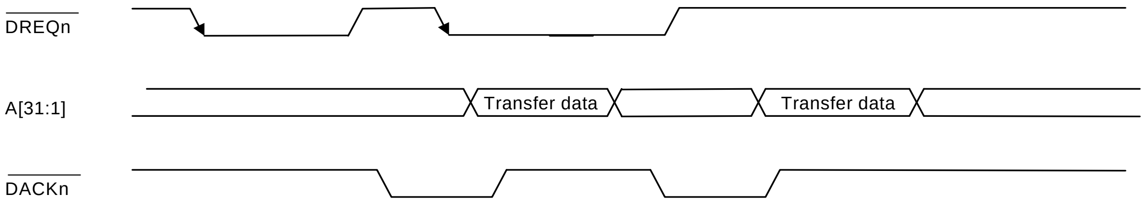


Fig. 10.14 Transfer Request Timing (Edge Mode)

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 TMP19A44 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.

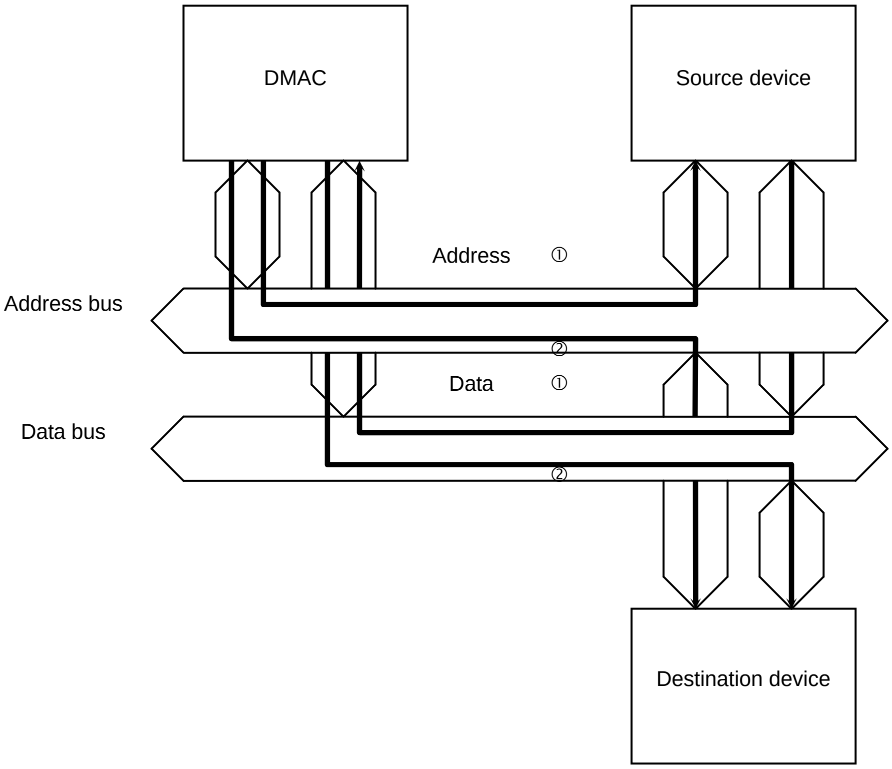


Fig. 10.15 Basic Concept of Data Transfer in the Dual Address Mode

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.

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:

  – 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.

  – 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 four 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 ($\overline{\text{INTDMAn}}$: DMA transfer completion interrupt) to the TX19A/ H1 processor core with two types of interrupts available: a normal completion interrupt and an abnormal completion interrupt.

$\overline{\text{INTDMA0}}$: 0ch, $\overline{\text{INTDMA1}}$: 1ch, $\overline{\text{INTDMA2}}$: 2ch, $\overline{\text{INTDMA3}}$: 3ch

$\overline{\text{INTDMA4}}$: 4ch, $\overline{\text{INTDMA5}}$: 5ch, $\overline{\text{INTDMA6}}$: 6ch, $\overline{\text{INTDMA7}}$: 7ch

- 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/ H1 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/ H1 processor core to authorize an interrupt.

(Note) The DMA transfer completion interrupt comes in two types: INTDMA0 for 0ch through 3ch and INTDMA1 for 4ch through 7ch.

10.5 Timing Diagrams

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

10.5.1 Dual Address Mode

- Continuous transfer

  Fig. 10.16 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."

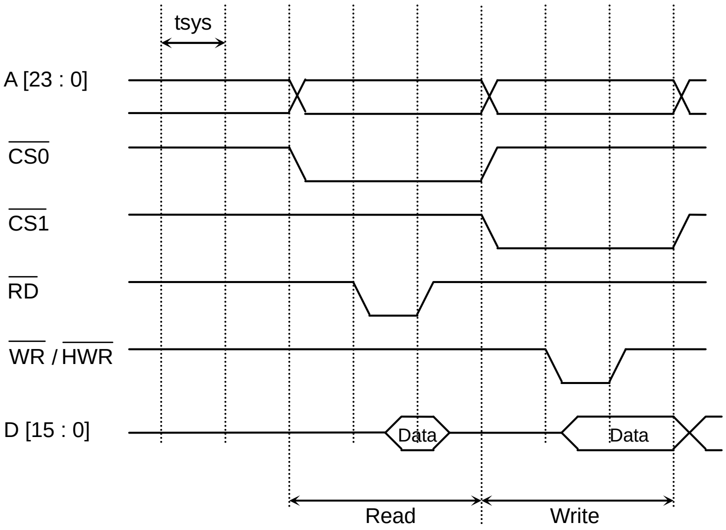


Fig. 10.16 Dual Address Mode (continuous)

- Single transfer (1)

  Fig. 10.17 shows an example of the timing with which data is transferred from one external memory to another if the unit of data to be transferred is set to 16 bits and if the device port size is set to 16 bits.

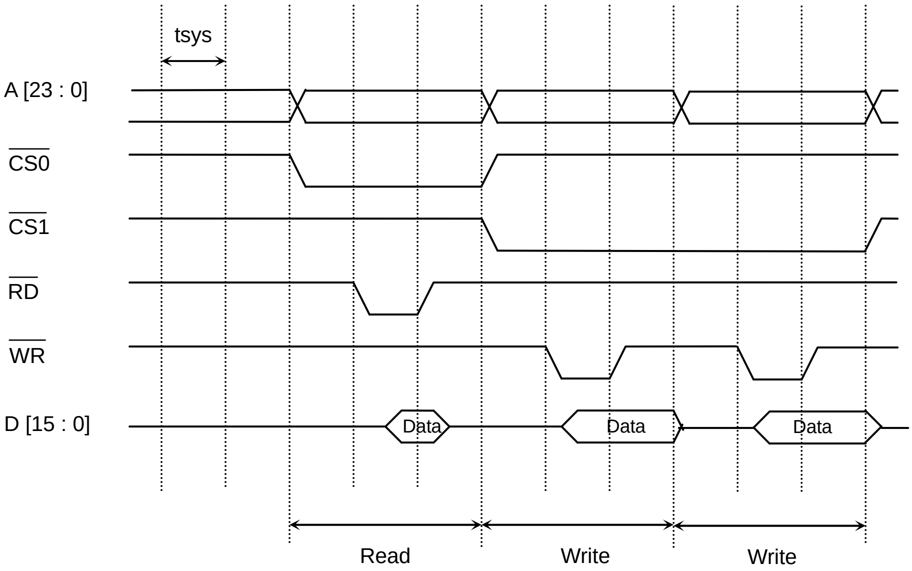


Fig. 10.17 Dual Address Mode (single transfer)

- Single transfer (2)

  Fig. 10.18 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.

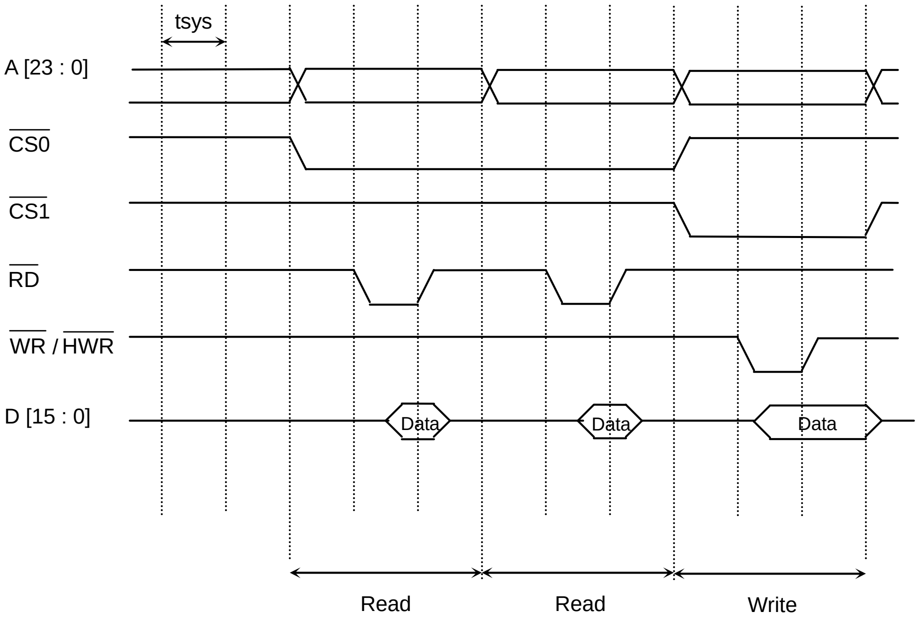


Fig. 10.18 Dual Address Mode (single transfer)

10.5.2 DREQn-Initiated Transfer Mode

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

  Fig. 10.19 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

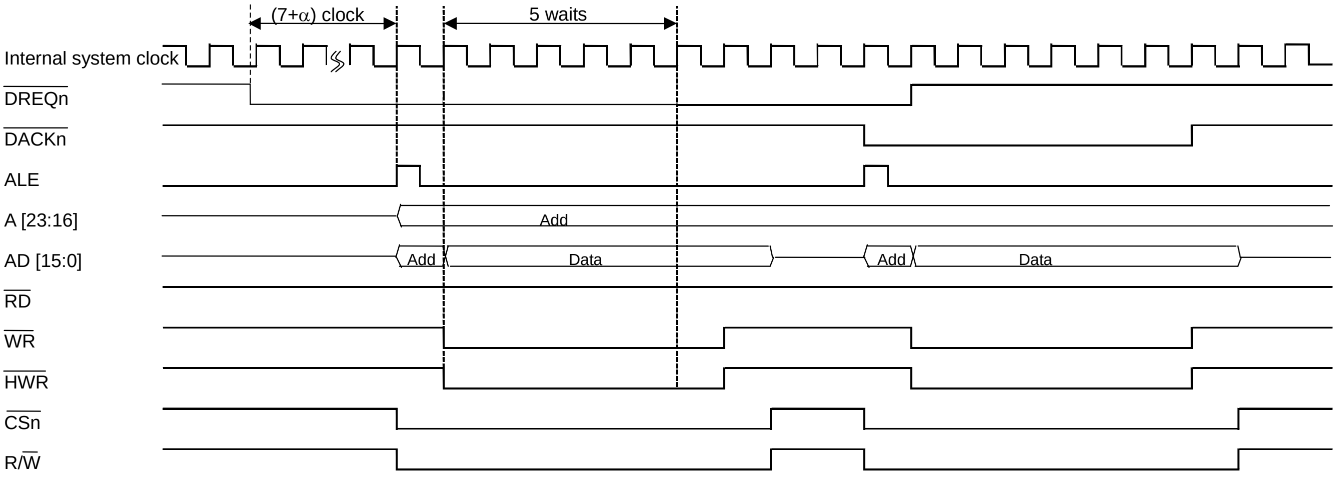


Fig. 10.19 Level Mode (from Internal RAM to External Memory)

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

  Fig. 10.20 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bit width) to internal RAM.

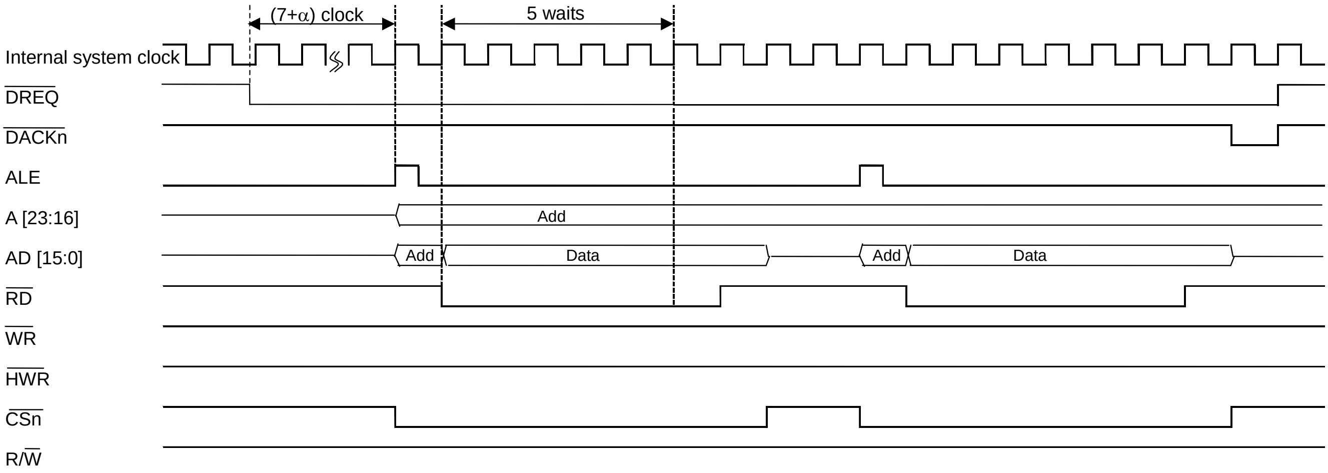


Fig. 10.20 Level Mode (from External Memory to Internal RAM)

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

  Fig. 10.21 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

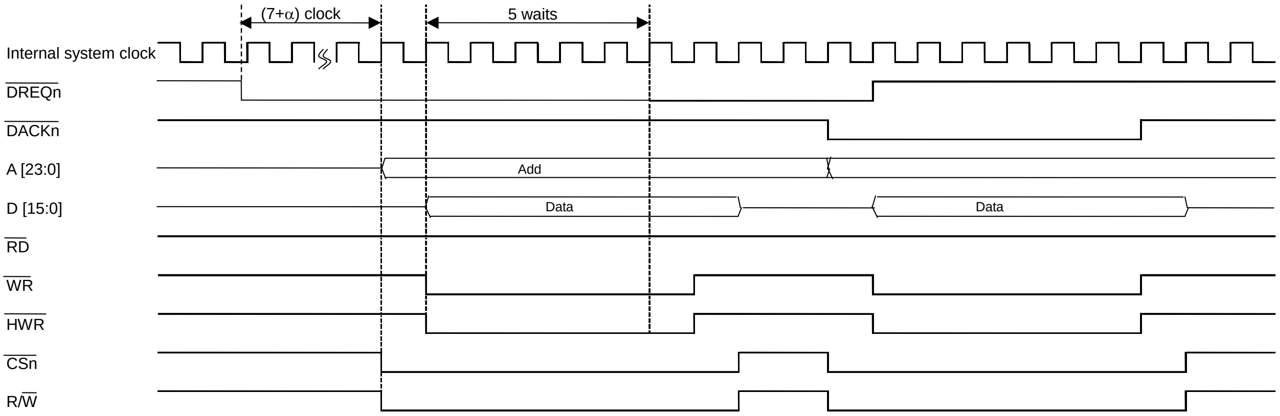


Fig. 10.21 Level Mode (Internal RAM to External Memory)

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

  Fig. 10.22 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bid width) to internal RAM.

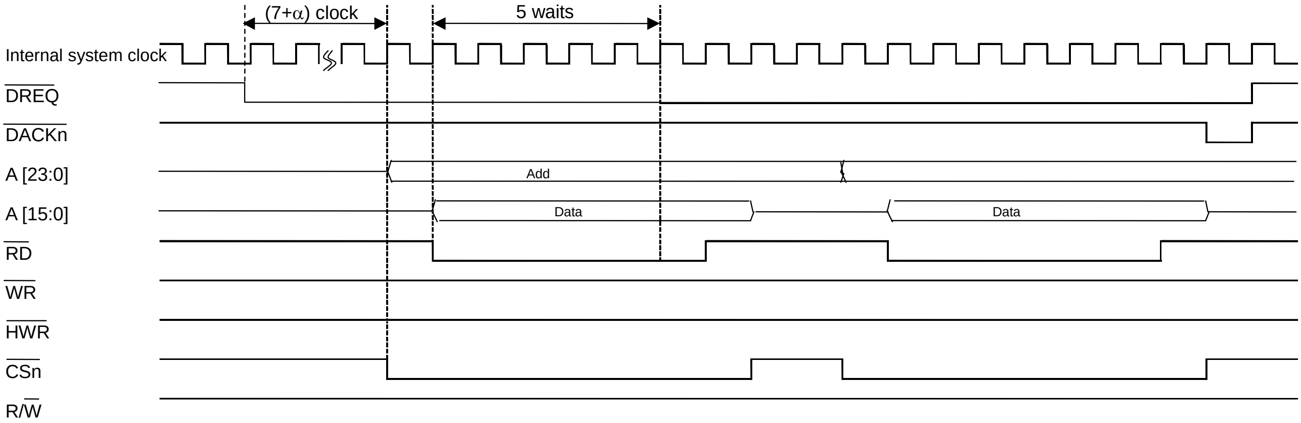


Fig. 10.22 Level Mode (from External Memory to Internal RAM)

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

  Fig. 10.23 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

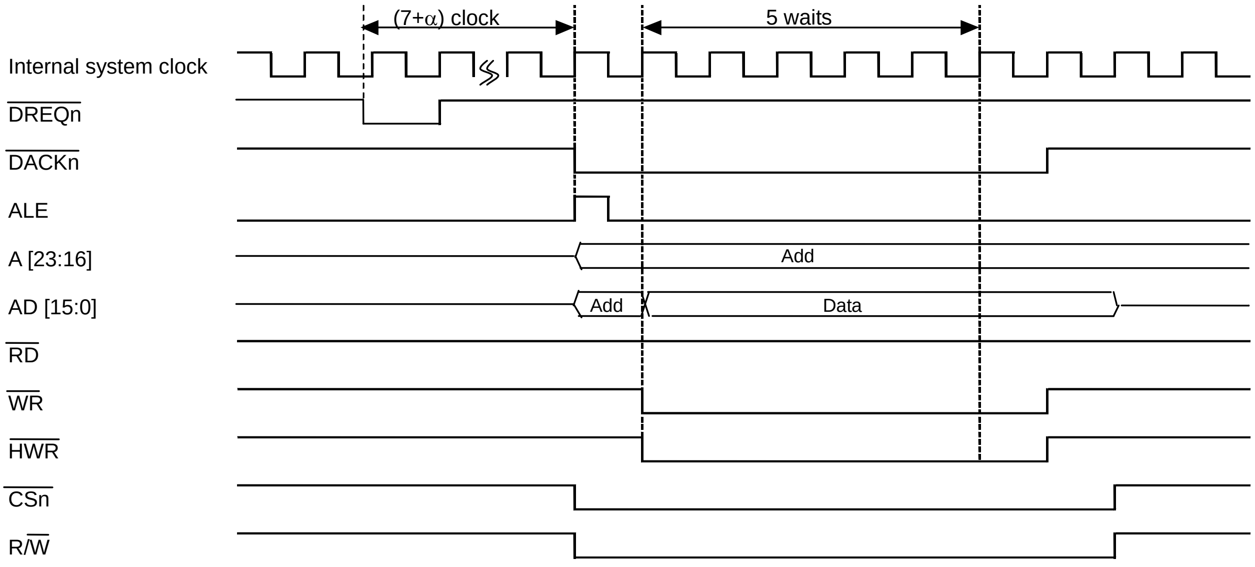


Fig. 10.23 Edge Mode (from Internal RAM to External Memory)

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

  Fig. 10.24 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

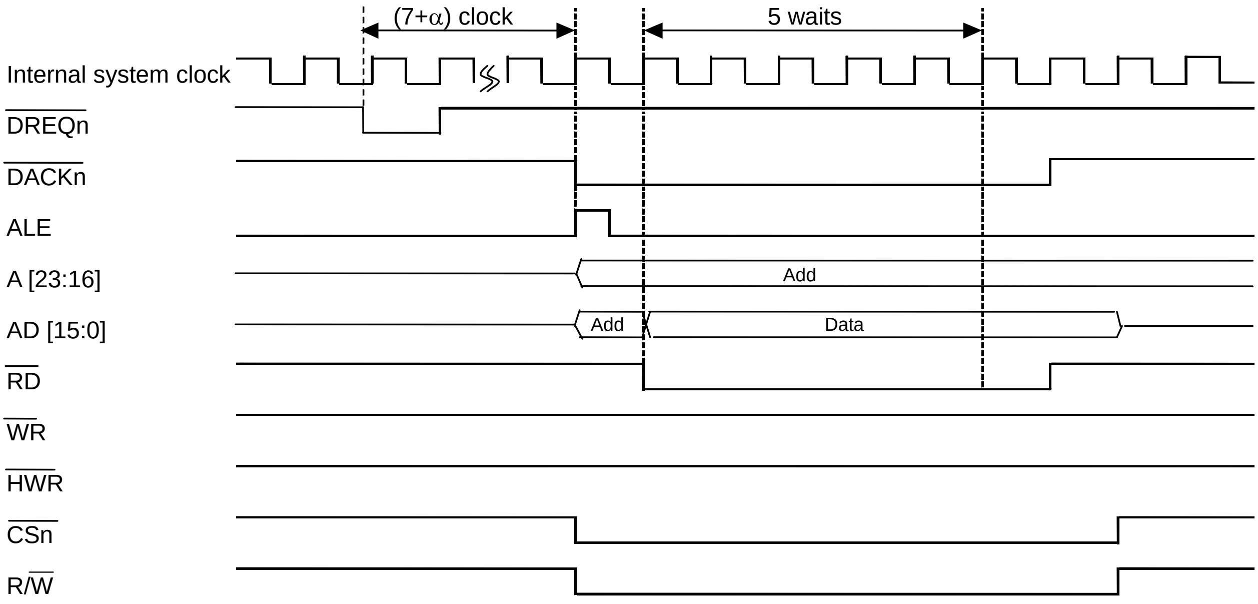


Fig. 10.24 Edge Mode (from External Memory to Internal RAM)

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

  Fig. 10.25 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

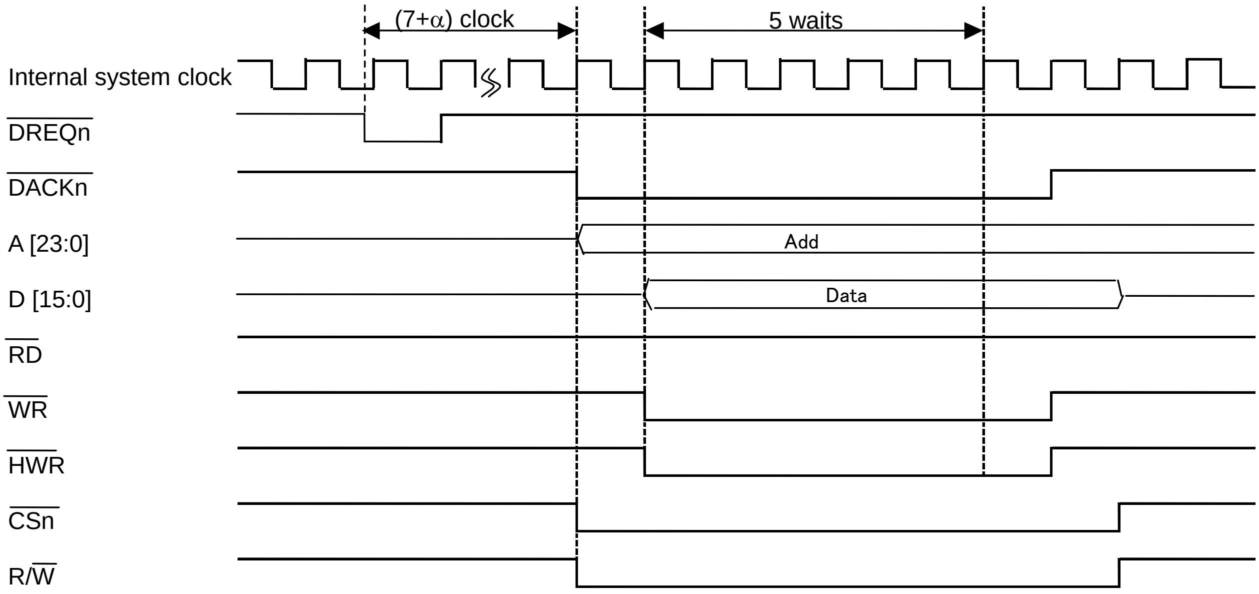


Fig. 10.25 Edge Mode (from Internal RAM to External Memory)

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

  Fig. 10.26 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

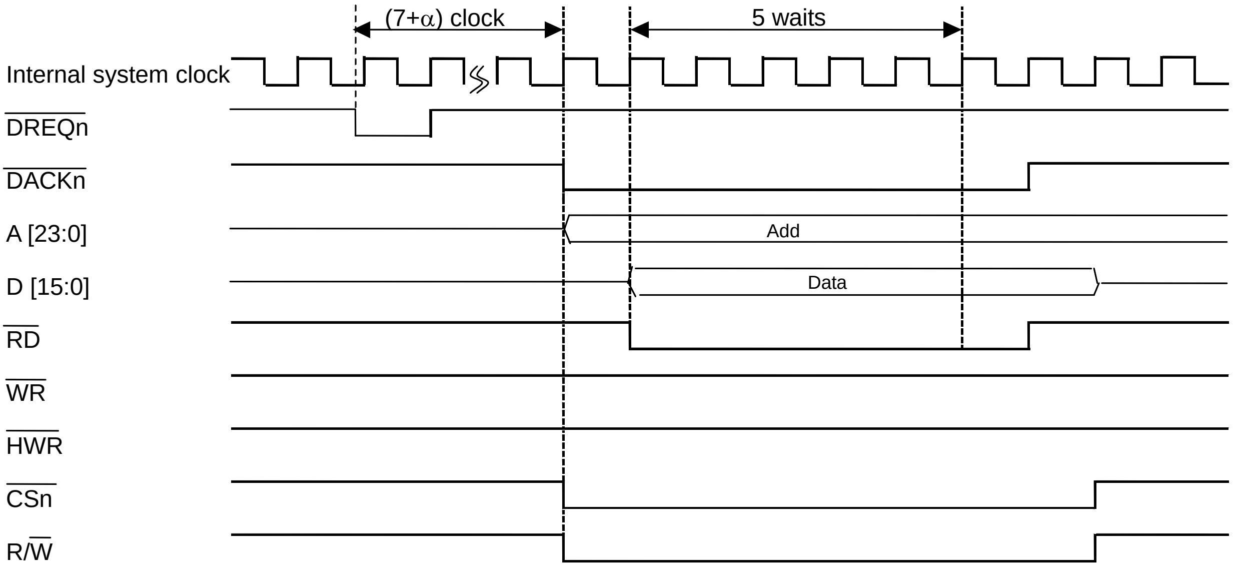


Fig. 10.26 Edge Mode (from External Memory to Internal RAM)

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_A000
- Number of bytes transferred: 256 bytes

<Serial channel setting>

- Data length 8 bits: UART
- Serial channel: 1 ch
- Transfer rate: 9600 bps

<SIO ch.1 setting>

IMC09	←	x111, x000	/* assigned to DMC0 activation factor * /
INTCLR	←	0x098	/* IVR [8:0], INTRX1 interrupt factor * /
SC1MOD0	←	0x29	/* UART mode, 8-bit length, baud rate generator * /
SC1CR	←	0x00	
BR1CR	←	0x1F	/*ϕT4, N=15*/

<DMA0 setting>

DCR	←	0x8000_0000	/* DMA reset * /
IMC17	←	x000, x000	/* disable interrupt * /
INTCLR	←	0x17C	/* IVR [8:0] value * /
IMC17	←	x000, x100	/* level = 4 (any given value) */
DTCR0	←	0x0000_0000	/* DACM = 000 * / /* SACM = 000 * /
SAR0	←	0xFF00_4C44	/* 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 */

(Contents)

31				27				23				19			
1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

15				11				7				3			
0	1	0	1	1	x	1	1	x	0	0	0	1	1	1	1

10.7 DMAC Transfer Request Clear Registers (DREQFLG)

Setting the DREQ [7:0] for the corresponding factor into the DREQFLG register enables to clear any DMAC transfer request.

DREQ [7:0] value setting to clear the DMAC transfer request

DREQFLG
(0xFF00_10C4)

	7	6	5	4	3	2	1	0
bit Symbol	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Corresponding DMAC transfer request is cleared.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” is read.							

Reading 0: DMAC transfer is requested.

1: DMAC transfer is not requested.

Writing 0: Invalid

1 :DMAC transfer request is cleared.

11. 16-bit Timer/Event Counters (TMRBs)

Each of the eighteen channels (TMRB0 through TMR11) 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 (simultaneous output in units of four channels can be programmed)
- Timer synchronous mode

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 (double-buffered), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit. Timer operation modes and the timer flip-flop are controlled by a 13-byte register.

Each channel (TMRB0 through TMRB11) functions independently and while the channels operate in the same way, there are differences in their specifications as shown in Table 11.1. Therefore, the operational descriptions here are for TMRB0 only.

The channels shown below are used as the capture or start trigger.

(1) The flip-flop output of TMRB 0, TMRB 8 and TMRB 10 can be used as the capture trigger of other channels.

- TB0OUT => available for TMRB 1 through TMRB 7
- TB8OUT => available for TMRB 9 through TMRB F
- TB10OUT => available for TMRB 11

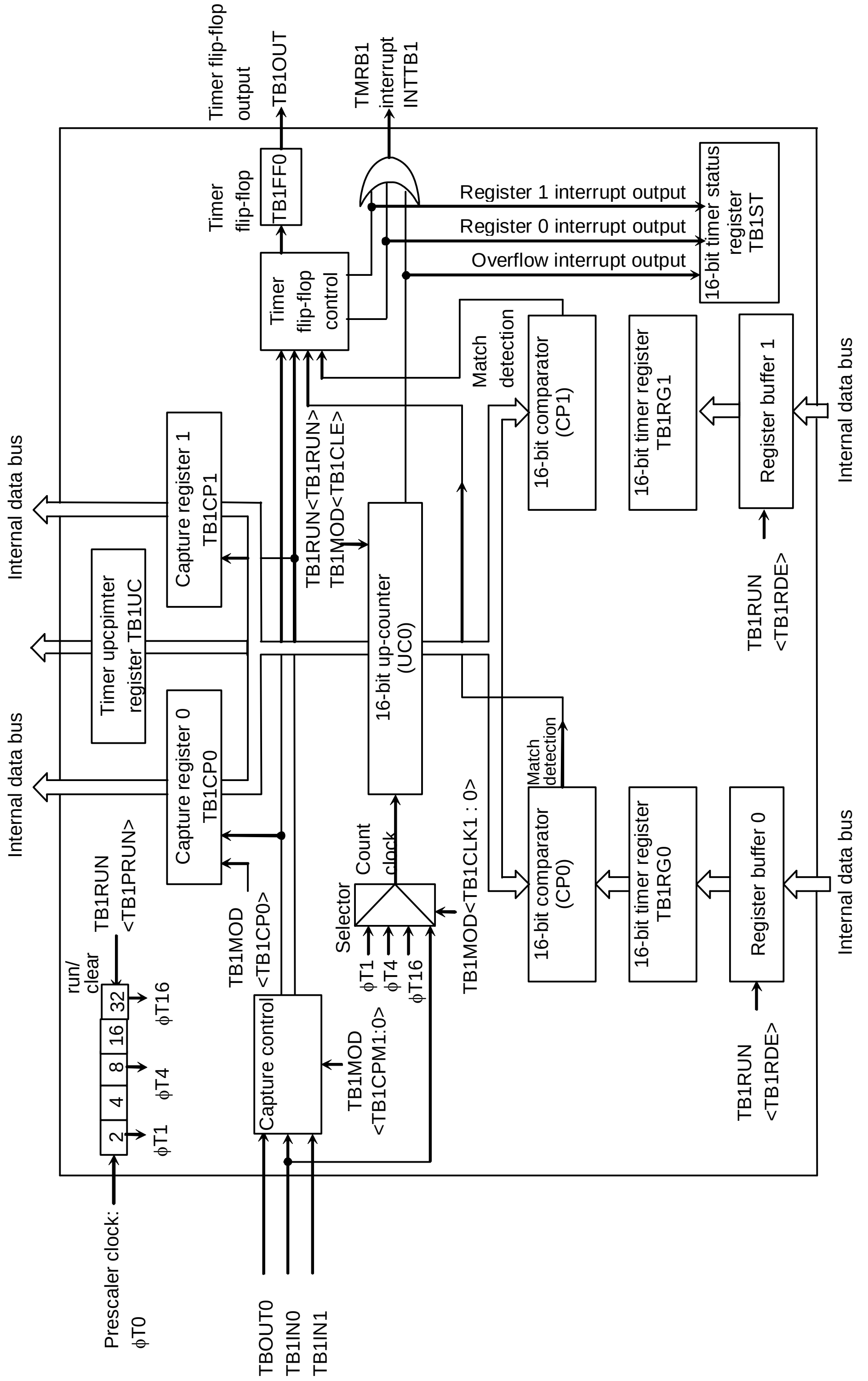
(2) The start trigger of the timer synchronous mode (with TBxRUN)

- TMRB0 => can start TMRB 0 through TMRB 3 synchronously
- TMRB4 => can start TMRB 4 through TMRB 7 synchronously
- TMRB8 => can start TMRB 8, 9, A and B synchronously
- TMRBC => can start TMRB C through TMRB F synchronously
- TMRB10 => can start TMRB 10 through TMRB 11 synchronously

Table 11.1 Differences in the Specifications of TMRB Modules

Specification / Channel	External pins		Internal signals	
	External clock/ capture trigger input pins	Timer flip-flop output pin	Timer for capture triggers	Timer for synchronous start
TMRB0	–	TB0OUT(shared with P54)	–	–
TMRB1	TB1IN0 (shared with P20) TB1IN1 (shared with P21)	TB1OUT (shared with P55)	TB0OUT	TMRB0
TMRB2	TB2IN0 (shared with P22) TB2IN1 (shared with P23)	TB2OUT (shared with P56)		
TMRB3	TB3IN0 (shared with P24) TB3IN1 (shared with P25)	TB3OUT (shared with P57)		
TMRB4	–	TB4OUT (shared with P63)		–
TMRB5	TB5IN0 (shared with P26) TB5IN1 (shared with P27)	TB5OUT (shared with P67)		TMRB4
TMRB6	TB6IN0 (shared with PA4) TB6IN1 (shared with PA5)	TB6OUT(shared with PB2)		
TMRB7	–	TB7OUT (shared with PB3)		
TMRB8	–	TB8OUT (shared with PB7)	–	–
TMRB9	TB9IN0 (shared with PH0) TB9IN1 (shared with PH1)	TB9OUT (shared with P93)	TB8OUT	TMRB8
TMRBA	TBAIN0 (shared with PH2) TBAIN1 (shared with PH3)	TBAOUT (shared with P97)		
TMRBB	TBBIN0 (shared with PH4) TBBIN1 (shared with PH5)	TBBOUT (shared with PD3)		
TMRBC	–	TBCOUT (shared with PD4)		–
TMRBD	TBDIN0 (shared with PH6) TBDIN1 (shared with PH7)	TBDOUT (shared with PD5)		TMRBC
TMRBE	–	TBEOUT (shared with P34)		
TMRBF	–	TBFOUT (shared with P47)		
TMRB10	–	TBFOUT (shared with PI5)	–	–
TMRB11	TB11IN0 (shared with PJ0) TB11IN1 (shared with PJ1)	TBFOUT (shared with PI6)	TB10OUT	TMRB10

11.1 Block Diagram of Each Channel



(Note) Channels 0, 8 and 10 do not have input pin. TB0OUT is input to channels 1 and 4. TB8OUT is input to channels 9 through F. TB9OUT is input to channels 11.

Fig. 11.1 TMRB1 Block Diagram (Same for Other Channels)

11.2 Register Description

11.2.1 TMRB registers

Table 11.2 shows the register names and addresses of each channel.

Table 11.2 TMRB registers

Channel Specification		TMRB0	TMRB1	TMRB2	TMRB3
Register names (addresses)	Timer enable register	TB0EN (0xFF00_4500)	TB1EN (0xFF00_4540)	TB2EN (0xFF00_4580)	TB3EN (0xFF00_45C0)
	Timer RUN register	TB0RUN (0xFF00_4504)	TB1RUN (0xFF00_4544)	TB2RUN (0xFF00_4584)	TB3RUN (0xFF00_45C4)
	Timer control register	TB0CR (0xFF00_4508)	TB1CR (0xFF00_4548)	TB2CR (0xFF00_4588)	TB3CR (0xFF00_45C8)
	Timer mode register	TB0MOD (0xFF00_450C)	TB1MOD (0xFF00_454C)	TB2MOD (0xFF00_458C)	TB3MOD (0xFF00_45CC)
	Timer flip-flop control register	TB0FFCR (0xFF00_4510)	TB1FFCR (0xFF00_4550)	TB2FFCR (0xFF00_4590)	TB3FFCR (0xFF00_45D0)
	Timer status register	TB0ST (0xFF00_4514)	TB1ST (0xFF00_4554)	TB2ST (0xFF00_4594)	TB3ST (0xFF00_45D4)
	Interrupt mask register	TB0IM (0xFF00_4518)	TB1IM (0xFF00_4558)	TB2IM (0xFF00_4598)	TB3IM (0xFF00_45D8)
	Timer up counter register	TB0UC (0xFF00_451C)	TB1UC (0xFF00_455C)	TB2UC (0xFF00_459C)	TB3UC (0xFF00_45DC)
	Timer register	TB0RG0 (0xFF00_4520) TB0RG1 (0xFF00_4524)	TB1RG0 (0xFF00_4560) TB1RG1 (0xFF00_4564)	TB2RG0 (0xFF00_45A0) TB2RG1 (0xFF00_45A4)	TB3RG0 (0xFF00_45E0) TB3RG1 (0xFF00_45E4)
	Capture register	TB0CP0 (0xFF00_4528) TB0CP1 (0xFF00_452C)	TB1CP0 (0xFF00_4568) TB1CP1 (0xFF00_456C)	TB2CP0 (0xFF00_45A8) TB2CP1 (0xFF00_45AC)	TB3CP0 (0xFF00_45E8) TB3CP1 (0xFF00_45EC)

Channel Specification		TMRB4	TMRB5	TMRB6	TMRB7
Register names (addresses)	Timer enable register	TB4EN (0xFF00_4600)	TB5EN (0xFF00_4640)	TB6EN (0xFF00_4680)	TB7EN (0xFF00_46C0)
	Timer RUN register	TB4RUN (0xFF00_4604)	TB5RUN (0xFF00_4644)	TB6RUN (0xFF00_4684)	TB7RUN (0xFF00_46C4)
	Timer control register	TB4CR (0xFF00_4608)	TB5CR (0xFF00_4648)	TB6CR (0xFF00_4688)	TB7CR (0xFF00_46C8)
	Timer mode register	TB4MOD (0xFF00_460C)	TB5MOD (0xFF00_464C)	TB6MOD (0xFF00_468C)	TB7MOD (0xFF00_46CC)
	Timer flip-flop control register	TB4FFCR (0xFF00_4610)	TB5FFCR (0xFF00_4650)	TB6FFCR (0xFF00_4690)	TB7FFCR (0xFF00_46D0)
	Timer status register	TB4ST (0xFF00_4614)	TB5ST (0xFF00_4654)	TB6ST (0xFF00_4694)	TB7ST (0xFF00_46D4)
	Interrupt mask register	TB4IM (0xFF00_4618)	TB5IM (0xFF00_4658)	TB6IM (0xFF00_4698)	TB7IM (0xFF00_46D8)
	Timer up counter register	TB4UC (0xFF00_461C)	TB5UC (0xFF00_465C)	TB6UC (0xFF00_469C)	TB7UC (0xFF00_46DC)
	Timer register	TB4RG0 (0xFF00_4620) TB4RG1 (0xFF00_4624)	TB5RG0 (0xFF00_4660) TB5RG1 (0xFF00_4664)	TB6RG0 (0xFF00_46A0) TB6RG1 (0xFF00_46A4)	TB7RG0 (0xFF00_46E0) TB7RG1 (0xFF00_46E4)
	Capture register	TB4CP0 (0xFF00_4628) TB4CP1 (0xFF00_462C)	TB5CP0 (0xFF00_4668) TB5CP1 (0xFF00_466C)	TB6CP0 (0xFF00_46A8) TB6CP1 (0xFF00_46AC)	TB7CP0 (0xFF00_46E8) TB7CP1 (0xFF00_46EC)

Channel Specification		TMRB8	TMRB9	TMRBA	TMRBB
Register names (addresses)	Timer enable register	TB8EN (0xFF00_4700)	TB9EN (0xFF00_4740)	TBAEN (0xFF00_4780)	TBBEN (0xFF00_47C0)
	Timer RUN register	TB8RUN (0xFF00_4704)	TB9RUN (0xFF00_4744)	TBARUN (0xFF00_4784)	TBBRUN (0xFF00_47C4)
	Timer control register	TB8CR (0xFF00_4708)	TB9CR (0xFF00_4748)	TBACR (0xFF00_4788)	TBBCR (0xFF00_47C8)
	Timer mode register	TB8MOD (0xFF00_470C)	TB9MOD (0xFF00_474C)	TBAMOD (0xFF00_478C)	TBBMOD (0xFF00_47CC)
	Timer flip-flop control register	TB8FFCR (0xFF00_4710)	TB9FFCR (0xFF00_4750)	TBAFFCR (0xFF00_4790)	TBBFFCR (0xFF00_47D0)
	Timer status register	TB8ST (0xFF00_4714)	TB9ST (0xFF00_4754)	TBAST (0xFF00_4794)	TBBST (0xFF00_47D4)
	Interrupt mask register	TB8IM (0xFF00_4718)	TB9IM (0xFF00_4758)	TBAIM (0xFF00_4798)	TBBIM (0xFF00_47D8)
	Timer up counter register	TB8UC (0xFF00_471C)	TB9UC (0xFF00_475C)	TBAUC (0xFF00_479C)	TBBUC (0xFF00_47DC)
	Timer register	TB8RG0 (0xFF00_4720) TB8RG1 (0xFF00_4724)	TB9RG0 (0xFF00_4760) TB9RG1 (0xFF00_4764)	TBARG0 (0xFF00_47A0) TBARG1 (0xFF00_47A4)	TBBRG0 (0xFF00_47E0) TBBRG1 (0xFF00_47E4)
	Capture register	TB8CP0 (0xFF00_4728) TB8CP1 (0xFF00_472C)	TB9CP0 (0xFF00_4768) TB9CP1 (0xFF00_476C)	TBACP0 (0xFF00_47A8) TBACP1 (0xFF00_47AC)	TBBCP0 (0xFF00_47E8) TBBCP1 (0xFF00_47EC)

Channel Specification		TMRBC	TMRBD	TMRBE	TMRBF
Register names (addresses)	Timer enable register	TBCEN (0xFF00_4800)	TBDEN (0xFF00_4840)	TBAEN (0xFF00_4880)	TBFEN (0xFF00_48C0)
	Timer RUN register	TBCRUN (0xFF00_4804)	TBDRUN (0xFF00_4844)	TBERUN (0xFF00_4884)	TBFRUN (0xFF00_48C4)
	Timer control register	TBCCR (0xFF00_4808)	TBDCR (0xFF00_4848)	TBECR (0xFF00_4888)	TBFCR (0xFF00_48C8)
	Timer mode register	TBCMOD (0xFF00_480C)	TBDMOD (0xFF00_484C)	TBEMOD (0xFF00_488C)	TBFMOD (0xFF00_48CC)
	Timer flip-flop control register	TBCFFCR (0xFF00_4810)	TBDFFCR (0xFF00_4850)	TBEFFCR (0xFF00_4890)	TBFFFCR (0xFF00_48D0)
	Timer status register	TBCST (0xFF00_4814)	TBDST (0xFF00_4854)	TBEST (0xFF00_4894)	TBFST (0xFF00_48D4)
	Interrupt mask register	TBCIM (0xFF00_4818)	TBDIM (0xFF00_4858)	TBEIM (0xFF00_4898)	TBFIM (0xFF00_48D8)
	Timer up counter register	TBCUC (0xFF00_481C)	TBDUC (0xFF00_485C)	TBEUC (0xFF00_489C)	TBFUC (0xFF00_48DC)
	Timer register	TBCRG0 (0xFF00_4820) TBCRG1 (0xFF00_4824)	TBDRG0 (0xFF00_4860) TBDRG1 (0xFF00_4864)	TBERG0 (0xFF00_48A0) TBERG1 (0xFF00_48A4)	TBFRG0 (0xFF00_48E0) TBFBRG1 (0xFF00_48E4)
	Capture register	TBCCP0 (0xFF00_4828) TBCCP1 (0xFF00_482C)	TBDCP0 (0xFF00_4868) TBDCP1 (0xFF00_486C)	TBECP0 (0xFF00_48A8) TBECP1 (0xFF00_48AC)	TBFCP0 (0xFF00_48E8) TBFCP1 (0xFF00_48EC)

Specification \ Channel		TMRB10	TMRB11
Register names (addresses)	Timer enable register	TB10EN (0xFF00_4900)	TB11EN (0xFF00_4940)
	Timer RUN register	TB10RUN (0xFF00_4904)	TB11RUN (0xFF00_4944)
	Timer control register	TB10CR (0xFF00_4908)	TB11CR (0xFF00_4948)
	Timer mode register	TB10MOD (0xFF00_490C)	TB11MOD (0xFF00_494C)
	Timer flip-flop control register	TB10FFCR (0xFF00_4910)	TB11FFCR (0xFF00_4950)
	Timer status register	TB10ST (0xFF00_4914)	TB11ST (0xFF00_4954)
	Interrupt mask register	TB10IM (0xFF00_4918)	TB11IM (0xFF00_4958)
	Timer up counter register	TB10UC (0xFF00_491C)	TB11UC (0xFF00_495C)
	Timer register	TB10RG0 (0xFF00_4920) TB10RG1 (0xFF00_4924)	TB11RG0 (0xFF00_4960) TB11RG1 (0xFF00_4964)
	Capture register	TB10CP0 (0xFF00_4928) TB10CP1 (0xFF00_492C)	TB11CP0 (0xFF00_4968) TB11CP1 (0xFF00_496C)

11.3 Description of Operations for Each Circuit

11.3.1 Prescaler

There is a 4-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<PRCK2:0> 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 11.3shows prescaler output clock resolutions.

Table 11.3 Prescaler Output Clock Resolutions (fsys = 80MHz)

Release peripheral clock <FPSEL>	**Clock gear value** <GEAR2:0> (fsys)	**Select prescaler clock** <PRCK2:0>	**Prescaler output clock resolutions**		
			φT1	**φT4**	**φT16**
0 (fgear)	000 (fc)	000(fperiph/2)	fc/2^{2}(0.05μs)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)
		001(fperiph/4)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
	100(fc/2)	000(fperiph/2)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)
		001(fperiph/4)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		010(fperiph/8)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		011(fperiph/16)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		100(fperiph/32)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
	101(fc/4)	000(fperiph/2)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		001(fperiph/4)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		010(fperiph/8)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
	110(fc/8)	000(fperiph/2)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		001(fperiph/4)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		010(fperiph/8)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		011(fperiph/16)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
		100(fperiph/32)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)
	111(fc/16)	000(fperiph/2)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		010(fperiph/8)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)	fc/2^{14}(204.8μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)

Release peripheral clock <FPSEL>	Clock gear value <GEAR2:0> (fsys)	Select prescaler clock <PRCK2:0>	Prescaler output clock resolutions		
			ϕT1	ϕT4	ϕT16
1 (fc)	000 (fc)	000(fperiph/2)	fc/2^{2}(0.05μs)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)
		001(fperiph/4)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
	100(fc/2)	000(fperiph/2)	—	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)
		001(fperiph/4)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
	101(fc/4)	000(fperiph/2)	—	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)
		001(fperiph/4)	—	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
	110(fc/8)	000(fperiph/2)	—	—	fc/2^{6}(0.8μs)
		001(fperiph/4)	—	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)
		010(fperiph/8)	—	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
	111(fc/16)	000(fperiph/2)	—	—	fc/2^{6}(0.8μs)
		001(fperiph/4)	—	—	fc/2^{7}(1.6μs)
		010(fperiph/8)	—	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		011(fperiph/16)	—	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)

(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.

11.3.2 Up-counter (UC0) and Time Up-counter Registers (TB0UC)

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 - ϕT1, ϕT4 and ϕT16 - 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 TB0RG1 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 TB0UC registers.

If UC0 overflow occurs, the INTTB0 overflow interrupt is generated.

11.3.3 Timer Registers (TB0RG0, TB0RG1)

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. TB0RG0/1 uses TB0CR<TB0WBF> to control the enabling/disabling of double buffering so that if < TB0WBF> = "0," double buffering is disabled and if < TB0WBF> = "1," it is enabled. If double buffering is enabled, data is transferred from register buffer 0 to the TB0RG0/1 timer register when there is a match between UC0 and TB0RG0/1.

A reset initializes TB0CR <TB0WBF> to "0" and sets double buffering to "disable." To use double buffering, write data to the timer register, set <TB0WBF> to "1" and then write the following data to the register buffers.

TB0RG0/1 and the register buffers are assigned to the same address: 0xFF00_4520/0xFF00_4524. If <TB0WBF> = "0," the same value is written to TB0RG0/1 and each register buffer; if <TB0WBF> = "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."

11.3.4 Capture Registers (TB0CP0, TB0CP1)

These are 16-bit registers for latching values from the UC0 up-counter.

11.3.5 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").

11.3.6 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.

11.3.7 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 P54). To enable timer output, the port 5 related registers P5CR and P5FC must be programmed beforehand.

11.4 Registers

TMRBn Enable Register（n=0~11）

TBnEN
(0xFF00_4xx0)

	7	6	5	4	3	2	1	0
bit Symbol	TBnEN							
Read/Write	R/W	R						
After reset	0	0						
Function	TMRBn operation 0: Disable 1: Enable	This can be read as “0”.						
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	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 consumption. (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 RUN Register（n=0~11）

TBnRUN
(0xFF00_4xx4)

	7	6	5	4	3	2	1	0
bit Symbol						TBnPRUN		TBnRUN
Read/Write	R					R/W	R	R/W
After reset	0					0	0	0
Function	This can be read as “0”.					Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as "0."		
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

＜TBnRUN＞ : Controls the TMRB0 count operation.

＜TBnPRUN＞: Controls the TMRB0 prescaler operation.

TMRBn Control Register (n=0~11)

TBnCR
(0xFF00_4xx8)

	7	6	5	4	3	2	1	0
bit Symbol	TBnWBF		TBnSYNC		I2TBn			
Read/Write	R/W	R/W	R/W	R	R/W	R		
After reset	0	0	0	0	0	0		
Function	Double Buffering 0:Disable 1:Enable	Write “0”.	Synchronization mode switch-over 0: Individual operation 1: Synchronous operation	This can be read as “0”.	IDLE 0:Stop 1:Operate	This can be read as “0”.		
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

＜I2TBm＞:Controls the operation in the IDLE mode.

＜TBnSYNC＞:Controls operation mode of timers.

“0”: timers operate individually.

“1”: timers operate synchronously.

＜TBmWBF＞:Controls enabling/disabling of double buffering.

TMRBn Mode Register (n=0~11)

TBnMOD
(0xFF00_4xx4)

	7	6	5	4	3	2	1	0
bit Symbol			TBnCP0	TBnCPM1	TBnCPM0	TBnCLE	TBnCLK1	TBnCLK0
Read/Write	R	R/W	W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	This can be read as “0”.	Write “0”.	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: TBnIN0 pin input 01: φT1 10: φT4 11: φT16	
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

＜TBnCLK1:0＞ : Clears and controls the TMRBn up-counter.

＜TBnCLE＞:Disables clearing of the 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 values into capture register 0 (TBnCP0) upon the rising of 16-bit timer match output (TB3OUT) and into capture register 1 (TBnCP1) upon the falling of TBxOUT

(TMRB1 through TMRB7 : TB0OUT, TMRB9 through F:TB8OUT and TMRB11 : TB10OUT).

＜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=0~11)

TBnFFCR
(0xFF00_4xx8)

	7	6	5	4	3	2	1	0
bit Symbol			TBnC1T1	TBnC0T1	TBnE1T1	TBnE0T1	TBnFF0C1	TBnFF0C0
Read/Write	R		R/W				R/W	
After reset	1	1	0	0	0	0	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 read 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		
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

<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

<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 (n=0~11)

TBnST
(0xFF00_4xxC)

	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
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

<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)
<INTTB0Fn>: 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.

TMRBn Interrupt Mask Register (n=0~11)

TBnIM
(0xFF00_4xx0)

	7	6	5	4	3	2	1	0
bit Symbol						TBIMOFn	TBIMn1	TBIMn0
Read/Write	R					R/W		
After reset	0					0	0	0
Function	This can be read as “0”.					1: Interrupt is masked	1: Interrupt is masked	1: Interrupt is masked
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

＜TBIMn0＞:Interrupt is masked if there is a match with timer register 0 (TBnRG0).
＜TBIMn1＞:Interrupt is masked if there is a match with timer register 1 (TBnRG1).
＜TBIMOFn＞:Interrupt is masked if an up-and-down counter overflow occurs.

TBnUC Time Up-counter Registers (n=0~11)

TBnUC
(0xFF00_4xx4)

	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit Symbol	UCn7	UCn6	UCn5	UCn4	UCn3	UCn2	UCn1	UCn0
Read/Write	R/W							
After reset	0							
Function	Data obtained by read capture: 7-0 bit data							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol	UCn15	UCn14	UCn13	UCn12	UCn11	UCn10	UCn9	UCn8
Read/Write	R/W							
After reset	0							
Function	Data obtained by read capture: 15-8 bit data							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

TBnRG0 Timer Register (n=0~11)

TBnRG0
(0xFF00_4xx8)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRG07	TBnRG06	TBnRG05	TBnRG04	TBnRG03	TBnRG02	TBnRG01	TBnRG00
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol	TBnRG015	TBnRG014	TBnRG013	TBnRG012	TBnRG011	TBnRG010	TBnRG09	TBnRG08
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

TBnRG1 Timer Register (n=0~11)

TBnRG1
(0xFF00_4xxC)

	7	6	5	4	3	2	1	0
bit Symbol	TBnRG17	TBnRG16	TBnRG15	TBnRG14	TBnRG13	TBnRG12	TBnRG11	TBnRG10
Read/Write	R/W							
After reset	0							
Function	Timer count value: 7-0 bit data							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol	TBnRG115	TBnRG114	TBnRG113	TBnRG112	TBnRG111	TBnRG110	TBnRG19	TBnRG18
Read/Write	R/W							
After reset	0							
Function	Timer count value: 15-8 bit data							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

TBnCP0 Capture Register (n=0~11)

TBnCP0
(0xFF00_4xx0)

	7	6	5	4	3	2	1	0
bit Symbol	TBnCP07	TBnCP06	TBnCP05	TBnCP04	TBnCP03	TBnCP02	TBnCP01	TBnCP00
Read/Write	R							
After reset	Undefined.							
Function	Timer capture value: 7-0 bit data							
	15	14	13	12	11	10	9	8
bit Symbol	TBnCP015	TBnCP014	TBnCP013	TBnCP012	TBnCP011	TBnCP010	TBnCP09	TBnCP08
Read/Write	R							
After reset	Undefined.							
Function	Timer capture value: 15-8 bit data							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

TBnCP1 Capture Register (n=0~11)

TBnCP1
(0xFF00_4xx4)

	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
bit Symbol	TBnCP17	TBnCP16	TBnCP15	TBnCP14	TBnCP13	TBnCP12	TBnCP11	TBnCP10
Read/Write	R							
After reset	Undefined.							
Function	Timer capture value: 7-0 bit data							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol	TBnCP115	TBnCP114	TBnCP113	TBnCP112	TBnCP111	TBnCP110	TBnCP19	TBnCP18
Read/Write	R							
After reset	Undefined.							
Function	Timer capture value: 15-8 bit data							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

11.5 Description of Operations for Each Mode

11.5.1 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	
TB0EN		1	X	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	X	X	X	X	X	0	X	0	Stops TMRB0.
IMC0D	←	0	1	1	0	0	1	0	0	Enables INTTB0, and sets it to level 4. (Setting of INTTB0 only is shown here. This is a 32-bit register and requires settings of other interrupts as well.)
TB0FFCR	←	1	1	0	0	0	0	1	1	Disables the trigger.
TB0MOD	←	0	0	1	0	0	1	*	*	Designates the prescaler input clock as the output clock,
TB0RG1	←	*	*	*	*	*	*	*	*	and specifies the time interval.
		*	*	*	*	*	*	*	*	(16 bits * This is a 32-bit register.)
TB0RUN	←	*	*	*	*	*	1	X	1	Starts TMRB0.

X; Don't care –; no change

11.5.2 16-bit Event Counter Mode

By using an input clock as an external clock (TB1IN0 pin input), it is possible to make it the event counter.

The up-counter counts up on the rising edge of TB1IN0 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	
TB1EN	←	1	X	X	X	X	X	X	X	Starts the TMRB0 module.
TB1RUN	←	X	X	X	X	X	0	X	0	Stops TMRB1.
P2CR	←	–	–	–	–	–	–	–	0	Sets P20 to the input mode.
P2FC3	←	–	–	–	–	–	–	–	1	
P2IE	←	–	–	–	–	–	–	–	1	
IMC0D	←	0	1	1	0	0	1	0	0	
		0	1	1	0	0	0	0	0	Enables INTTB0, and sets it to level 4.
		0	1	1	0	0	1	0	0	(Setting of INTTB0 only is shown here. This is a 32-bit register and requires settings of other interrupts as well.)
TB1FFCR	←	1	1	0	0	0	0	1	1	Disables the trigger.
TB1MOD	←	0	0	1	0	0	1	0	0	Designates the TB1IN0 pin input as the input clock.
TB1RUN	←	X	X	X	X	X	1	X	1	Starts TMRB1.
TB1MOD	←	X	X	0	0	0	1	0	0	Captures a value using software.
TB1RG1	←	*	*	*	*	*	*	*	*	Specifies the time interval.
		*	*	*	*	*	*	*	*	(16 bits * This is a 32-bit register.)

X; Don't care –; no change
To be used as the event counter, put the prescaler in a "RUN" state (TB1RUN<TB1PRUN> = "1").

11.5.3 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 TB1OUT pin by triggering the timer flip-flop (TB1FF) to reverse when the set value of the up-counter (UC1) matches the set values of the timer registers (TB1RG0 and TB1RG1). Note that the set values of TB1RG0 and TB1RG1 must satisfy the following requirement:

(Set value of TB1RG0) < (Set value of TB1RG1)

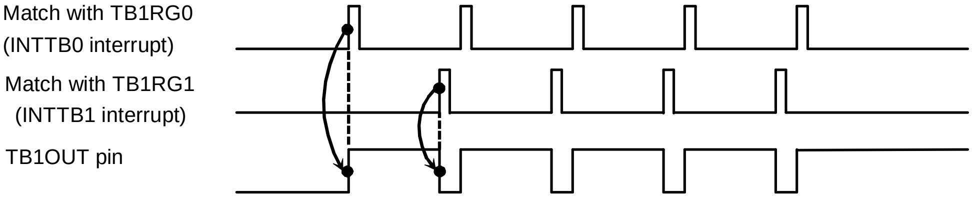


Fig. 11.2 Example of Output of Programmable Square Wave (PPG)

In this mode, by enabling the double buffering of TB1RG0, the value of register buffer 0 is shifted into TB1RG0 when the set value of the up-counter matches the set value of TB1RG1. This facilitates handling of small duties.

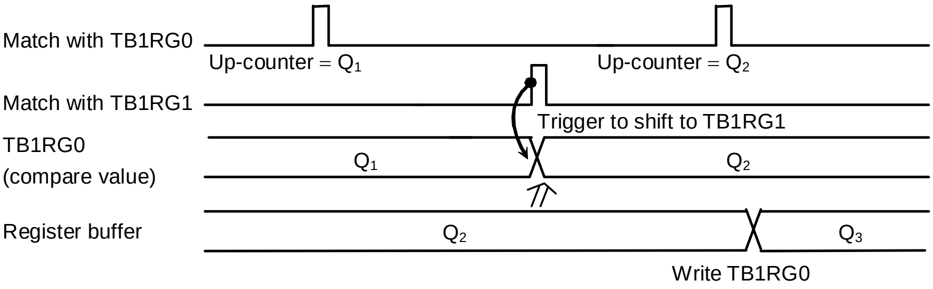


Fig. 11.3 Register Buffer Operation

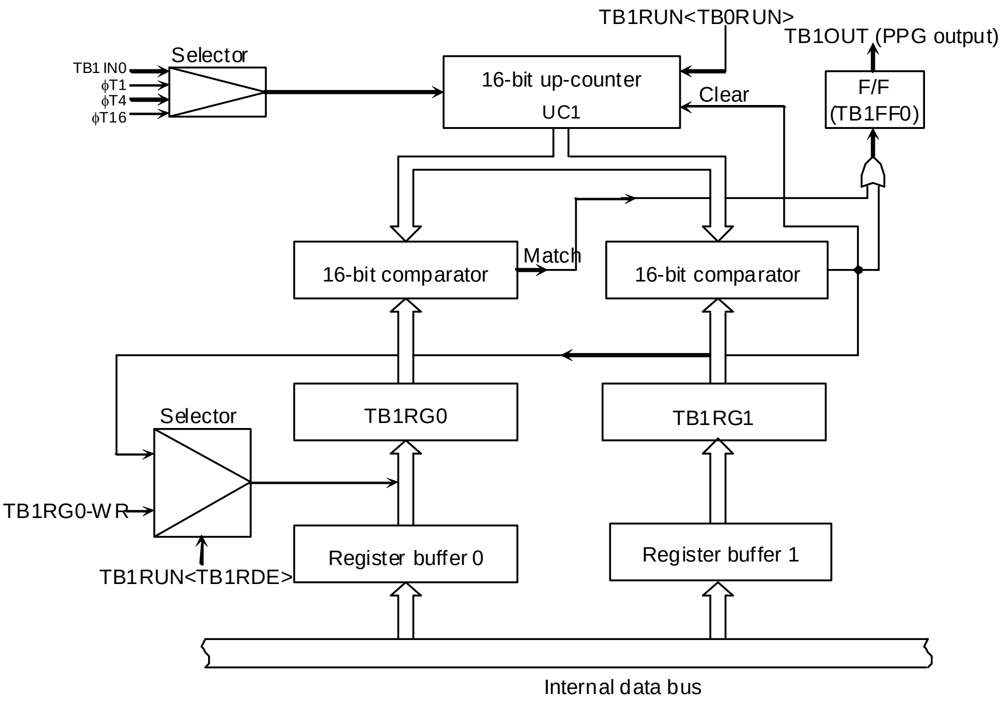


Fig. 11.4 Block Diagram of 16-bit PPG Mode

Each register in the 16-bit PPG output mode must be programmed as listed below.

		7	6	5	4	3	2	1	0	
TB1EN	←	1	X	X	X	X	X	X	X	Starts the TMRB1 module.
TB1RUN	←	X	X	X	X	X	0	X	0	Disables the TB1RG0 double buffering and stops TMRB1.
TB1RG0	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits *This is a 32-bit register)
	←	*	*	*	*	*	*	*	*	
TB1RG1	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits *This is a 32-bit register)
	←	*	*	*	*	*	*	*	*	
TB1CR	←	1	0	X	0	0	0	0	0	Enables the TB1RG0 double buffering. (Changes the duty/cycle when the INTTB1 interrupt is generated)
TB1FFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TB1FF0 to reverse when a match with TB1RG0 or TB1RG1 is detected, and sets the initial value of TB1FF0 to "0."
TB1MOD	←	0	0	1	0	0	1	*	*	Designates the prescaler input clock as the output clock, and disables the capture function.
					(** = 01, 10, 11)					
P5CR	←	–	–	–	1	–	–	–	–	Assigns P55 to TB1OUT.
	←	–	–	–	1	–	–	–	–	
P5FC2	←	–	–	–	0	–	–	–	–	
TB1RUN	←	1	0	0	0	–	1	X	1	Starts TMRB1.

X; Don't care –; no change

11.5.4 Timer Synchronization Mode

The timers can be started synchronously by using the timer synchronization mode.

The synchronization mode can be used for PPG output, for example, for application to driving a motor.

TBnCR<TBnSYC> is used to turn the synchronization mode on/off.

<TBnSYC> ="0": Operates the timers at the timing specified for each channel.

<TBnSYC> ="1": Enables the synchronous output.

There are five blocks, TMRB0 through TMRB3, TMRB4 through TMRB7, TMRB8 through TMRBB, TMRBC through TMRBF and TMRB10 through TMRB11.

If <TBnSYC> is set to "1," the timers will not start at the timing specified for each channel by setting TBmRUN<TBmPRUN,TBmRUN> to "1,1", but the timers in each block will start in synchronization with TMRB0, TMRB4, TMRB8, TMRBC or TMRB10.

(Note 1) For the channels to be output synchronously, set TBmRUN<TBmPRUN,TBmRUN> to "1,1" to enable simultaneous start before starting TMRB0, TMRB4, TMRB8, TMRBC or TMRB10.

(Note 2) Set TBnCR<TBnSYC> to "0" unless the synchronous output mode is selected. When the synchronous output mode is selected, other channels will not start until TMRB0, 4 , 8, C and 10 start.

Note) Master: TMRB0, TMRB4, TMRB8, TMRBC and TMRB10 write TBnSYC"0"

TBnCR
(0xFF00_4xx8)

	7	6	5	4	3	2	1	0
Bit symbol	TBnWBF		TBnSYNC		I2TBn			
Read/Write	R/W	R/W	R/W	R	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Double buffering 0: Disable 1: Enable	Write "0."	Synchronization mode 0: Individual operation	This can be read as "0."	IDLE 0:Stop 1:Operate	This can be read as "0."	This can be read as "0."	This can be read as "0."

Slave: Write TBnSYC "1"

TBnCR
(0xFFFF_4xx8)

	7	6	5	4	3	2	1	0
Bit symbol	TBnWEN		TBnSYNC		I2TBn			
Read/Write	R/W	R/W	R/W	R	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	Double buffering 0: Disable 1: Enable	Write "0."	Synchronization mode 0: Individual operation	This can be read as "0."	IDLE 0:Stop 1:Operate	This can be read as "0."	This can be read as "0."	This can be read as "0."

11.6 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 (UC6) 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 TB6IN0 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 (TB6CP0).

The INTC must be programmed so that an interrupt INT4 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TB6RG0) to the sum of the TB6CP0 value (c) and the delay time (d), (c + d), and set the timer registers (TB6RG1) to the sum of the TB6RG0 values and the pulse width (p) of one-shot pulse, (c + d + p).

In addition, the timer flip-flop control registers (TB6FFCR<TB6E1T1, TB6E0T1>) must be set to "11." This enables triggering the timer flip-flop (TB6FF0) to reverse when UC6 matches TB6RG0 and TB6RG1. This trigger is disabled by the INTTB6 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. 11.3.

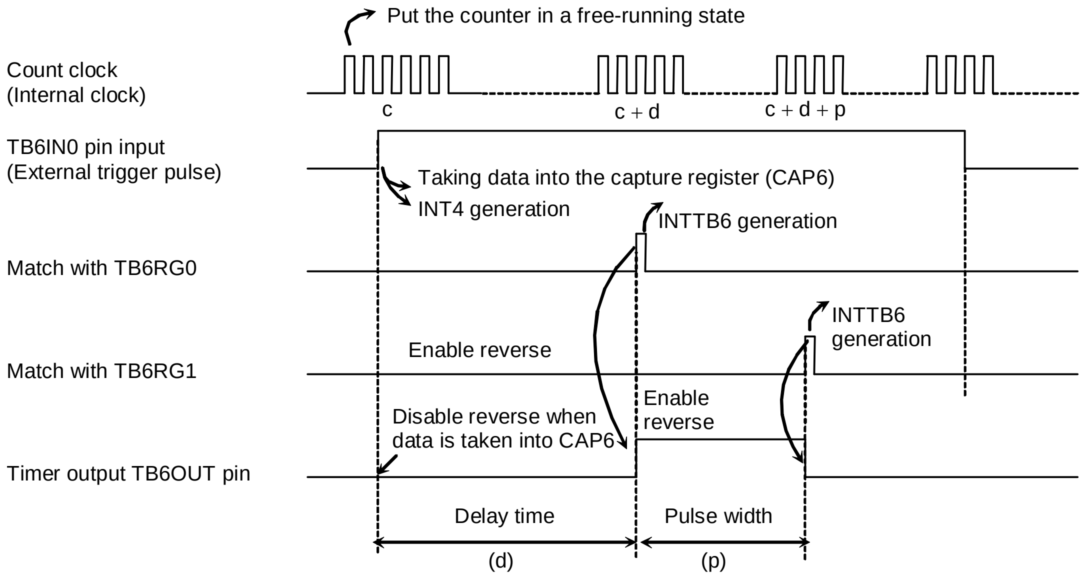


Fig. 11.5 One-shot Pulse Output (With Delay)

Programming example: Output a 2-ms one-shot pulse triggered by an external pulse from the TB6IN0 pin with a 3-ms delay

* Clock condition

System clock : High speed (fc)
High-speed clock gear : 1X (fc)
Prescaler clock : fperiph/8 (fperiph fsys)

Main programming

		7	6	5	4	3	2	1	0	
										Puts to a free-running state Sets TB6IN0.
PACR		X	X	X	0	X	X	X	X	
PAFC3		X	X	X	1	X	X	X	X	
PAPE		X	X	X	1	X	X	X	X	
PAIE		X	X	X	1	X	X	X	X	
TB6MOD	←	X	X	1	0	1	0	0	1	Uses ϕT1 for counting. Takes data into TB6CP0 at the rising of TB6IN0 input
TB6FFCR	←	X	X	0	0	0	0	1	0	Clears TB6FF0 to zero Disables TB6FF0 to reverse
PBCR	←	–	–	–	–	–	1	–	–	Assigns PB2 pin to TB6OUT
PBFC1	←	–	–	–	–	–	1	–	–	
IMC01	←	X	1	1	0	X	1	0	0	
IMC0F	←	X	1	1	0	X	0	0	0	Enables INT0 and disables INTTB6
TB6RUN	←	X	X	X	X	X	1	X	1	Starts TMRB6

INT4 programming

TB6RG0	←	TB0CP0 + 3 ms/ϕT1								
TB6RG1	←	TB0RG0 + 2 ms/ϕT1								
TB6FFCR	←	X	X	–	–	1	1	–	–	Enables TB2FF0 to reverse when there is a match with TB6RG0, 1
IMC0D	←	X	X	1	1	0	1	0	0	Enables INTTB6

INTTB6 programming

TB6FFCR	←	X	X	–	–	0	0	–	–	Disables TB6FF0 to reverse when there is a match with TB6RG0, 1
IMC0D	←	X	X	1	1	0	0	0	0	Disables INTTB6

X; Don't care —;no change

If a delay is not required, TB6FF0 is reversed when data is taken into TB6CP0, and TB6RG1 is set to the sum of the TB6CPO value (c) and the one-shot pulse width (p), (c + p), by generating the INT4 interrupt. TB6FF0 is enabled to reverse when UC6 matches with TB6RG1, and is disabled by generating the INTTB6 interrupt.

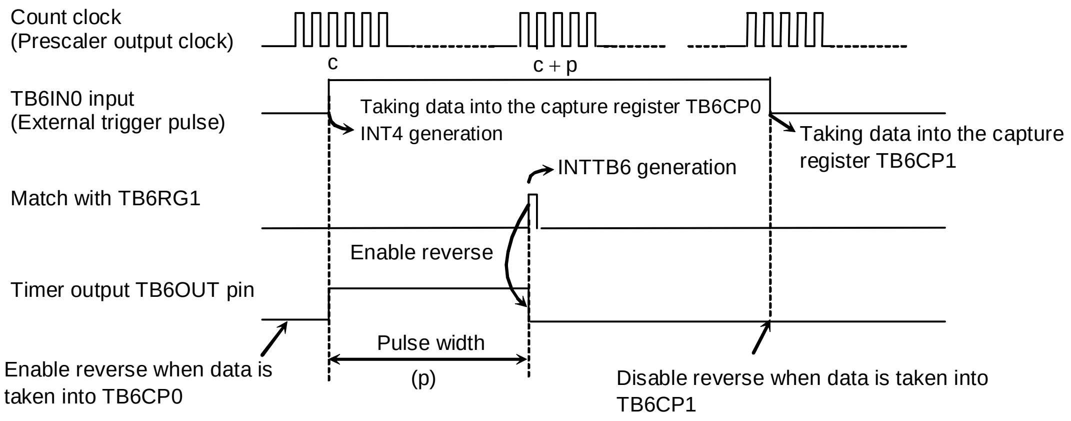

Count clock
(Prescaler output clock)
c
c + p
TB6IN0 input
(External trigger pulse)
Taking data into the capture register TB6CP0
INT4 generation
Taking data into the capture register TB6CP1
INTTB6 generation
Match with TB6RG1
Enable reverse
Timer output TB6OUT pin
Pulse width
(p)
Enable reverse when data is taken into TB6CP0
Disable reverse when data is taken into TB6CP1

Fig. 11.6 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 (TMRB0) is used in combination with the 16-bit event counter mode (TMRB0 reverses TB0FFCR to specify the measurement time).

The TB3IN0 pin input is selected as the TMRB3 count clock to perform the count operation using an external input clock. TB3MOD<TB3CPM1:0> is set to "11." This setting allows a count value of the 16-bit UC0 up-counter to be taken into the capture register (TB3CP0) upon the rising of a timer flip-flop (TB0FFCR) of the 16-bit timer (TMRB0), and an UC0 counter value to be taken into the capture register (TB3CP1) upon the falling of TB3FF of the 16-bit timer (TMRB0).

A frequency is then obtained from the difference between TB0CP0 and TB0CP1 based on the measurement, by generating the INTTB3 16-bit timer interrupt.

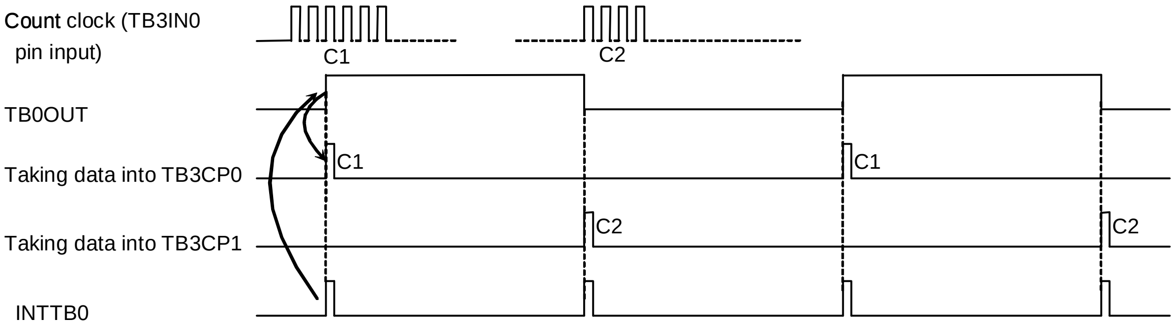


Fig. 11.7 Frequency Measurement

For example, if the set width of TB3FF level "1" of the 16-bit timer is 0.5 s and if the difference between TB0CP0 and TB0CP1 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 (UC6) 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 (TB6CP0, TB6CP1). The INTC must be programmed so that INT4 is generated at the falling edge of an external pulse input through the TB6IN0 pin.

The "H" level pulse width can be calculated by multiplying the difference between TB6CP0 and TB6CP1 by the clock cycle of an internal clock.

For example, if the difference between TB6CP0 and TB6CP1 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 UC6 maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.

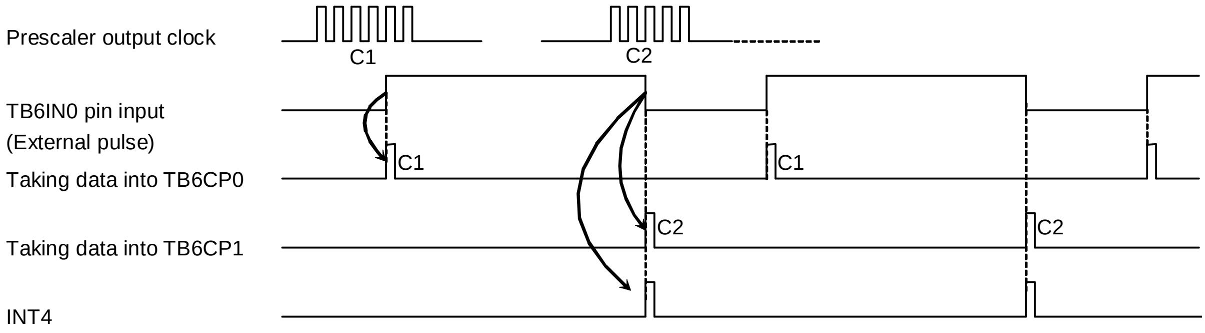


Fig. 11.8 Pulse Width Measurement

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 INT4 interrupt processing as shown in Fig. 11.78 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 (UC6) is made to count up by putting it in a free-running state using the prescaler output clock. The value of UC6 is taken into the capture register (TB6CP0) at the rising edge of the TB6IN0 pin input pulse. The INTC must be programmed to generate INT4 interrupt at this time.

The value of UC6 is taken into the capture register TB6CP1 at the rising edge of the TB6IN1 pin input pulse. The INTC must be programmed to generate INT5 interrupt at this time.

The time difference can be calculated by multiplying the difference between TB6CP1 and TB6CP0 by the clock cycle of an internal clock.

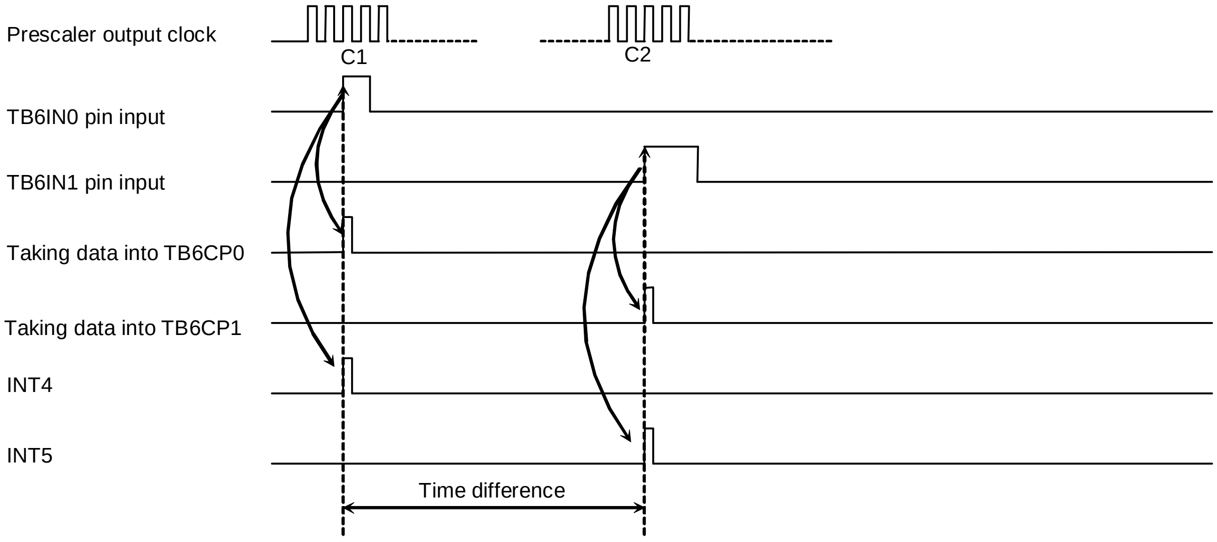


Fig. 11.9 Time Difference Measurement

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 eight channels (TCCMP0 through TCCMP7) each with a 32-bit compare register.

Fig. 12.1 shows the TMRC block diagram.

12.1 TMRC Block Diagram

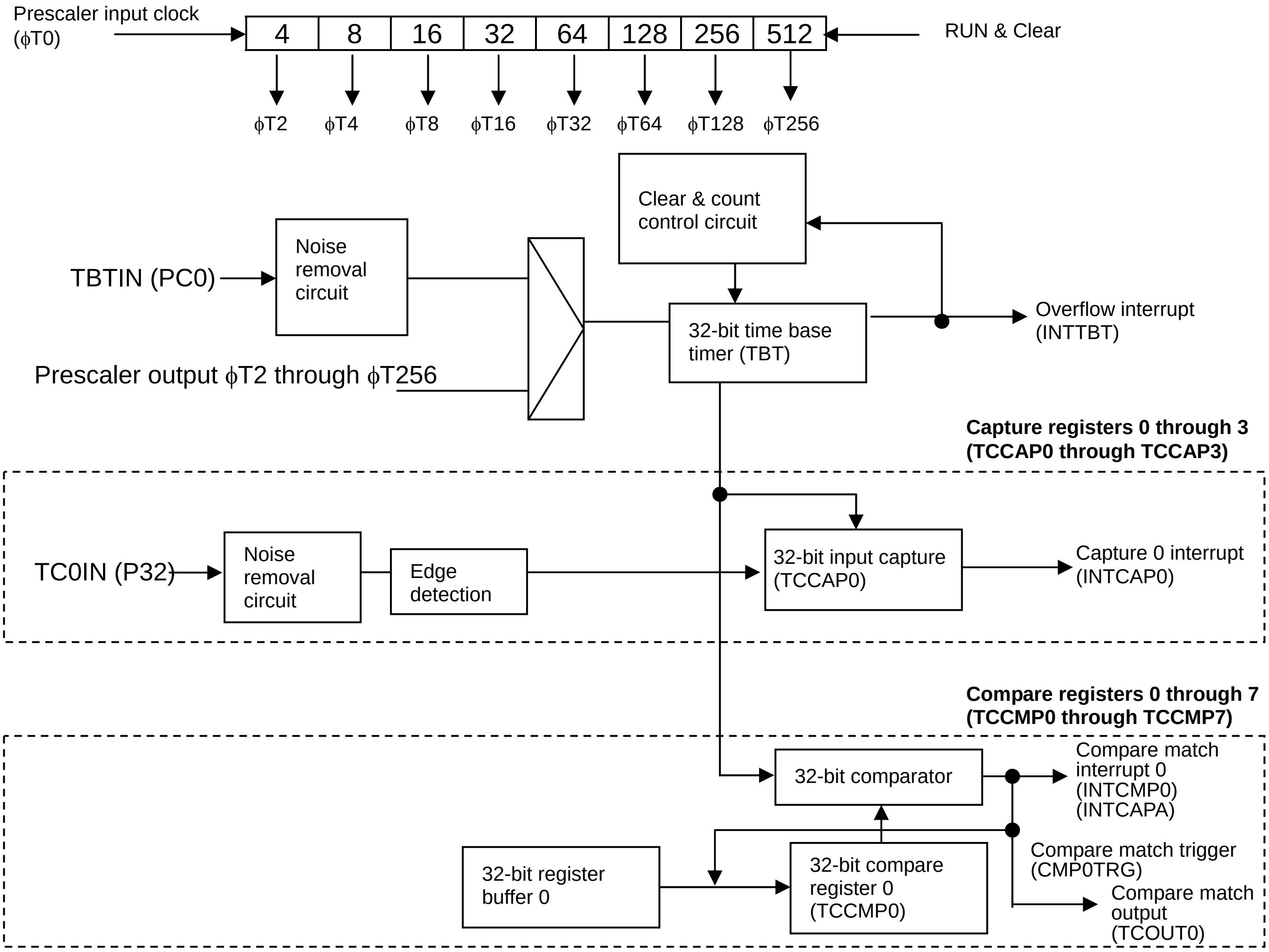

Prescaler input clock
(ϕT0)
4
8
16
32
64
128
256
512
RUN & Clear
ϕT2
ϕT4
ϕT8
ϕT16
ϕT32
ϕT64
ϕT128
ϕT256
Clear & count control circuit
TBTIN (PC0)
Noise removal circuit
32-bit time base timer (TBT)
Overflow interrupt (INTTBT)
Prescaler output ϕT2 through ϕT256
Capture registers 0 through 3 (TCCAP0 through TCCAP3)
TC0IN (P32)
Noise removal circuit
Edge detection
32-bit input capture (TCCAP0)
Capture 0 interrupt (INTCAP0)
Compare registers 0 through 7 (TCCMP0 through TCCMP7)
32-bit comparator
Compare match interrupt 0 (INTCMP0) (INTCAPA)
Compare match trigger (CMP0TRG)
32-bit register buffer 0
32-bit compare register 0 (TCCMP0)
Compare match output (TCOUT0)

Fig. 12.1 Timer C Block Diagram

12.2 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, fperiph/16 or fperiph/32 selected by SYSCR1<PRCK2:0> 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 @fc＝80.0MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK2:0>	Prescaler output clock resolution			
			ΦT2	ΦT4	ΦT8	ΦT16
0(fgear)	000(fc)	000(fperiph/2)	fc/2^{3}(0.10μs)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)
		001(fperiph/4)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		010(fperiph/8）	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		011(fperiph/16)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)
		100(fperiph/32)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)
	100(fc/2)	000(fperiph/2)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		001(fperiph/4)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		010(fperiph/8）	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.32μs)
		011(fperiph/16)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.6μs)
		100(fperiph/32)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
	101(fc/4)	000(fperiph/2)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		001(fperiph/4)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.32μs)
		010(fperiph/8）	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.6μs)
		011(fperiph/16)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
	110(fc/8)	000(fperiph/2)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.32μs)
		001(fperiph/4)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.6μs)
		010(fperiph/8）	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		011(fperiph/16)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		100(fperiph/32)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
	111(fc/16)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.6μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
1(fc)	000(fc)	000(fperiph/2)	fc/2^{3}(0.10μs)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)
		001(fperiph/4)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		010(fperiph/8）	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		011(fperiph/16)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)
		100(fperiph/32)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)
	100(fc/2)	000(fperiph/2)	fc/2^{3}(0.10μs)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)
		001(fperiph/4)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		010(fperiph/8）	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		011(fperiph/16)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)
		100(fperiph/32)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)
	101(fc/4)	000(fperiph/2)	—	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)
		001(fperiph/4)	fc/2^{4}(0.20μs)	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		010(fperiph/8）	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		011(fperiph/16)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)
		100(fperiph/32)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)
	110(fc/8)	000(fperiph/2)	—	—	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)
		001(fperiph/4)	—	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		010(fperiph/8）	fc/2^{5}(0.40μs)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		011(fperiph/16)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)
		100(fperiph/32)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)
	111(fc/16)	000(fperiph/2)	—	—	—	fc/2^{6}(0.80μs)
		001(fperiph/4)	—	—	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)
		010(fperiph/8）	—	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)
		011(fperiph/16)	fc/2^{6}(0.80μs)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)
		100(fperiph/32)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK2:0>	Prescaler output clock resolution			
			ΦT32	ΦT64	ΦT128	ΦT256
0(fgear)	000(fc)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.4μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
	100(fc/2)	000(fperiph/2)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		001(fperiph/4)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		010(fperiph/8）	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		011(fperiph/16)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
		100(fperiph/32)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)
	101(fc/4)	000(fperiph/2)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		001(fperiph/4)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		010(fperiph/8）	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
		011(fperiph/16)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)
		100(fperiph/32)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)	fc/2^{16}(819.2μs)
	110(fc/8)	000(fperiph/2)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		001(fperiph/4)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
		010(fperiph/8）	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)
		011(fperiph/16)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)	fc/2^{16}(819.2μs)
		100(fperiph/32)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)	fc/2^{16}(819.2μs)	fc/2^{17}(1638.4μs)
	111(fc/16)	000(fperiph/2)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
		001(fperiph/4)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)
		010(fperiph/8）	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)	fc/2^{16}(819.2μs)
		011(fperiph/16)	fc/2^{14}(204.8μs)	fc/2^{15}(409.6μs)	fc/2^{16}(819.2μs)	fc/2^{17}(1638.4μs)
		100(fperiph/32)	fc/2^{15}(409.6μs)	fc/2^{16}(819.2μs)	fc/2^{17}(1638.4μs)	fc/2^{18}(3276.8μs)
1(fc)	000(fc)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.4μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
	100(fc/2)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.4μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
	101(fc/4)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.4μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
	110(fc/8)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.4μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)
	111(fc/16)	000(fperiph/2)	fc/2^{7}(1.60μs)	fc/2^{8}(3.20μs)	fc/2^{9}(6.4μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{8}(3.20μs)	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)
		010(fperiph/8）	fc/2^{9}(6.40μs)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{10}(12.8μs)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{11}(25.6μs)	fc/2^{12}(51.2μs)	fc/2^{13}(102.4μs)	fc/2^{14}(204.8μs)

(Note 1) Do not change the clock gear while the timer is operating.

(Note 2) "-" 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.

This counter can perform a read capture operation.

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 shows capture inputs, outputs (capture factor outputs) produced by the edge detection circuit, and specific detection circuit settings.

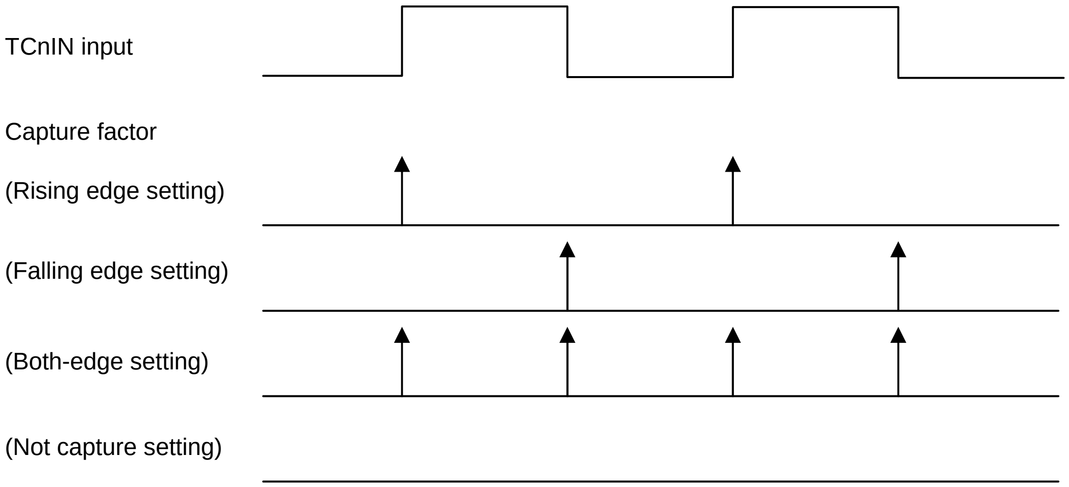


Fig. 12.2 Capture Inputs and Capture Factor Outputs (Outputs Produced by the Edge Detection Circuit)

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 then notified to the interrupt controller.

12.2.6 32-bit Compare Register

This is a 32-bit register for specifying a compare value. TMRC has eight built-in compare registers, TCCMP0 through TCCMP7. 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<CMPENn>.

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 Enable Register

		7	6	5	4	3	2	1	0
TCEN (0xFF00_4A00)	bit Symbol	TCEN	–						
	Read/Write	R/W	R						
	After reset	0	0						
	Function	TMRC operation 0: Disable 1: Enable	“0” is read.						
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	R							
	After reset	0							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0							
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0							

<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.

TMRC RUN Register

TBTRUN
(0xFF00_4A04)

	7	6	5	4	3	2	1	0
bit Symbol		I2TBT				TBTCAP	TBTPRUN	TBTRUN
Read/Write	R	R/W	R		R/W			
After reset	0	0	0		0	0	0	0
Function	"0" is read.	IDLE 0:Stop 1:Operation	"0" is read.		Ensure this is set to "0".	TBT counter software capture 0: D'ont Care 1: Software capture	TimerRun/Stop Control 0: Stop & clear 1: Count	

	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							

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

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

<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.

<I2TBT> : Controls the TMRC operation in IDLE mode.

Fig. 12.3 TMRC-related Registers

TMRC Control Register

		7	6	5	4	3	2	1	0
TBTCR (0xFF00_4A08)	bit Symbol	TBTNF				TBTCLK 3	TBTCLK 2	TBTCLK 1	TBTCLK0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	TBTIN input noise removal 0:2/fsys or more 1:6/fsys or more	Ensure to write “0”.			TBT source clock 0000: ϕT2 0001: ϕT4 0010: ϕT8 0011: ϕT16 0100: ϕT32 0101: ϕT64 0110: ϕT128 0111: ϕT256 1111: TBTIN pin input			
		15	14	13	12	11	10	9	8
	bit Symbol								
	Read/Write	0							
	After reset	R							
		23	22	21	20	19	18	17	16
	bit Symbol								
	Read/Write	R							
	After reset	0							
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0							

<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 (25ns@fperiph=fc=80MHz) 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 (75ns@fperiph=fc=80MHz) 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
(0xFF00_4A0C)

	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							
	15	14	13	12	11	10	9	8
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							
	23	22	21	20	19	18	17	16
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							
	31	30	29	28	27	26	25	24
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							

Fig. 12.4 TMRC-related Registers

TBTRead Capture Register（TBTRDCAP）

TBTRDCAP
(0xFF00_4A10)

	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							
	15	14	13	12	11	10	9	8
bit Symbol	RDCAP15	RDCAP14	RDCAP13	RDCAP12	RDCAP11	RDCAP10	RDCAP09	RDCAP08
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data							
	23	22	21	20	19	18	17	16
bit Symbol	RDCAP23	RDCAP22	RDCAP21	RDCAP20	RDCAP19	RDCAP18	RDCAP17	RDCAP16
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data							
	31	30	29	28	27	26	25	24
bit Symbol	RDCAP31	RDCAP30	RDCAP29	RDCAP28	RDCAP27	RDCAP26	RDCAP25	RDCAP24
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data							

Fig. 12.5 TMRC-related Registers

TMRC Capture Control Register

CAP0CR
(0xFF00_4Ax0)

	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
Function	TC0IN input noise removal 0:2/fsys or more 1:6/fsys or more	"0" is read.					Select effective edge of TC0IN input 00: Not captured 01: Rising edge 10: Falling edge 11: Both edges	
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol								
Read/Write	R							
After reset	0							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit Symbol								
Read/Write	R							
After reset	0							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
bit Symbol								
Read/Write	R							
After reset	0							

<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 (25ns@fperiph=fc=80MHz) 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 (75ns@fperiph=fc=80MHz) 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.

Fig. 12.6 TMRC-related Registers

TMRC Capture 0 Register (TCCAP0)

TCCAP0
(0xFF00_4Ax4)

	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							
	15	14	13	12	11	10	9	8
bit Symbol	CAP015	CAP014	CAP013	CAP012	CAP011	CAP010	CAP009	CAP008
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data							
	23	22	21	20	19	18	17	16
bit Symbol	CAP023	CAP022	CAP021	CAP020	CAP019	CAP018	CAP017	CAP016
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data							
	31	30	29	28	27	26	25	24
bit Symbol	CAP031	CAP030	CAP029	CAP028	CAP027	CAP026	CAP025	CAP024
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 0 data							

TMRC Compare Control Register (CMPCTL)

CMPCTL0
(0xFF00_4Ax0)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN0	TCFFC01	TCFFC00			CMPRDE0	CMPEN0
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	“0” is read.	TCFF0 reversal 0:Disable 1:Enable	TCFF0 control 00:Reversal 01:Set 10:Clear 11:D’ont care		“0” is read.		Double buffer 0 0:Disable 1:Enable	Compare 0 enable 0:Disable 1:Enable
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

<CMPENn>: Controls enabling/disabling of the compare match detection.

<CMPRDEn>: Controls enabling/disabling of double buffers of the compare register.

<TCFFCn1:0>: Controls F/F of the compare match output.

<TCFFENn>: Controls enabling/disabling of F/F reversal of the compare match output.

Fig. 12.7 TMRC-related Registers

TMRC Compare Register (TCCMP0)

TCCMP0
(0xFF00_4Ax4)

	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							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
bit Symbol	CMP015	CMP014	CMP013	CMP012	CMP011	CMP010	CMP009	CMP008
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
bit Symbol	CMP023	CMP022	CMP021	CMP020	CMP019	CMP018	CMP017	CMP016
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
bit Symbol	CMP031	CMP030	CMP029	CMP028	CMP027	CMP026	CMP025	CMP024
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data							

13 Two-phase Pulse Input Counter (PHCNT)

Each of the six channels (PHCNT0 through PHCNT5) has a two-phase input counter.

(The six channels operate in the same way. This section describes PHCNT0 only.)

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 PHC0IN0 and PHC0IN1 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. Interrupt is output in the ups and downs counter mode by the count operation.

There are two counting operation modes, which are switched by the register setting.

1) Normal operation mode (up/down at the fourth count)

2) Quadruple mode (up/down at each count)

13.1 Overview

1) PHCNT incorporates 16-bit up-and-down counter of which default value is 0x7fff.

2) PHCNT counts up or down according to the combination of asynchronous two-phase pulse inputs.

3) Two-phase pulse input pins incorporate a noise filter that can be enabled or disabled.

4) Counting mode is selectable from normal mode or quadruple mode.

5) PHCNT can control generation of two kinds of compare interrupts and an interrupt that occurs by each count.

6) The control register can control an interrupt output.

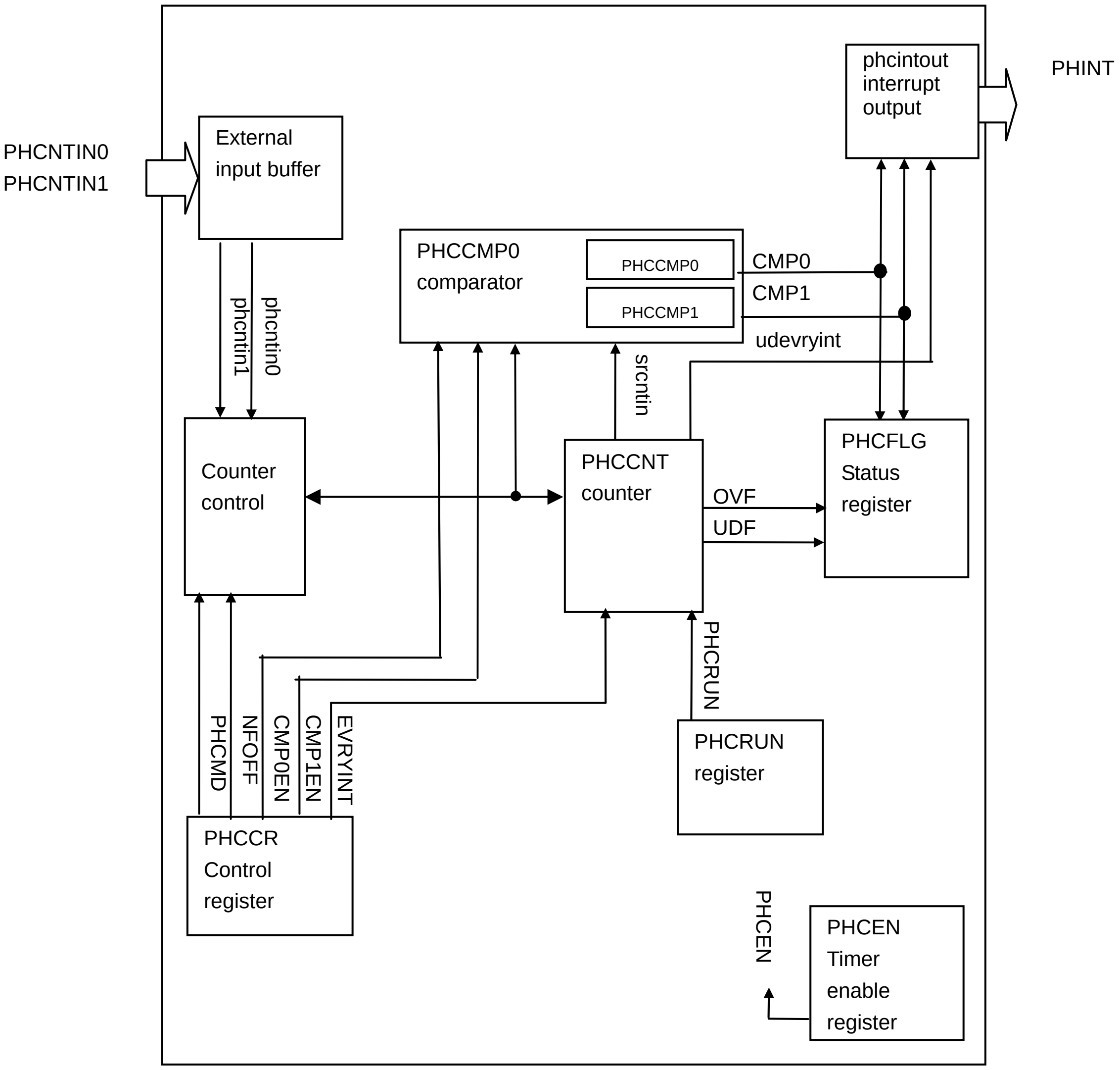
7) The status register can distinguish an overflow interrupt, an underflow interrupt and a compare interrupt.

Table 13.1 PHCNT Registers

Channel Specification		PHCNT0	PHCNT1	PHCNT2
External pin	Two-phase pulse input pin	PHC0IN0 (shared with PA0) PHC0IN1 (shared with PA1)	PHC1IN0 (shared with PA2) PHC1IN1 (shared with PA3)	PHC2IN0 (shared with PA6) PHC2IN1 (shared with PA7)
Register names (addresses)	Timer RUN register	PHC0RUN (0xFF00_1600)	PHC1RUN (0xFF00_1640)	PHC2RUN (0xFF00_1680)
	Timer control register	PHC0CR (0xFF00_1604)	PHC1CR (0xFF00_1644)	PHC2CR (0xFF00_1684)
	Timer mode register	PHC0EN (0xFF00_1608)	PHC1EN (0xFF00_1648)	PHC2EN (0xFF00_1688)
	Timer flip-flop control register	PHC0FLG (0xFF00_160C)	PHC1FLG (0xFF00_164C)	PHC2FLG (0xFF00_168C)
	Timer register	PHC0CMP0 (0xFF00_1610)	PHC1CMP0 (0xFF00_1650)	PHC2CMP0 (0xFF00_1690)
	Capture register	PHC0CMP1 (0xFF00_1614)	PHC1CMP1 (0xFF00_1654)	PHC2MP1 (0xFF00_1694)
	Count read register	PHC0CNT (0xFF00_1618)	PHC1CNT (0xFF00_1658)	PHC2CNT (0xFF00_1698)

Channel Specification		PHCNT3	PHCNT4	PHCNT5
External pin	External clock/ capture trigger input pins	PHC3IN0 (shared with PB0) PHC3IN1 (shared with PB1)	PHC4IN0 (shared with PI0) PHC4IN1 (shared with PI1)	PHC5IN0 (shared with PI1) PHC5IN1 (shared with PI2)
Register names (addresses)	Timer RUN register	PHC3RUN (0xFF00_16C0)	PHC4RUN (0xFF00_1700)	PHC5RUN (0xFF00_1740)
	Timer control register	PHC3CR (0xFF00_16C4)	PHC4CR (0xFF00_1704)	PHC5CR (0xFF00_1744)
	Timer mode register	PHC3EN (0xFF00_16C8)	PHC4EN (0xFF00_1708)	PHC5EN (0xFF00_1748)
	Timer flip-flop control register	PHC3FLG (0xFF00_16CC)	PHC4FLG (0xFF00_170C)	PHC5FLG (0xFF00_174C)
	Timer register	PHC3CMP0 (0xFF00_16D0)	PHC4CMP0(0xFF00_1710)	PHC5CMP0 (0xFF00_1750)
	Capture register	PHC3CMP1 (0xFF00_16D4)	PHC4CMP1(0xFF00_1714)	PHC5CMP1 (0xFF00_1754)
	Count read register	PHC3CNT (0xFF00_16D8)	PHC4CNT (0xFF00_1718)	PHC5CNT (0xFF00_1758)

13.2 Block Diagram (PHCNT0)



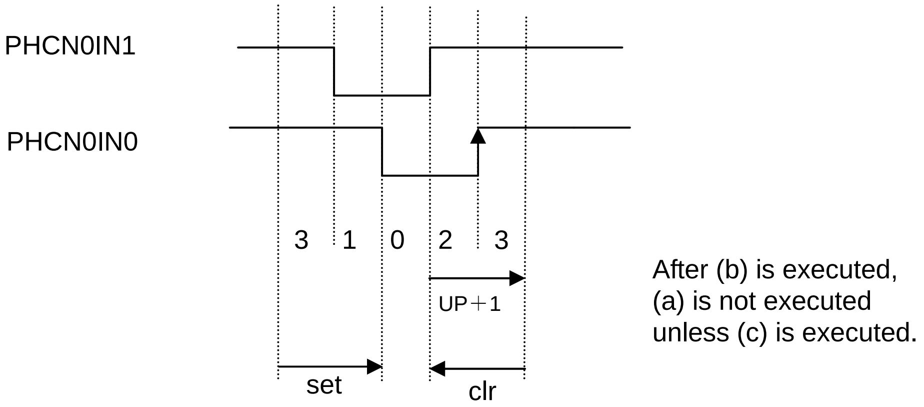
13.3 Operation Mode

Counting mode is selected from normal mode or quadruple mode according to PHCMD of the control register (PHCCR). The counter is incremented or decremented by one depending on the state transition of the asynchronous two-phase pulse that is input through PHCNTIN0 and PHCNTIN1. An interrupt can be generated by each count or when counter value matches with a value set in the compare register 0 or 1. The timing to generate an interrupt is selectable with the control register.

1) Normal mode

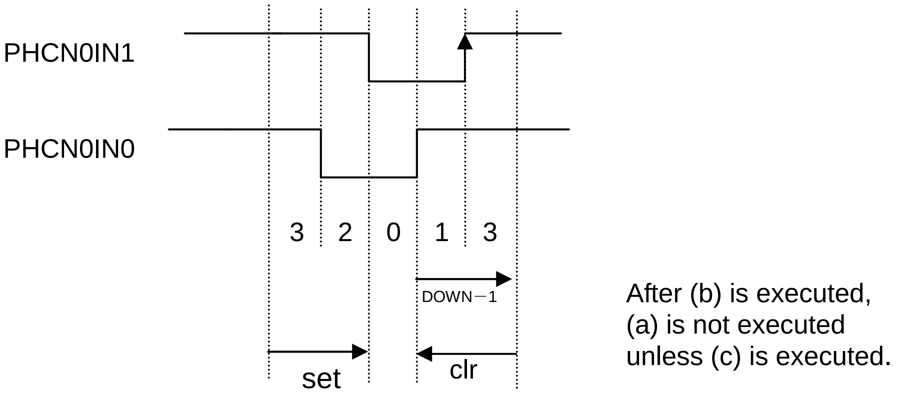
・**Count up**

(a) Count value is incremented by one when “2” is input at the previous clock and the current state is “3”.
(b) Count value is cleared when “3” is input at the previous clock and the current state is “2”.
(c) Count value is set when “3” is input at the previous clock and the current state is “1”.



・**Count down**

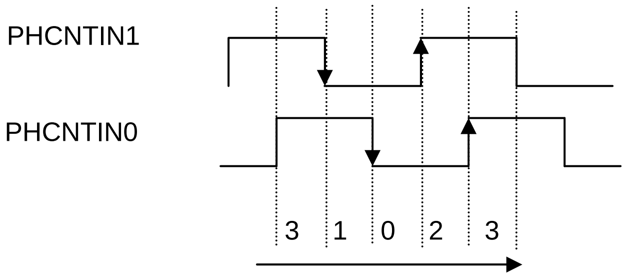
(a) Count value is decremented by one when “1” is input at the previous clock and the current state is “3”.
(b) Count value is cleared when “3” is input at the previous clock and the current state is “1”.
(c) Count value is set when “3” is input at the previous clock and the current state is “2”.



2) Quadruple mode

Count up

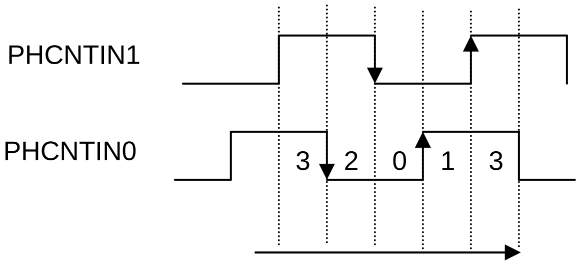
Count value is incremented by one when:
"3" is input at the previous clock and the current state is "1".
"1" is input at the previous clock and the current state is "0".
"0" is input at the previous clock and the current state is "2".
"2" is input at the previous clock and the current state is "3".



Count value incremented by one at each edge

Count down

Count value is decremented by one when:
"3" is input at the previous clock and the current state is "2".
"2" is input at the previous clock and the current state is "0".
"0" is input at the previous clock and the current state is "1".
"1" is input at the previous clock and the current state is "3".



Count value decremented by one at each edge

13.4 Registers

13.4.1 RUN register (PHCnRUN)

PHCnRUN
(0xFF00_1xx0)

	7	6	5	4	3	2	1	0
bit Symbol	–							PHCnRUN
Read/Write	R							R/W
After reset	0							0
Function	“0” can be read.							Timer control 0:Stop 1:Run
	15	14	13	12	11	10	9	8
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	“0” can be read.							

<PHCnRUN>: Controls PHCNTn count operation.

13.4.2 Control register (PHCnCR)

PHCnCR
(0xFF00_1xx4)

	7	6	5	4	3	2	1	0
bit Symbol	–			EVRYINTn	CMP1ENn	CMP0ENn	NFOFFn	PHCMDn
Read/Write	R			R/W	R/W	R/W	R/W	R/W
After reset	0			0	0	0	0	0
Function	“0” can be read.			Interrupt by each count 0:Disabled 1:Enabled	Compare interrupt 1 0:Disabled 1:Enabled	Compare interrupt 0 0:Disabled 1:Enabled	Noise filter 0: No use 1: Use	Mode switch-over 0: Normal 1: Quadruple
	15	14	13	12	11	10	9	8
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	“0” can be read.							

<PHCMD>: Controls mode switching.

0: normal mode…an interrupt is generated when count-up or count-down is selected.

1 :quadruple mode… an interrupt is generated by each count.

<NFOFF>: Controls noise cancellation.

<CMP0EN>: Generates an interrupt if counter value matches with a value set in the compare register 0.

<CMP1EN>: Generates an interrupt if counter value matches with a value set in the compare register 1.

<EVRYINT>: Controls interrupt generation.

Enables to prohibit generating an interrupt by each count when using a compare interrupt.

13.4.3 Timer Enable Register (PHCnEN)

PHCnEN
(0xFF00_1xx8)

	7	6	5	4	3	2	1	0
bit Symbol	-							PHCnEN
Read/Write	R							R/W
After reset	0							0
Function	"0" can be read.							Timer operation 0:Disabled 1:Enabled
	15	14	13	12	11	10	9	8
Read/Write	R							
After reset	0							
Function	"0" can be read.							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	"0" can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	"0" can be read.							

<PHCEN>: Enables or disables PHCEN operation. Disabling PHCEN operation stops providing register clock to other registers of PHCNT module, and it can reduce power consumption. (Neither reading nor writing is enabled with other registers). To use PHCNT, enable PHCNT by setting this bit to "1" before setting other PHCNT registers. When disabling PHCNT after temporary operation, setting in each register is kept.

13.4.4 Status register (PHCnFLG)

PHCnFLG
(0xFF00_1xxC)

	7	6	5	4	3	2	1	0
bit Symbol	–				UDFn	OVFn	CMPn1	CMPn0
Read/Write	R				R/W	R/W	R/W	R/W
After reset	0				0	0	0	0
Function	“0” can be read.				Underflow interrupt 0: Not generated 1: Generated	Overflow interrupt 0: Not generated 1: Generated	An interrupt generated if there is a match with compare register 1 0: Not generated 1: Generated	An interrupt generated if there is a match with compare register 0 0: Not generated 1: Generated
	15	14	13	12	11	10	9	8
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	“0” can be read.							

* These bits are not automatically cleared. Initialize them before use. Writing “1” to each bit clears its flag.

<CMP0>: Flag for an interrupt generated if there is a match with compare register 0 (PHCCMP0)
<CMP1>: Flag for an interrupt generated if there is a match with compare register 1 (PHCCMP1)
<OVF> : Flag for an overflow interrupt of an up-and-down counter.
<UDF> : Flag for an underflow interrupt of an up-and-down counter.

13.4.5 Compare register 0 (PHCnCMP0)

PHCnCMP0
(0xFF00_1xx0)

	7	6	5	4	3	2	1	0
bit Symbol	PHCCMP0							
Read/Write	R/W							
After reset	0x00							
Function	Set compare value.							
	15	14	13	12	11	10	9	8
bit Symbol	PHCCMP0							
Read/Write	R/W							
After reset	0x00							
Function	Set compare value.							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	“0” can be read.							

13.4.6 Compare register 1 (PHCnCMP1)

PHCnCMP1
(0xFF00_1xx4)

	7	6	5	4	3	2	1	0
bit Symbol	PHCCMP1							
Read/Write	R/W							
After reset	0x00							
Function	Set compare value.							
	15	14	13	12	11	10	9	8
bit Symbol	PHCCMP1							
Read/Write	R/W							
After reset	0x00							
Function	Set compare value.							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	“0” can be read.							

* By using both PHCnCMP0 and PHCnCMP1, up to two compare values can be set.

13.4.7 Counter Read Register (PHCnCNT)

PHCnCNT
(0xFF00_1xx8)

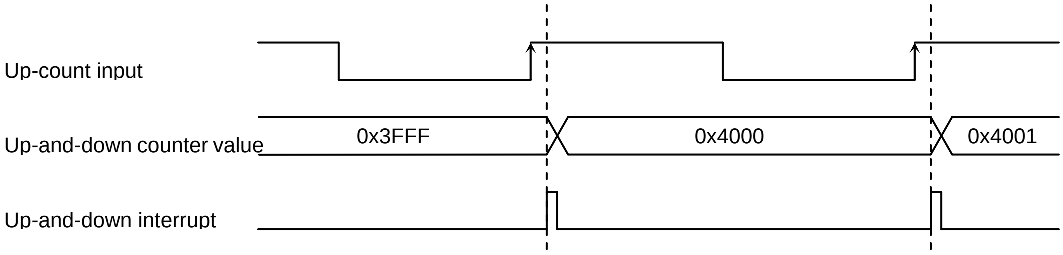
	7	6	5	4	3	2	1	0
bit Symbol	PHCCNT							
Read/Write	R							
After reset	0x00							
Function	Data read from counter							
	15	14	13	12	11	10	9	8
bit Symbol	PHCCNT							
Read/Write	R							
After reset	0x00							
Function	Data read from counter							
	23	22	21	20	19	18	17	16
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Read/Write	R							
After reset	0							
Function	“0” can be read.							

* Reading twice is recommended.

This register is initialized when the RUN register is cleared to "0".

13.5 Up-and-down counter

When starting the two-phase input count (PHCRUN<PHCRUN> = "1"), the up-counter is initialized to 0x7FFF and becomes ready for receiving counts. 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 PHCFLG 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 PHCNT0 interrupt handling.

13.6 Interrupt

- In the NORMAL or SLOW mode

The PHCNT0 interrupt is enabled using the interrupt controller (INTC). The PHCNT0 interrupt is generated by counting up or down. Reading the status register PHCFLG during interrupt handling allows simultaneous check for occurrences of an overflow and an underflow. If PHCnFLG<OVFn> is "1," it indicates that an overflow has occurred. If <UDFn> is "1," it indicates that an underflow has occurred. This register is cleared after “1” is written. The counter becomes 0x0000 when an overflow occurs, and it becomes 0xFFFF when an underflow occurs. After that, the counter continues the counting operation.

- In the SLEEP/ backup SLEEP mode

The two-phase input pulse input counter operates. The PHCNT0 interrupt is generated by the count-up or count-down input, and the system recovers from the SLEEP mode. Reading the status register PHCFLG during interrupt handling allows simultaneous check for occurrences of an overflow, an underflow and a compare interrupt. This register is cleared after “1” is written. The counter becomes 0x0000 when an overflow occurs, it and becomes 0xFFFF when an underflow occurs. After that, the counter continues the counting operation.

14. Serial Channel (SIO)

14.1 Features

This device has three serial I/O channels: SIO0 to SIO2.

14.1.1 Operation Modes

Four operation modes (mode 0 through mode 3) are provided to SIO. Table 14.1 shows data format of each mode.

Table 14.1 Data Format

Mode	Type	Data length	Transfer	Add parity	Stop bit length (transfer)
Mode 0	Synchronous mode (I/O interface mode)	8 bit	LSB first or MSB first	×	-
Mode 1	Asynchronous mode (UART mode)	7 bit	LSB first	○	1 or 2
Mode 2		8 bit		○	
Mode 3		9 bit		×	

Mode 0 is to send and receive I/O data and associated synchronization signals (SCLK) to extend I/O. SCLK can be used for input and output.
You can select either of LSB or MSB to send first. Neither adding parity nor stop bit is available.

Modes 1 through 3 execute asynchronous communication and are designed to send LSB first.

In the 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).

Stop bit length at transmission can be selected from 1 bit or 2 bits. At reception, stop bit is 1 bit.

Hereinafter synchronous mode is referred to as I/O interface mode, asynchronous mode is referred to as UART mode or 7 bit/ 8 bit/ 9 bit UART mode, which includes TX/RX data length.

14.1.2 Data Format

Table 14.1 shows data format of each mode.

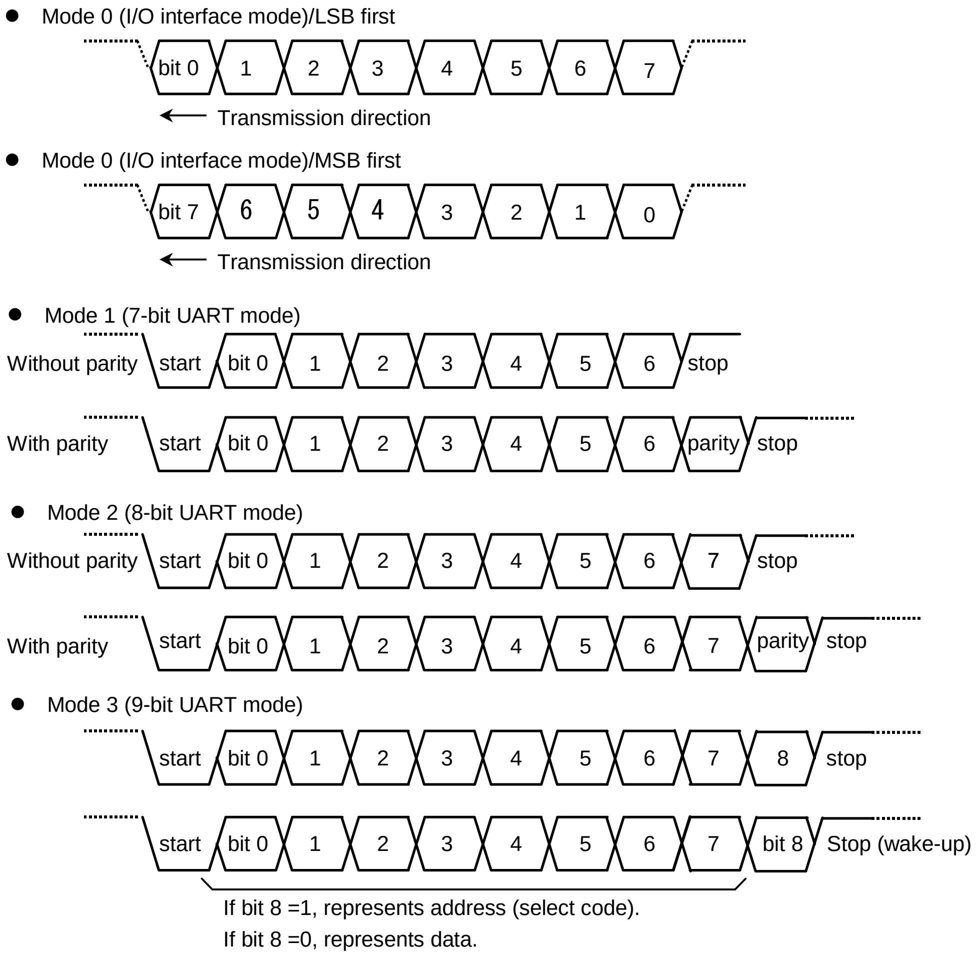


Fig. 14.1 Data Format

14.2 Block Diagram (Channel 0)

Fig. 14.2 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.

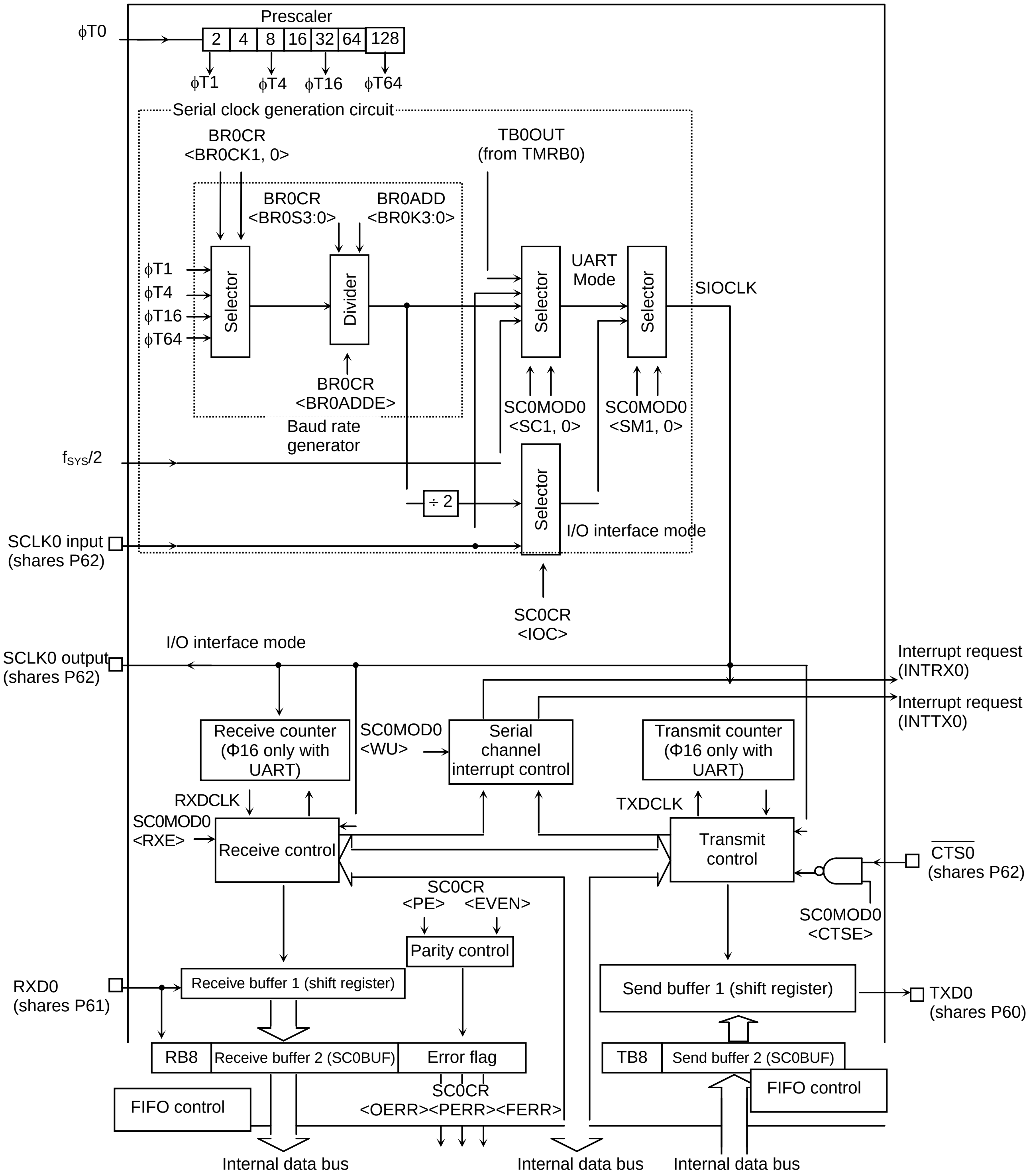


Fig. 14.2 SIO0 Block Diagram

14.2.1 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 625 kbps when ϕT1 is 20 MHz.

The fsys/2 frequency, which is independent of the baud rate generator, can be used as the serial clock. In this case, the highest baud rate will be 2.5 Mbps when fsys is 80 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 20 MHz. If double buffering is used, the divide ratio can be set to "1" and the resulting output baud rate will be 10 Mbps. (If double buffering is not used, the highest baud rate will be 5 Mbps applying the divide ratio of "2.")

- Example baud rate setting:

1) Division by an integer (divide by N):

Selecting fc = 39.321 MHz for fperiph, setting ϕT0 to fperiph/8, 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	:	1/2 (fc)
Prescaler clock	:	$f_{periph}/8$ ($f_{periph} = f_{sys}$)

$$\text{Baud rate} = \frac{fc/32}{4} /16$$

$= 39.321 \text{ (bps)} \times 10^6 / 32 / 4 / 16 \approx 19200 \text{ (bps)}$

(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 = 19.2 MHz for fperiph, setting ϕT0 to fperiph/8, using the baud rate generator input clock ϕT2, setting the divide ratio N (BR0CR<BR0S3:0>) = 7, setting K (BR0ADD<BR0K3:0>) = 3, and selecting BR0CR<BR0ADDE> = 1, the resulting baud rate is calculated as follows:

* Clocking conditions

System clock	:	High-speed (fgear)
High-speed clock gear	:	1/4 (fgear)
Prescaler clock	:	$f_{periph}/4$ ($f_{periph} = f_{sys}$)

$$\text{Baud rate} = \frac{fc/32}{7 + \frac{(16-3)}{16}} /16$$

$= 19.2 \times 10^6 / 32 / (7 + \frac{13}{16}) / 16 = 4800 \text{ (bps)}$

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 = 80 MHz, the highest baud rate will be 80 / 4 / 16 = 1.25 (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 = 80 MHz, the baud rate must be set to a rate lower than 80 / 12 = 6.67 (Mbps).

When double buffering is not used, it is necessary to satisfy the following relationship:

External clock input period > 16/fsys

Therefore, when fsys = 80 MHz, the baud rate must be set to a rate lower than 80 / 16 = 5 (Mbps).

Example baud rates for the UART mode are shown in Table 14-2 andTable 14- 3.

Table 14-2 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 f_{periph}/2.

Table 14-1 Selection of UART Baud Rate

(The TMRB2 timer output (internal TB2OUT) is used with the timer input clock set to ϕT1.)

Unit (kbps)

Fperiph/4 / TB0REG	19.6608 MHz	16 MHz
2H	153.6	125
3H	76.8	62.5
4H	51.2	41.67
5H	38.4	31.25

Baud rate calculation to use the TMRB2 timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by SYSCR0} < \text{PRCK2} : 0 >}{\text{TB2REG} \times \underline{2} \times \underline{2} \times 16}$$

(When input clock to the timer TMRB2 is ϕT1)

(Note 1) In the I/O interface mode, the TMRB0 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 f_{periph}/4.

14.2.2 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 TMRB2 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

14.2.3 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.

14.2.4 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.

14.2.5 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 <FDPX1:0> =01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SCNFCNF <CNFG> = 1 and <FDPX1:0> = 01/11), an interrupt will be generated according to the SC0RFC <RIL1:0> 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 <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."

14.2.6 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.

14.2.7 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:

SC0FCNF <4:0>=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.

SC0RFC<1:0>=00: Sets the interrupt to be generated at fill level 4.I

SC0RFC<7:6>=01: Clears receive FIFO and sets the condition of interrupt generation.

n 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).

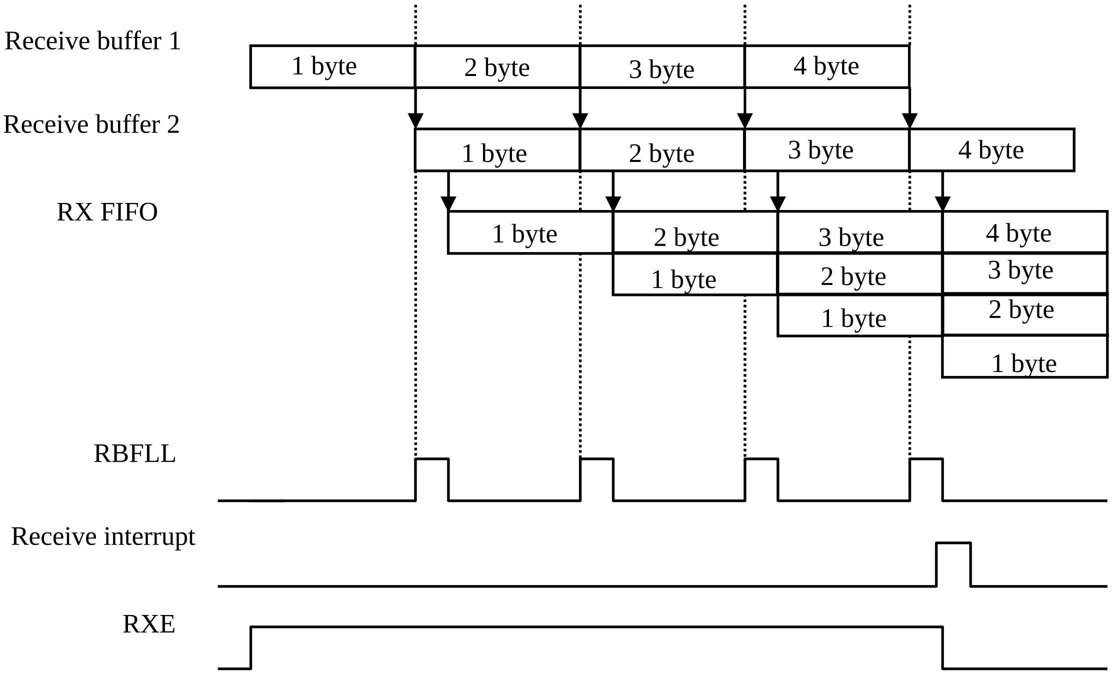


Fig. 14-3 Receive FIFO Operation

② I/O interface mode with SCLK input:

The following example describes the case a 4-byte data stream is received:

SC0FCNF <4:0> = 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.

SC0RFC <1:0> = 00: Sets the interrupt to be generated at fill level 4.

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

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.

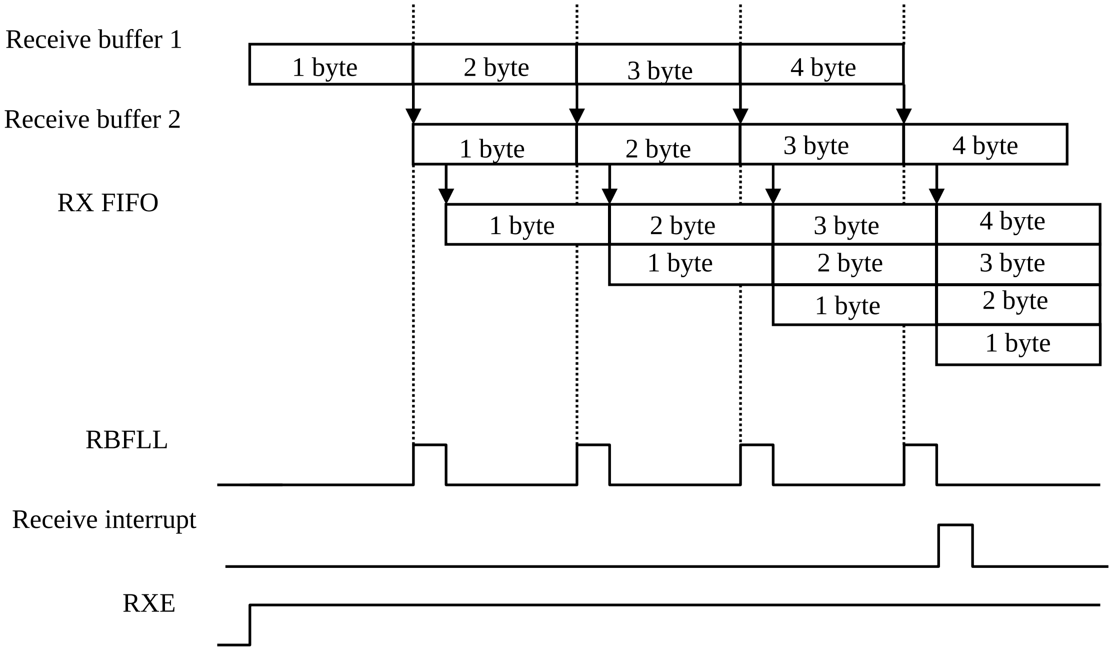


Fig. 14-4 Receive FIFO Operation

14.2.8 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.

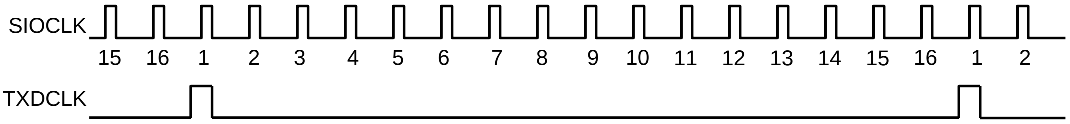


Fig. 14-5 Transmit Clock Generation

14.2.9 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 $\overline{\text{CTS}}$ 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 $\overline{\text{CTS0}}$ pin is set to the "H" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTS0}}$ 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 $\overline{\text{RTS}}$ pin is provided, a handshake control function can be easily implemented by assigning a port for the $\overline{\text{RTS}}$ 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.

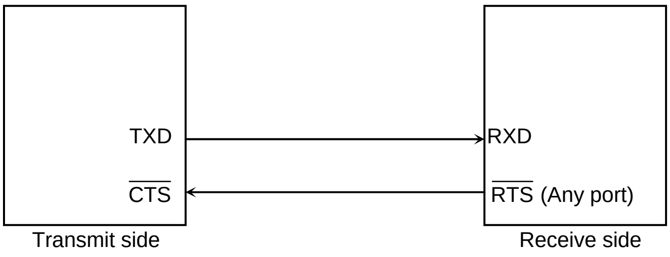
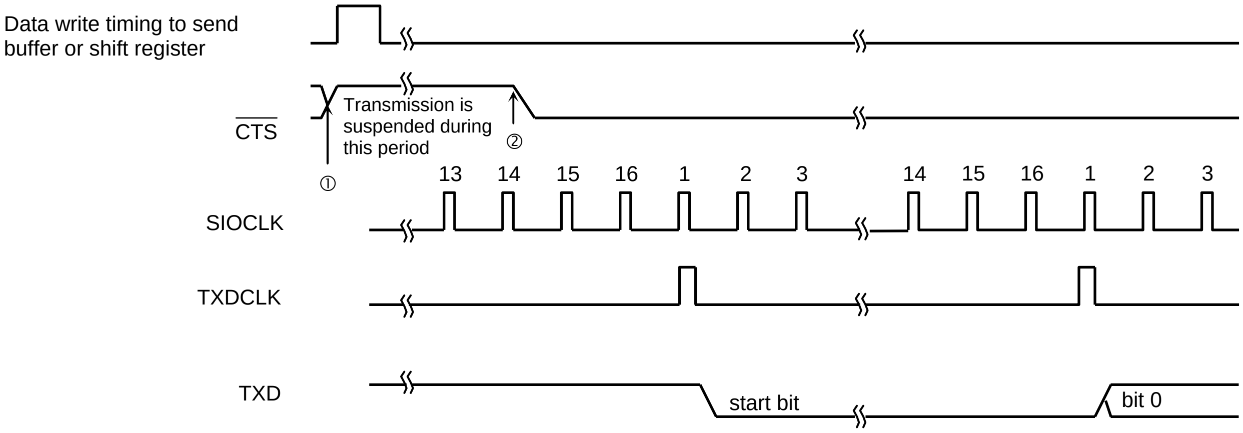


Fig. 14-6 Handshake Function


Data write timing to send buffer or shift register
$\overline{\text{CTS}}$
Transmission is suspended during this period
①
②
13
14
15
16
1
2
3
14
15
16
1
2
3
SIOCLK
TXDCLK
TXD
start bit
bit 0

(Note) ① If the $\overline{\text{CTS}}$ 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 $\overline{\text{CTS}}$ is set to "L."

Fig. 14-7 $\overline{\text{CTS}}$ (Clear to Send) Signal Timing

14.2.10 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> = 01/11), 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 underrun 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.

14.2.11 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.

note. Please execute clear the transmit FIFO after the forwarding mode setting
and the permission of FIFO of SIO when you use the transmit FIFO buffer.

14.2.12 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:

SC0FCNF <4:0> = 01011: Inhibits continued transmission after reaching the fill level.

SC0TFC <1:0> = 00: Sets the interrupt to be generated at fill level 0.

SC0TFC <7:6> = 11: Clears transmit FIFO and sets the condition of interrupt generation

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.

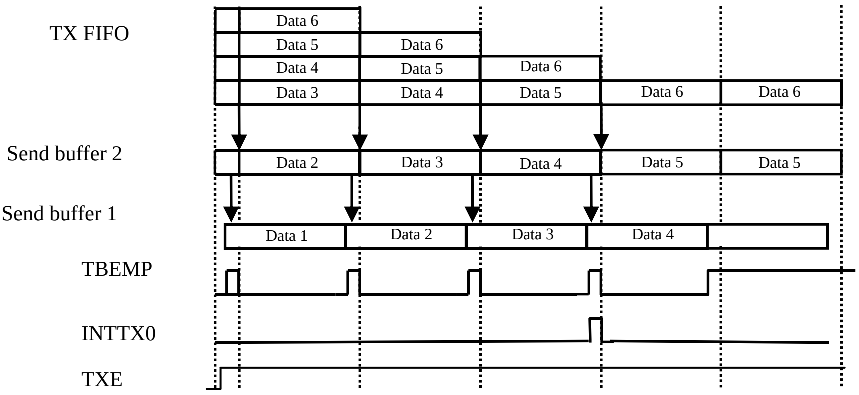


Fig. 14-8 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

SC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

SC0TFC <1:0> = 00: Clears the transmit FIFO and sets the condition of interrupt generation.

SC0TFC <7:6> = 11: Sets the interrupt to be generated at fill level 0.

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.

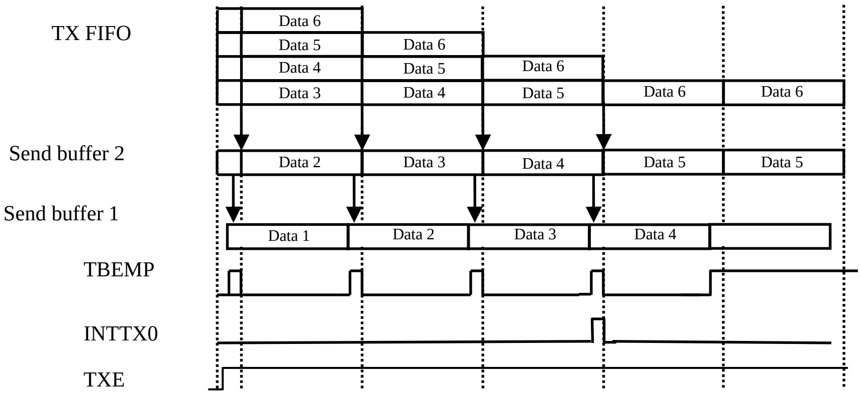


Fig. 14-9 Transmit FIFO Operation

14.2.13 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.

14.2.14 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

14.2.15 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.

14.2.16 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.

14.2.17 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.

14.2.18 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

14.2.19 software reset

Software reset is HSC0MOD2 <SWRST1:0>“10” → “01”
SC0MOD0＜RXE＞、SC0MOD1<TXE>，SC0MOD2＜TBEMP＞,＜RBFLL＞,＜TXRUN＞、
SC0CR＜OERR＞、＜PERR＞、＜FERR＞and internal circuit is initialized.
Other states are maintained.

14.2.20 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.

14.3 Register Description

14.3.1 Operation of Each Channel

Registers and addresses of each channel are shown below.

Table 14.4 Register List

		SIO0		SIO1		SIO2	
Register name (address)	Enable Register	SC0EN	0xFF00_4C00	SC1EN	0xFF00_4C40	SC2EN	0xFF00_4C80
	TX/ RX Buffer Register	SC0BUF	0xFF00_4C04	SC1BUF	0xFF00_4C44	SC2BUF	0xFF00_4C84
	Control Register	SC0CR	0xFF00_4C08	SC1CR	0xFF00_4C48	SC2CR	0xFF00_4C88
	Mode control Register 0	SC0MOD0	0xFF00_4C0C	SC1MOD0	0xFF00_4C4C	SC2MOD0	0xFF00_4C8C
	Baud Rate Generator Control Register	BR0CR	0xFF00_4C10	BR1CR	0xFF00_4C50	BR2CR	0xFF00_4C90
	Baud Rate Generator Control Register 2	BR0ADD	0xFF00_4C14	BR1ADD	0xFF00_4C54	BR2ADD	0xFF00_4C94
	Mode Control Register 1	SC0MOD1	0xFF00_4C18	SC1MOD1	0xFF00_4C58	SC2MOD1	0xFF00_4C98
	Mode Control Register 2	SC0MOD2	0xFF00_4C1C	SC1MOD2	0xFF00_4C5C	SC2MOD2	0xFF00_4C9C
	Receive FIFO Configuration Register	SC0RFC	0xFF00_4C20	SC1RFC	0xFF00_4C60	SC2RFC	0xFF00_4CA0
	Transmit FIFO Configuration Register	SC0TFC	0xFF00_4C24	SC1TFC	0xFF00_4C64	SC2TFC	0xFF00_4CA4
	Receive FIFO Status Register	SC0RST	0xFF00_4C28	SC1RST	0xFF00_4C68	SC2RST	0xFF00_4CA8
	Transmit FIFO Status Register	SC0TST	0xFF00_4C2C	SC1TST	0xFF00_4C6C	SC2TST	0xFF00_4CAC
	FIFO Configuration Register	SC0FCNF	0xFF00_4C30	SC1FCNF	0xFF00_4C70	SC2FCNF	0xFF00_4CB0

14.3.2 Detailed Description of Registers

As channels 0 to 3 have same register set, only channel 0 is described here.

14.3.2.1 Enable Register

SC0EN

	7	6	5	4	3	2	1	0
bit Symbol								SIOE
Read/Write	R							R/W
After reset	0							0
Function	Always reads "0."							SIO operation 0: Disable 1: Enable

<SIOE>: It specifies SIO 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. When SIO operation is disabled, the clock will not be supplied to the SIO module except for the register part and thus power consumption can be reduced. If SIO is enabled once and then disabled, any register setting is maintained.

14.3.2.2 TX/ RX Buffer Register

SC0BUF

	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	TB7~0 : TX buffer/ FIFO RB7~0 : RX buffer/ FIFO							

The buffer register (SC0BUF) functions as TX buffer at writing and RX buffer at reading.

<TB7:0> TX buffer (only for writing)

<RB7:0> RX buffer (only for reading)

14.3.2.3 Control Register

SC0CR

	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 **(for UART)**	Parity **(for UART)** 0: Odd 1: Even	Add parity **(for UART)** 0: Disable 1: Enable	0: Normal operation 1: Error			0: SCLK0 ↑ 1: SCLK0 ↓	**(for I/O interface)** 0: Baud rate generator 1: SCLK0 pin input
				Overrun	Parity/ under-run	Framing		

<RB8>: Indicates 9th received bit in 9 bit UART mode.

<EVEN>: Specifies parity condition.
"0": Odd parity
"1": Even parity
Parity is available for 7 bit/ 8 bit UART mode.

<PE>: Enables or disables parity.
Parity is available for 7 bit/ 8 bit UART mode.

<OERR>: Indicates error flags (overrun error flag, parity error/ underrun error flag and framing error flag).
<PERR>: (Note)
<FERR>:

<SCLKS>: Clock edge selection for data transmission/ reception
"0": Data send/receive at rising edges of SCLK0
"1": Data send/receive at falling edges of SCLK0

<IOC>: I/O interface input clock selection.
"0": Baud rate generator
"1": SCLK0 pin input

(Note) Every error flag is cleared when read.

14.3.2.4 Mode Control Register 0

SC0MOD0

	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 TB2OUT 01: Baud rate generator 10: Internal f_{SYS}/2 clock 11: External clock (SCLK0 input)	

<TB8>: Sets 9th transmit bit in 9 bit UART mode.

<CTSE>: Controls handshake function.
Setting this bit to “1” enables handshake function using $\overline{\text{CTS}}$ pin.

<RXE>: Executes receive control. **(Note)**
Enable this bit after setting each mode register (SC0MOD0, SC0MOD1 and SC0MOD2).

<WU>: Controls wake-up function.
This function is available only for 9 bit UART mode. Setting this bit is ignored in other modes.

	9 bit UART mode	Others
0	An interrupt is generated whenever data is received	don’t care
1	An interrupt is generated when 9th received bit is “1”.	

<SM1:0>: Specifies transfer mode.

<SC1:0>: Specifies transfer clock in UART mode.
In the I/O interface mode, the serial control register (SC0CR) is used for clock.

(Note) With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2).
Then set <RXE> to "1."

14.3.2.5 Mode Control Register 1

SC0MOD1

	7	6	5	4	3	2	1	0
bit Symbol	I2S0	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	–
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	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 (for I/O interface) 000: None 100: 8SCLK 001: 1SCLK 101:16SCLK 010: 2SCLK 110: 32SCLK 011: 4SCLK 111: 64SCLK			Write "0."

<I2S0>: Specifies the IDLE mode operation.

<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.

<TXE>: This bit enables transmission and is valid for all the transfer modes. (Note) If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<SINT2:0>: This parameter is valid for I/O interface mode when a clock is not input from SCLK0 pin (invalid for other modes).
Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.

(Note) Enable <TXE> bit after setting other bits.

14.3.2.6 Mode Control Register 2

SC0MOD2

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R			R/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 (for UART) 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	

<TBEMP>: If double buffering is disabled, this flag is insignificant.
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."

<RBFLL>: If double buffering is disabled, this flag is insignificant.
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."

<TXRUN>: This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> show the following status.

<TXRUN>	<TBEMP>	Status
1	-	Transmission in progress
0	1	Transmission completed
	0	Wait state for transmission with next data in a send buffer.

<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.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, you need to fix it to LSB first.

<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 when receiving data in I/O interface mode (SCLK input) and UART mode regardless of the <WBUF> setting.

<SWRST1:0>: Overwriting "01" in place of "10" generates a software reset.
When this software reset is executed, the following bits and their internal circuits are initialized.
(Note 1, 2 and 3)

Register name	Bit
SC0MOD0	RXE
SC0MOD1	TXE
SECMOD2	TBEMP,RBFLL,TXRUN,
SC0CR	OERR,PERR,FERR

(Note 1) While data transmission is in progress, any software reset operation must be executed twice in succession.

(Note 2) To complete software reset, it takes 2 clocks after executing an instruction.
Executing SYNC and NOP instructions after software reset is recommended.

(Note 3) When software reset is executed, the bits listed in the <SWRST1:0> description are initialized. It requires resetting of the mode registers and control register.

14.3.2.7 Baud Rate Generator Control Register (BR0CR)
Baud Rate Generator Control Register 2 (BR0ADD)

BR0CR

	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	Specify baud rate generator input clock 00: ϕT1 01: ϕT4 10: ϕT16 11: ϕT64		Divide ratio "N" 0000: 16 0001: 1 0010: 2 : 1111: 15			

BR0ADD

	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
Function	Always reads "0."				Specify K for the "N + (16 - K)/16" division 0000: Setting prohibited 0001: K=1 0010: K=2 : 1111: K=15			

<RB0ADDE>: Enables or disables N+(16-K)/16 division function.
This function is available only for UART mode.

<RB0CK1:0>: Selects input clock to the baud rate generator.

<RB0S3:0>: Sets divide ratio “N” of the baud rate generator.

<RB0K3:0>: Specifies K for the "N + (16 - K)/16" division.

Divide ratio of baud rate generator is set in the above two registers.
Table 14. shows the divide ratio configuration.

Table 14.5 Divide ratio setting

	BR0ADDE=0	BR0ADDE=1 **(Note 1)** (available for UART mode)
BR0S	Set divide ratio “N” **(Note 2, 3)**	
BR0K	No setting required	Set “K” **(Note 4)**
Divide ratio	N	$N + \frac{(16 - K)}{16}$

(Note 1) To use the "N + (16 - K)/16" division function, be sure to set BR0ADDE <BR0ADDE> to "1" after setting the K value (K = 1 to 15) to BR0ADD <BR0K3:0>.

(Note 2) The division ratio "1" (“0001”) of the baud rate generator can be specified only when

・the "N + (16 - K)/16" division function is not used In the UART mode.

・double buffering is used in the I/O interface mode.

(Note 3) To use the "N + (16 - K)/16" division function, neither the division ratio "1" (“0001”) nor “16” (“0000”) can be set.

(Note 4) “0” cannot be set as K value.

14.3.2.8 FIFO Configuration Register

SC0FCNF

	7	6	5	4	3	2	1	0
bit Symbol	Reserved	Reserved	Reserved	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

<RFST>: Specifies bytes used in RX FIFO. **(Note)**
0: The maximum number of bytes of the FIFO configured is available.
(See description of <CNFG> bit.)
1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL1:0>.

<TFIE>: When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFIE>: When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<RXTXCNT>: The function to automatically disable RXE/TXE bits is disabled.
If “1” is set, it functions as shown below according to the communication method configured (the communication method can be set in the mode control register 1 SC0MOD1<FDPX1:0>).

Half duplex RX	When the RX FIFO is filled up to the specified number of valid bytes, SC0MOD0<RXE> is automatically set to "0" to inhibit further reception.
Half duplex TX	When the TX FIFO is empty, SC0MOD1<TXE> is automatically set to "0" to inhibit further transmission.
Full duplex	When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<CNFG>: This parameter is to enable FIFO.
Setting “1” enables FIFO. If enabled, the SCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows: (the communication method can be set in the mode control register 1 SC0MOD1<FDPX1:0>).

Half duplex RX	RX FIFO 4byte
Half duplex TX	TX FIFO 4byte
Full duplex	RX FIFO 2byte＋TX FIFO 2byte

(Note) 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.

14.3.2.9 Receive FIFO Configuration Register

SC0RFC

	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
Function	Clear RX FIFO 1: Clear Always reads "0."	Select interrupt generation condition	Always reads "0."				FIFO fill level to generate RX interrupts 00: 4 bytes (2 bytes if full duplex) 01:1byte 10:2byte 11:3byte	

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.

<RFCS>: Clears RX FIFO.
Writing "1" clears FIFO. If not, "0" is read.

<RFIS>: Selects interrupt generation condition.
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.

<RIL1:0>: Specifies FIFO fill level. **(Note)**

	Full duplex	Others
00	2byte	4byte
01	1byte	1byte
10	2byte	2byte
11	1byte	3byte

(Note) RIL1 is ignored if FDPX1:0=11 (full duplex).

14.3.2.10 Transmit FIFO Configuration Register

SC0TFC

	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
Function	Clear TX FIFO 1: Clear Always reads "0."	Select interrupt generation condition	Always reads "0."				FIFO fill level to generate TX interrupts 00:Empty 01:1byte 10:2byte 11:3byte **Note: TIL1 is ignored if 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.

<TFCS>: Clears TX FIFO.
Writing “1” clears FIFO. If not, “0” is read.

<TFIS>: Selects interrupt generation condition.
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.

<TIL1:0>: Specifies FIFO fill level. **(Note)**

	Full duplex	Others
00	Empty	Empty
01	1byte	1byte
10	Empty	2byte
11	1byte	3byte

(Note) TIL1 is ignored if FDPX1:0=11 (full duplex).

14.3.2.11 Receive FIFO Status Register

SC0RST

	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
Function	RX FIFO Overrun 1: Generated	Always reads "0."				Status of RX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

<ROR>: Flag for RX FIFO overrun.
This parameter is set to “1” if overrun occurs **(note).**

<RLVL2:0>: Indicates status of RX FIFO fill level.

(Note) The <ROR> bit is cleared to "0" when receive data is read from the SC0BUF register.

14.3.2.12 Transmit FIFO Status Register

		7	6	5	4	3	2	1	0
	bit Symbol	TUR					TLVL2	TLVL1	TLVL0
SC0TST	Read/Write	R	R				R		
	After reset	1	0				0	0	0
	Function	TX FIFO Underrun 1: Generated. Cleared to “0” when FIFO is written by received data.	Always reads "0."				Status of TX FIFO fill level 000:Empty 001:1Byte 010:2Byte 011:3Byte 100:4Byte		

<TUR>: Flag for TX FIFO underrun.
This parameter is set to “1” if underrun occurs **(note).**

<TLVL2:0>: Indicates status of TX FIFO fill level.

(Note) The <TUR> bit is cleared to "0" when receive data is read from the SC0BUF register.

14.4 Operation of Each Circuit (Channel 0)

As channels 0 to 3 operate identically, only channel 0 is described here.

14.4.1 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 <PRCK2:0> to provide the frequency of either fperiph/2, fperiph/4, fperiph/8, fperiph/16 or fperiph/32.

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 in the mode control register 0 (SC0MOD0<SC1:0>).

Table 14. shows Clock Resolution to the Baud Rate Generator.

The serial interface baud rate generator uses four different clocks, i.e., ϕT1, ϕT4, ϕT16 and ϕT64, supplied from the prescaler output clock.

Table 14.6 Clock Resolution to the Baud Rate Generator (fsys=80MHz)

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000 (fc)	000(fperiph/2)	—	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		001(fperiph/4)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
	100(fc/2)	000(fperiph/2)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		001(fperiph/4)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		010(fperiph/8)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		011(fperiph/16)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
		100(fperiph/32)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)
	101(fc/4)	000(fperiph/2)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		001(fperiph/4)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		010(fperiph/8)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
		011(fperiph/16)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)
		100(fperiph/32)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)	fc/2^{14}(204.8μs)
	110(fc/8)	000(fperiph/2)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		001(fperiph/4)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
		010(fperiph/8)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)
		011(fperiph/16)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)	fc/2^{14}(204.8μs)
		100(fperiph/32)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)	fc/2^{15}(409.6μs)
	111(fc/16)	000(fperiph/2)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
		001(fperiph/4)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)
		010(fperiph/8)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)	fc/2^{14}(204.8μs)
		011(fperiph/16)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)	fc/2^{13}(102.4μs)	fc/2^{15}(409.6μs)
		100(fperiph/32)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)	fc/2^{14}(204.8μs)	fc/2^{16}(819.2μs)
1 (fc)	000 (fc)	000(fperiph/2)	—	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		001(fperiph/4)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
	100(fc/2)	000(fperiph/2)	—	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		001(fperiph/4)	fc/2^{3}(0.1μs)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
	101(fc/4)	000(fperiph/2)	—	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		001(fperiph/4)	—	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		010(fperiph/8)	fc/2^{4}(0.2μs)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
	110(fc/8)	000(fperiph/2)	—	—	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)
		001(fperiph/4)	—	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)
		010(fperiph/8)	—	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	fc/2^{5}(0.4μs)	fc/2^{7}(1.6μs)	fc/2^{9}(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)
	111(fc/16)	000(fperiph/2)	—	—	fc/2^{6}(0.8μs)	fc/2^{8}(3.2μs)

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK2:0>	Prescaler output clock resolution			
			ϕT1	ϕT4	ϕT16	ϕT64
		001(fperiph/4)	—	—	fc/2^7(1.6μs)	fc/2^9(6.4μs)
		010(fperiph/8)	—	fc/2^6(0.8μs)	fc/2^8(3.2μs)	fc/2^{10}(12.8μs)
		011(fperiph/16)	—	fc/2^7(1.6μs)	fc/2^9(6.4μs)	fc/2^{11}(25.6μs)
		100(fperiph/32)	fc/2^6(0.8μs)	fc/2^8(3.2μs)	fc/2^{10}(12.8μs)	fc/2^{12}(51.2μs)

(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.

14.4.2 Serial Clock Generation Block

This block generates basic transmit and receive clocks.

In this block, clock is determined according to baud rate generator and specified mode and register setting.

14.4.2.1 Clock Selection Circuit

Serial clock is determined according to mode and register setting specified.

Specify mode in the mode control register 0 (SC0MOD0<SM1:0>).

To use I/O interface mode, clock is specified in the control register SC0CR.

To use UART mode, clock is specified in the mode control register0 (SC0MOD0<SC1:0>).

Table 14. and Table 14. show details of clock selection in I/O interface mode and UART mode respectively.

Table 14.7 Clock selection in I/O interface mode

Mode SC0MOD0<SM1:0>	Input/ output selection SC0CR<IOC>	Clock edge selection SC0CR<SCLKS>	Clock selection
I/O interface mode	SCLK output	- (Fixed to rising edge)	Baud rate generator output divided by 2
	SCLK input	Rising edge	Rising edge of SCLK input
		Falling edge	Falling edge of SCLK input

Table 14.8 Clock selection in UART mode

Mode SC0MOD0<SM1:0>	Clock selection SC0MOD0<SC1:0>
UART mode	Timer output
	Baud rate generator
	fsys/2
	SCLK input

14.4.2.2 Baud Rate Generator

The baud rate generator divides clocks input from prescaler, and generates transmit and receive clocks.

(1) Input clock

The baud rate generator uses one of four clocks (ϕT1, ϕT4, ϕT16 or ϕT64) supplied from the prescaler output clock. This input clock selection is made by setting the baud rate generator control register, BR0CR <BR0CK1:0>.

Specify the divide ratio of the output clock with the baud rate generator control register BR0CR<BR0ADDE><BR0S3:0> and the baud rate generator control register 2 BR0ADD<BR0K3:0>.

(2) UART mode

Table 14.9 UART mode

BR0CR<BR0ADDE>	BR0ADD<BR0K3:0>	BR0CR<BR0S3:0>
“0”	Setting invalid	Setting valid Set divide ratio “N” （N=1,2,3…16）
“1” N+ $\frac{(16-K)}{16}$ division	Setting valid Set divide ratio “K” （K=1,2,3…15）	Setting valid Set divide ratio “N” （N=2,3…15）
	–	N=1,16: setting prohibited

15. Serial Channel (HSIO)

15.1 Features

This device has three serial I/O channels: HSIO0 to HSIO2.

15.1.1 Operation Modes

Four operation modes (mode 0 through mode 3) are provided to HSIO. Table 15.1 shows data format of each mode.

Table 15.1 Data Format

Mode	Type	Data length	Transfer	Add parity	Stop bit length (transfer)
Mode 0	Synchronous mode (I/O interface mode)	8 bit	LSB first or MSB first	×	-
Mode 1	Asynchronous mode (UART mode)	7 bit	LSB first	○	1 or 2
Mode 2		8 bit		○	
Mode 3		9 bit		×	

Mode 0 is to send and receive I/O data and associated synchronization signals (HSCLK) to extend I/O. HSCLK can be used for input and output.
You can select either of LSB or MSB to send first. Neither adding parity nor stop bit is available.

Modes 1 through 3 execute asynchronous communication and are designed to send LSB first.

In the 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).

Stop bit length at transmission can be selected from 1 bit or 2 bits. At reception, stop bit is 1 bit.

Hereinafter synchronous mode is referred to as I/O interface mode, asynchronous mode is referred to as UART mode or 7 bit/ 8 bit/ 9 bit UART mode, which includes TX/RX data length.

15.1.2 Data Format

Table 15.1 shows data format of each mode.

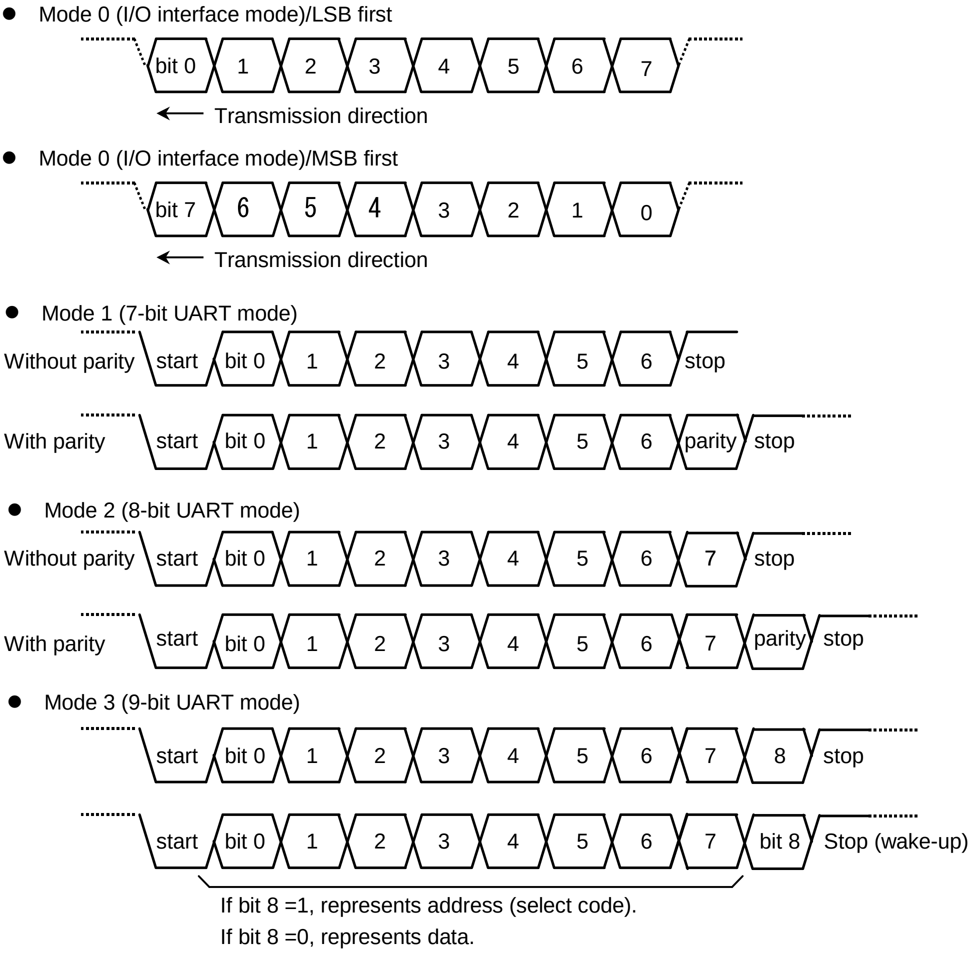


Fig. 15.1 Data Format

15.2 Block Diagram (Channel 0)

Fig. 15.2 shows the block diagram of HSIO0.

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.

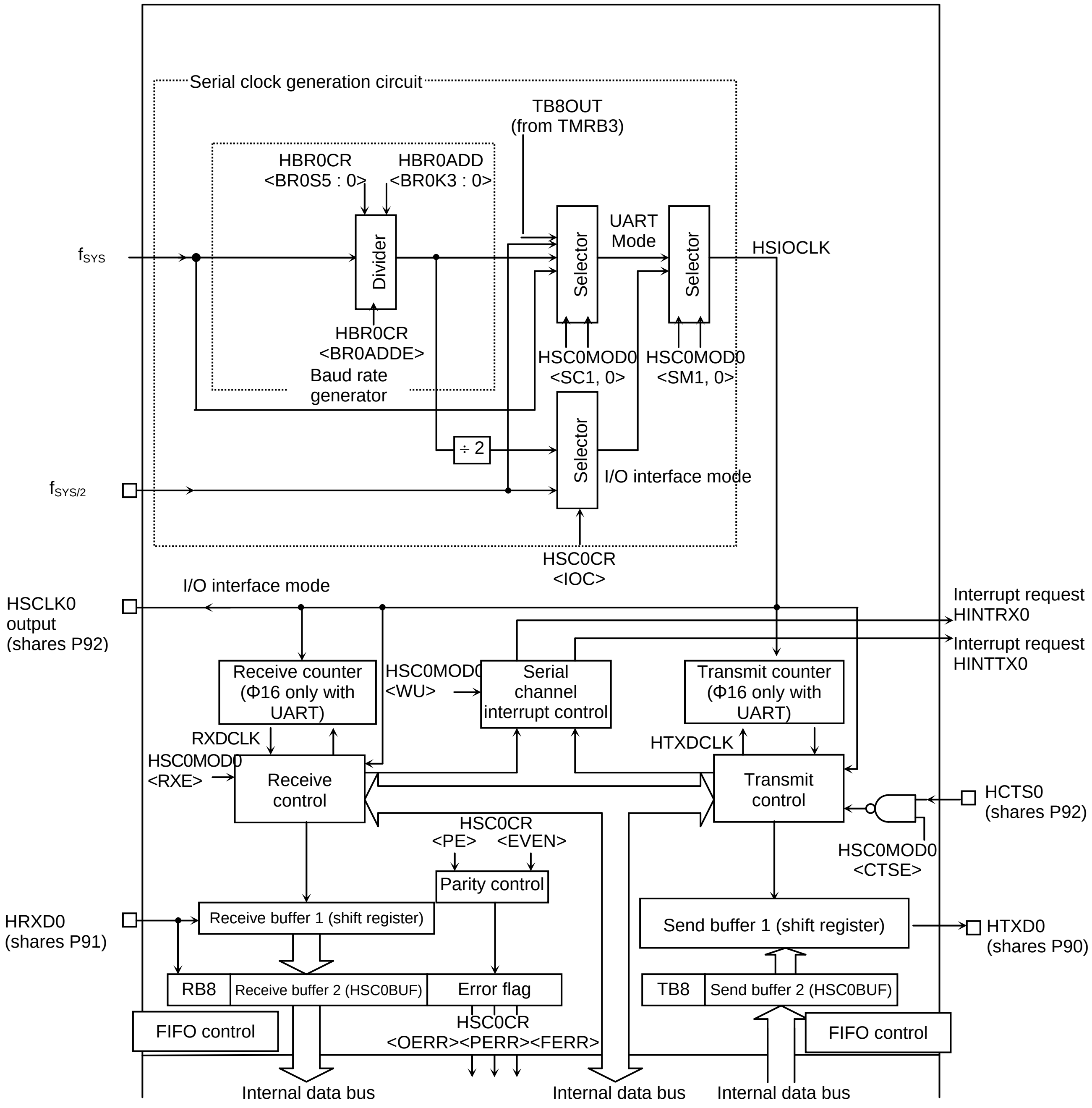


Fig. 15.2 HSIO0 Block Diagram

15.3 Register Description

15.3.1 Operation of Each Channel

Registers and addresses of each channel are shown below.

Table 15.2 Register List

		HSIO0	HSIO1	HSIO2
Register name (address)	TX/ RX Buffer Register	HSC0BUF 0xFF00_1800	HSC1BUF 0xFF00_1810	HSC2BUF 0xFF00_1820
	Baud Rate Generator Control Register 2	HBR0ADD 0xFF00_1804	HBR1ADD 0xFF00_1814	HBR2ADD 0xFF00_1824
	Mode control Register 1	HSC0MOD1 0xFF00_1805	HSC1MOD1 0xFF00_1815	HSC2MOD1 0xFF00_1825
	Mode control Register 2	HC0MOD2 0xFF00_1806	HSC1MOD2 0xFF00_1816	HSC2MOD2 0xFF00_1826
	Enable Register	HSC0EN 0xFF00_1807	HSC1EN 0xFF00_1817	HSC2EN 0xFF00_1827
	Receive FIFO Configuration Register	HSC0RFC 0xFF00_1808	HSC1RFC 0xFF00_1818	HSC2RFC 0xFF00_1828
	Transmit FIFO Configuration Register	HSC0TFC 0xFF00_1809	HSC1TFC 0xFF00_1819	HSC2TFC 0xFF00_1829
	Receive FIFO Status Register	HSC0RST 0xFF00_180A	HSC1RST 0xFF00_181A	HSC2RST 0xFF00_182A
	Transmit FIFO Status Register	HSC0TST 0xFF00_180B	HSC1TST 0xFF00_181B	HSC2TST 0xFF00_182B
	FIFO Configuration Register	HSC0FCNF 0xFF00_180C	HSC1FCNF 0xFF00_181C	HSC2FCNF 0xFF00_182C
	Control Register	HC0CR 0xFF00_180D	HSC1CR 0xFF00_181D	HSC2CR 0xFF00_182D
	Mode control Register 0	HSC0MOD0 0xFF00_180E	HSC1MOD0 0xFF00_181E	HSC2MOD0 0xFF00_182E
	Baud Rate Generator Control Register	HBR0CR 0xFF00_180F	HBR1CR 0xFF00_181F	HBR2CR 0xFF00_182F

Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses the sys/2 clock.

The baud rate generator contains built-in dividers for divide by 1, (N + m/16), and 64 where N is a number from 2 to 63 and m is a number from 0 to 15. The division is performed according to the settings of the baud rate control registers HBR0CR <BR0ADDE> <BR3S3:0> and HBR0ADD <BR0K3:0> to determine the resulting transfer rate.

- UART Mode:

1) If HBR0CR <BR0ADDE> = 0,

   The setting of HBR0ADD <BR0K3:0> is ignored and the counter is divided by N where N is the value set to HBR0CR <BR0S5:0>. (N = 1 to 64).

2) If HBR0CR <BR0ADDE> = 1,

   The N + (16 - K)/16 division function is enabled and the division is made by using the values N (set in HBR0CR <BR0S3:0>) and K (set in HBR0ADD<BR0K3:0>). (N = 2 to 63, 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 HBR0CR<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 HBR0CR <BR0ADDE> to "0."

- Baud rate calculation to use the baud rate generator:

1) UART mode

$$\text{Baud rate} = \frac{fsys}{\text{Frequency divided by the divide ratio}} /16$$

The highest baud rate out of the baud rate generator is 5 Mbps when fsys is 80 MHz.

2) I/O interface mode

$$\text{Baud rate} = \frac{fsys}{\text{Frequency divided by the divide ratio}} \; /2$$

The highest baud rate will be generated when fsys is 80 MHz. If double buffering is used, the divide ratio can be set to "2" and the resulting output baud rate will be 10 Mbps. (If double buffering is not used, the highest baud rate will be 5 Mbps applying the divide ratio of "2.")

- <u>Example baud rate setting:</u>

1) Division by an integer (divide by N):

Using the baud rate generator input clock sys, setting the divide ratio N (HBR0CR<BR0S5:0>) = 4, and setting HBR0CR<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)

$$\text{Baud rate} = \frac{\text{fsys}}{4} \; /16$$

$= 80 \times 10^6 / 4 / 16 =$1250 k (bps)

> **(Note) The divide by (N + (16-K)/16) function is inhibited and thus HBR0ADD <BR0K3:0> is ignored.**

2) For divide by N + (16-K)/16 (only for UART mode):

Using the baud rate generator fsys, setting the divide ratio N (HBR0CR<BR3S5:0>) = 4, setting K (HBR0ADD<BR3K3:0>) = 14, and selecting HBR0CR<BR3ADDE> = 1, the resulting baud rate is calculated as follows:

* Clocking conditions
System clock : High-speed (fc)
High-speed clock gear : x 1 (fc)

$$\text{Baud rate} = \frac{\text{Fsys}}{4 + \frac{(16 - 14)}{16}} \; /16$$

$= 80\times 10^6 / (4 + \frac{2}{16}) / 16 =$ 121.2 k (bps)

High-speed Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

- <u>I/O interface mode:</u>

  In the HSCLK output mode with the HC0CR <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 HSCLK input mode with HC0CR <IOC> set to "1," rising and falling edges are detected according to the HC0CR <HSCLKS> setting to generate the basic clock.

- <u>Asynchronous (UART) mode:</u>

  According to the settings of the serial control mode register HSC0MOD0 <SC1:0>, either the clock from the baud rate register, the system clock (f_{SYS}), the internal output signal of the TMRB3 timer, or the external clock (HSCLKO pin) is selected to generate the basic clock, HSIOCLK.

Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by HSIOCLK. Sixteen HSIOCLK 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.

Receive Control Unit

- <u>I/O interface mode:</u>

  In the HSCLK output mode with HC0CR <IOC> set to "0," the HRXD0 pin is sampled on the rising edge of the shift clock output to the HSCLK0 pin.

  In the HSCLK input mode with HC0CR <IOC> set to "1," the serial receive data HRXD0 pin is sampled on the rising or falling edge of HSCLK input depending on the HC0CR <HSCLKS> setting.

- <u>Asynchronous (UART) mode:</u>

  The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

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 (HSC0BUF). At the same time, the receive buffer full flag (HC0MOD2 "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 (HSCOFCNF <CNFG> = 0 and <FDPX1:0> = 01/11), the HINTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (HSCNFCNF <CNFG> = 1 and SCOMOD1<FDPX1:0> = 01/11), an interrupt will be generated according to the HSC0RFC <RIL1:0> setting.

The CPU will read the data from either the second receive buffer (HSC0BUF) 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 <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 (HSC0BUF) or the receive FIFO.

If HSCLK is set to generate clock output in the I/O interface mode, the double buffer control bit HC0MOD2 <WBUF> can be programmed to enable or disable the operation of the second receive buffer (HSCOBUF).

By disabling the second receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (HSCOFCNF <CNFG> = 0 or 1 and <FDPX1:0> = 10), handshaking with the other side of communication can be enabled and the HSCLK 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 HSCLK output resumes.

If the second receive buffer (i.e., double buffering) is enabled but the receive FIFO is not enabled, the HSCLK 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 HSCLK output is resumed upon generation of the receive interrupt HINTRX. Therefore, no buffer overrun error will be caused in the I/O interface HSCLK output mode regardless of the setting of the double buffer control bit HC0MOD2 <WBUF>.

If the second receive buffer (double buffering) is enabled and the receive FIFO is also enabled (HSCNFCNF <CNFG> = 1 and <FDPX1:0> = 01/11), the HSCLK output will be stopped when the receive FIFO is full (according to the setting of HSCOFNCF <RFST>) and both the first and second receive buffers contain valid data. Also in this case, if HSCOFCNF <RXTXCNT> has been set to "1," the receive control bit RXE will be automatically cleared upon suspension of the HSCLK output. If it is set to "0," automatic clearing will not be performed.

(Note) In this mode, the HC0CR <OERR> flag is insignificant and the operation is undefined. Therefore, before switching from the HSCLK output mode to another mode, the HC0CR 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 (HSC0BUF) 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 HC0CR <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 HC0CR <RB8>.

In the 9-bit UART mode, the slave controller can be operated in the wake-up mode by setting the wake-up function HSC0MOD0 <WU> to "1." In this case, the interrupt HINTRX0 will be generated only when HC0CR <RB8> is set to "1."

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 HSC0FCNF register and <FDPX1:0> of the HSC0MOD1 register, the 32-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.

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 HSC0FCNF register and <FDPX1:0> of the HSC0MOD1 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.

Receive FIFO Operation

① I/O interface mode with HSCLK output:

The following example describes the case a 32-byte data stream is received in the half duplex mode:

HSC0FCNF <4:0>=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.

HSC0RFC<4:0>=11111: Sets the interrupt to be generated at fill level 32.

HSC0RFC<7:6>=01: Clears receive FIFO and sets the condition of interrupt generation.

n this condition, 32-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 (HSCLK is stopped).

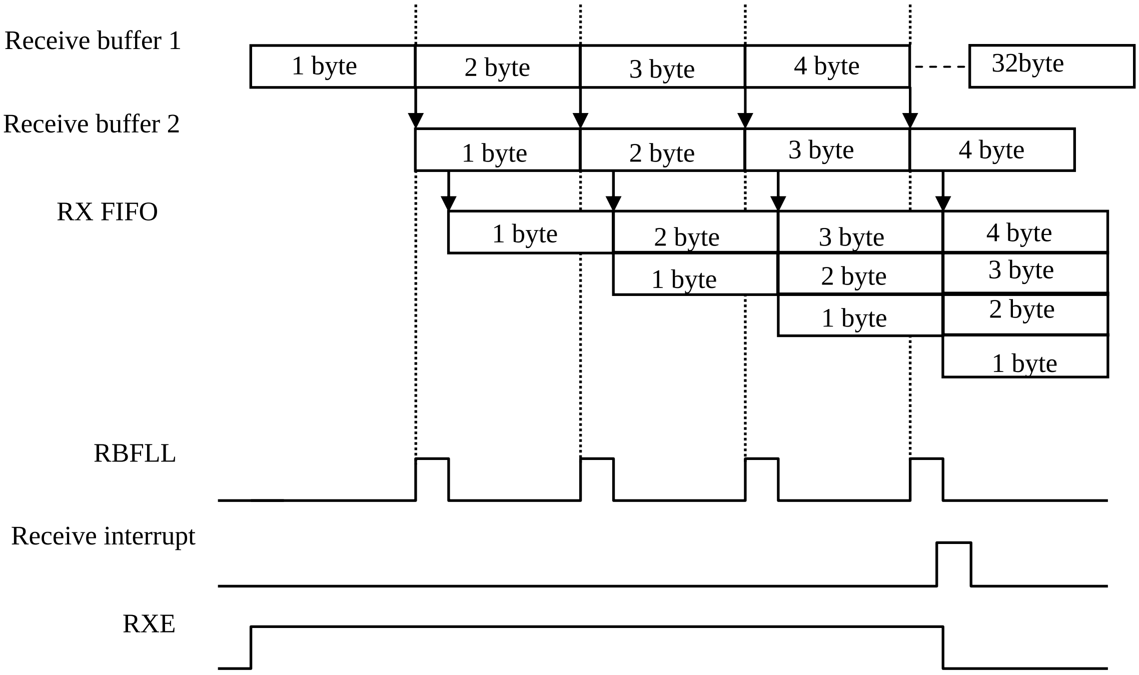


Fig. 15-3 Receive FIFO Operation

② I/O interface mode with HSCLK input:

The following example describes the case a 32-byte data stream is received:

HSC0FCNF <4:0> = 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.

SC0RFC <4:0> = 11111: Sets the interrupt to be generated at fill level 32.

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

In this condition, 32-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 32-byte data can be received before data is fully read from the FIFO.

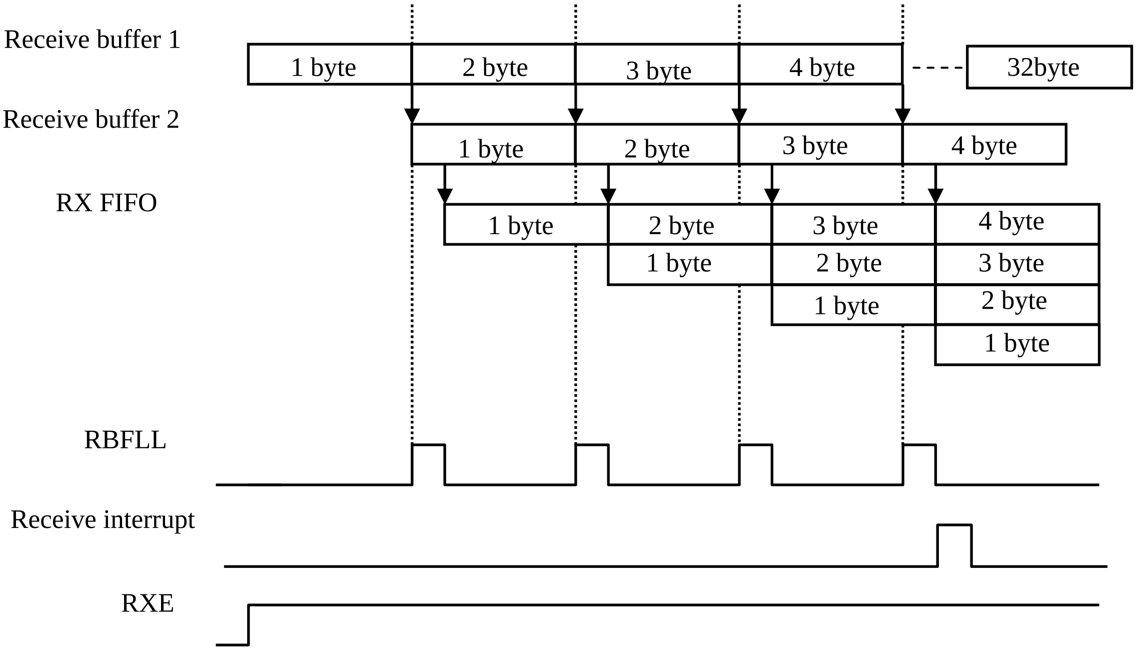


Fig. 15-4 Receive FIFO Operation

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.

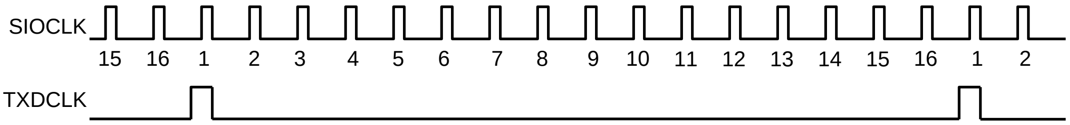


Fig. 15-5 Transmit Clock Generation

Transmit Control Unit

- I/O interface mode:

  In the HSCLK output mode with HC0CR <IOC> set to "0," each bit of data in the send buffer is output to the HTXD0 pin on the rising edge of the shift clock output from the HSCLK0 pin.

  In the HSCLK input mode with HC0CR <IOC> set to "1," each bit of data in the send buffer is output to the HTXD0 pin on the rising or falling edge of the input HSCLK signal according to the HC0CR <HSCLKS> setting.

- Asynchronous (UART) mode:

  When the CPU writes data to the send buffer, data transmission is initiated on the rising edge of the next HTXDCLK and the transmit shift clock (HTXDSFT) is also generated.

- Handshake function

The H$\overline{\text{CTS}}$ pin enables frame by frame data transmission so that overrun errors can be prevented. This function can be enabled or disabled by HSC0MOD0 <CTSE>.

When the H$\overline{\text{CTS0}}$ pin is set to the "H" level, the current data transmission can be completed but the next data transmission is suspended until the H$\overline{\text{CTS0}}$ pin returns to the "L" level. However in this case, the HINTTX0 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 H$\overline{\text{RTS}}$ pin is provided, a handshake control function can be easily implemented by assigning a port for the H$\overline{\text{RTS}}$ 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.

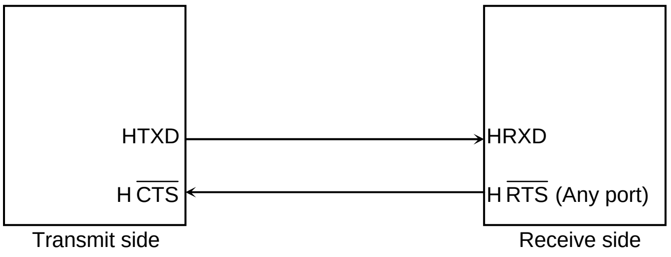
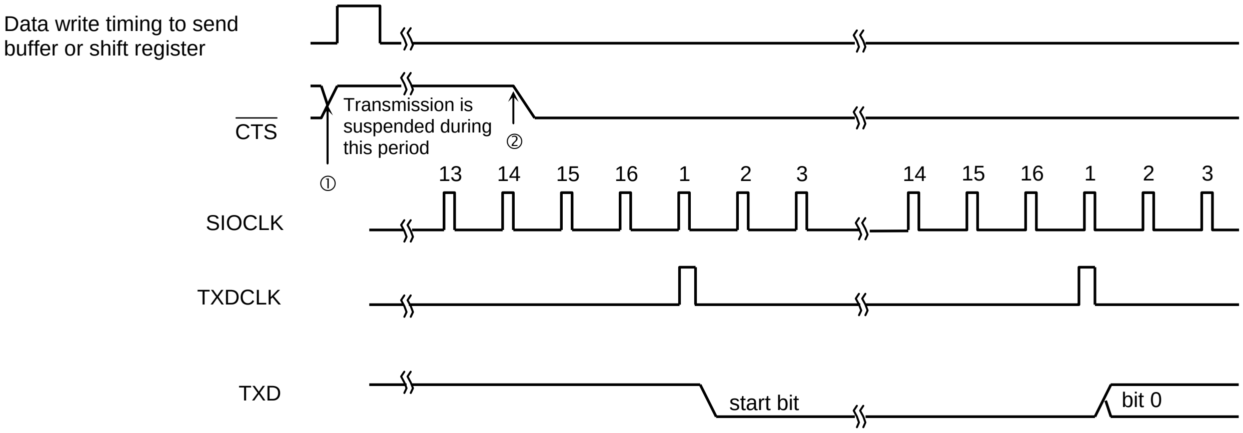


Fig. 15-6 Handshake Function


Data write timing to send buffer or shift register
$\overline{\text{CTS}}$
Transmission is suspended during this period
①
②
13
14
15
16
1
2
3
14
15
16
1
2
3
SIOCLK
TXDCLK
TXD
start bit
bit 0

(Note)

① **If the $\overline{\text{CTS}}$ 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 $\overline{\text{CTS}}$ is set to "L."**

Fig. 15-7 $\overline{\text{CTS}}$ (Clear to Send) Signal Timing

Transmit Buffer

The send buffer (HSC0BUF) 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 (HC0MOD2). 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 (HSCOFCNF <CNFG> = 0 or 1 and <FDPX1:0> = 01/11), the HINTTX interrupt is generated at the same time and the send buffer empty flag <TBEMP> of HC0MOD2 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 (HSCNFCNF <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 HSCLK 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 (HC0CR) <PERR> parity/under-run flag is set.

If the transmit FIFO is enabled in the I/O interface HSCLK 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 HSCLK output mode, when data in send buffer 2 is moved to send buffer 1 and the data transmission is completed, the HSCLK output stops. So, no under-run errors can be generated.

If the transmit FIFO is enabled in the I/O interface HSCLK output mode, the HSCLK 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 HSCLK output mode, the HC0CR <PEER> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the HSCLK output mode to another mode, HC0CR 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 HINTTX 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.

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 HSC0FCNF register and <FDPX1:0> of the HSC0MOD1 register, the 32-byte send buffer can be enabled. In the UART mode or I/O interface mode, up to 32bytes 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.

note. Please execute clear the transmit FIFO after the forwarding mode setting
and the permission of FIFO of SIO when you use the transmit FIFO buffer.

Transmit FIFO Operation

① I/O interface mode with HSCLK output (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

HSC0FCNF <4:0> = 01011: Inhibits continued transmission after reaching the fill level.

HSC0TFC <5:0> = 00000: Sets the interrupt to be generated at fill level 0.

HSC0TFC <7:6> = 01: Clears transmit FIFO and sets the condition of interrupt generation

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.

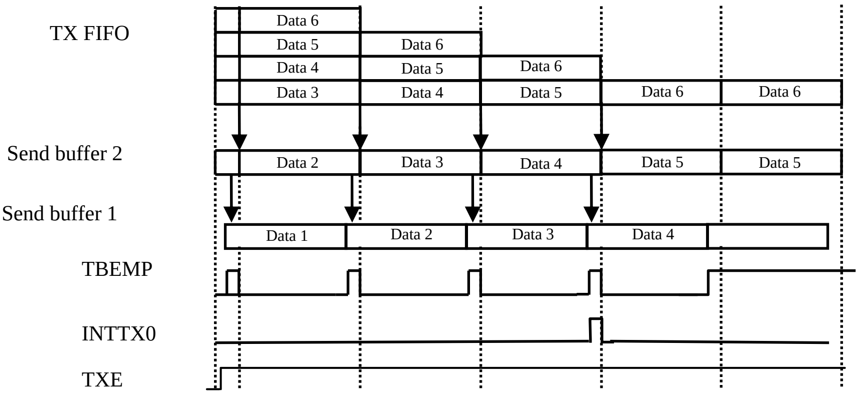


Fig. 15-8 Transmit FIFO Operation

② I/O interface mode with HSCLK input (normal mode):

The following example describes the case a 32-byte data stream is transmitted:

HSC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

HSC0TFC <5:0> = 000000: Clears the transmit FIFO and sets the condition of interrupt generation.

HSC0TFC <7:6> = 11: Sets the interrupt to be generated at fill level 0.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 32 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.

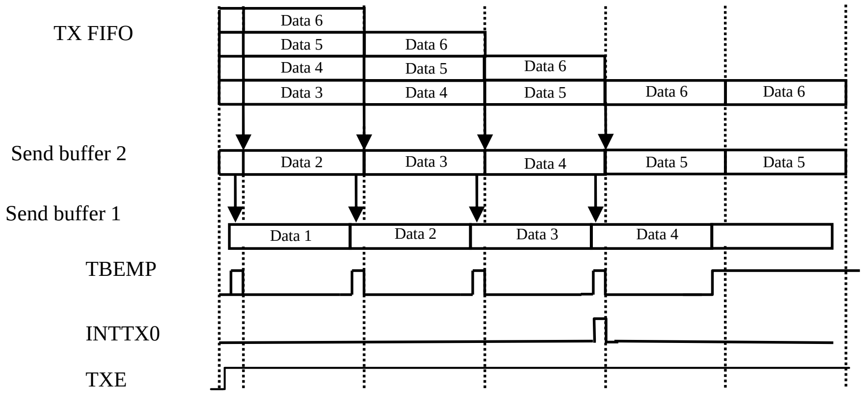


Fig. 15-9 Transmit FIFO Operation

Parity Control Circuit

If the parity addition bit <PE> of the serial control register HC0CR 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 HC0CR 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 (HSC0BUF). After data transmission is complete, the parity bit will be stored in HSC0BUF 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 (HSC0BUF). In the 7-bit UART mode, the parity generated is compared with the parity stored in HSC0BUF <RB7>, while in the 8-bit UART mode, it is compared with the bit 7 <RB8> of the HC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the HC0CR register is set.

In the I/O interface mode, the HC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

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 HC0CR

   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 HSCLK 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 HC0CR 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 HC0MOD2 is set to "1" in the HSCLK 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 HSCLK 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 HC0CR 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, HC0MOD2, 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 (HSCLK input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O interface (HSCLK output)	OERR	Operation undefined
	PERR	Operation undefined
	FERR	Fixed to 0

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 HC0MOD2 serial mode control register 2. Don't switch the direction when data is being transferred.

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 HC0MOD2 register.

Status Flag

If the double buffer function is enabled (HC0MOD2 <WBUF> = "1"), the bit 6 flag <RBFLL> of the HC0MOD2 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 (HC0MOD2 <WBUF> = "1"), the bit 7 flag <TBEMP> of the HC0MOD2 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.

Configurations of Send/Receive Buffers

		<WBUF> = 0	<WBUF> = 1
UART	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O interface (HSCLK input)	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O interface (HSCLK output)	Transmit buffer	Single	Double
	Receive buffer	Single	Double

software reset

Software reset is HC0MOD2 <SWRST1:0>"10" → "01"

SC0MOD0＜RXE＞、HSC0MOD1<TXE>，HC0MOD2＜TBEMP＞,＜RBFLL＞,＜TXRUN＞、HC0CR＜OERR＞、＜PERR＞、＜FERR＞and internal circuit is initialized.
Other states are maintained.

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)	HSCLK output mode	Immediately after the rising edge of the last HSCLK
	HSCLK input mode	Immediately after the rising or falling edge of the last HSCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (WBUF = 1)	HSCLK output mode	Immediately after the rising edge of the last HSCLK (just after data transfer to receive buffer 2) or just after receive buffer 2 is read
	HSCLK input mode	Immediately after the rising edge or falling edge of the last HSCLK depending on the rising or falling edge triggering mode, respectively (right after data is moved to receive buffer 2)
Overrun error generation timing	HSCLK input mode	Immediately after the rising or falling edge of the last HSCLK (for rising or falling edge mode, respectively)

Transmit Side

Interrupt generation timing (WBUF = 0)	HSCLK output mode	Immediately after the rising edge of the last HSCLK
	HSCLK input mode	Immediately after the rising or falling edge of the last HSCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (WBUF = 1)	HSCLK output mode	Immediately after the rising edge of the last HSCLK or just after data is moved to send buffer 1
	HSCLK input mode	Immediately after the rising or falling edge of the last HSCLK (for the rising or falling edge mode, respectively) or just after data is moved to send buffer 1

Under-run error generation timing	HSCLK input mode	Immediately after the falling or rising edge of the next HSCLK (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 HSC0MOD1 <TXE> = "0") when data is being transmitted.

Detailed Description of Registers

As channels 0 to 3 have same register set, only channel 0 is described here.

15.3.1.1 Enable Register

HSC0EN

	7	6	5	4	3	2	1	0
bit Symbol								SIOE
Read/Write	R							R/W
After reset	0							0
Function	Always reads "0."							HSIO operation 0: Disable 1: Enable

<SIOE>: It specifies HSIO operation. When HSIO is to be used, be sure to enable HSIO by setting "1" to this register before setting any other registers of the HSIO module. When HSIO operation is disabled, the clock will not be supplied to the HSIO module except for the register part and thus power consumption can be reduced. If HSIO is enabled once and then disabled, any register setting is maintained.

15.3.1.2 TX/ RX Buffer Register

HSC0BUF

	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	TB7~0 : TX buffer/ FIFO RB7~0 : RX buffer/ FIFO							

The buffer register (HSC0BUF) functions as TX buffer at writing and RX buffer at reading.

<TB7:0> TX buffer (only for writing)

<RB7:0> RX buffer (only for reading)

15.3.1.3 Control Register

HSC0CR

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	HSCLKS	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 **(for UART)**	Parity **(for UART)** 0: Odd 1: Even	Add parity **(for UART)** 0: Disable 1: Enable	0: Normal operation 1: Error			0: HSCLK0 ↑ 1: HSCLK0 ↓	**(for I/O interface)** 0: Baud rate generator 1: HSCLK0 pin input
				Overrun	Parity/ under-run	Framing		

<RB8>: Indicates 9th received bit in 9 bit UART mode.

<EVEN>: Specifies parity condition.
"0": Odd parity
"1": Even parity
Parity is available for 7 bit/ 8 bit UART mode.

<PE>: Enables or disables parity.
Parity is available for 7 bit/ 8 bit UART mode.

<OERR>: Indicates error flags (overrun error flag, parity error/ underrun error flag and framing error flag).
<PERR>: (Note)
<FERR>:

<HSCLKS>: Clock edge selection for data transmission/ reception
"0": Data send/receive at rising edges of HSCLK0
"1": Data send/receive at falling edges of HSCLK0

<IOC>: I/O interface input clock selection.
"0": Baud rate generator
"1": HSCLK0 pin input

(Note) Every error flag is cleared when read.

15.3.1.4 Mode Control Register 0

		7	6	5	4	3	2	1	0
HSC0MOD0	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 TB3OUT 01: Baud rate generator 10: Internal $f_{SYS}/2$ clock 11: External clock (HSCLK0 input)	

<TB8>: Sets 9th transmit bit in 9 bit UART mode.

<CTSE>: Controls handshake function.
Setting this bit to “1” enables handshake function using H$\overline{\text{CTS}}$ pin.

<RXE>: Executes receive control. **(Note)**
Enable this bit after setting each mode register (HSC0MOD0, HSC0MOD1 and HC0MOD2).

<WU>: Controls wake-up function.
This function is available only for 9 bit UART mode. Setting this bit is ignored in other modes.

	9 bit UART mode	Others
0	An interrupt is generated whenever data is received	don’t care
1	An interrupt is generated when 9th received bit is “1”.	

<SM1:0>: Specifies transfer mode.

<SC1:0>: Specifies transfer clock in UART mode.
In the I/O interface mode, the serial control register (HC0CR) is used for clock.

**(Note) With <RXE> set to "0," set each mode register (HSC0MOD0, HSC0MOD1 and HC0MOD2).
Then set <RXE> to "1."**

15.3.1.5 Mode Control Register 1

HSC0MOD1

	7	6	5	4	3	2	1	0
bit Symbol	I2S0	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	–
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	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 (for I/O interface) 000: None 100: 8SCLK 001: 1SCLK 101:16SCLK 010: 2SCLK 110: 32SCLK 011: 4SCLK 111: 64SCLK			Write "0."

<I2S0>: Specifies the IDLE mode operation.

<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.

<TXE>: This bit enables transmission and is valid for all the transfer modes. (Note) If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<SINT2:0>: This parameter is valid for I/O interface mode when a clock is not input from HSCLK0 pin (invalid for other modes).
Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode.

(Note) Enable <TXE> bit after setting other bits.

15.3.1.6 Mode Control Register 2

HSC0MOD2

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R			R/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 (for UART) 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	

<TBEMP>: If double buffering is disabled, this flag is insignificant.
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."

<RBFLL>: If double buffering is disabled, this flag is insignificant.
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."

<TXRUN>: This is a status flag to show that data transmission is in progress.
<TXRUN> and <TBEMP> show the following status.

<TXRUN>	<TBEMP>	Status
1	-	Transmission in progress
0	1	Transmission completed
	0	Wait state for transmission with next data in a send buffer.

<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.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, you need to fix it to LSB first.

<WBUF>: This parameter enables or disables the send/receive buffers to send (in both HSCLK output/input modes) and receive (in HSCLK output mode) data in the I/O interface mode and to transmit data in the UART. In all other modes, double buffering is enabled when receiving data in I/O interface mode (HSCLK input) and UART mode regardless of the <WBUF> setting.

<SWRST1:0>: Overwriting "01" in place of "10" generates a software reset.
When this software reset is executed, the following bits and their internal circuits are initialized.
(Note 1, 2 and 3)

Register name	Bit
HSC0MOD0	RXE
HSC0MOD1	TXE
HSECMOD2	TBEMP,RBFLL,TXRUN,
HC0CR	OERR,PERR,FERR

(Note 1) **While data transmission is in progress, any software reset operation must be executed twice in succession.**

(Note 2) **To complete software reset, it takes 2 clocks after executing an instruction.**
Executing SYNC and NOP instructions after software reset is recommended.

(Note 3) **When software reset is executed, the bits listed in the <SWRST1:0> description are initialized. It requires resetting of the mode registers and control register.**

15.3.1.7 Baud Rate Generator Control Register (HBR0CR)
Baud Rate Generator Control Register 2 (HBR0ADD)

HBR0CR

	7	6	5	4	3	2	1	0
bit Symbol	—	HBR0ADDE	HBR0S5	HBR0S4	HBR0S3	HBR0S2	HBR0S1	HBR0S0
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	Divide ratio "N" 000000: 64 000001: 1 000010: 2 : 111111: 63					

BR0ADD

	7	6	5	4	3	2	1	0
bit Symbol					HBR0K3	HBR0K2	HBR0K1	HBR0K0
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	Always reads "0."				Specify K for the "N + (16 - K)/16" division 0000: Setting prohibited 0001: K=1 0010: K=2 : 1111: K=15			

<HRB0ADDE>: Enables or disables N+(16-K)/16 division function.
This function is available only for UART mode.

<HRB0S5:0>: Sets divide ratio “N” of the baud rate generator.

<HRB0K3:0>: Specifies K for the "N + (16 - K)/16" division.

Divide ratio of baud rate generator is set in the above two registers.
Table 15.3 shows the divide ratio configuration.

Table 15.3 Divide ratio setting

	HBR0ADDE=0	HBR0ADDE=1 **(Note 1)** (available for UART mode)
HBR0S	Set divide ratio “N” **(Note 2, 3)**	
HBR0K	No setting required	Set “K” **(Note 4)**
Divide ratio	N	$N + \frac{(16-K)}{16}$

(Note 1) To use the "N + (16 - K)/16" division function, be sure to set HBR0ADDE <HBR0ADDE> to "1" after setting the K value (K = 1 to 15) to HBR0ADD <HBR0K3:0>.

(Note 2) The division ratio "1" (“000001”) of the baud rate generator can be specified only when

- the "N + (16 - K)/16" division function is not used In the UART mode.
- double buffering is used in the I/O interface mode.

(Note 3) To use the "N + (16 - K)/16" division function, neither the division ratio "1" (“0001”) nor “16” (“0000”) can be set.

(Note 4) “0” cannot be set as K value.

15.3.1.8 FIFO Configuration Register

		7	6	5	4	3	2	1	0
HSC0FCNF	bit Symbol	Reserved	Reserved	Reserved	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

<RFST>: Specifies bytes used in RX FIFO. **(Note)**

0: The maximum number of bytes of the FIFO configured is available. (See description of <CNFG> bit.)

1: Same as the fill level for receive interrupt generation specified by HSC0RFC <RIL1:0>.

<TFIE>: When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFIE>: When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<RXTXCNT>: The function to automatically disable RXE/TXE bits is disabled. If “1” is set, it functions as shown below according to the communication method configured (the communication method can be set in the mode control register 1 HSC0MOD1<FDPX1:0>).

Half duplex RX	When the RX FIFO is filled up to the specified number of valid bytes, HSC0MOD0<RXE> is automatically set to "0" to inhibit further reception.
Half duplex TX	When the TX FIFO is empty, HSC0MOD1<TXE> is automatically set to "0" to inhibit further transmission.
Full duplex	When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<CNFG>: This parameter is to enable FIFO. Setting “1” enables FIFO. If enabled, the HSCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows: (the communication method can be set in the mode control register 1 HSC0MOD1<FDPX1:0>).

Half duplex RX	RX FIFO 32byte
Half duplex TX	TX FIFO 32byte
Full duplex	RX FIFO 16byte＋TX FIFO 16byte

(Note) 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.

15.3.1.9 Receive FIFO Configuration Register

HSC0RFC

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS		RIL4	RIL3	RIL2	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	Always reads "0."	FIFO fill level to generate RX interrupts (0_0000::32byte) 0_0001:1byte 0_0010:2byte To 0_1110:30byte 1_1111:31byte				

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.

<RFCS>: Clears RX FIFO.
Writing "1" clears FIFO. If not, "0" is read.

<RFIS>: Selects interrupt generation condition.
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.

<RIL1:0>: Specifies FIFO fill level.

15.3.1.10 Transmit FIFO Configuration Register

HSC0TFC

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS	TIL5	TIL4	TIL3	TIL2	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_0000:Empty) 00_0001:1byte 00_0010:2byte To 01_1111:31byte 10_0000:32byte					

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.

<TFCS>: Clears TX FIFO.
Writing “1” clears FIFO. If not, “0” is read.

<TFIS>: Selects interrupt generation condition.
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.

<TIL1:0>: Specifies FIFO fill level.

15.3.1.11 Receive FIFO Status Register

HSC0RST

	7	6	5	4	3	2	1	0
bit Symbol	ROR		RLVL5	RLVL4	RLVL3	RLVL2	RLVL1	RLVL0
Read/Write	R		R					
After reset	0		0	0	0	0	0	0
Function	RX FIFO Overrun 1: Generated	Always reads "0."	Status of RX FIFO fill level (00_0000:32byte) 00_0000:32byte 00_0001:1byte 00_0010:2byte To 01_1110:30byte 01_1111:31byte					

<ROR>: Flag for RX FIFO overrun.
This parameter is set to “1” if overrun occurs **(note).**

<RLVL2:0>: Indicates status of RX FIFO fill level.

(Note) The <ROR> bit is cleared to "0" when receive data is read from the HSC0BUF register.

15.3.1.12 Transmit FIFO Status Register

HSC0TST

	7	6	5	4	3	2	1	0
bit Symbol	TUR					TLVL2	TLVL1	TLVL0
Read/Write	R		R					
After reset	1		0	0	0	0	0	0
Function	TX FIFO Underrun 1: Generated. Cleared to “0” when FIFO is written by received data.	Always reads "0."	Status of TX FIFO fill level (00_0000:Empty) 00_0001:1byte 00_0010:2byte To 01_1111:31byte 10_0000:32byte					

<TUR>: Flag for TX FIFO underrun.
This parameter is set to “1” if underrun occurs **(note).**

<TLVL2:0>: Indicates status of TX FIFO fill level.

(Note) The <TUR> bit is cleared to "0" when receive data is read from the HSC0BUF register.

15.4 Operation of Each Circuit (Channel 0)

As channels 0 to 3 operate identically, only channel 0 is described here.

15.4.1 Serial Clock Generation Block

This block generates basic transmit and receive clocks.
In this block, clock is determined according to baud rate generator and specified mode and register setting.

15.4.1.1 Clock Selection Circuit

Serial clock is determined according to mode and register setting specified.

Specify mode in the mode control register 0 (HSC0MOD0<SM1:0>).

To use I/O interface mode, clock is specified in the control register HC0CR.

To use UART mode, clock is specified in the mode control register0 (HSC0MOD0<SC1:0>).

Table 15.4 and Table 15.5 show details of clock selection in I/O interface mode and UART mode respectively.

Table 15.4 Clock selection in I/O interface mode

Mode HSC0MOD0<SM1:0>	Input/ output selection HC0CR<IOC>	Clock edge selection HC0CR<HSCLKS>	Clock selection
I/O interface mode	HSCLK output	- (Fixed to rising edge)	Baud rate generator output divided by 2
	HSCLK input	Rising edge	Rising edge of HSCLK input
		Falling edge	Falling edge of HSCLK input

Table 15.5 Clock selection in UART mode

Mode HSC0MOD0<SM1:0>	Clock selection HSC0MOD0<SC1:0>
UART mode	Timer output
	fsys
	fsys/2
	HSCLK input

15.4.1.2 Baud Rate Generator

The baud rate generator generates transmit and receive clocks.

(1) Input clock

The input clock selection is made by setting the baud rate generator control register, HBR0CR <BR0CK5:0>.

Specify the divide ratio of the output clock with the baud rate generator control register HBR0CR<BR0ADDE><BR0S3:0> and the baud rate generator control register 2 HBR0ADD<BR0K3:0>.

(2) UART mode

Table 15.6 UART mode

HBR0CR<BR0ADDE>	HBR0ADD<BR0K3:0>	HBR0CR<BR0S3:0>
“0”	Setting invalid	Setting valid Set divide ratio “N” （N=1,2,3…64）
“1” N+ $\frac{(16-K)}{16}$ division	Setting valid Set divide ratio “K” （K=1,2,3…15）	Setting valid Set divide ratio “N” （N=2,3…63）
	―	N=1,64: setting prohibited

16. Serial Bus Interface (SBI)

The TMP19A44 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 PC4 (SDA) and PC5 (SCL). In the clock-synchronous 8-bit SIO mode, the SBI is connected to external devices via PC6 (SCK), PC4 (SO) and PC5 (SI).

The following table shows the programming required to put the SBI in each operating mode.

	Port open drain output	Port control register	Port function register
I2C bus mode	PCODE <6:4> = 11	PCCR<6:4> = x11	PCFC1<6:4> = 011
Clock-synchronous 8-bit SIO mode	PCODE <6:4> = x0x	PCCR<6:4> = 101 (clock output) PCCR<6:4> = 001(clock input)	PCFC1<6:4> = 111

16.1 Configuration

The configuration is shown in Fig. 16.1.

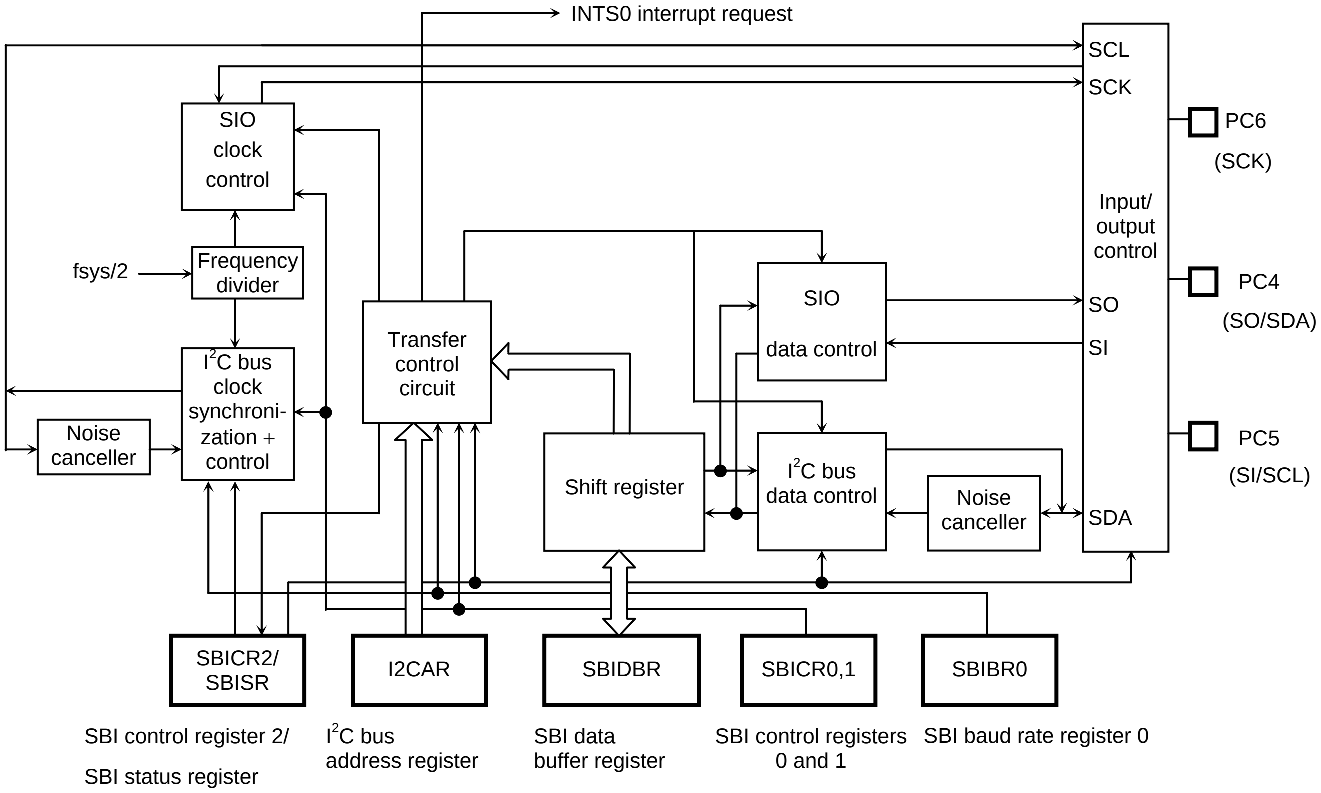


Fig. 16.1 SBI Block Diagram

16.2 Control

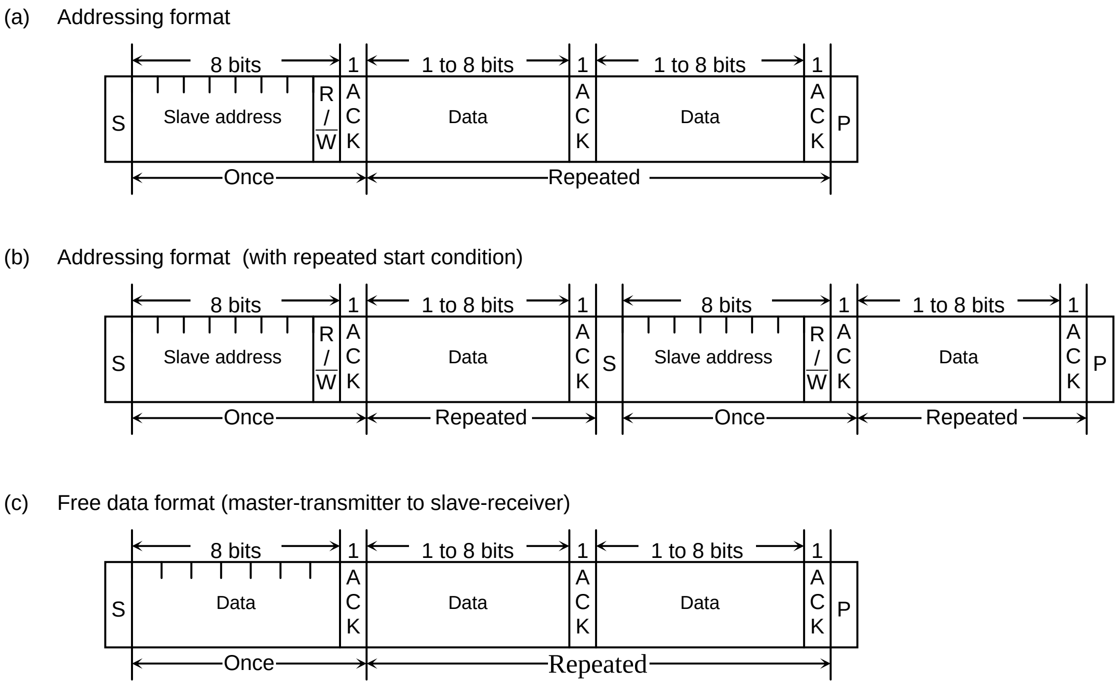
The following registers control the serial bus interface and provide its status information for monitoring.

- 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^{2}C 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 "16.4 Control in the I^{2}C Bus Mode" and "16.7 Control in the Clock-synchronous 8-bit SIO Mode."

16.3 I^{2}C Bus Mode Data Formats

Fig. 16.2 shows the data formats used in the I^{2}C bus mode.



Note: S: Start condition
R/$\overline{W}$: Direction bit
ACK: Acknowledge bit
P: Stop condition

Fig. 16.2 I^{2}C Bus Mode Data Formats

16.4 Control Registers in the I^{2}C Bus Mode

The following registers control the serial bus interface (SBI) in the I^{2}C bus mode and provide its status information for monitoring.

Serial bus interface control register 0

SBICR0
(0xFF00_4B00)

	7	6	5	4	3	2	1	0
Bit symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	This can be read as "0."						
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

<SBIEN>: To use the SBI, enable the SBI operation ("1") before setting each register in the SBI module.

Fig. 16.3 I^{2}C Bus Mode Register

Serial bus interface control register 1

SBICR1
(0xFF00_4B04)

	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)			Acknowledgment clock 0: Not generate 1: Generate	This can be read as "1."	Select internal SCL output clock frequency (Note 2) and reset monitor		
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

<Bit 2:0><SCK2:0> : Selects internal SCL output clock frequency.

On writing <SCK2:0>: Select internal SCL output clock frequency

000	n=5	384 kHz	
001	n=6	294 kHz	System clock: fsys
010	n=7	200 kHz	(=80 MHz)
011	n=8	121 kHz	Clock gear: fc/1
100	n=9	68 kHz	
101	n=10	36 kHz	Frequency = [Hz]
110	n=11	18 kHz	
111		reserved	

<Bit 0>< SWRMON:0> : Software reset status monitor

On reading <SWRMON>: Software reset status monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

<Bit 7:5><BC2:0>Select the number of bits per transfer

<BC2:0>	<ACK> = 0		<ACK> = 1	
	# of clock cycles	Data length	# 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 "3.12.5 (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. 16.4 I^2C Bus Mode Register

Serial bus interface control register 2

SBICR2
(0xFF00_4B10)

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 2)		W (Note 1)	
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 INTS0 interrupt request 0: – 1: Clear interrupt request	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I^{2}C bus mode 11: (Reserved)		Software reset generation Write "10" followed by "01" to generate a reset.	
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							

<Bit 1:0><SCK2:0>: Write "10" followed by "01" to generate a reset.

<Bit 3:2><SBIM1:0>: Select serial bus interface operating mode

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^{2}C bus mode
11	(Reserved)

<Bit 4><PIN>	: Clear INTS0 interrupt request
<Bit 5><BB>	: Start/stop condition generation
<Bit 6><TRX>	: Select transmit/ receive
<Bit 7><MST>	: Select master/slave

(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^{2}C bus or clock-synchronous 8-bit SIO mode.

(Note 3) Ensure that serial transfer is completed before switching the mode.

Fig. 16.5 I^{2}C Bus Mode Register

Table 16.1 Base Clock Resolution

@fsys = 80 MHz

Clock gear value <GEAR2:0>	Base clock resolution
000 (fc)	fsys/2^2 (0.05 μs)
100 (fc/2)	fsys/2^3 (0.1 μs)
101 (fc/4)	fsys/2^4 (0.2 μs)
110 (fc/8)	fsys/2^5 (0.4 μs)
111 (fc/16)	fsys/2^5 (0.8 μs)

Serial bus interface status register

SBISR
(0xFF00_4B10

	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	INTS0 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 (When the General call is detected, it is set.)	Last received bit monitor 0: "0" 1: "1"
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

(Note) Writing to this register causes it to function as SBICR2.

Fig. 16.6 I^2C Bus Mode Register

Serial bus interface baud rate register 0

SBIBR
(0xFF00_4B14)

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI0						
Read/Write	R	R/W	R					R/W
After reset	1	0	1					0
Function	This can be read as “0”.	IDLE 0: Stop 1: Operate	This can be read as “0”.					Make sure that you write "0." (Note)
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							

(Note) This is read as "1" in the SIO mode.

Serial bus interface data buffer register

SBIDBR
(0xFF00_4B08)

	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							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

(Note) Transmit data must be written to this register, with bit 7 being the most-significant bit (MSB).

I^{2}C bus address register

I2CAR
(0xFF00_4B0C)

	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
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

<Bit 0><ALS> : Specify address recognition mode

0	Recognizes the slave address.
1	Does not recognize slave address.

(Note) Please set the bit of I2C bus address register I2CAR to "0" about 0< ALS >, except when you use the free data format.
It operates as a free data format when setting it to “1", it fixes to the transmission at the master, and the direction of forwarding is fixed to the reception at the slave.

Fig. 16.7 I^{2}C Bus Mode Register

16.5 Control Registers in the I^{2}C Bus Mode

16.5.1 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.

16.5.2 Setting the Number of Bits per Transfer

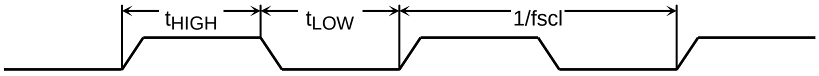
SBICR1 <BC2:0> specifies the number of bits of the next data to be transmitted or received.

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.

16.5.3 Serial Clock

① Clock source

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. 16.3 Clock Source

The highest speeds in the standard and high-speed modes are specified to 100KHz and 400KHz 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^{2}C 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.

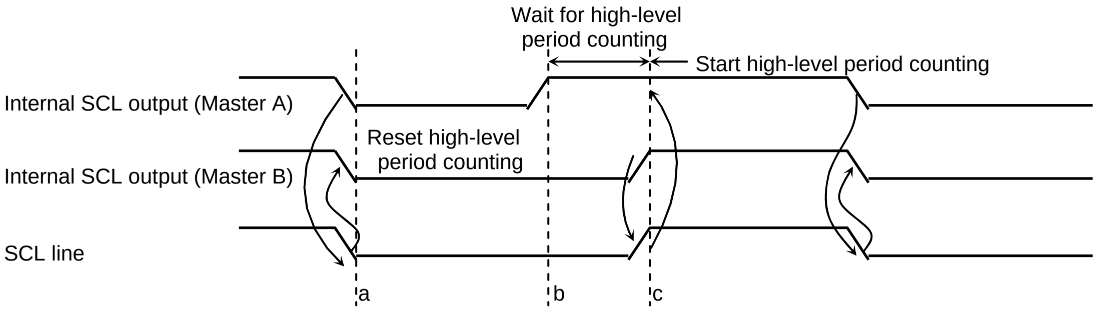

Wait for high-level period counting
Start high-level period counting
Internal SCL output (Master A)
Reset high-level period counting
Internal SCL output (Master B)
SCL line
a
b
c

Fig. 16.4 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.

16.5.4 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.

16.5.5 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.

16.5.6 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.

16.5.7 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.

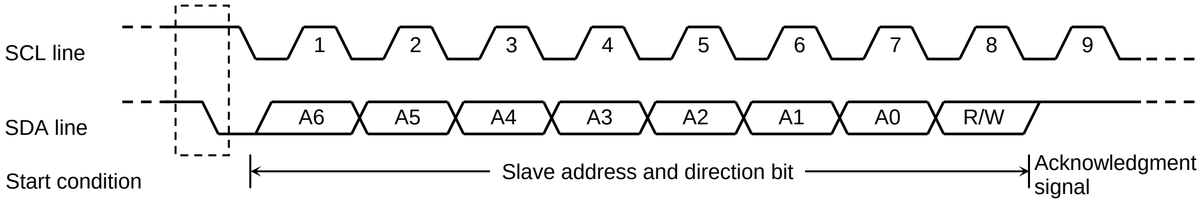


Fig. 16.5 Generating the Start Condition and a Slave Address

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, and PIN> should not be altered until the stop condition appears on the bus.

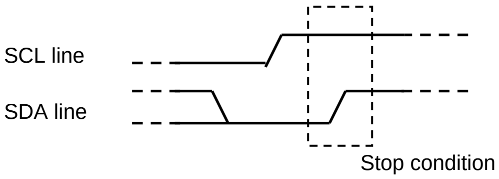


Fig. 16.6 Generating the Stop Condition

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).

16.5.8 Interrupt Service Request and Release

When a serial bus interface interrupt request (INTS0) 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."

16.5.9 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.

16.5.10 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.

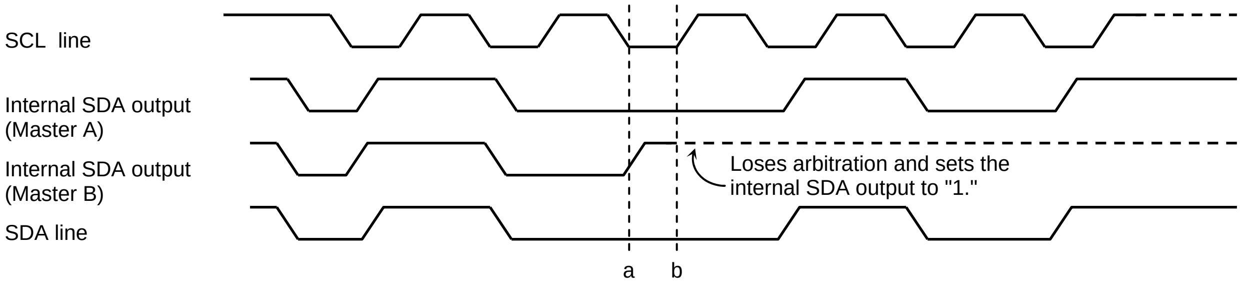

SCL line
Internal SDA output (Master A)
Internal SDA output (Master B)
SDA line
Loses arbitration and sets the internal SDA output to "1."
a
b

Fig. 16.7 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.

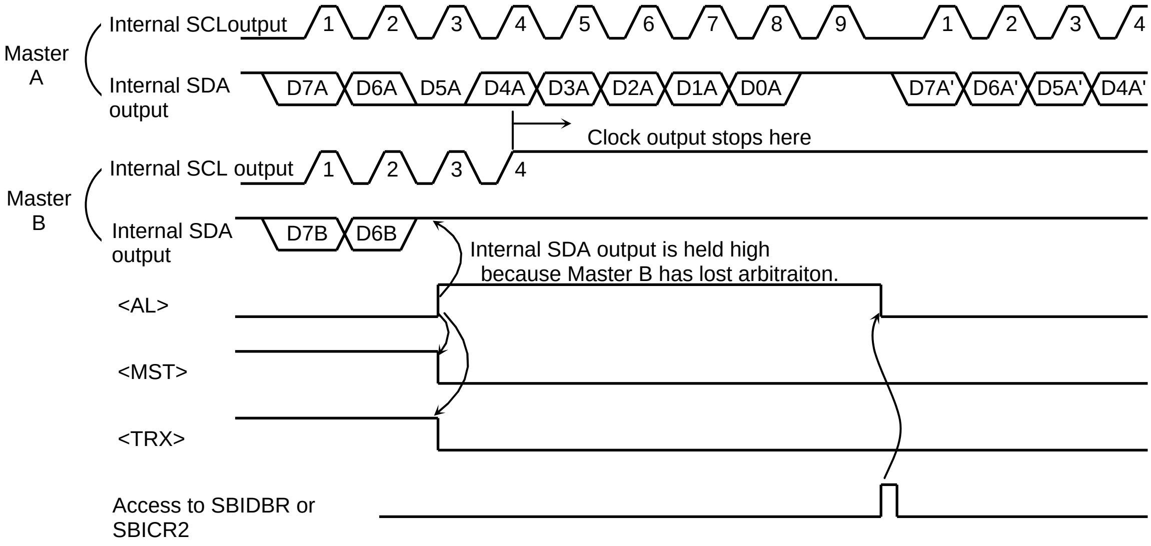


Fig. 16.8 Example of Master B Losing Arbitration (D7A = D7B, D6A = D6B)

16.5.11 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.

16.5.12 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.

16.5.13 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 INTS0 interrupt request causes ACK signal to be read.

16.5.14 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) A software reset causes the SBI operating mode to switch from the I^{2}C mode to the port mode.

16.5.15 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.

16.5.16 I^{2}C 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.

16.5.17 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.

16.6 Data Transfer Procedure in the I²C Bus Mode

16.6.1 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
```

16.6.2 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 INTS0 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 INTS0 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 INTS0 interrupt routine

```
INTCLR ← 0xbc                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 INTS0 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.**

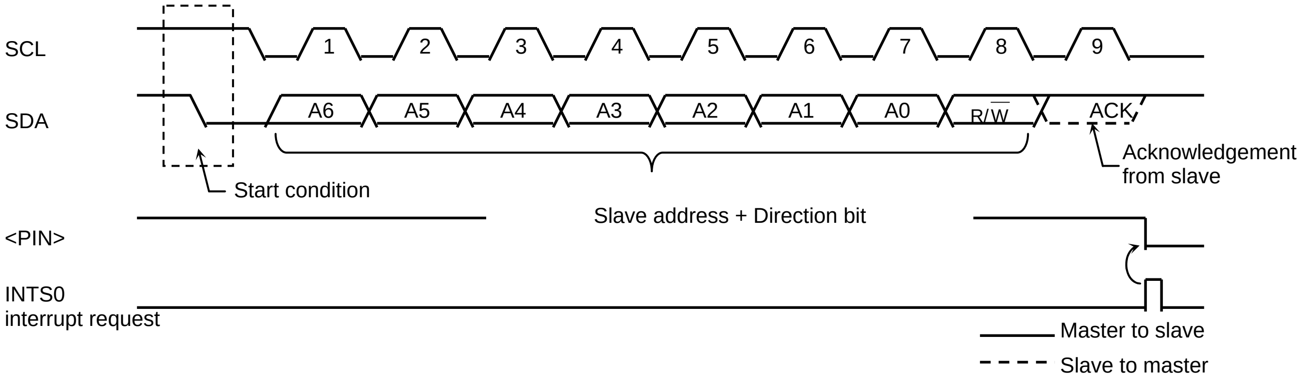


Fig. 16.9 Generation of the Start Condition and a Slave Address

16.6.3 Transferring a Data Word

At the end of a data word transfer, the INTS0 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 INTS0 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.

```
INTS0 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
```

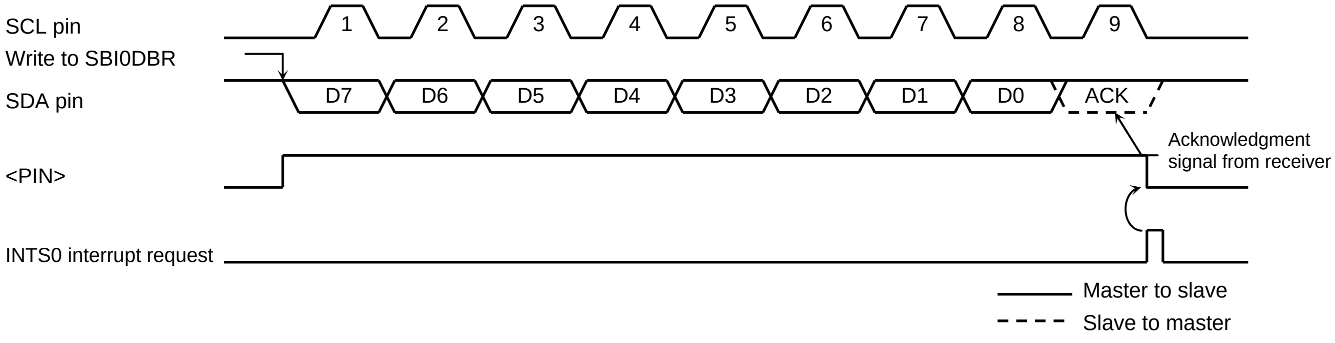


Fig. 16.10 <BC2:0> = "000" and <ACK> = "1" (Transmitter Mode)

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 INTS0 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.

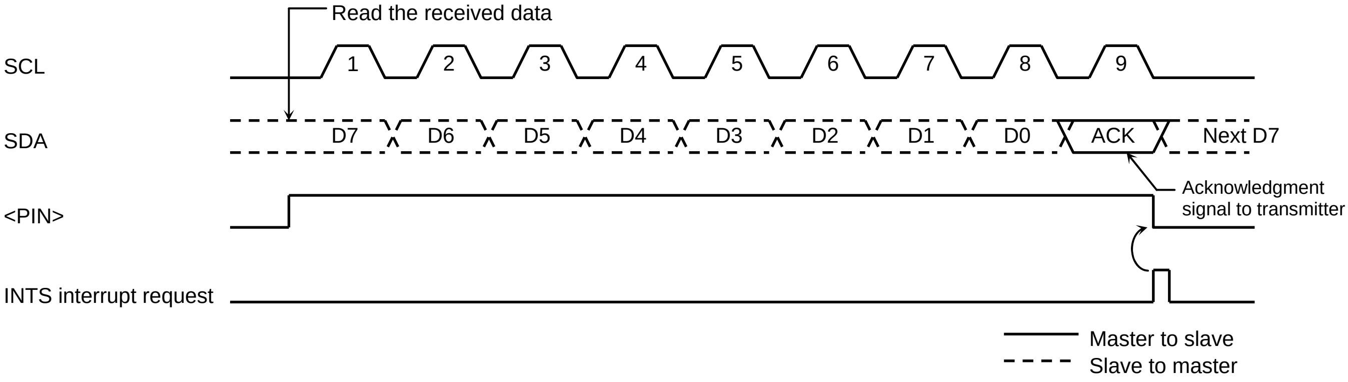


Fig. 16.11<BC2:0> = "000" and <ACK> = "1" (Receiver Mode)

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.

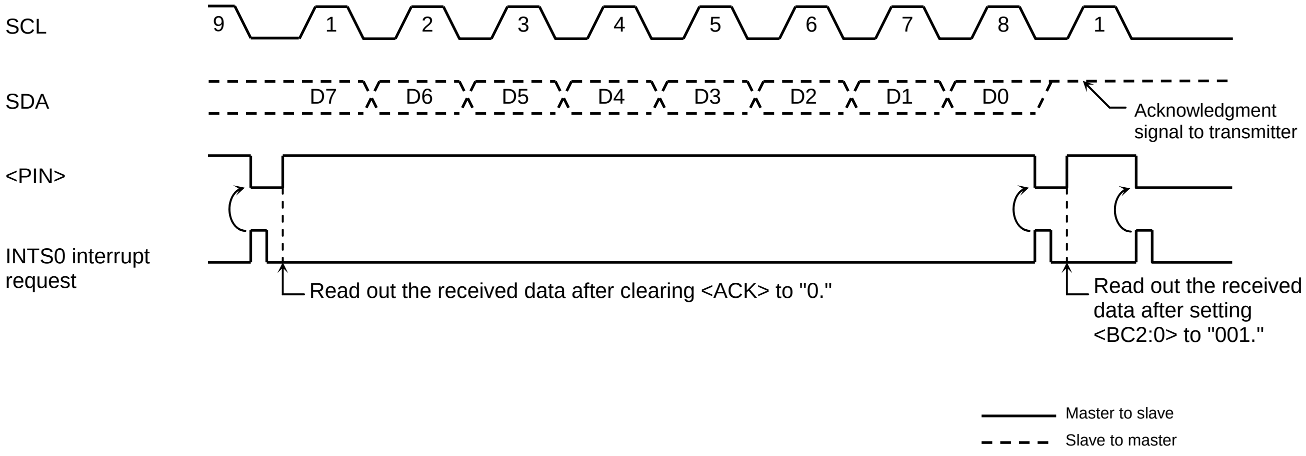

SCL
9
1
2
3
4
5
6
7
8
1
SDA
D7
D6
D5
D4
D3
D2
D1
D0
Acknowledgment signal to transmitter
<PIN>
INTS0 interrupt request
Read out the received data after clearing <ACK> to "0."
Read out the received data after setting <BC2:0> to "001."
Master to slave
Slave to master

Fig. 16.12 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data words

INTS0 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
```

INTS0 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
```

INTS0 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
```

INTS0 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
```

INTS0 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 INTS0 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 INTS0 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 16.2 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

INTS0 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 16.2 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 SBIDBR.
	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.

16.6.4 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.
```

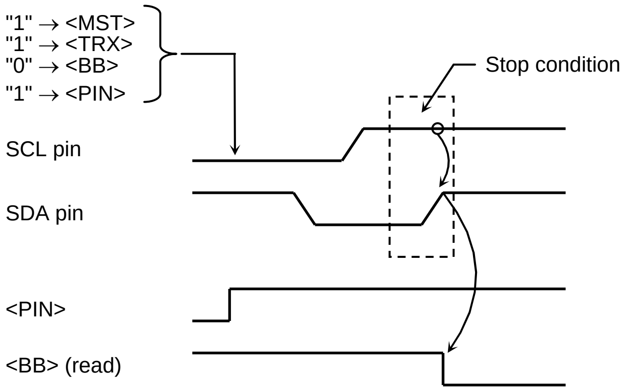


Fig. 16.13 Generating the Stop Condition

16.6.5 Repeated Start Procedure

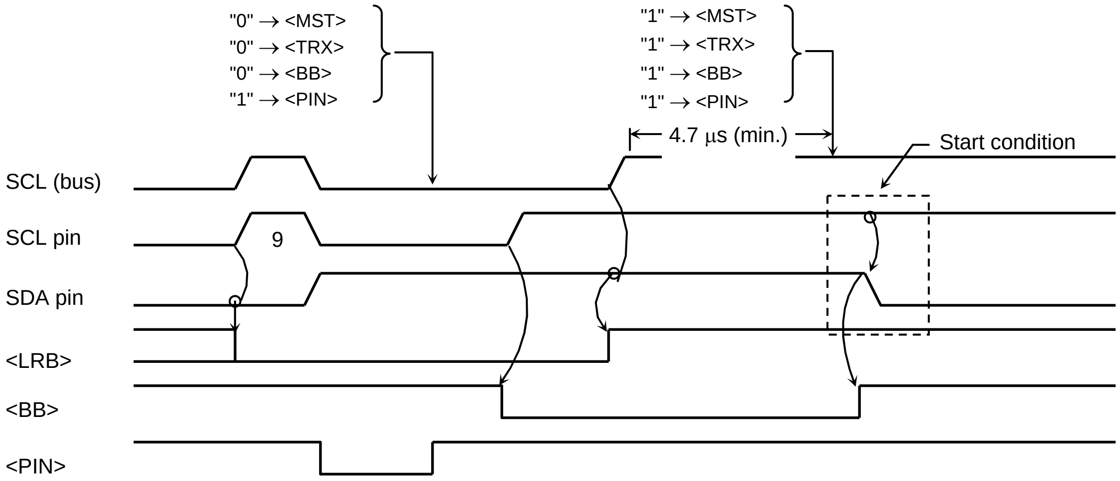
Repeated start is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a repeated start in the master mode is described below.

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 (16.6.2 Generating the Start Condition and a Slave Address) 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. 16.14 Timing Chart of Generating a Repeated Start

16.7 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
(0xFF00_4B00)

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	This can be read as “0”.						
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

<SBIEN>: To use the SBI, enable the SBI operation ("1") before setting each register of SBI module.

Serial bus interface control register 1

SBICR1
(0xFF00_4B04)

	7	6	5	4	3	2	1	0
bit Symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	W				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		This can be read as "1."	Select serial clock frequency		
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

On writing <SCK2:0>: Select serial clock frequency

000	n = 3	2.5 MHz	System clock : fsys (=80 MHz) Clock gear : fc/1 Frequency $= \frac{fsys/4}{2^n}$ [Hz]
001	n = 4	1.25 kHz	
010	n = 5	625 kHz	
011	n = 6	313 kHz	
100	n = 7	156 kHz	
101	n = 8	78 kHz	
110	n = 9	40 kHz	
111	—	External clock	

(Note) Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.

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	Undefined							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

Fig. 16.8 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				0	0	1	
Function	This can be read as “1”.				Select serial bus interface operating mode 00: Port mode 01: Clock-synchronous 8-bit SIO mode 10: I^{2}C bus mode 11: (Reserved)		This can be read as “1”.	
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

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				0	0	1	
Function	This can be read as “1”.				Serial transfer status monitor 0: Terminated 1: In progress	Shift operation status monitor 0: Terminated 1: In progress	This can be read as “1”.	
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0”.							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

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					W
After reset	1	0	1					0
Function	This can be read as "1."	IDLE 0: Stop 1: Operate	This can be read as "1."					Make sure that you write "0."
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as "0."							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

16.7.1 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.

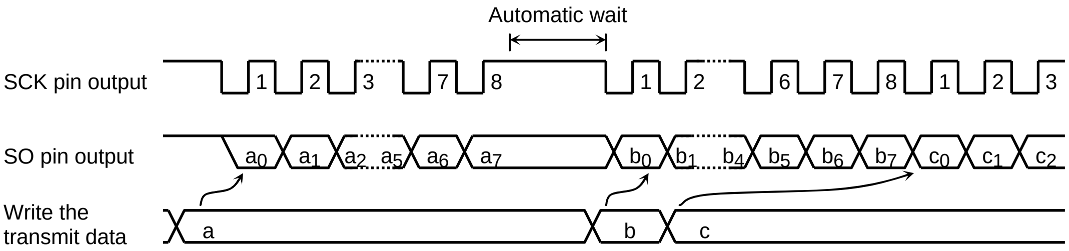

Automatic wait
SCK pin output
1
2
3
7
8
1
2
6
7
8
1
2
3
SO pin output
a_0 a_1 a_2 a_5 a_6 a_7 b_0 b_1 b_4 b_5 b_6 b_7 c_0 c_1 c_2
Write the transmit data
a
b
c

Fig. 16.15 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.

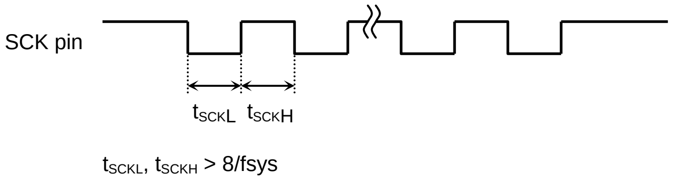


Fig. 16.16 9 Maximum Transfer Frequency of External Clock Input

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).

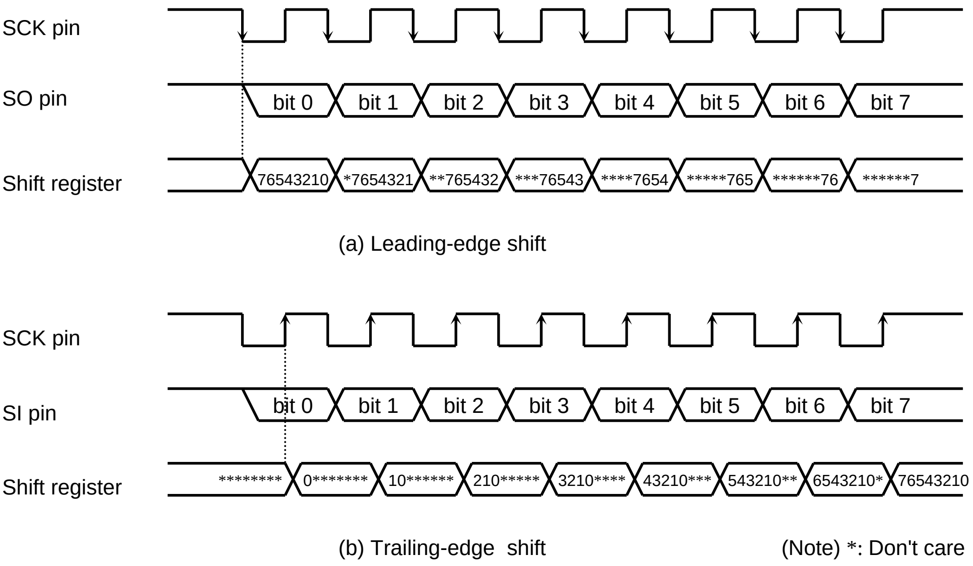


Fig. 16.17 Shift Edge

16.7.2 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 INTS0 (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 INTS0 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."

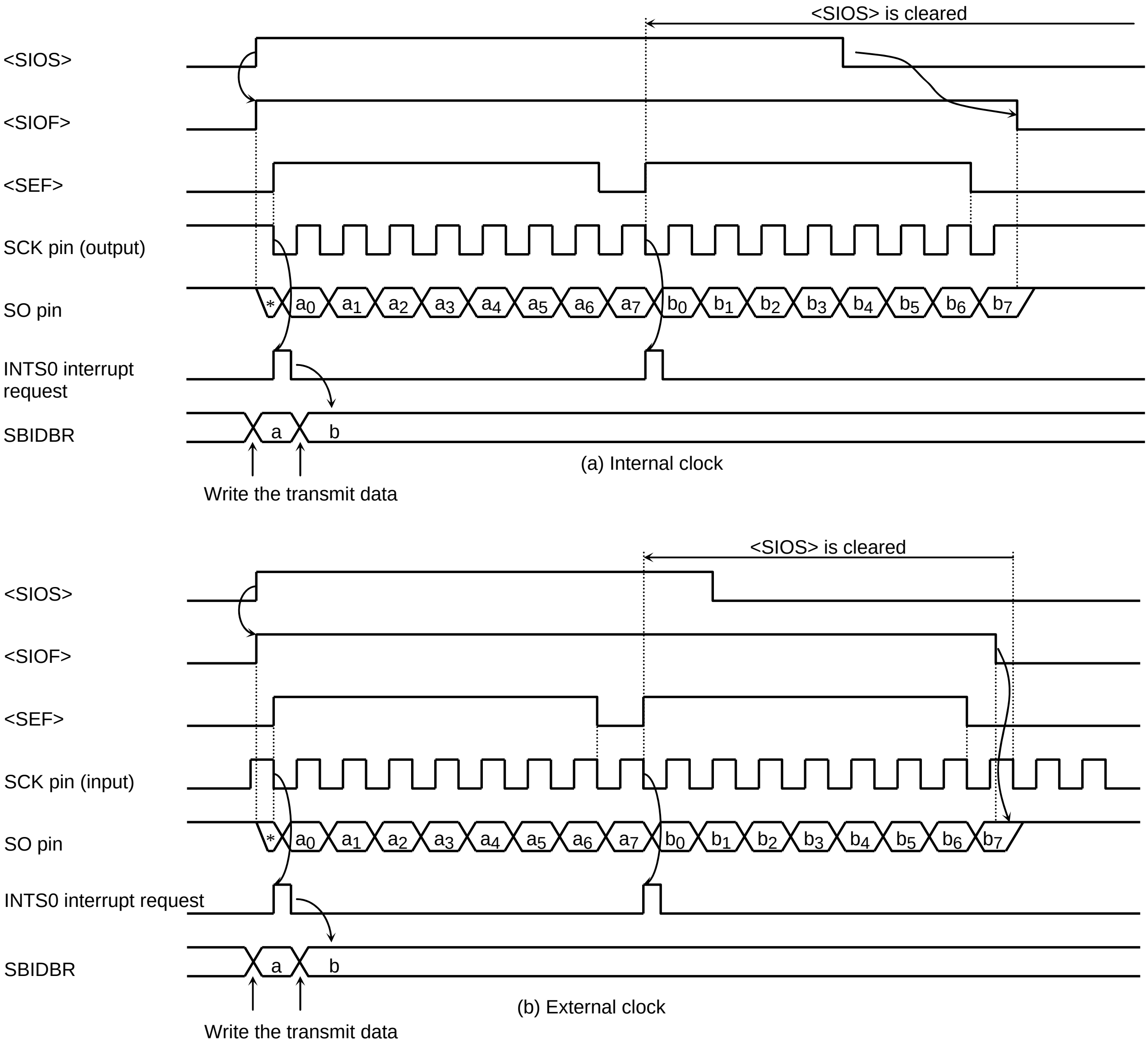
In the external clock mode, <SIOS> must be set to "0" before the next transmit data shift operation is started. Otherwise, operation will stop after dummy data is transmitted.

```
            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.
```

INTS0 interrupt

```
SBIDBR  ←   X X X X X X X X      Writes the transmit data.
```



(a) Internal clock

(b) External clock

Fig. 16.18 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,    0x40
STEST2     :    LB        r2, (PC)                     ;  If SCK = 0 then loop
                AND       r2,   r3
                BEQZ      r2,   STEST2
                ADDIU     r3,   r0,    0y00000111
                SB        r3, (SBICR1)                 ;  <SIOS> ← 0
```

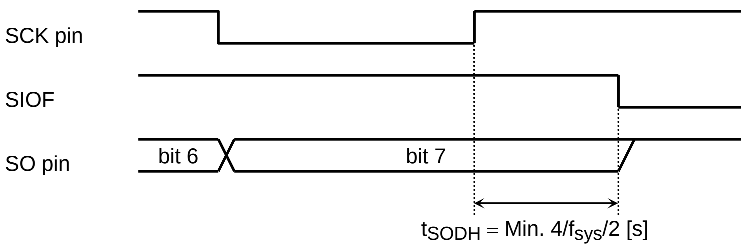


Fig. 16.19 Transmit Data Retention Time at the End of Transmission

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 INTS0 (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 INTS0 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.
```

INTS0 interrupt

```
Reg.      ← SBIDBR                   Reads the received data.
```

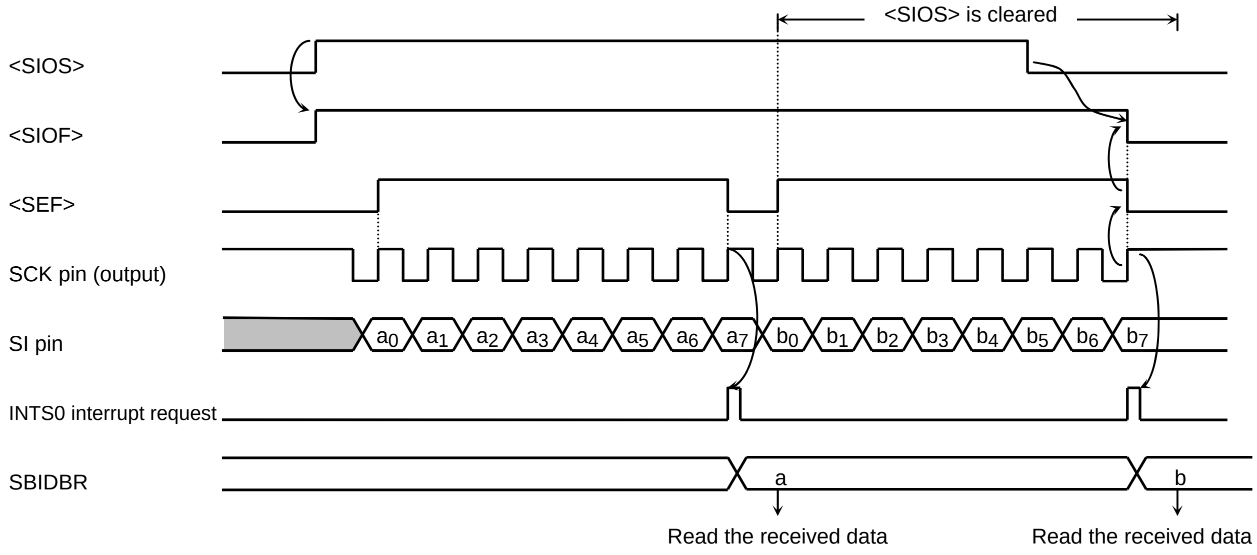


Fig. 16.20 Receive Mode (Example: Internal Clock)

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 INTS0 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 INTS0 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.

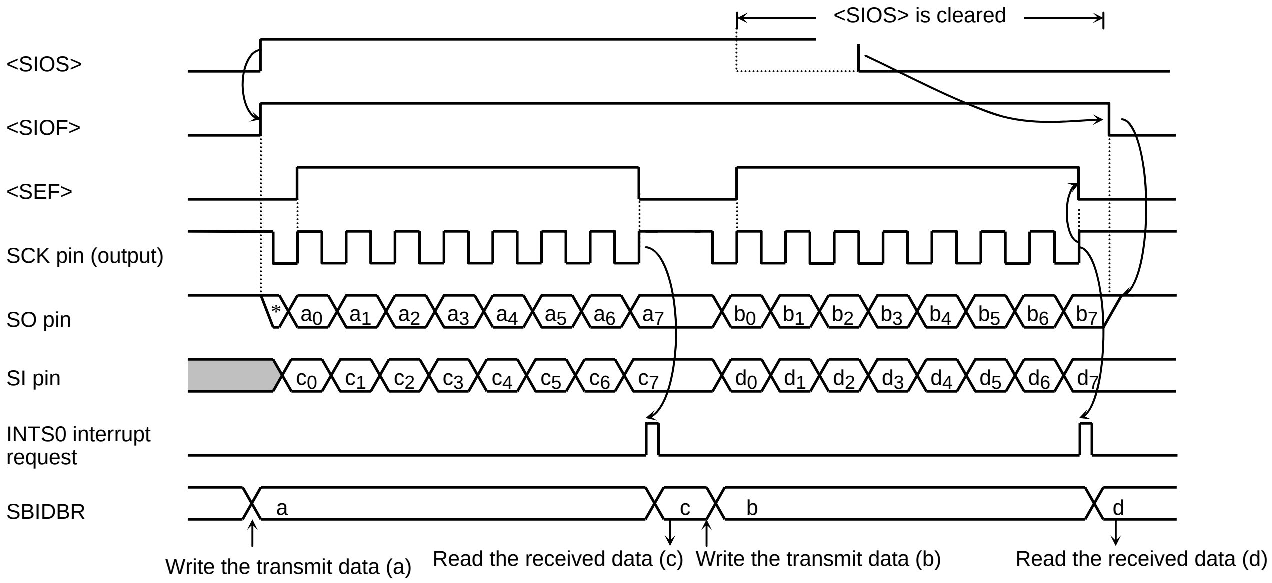

<SIOS> is cleared
<SIOS>
<SIOF>
<SEF>
SCK pin (output)
SO pin
SI pin
INTS0 interrupt request
SBIDBR
* a0 a1 a2 a3 a4 a5 a6 a7 b0 b1 b2 b3 b4 b5 b6 b7
c0 c1 c2 c3 c4 c5 c6 c7 d0 d1 d2 d3 d4 d5 d6 d7
a c b d
Write the transmit data (a)
Read the received data (c)
Write the transmit data (b)
Read the received data (d)

Fig. 16.21 Transmit/Receive Mode (Example: Internal Clock)

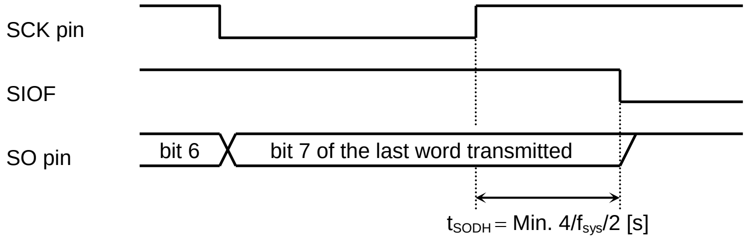

SCK pin
SIOF
SO pin
bit 6
bit 7 of the last word transmitted
t_{SODH} = Min. 4/f_{sys}/2 [s]

Fig. 16.22 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.

INTS0 interrupt

Reg.	←	SBIODBR	Reads the received data.
SBIDBR	←	X X X X X X X X	Writes the transmit data.

17. Analog/Digital Converter

A 10-bit, sequential-conversion analog/digital converter (A/D converter) is built into the TMP19A44. This A/D converter is equipped with 16 analog input channels, which are divided into three units (4 channels, 4 channels and 8 channels).

Fig. 17.1 shows the block diagram of this A/D converter.

These 16 analog input channels (pins ANA0 through AINA3, ANB0 through AINB3 and ANC0 through AINC8) are also used as input ports.

(Note) If it is necessary to reduce a power current by operating the TMP19A44 in IDLE, SLEEP, SLOW, STOP or BACKUP 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 TMP19A44 into standby mode:

1) The TMP19A44 must be put into IDLE mode when ADMOD1<I2AD> is "0."
2) The TMP19A44 must be put into SLEEP,SLOW,STOP,BACKUP SLEEP, BACKUP STOP mode.

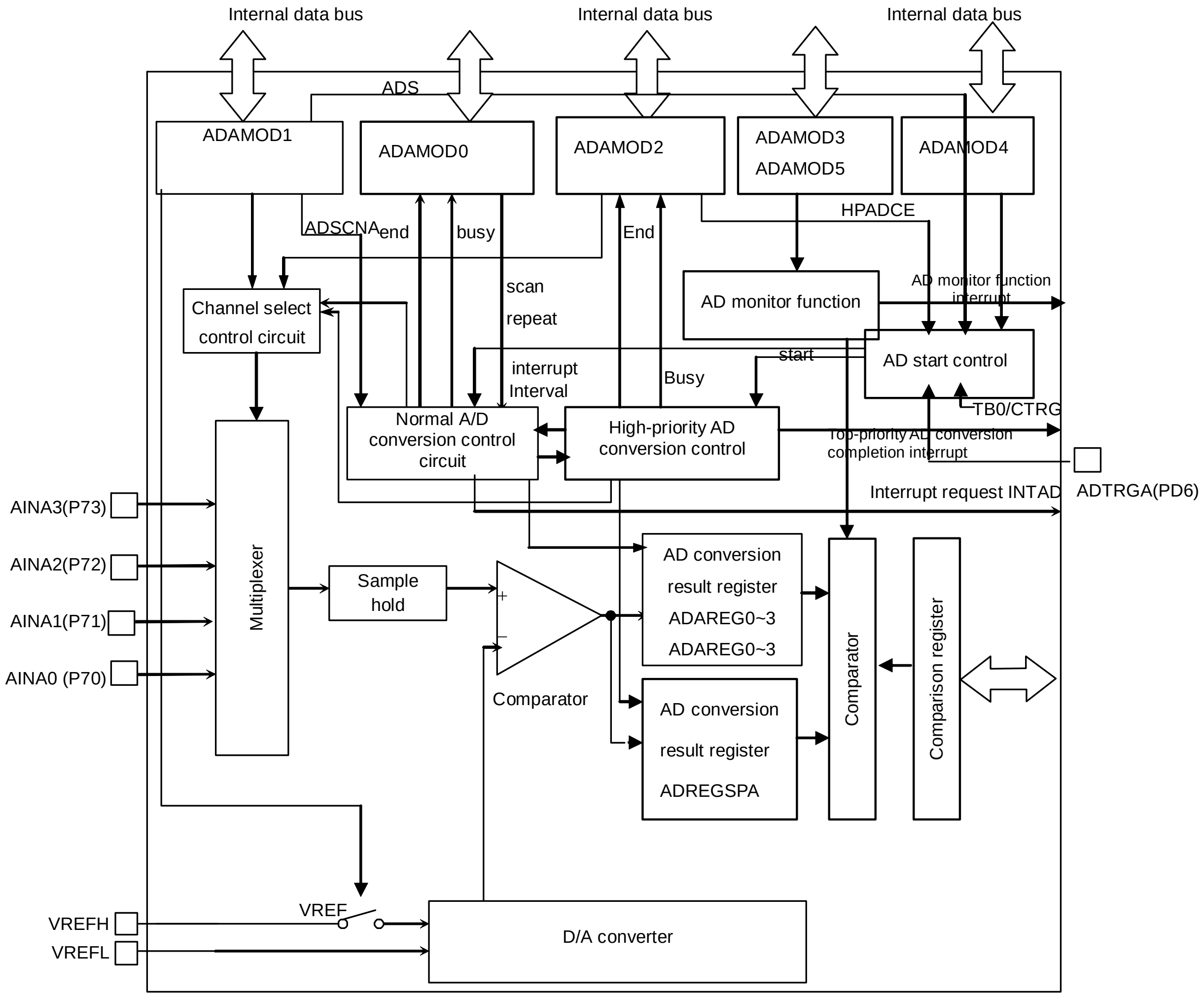


Fig. 17.1 Block Diagram of A/D Converter Unit A (the same applies to Unit B and C)

17.1 Control Register

17.1.1 Unit A, Unit B

The A/D converter is controlled by A/D mode control registers (ADnMOD0, ADnMOD1, ADnMOD2, ADnMOD3 and ADnMOD4: n=A, B). Results of A/D conversion are stored in A/D conversion result registers ADAREG0 through ADAREG3 and ADBREG0 through ADBREG3. Results of top-priority conversion are stored in ADAREGSP and ADBREGSP.

Fig. 17.2 through Fig. 17.4 show the registers related to the A/D converter.

A/D Mode Control Register 0

ADMOD0
(0x FF00_4D04)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFNA	ADBFNA		ITMA1	ITMA0	REPEATA	SCANA	ADSA
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	Setting prohibited
11	Setting prohibited

Fig. 17.2 Registers related to the A/D Converter

A/D Mode Control Register 1

ADMOD1
(0xFF00_ 4D08)

	7	6	5	4	3	2	1	0
bit Symbol	VREFONA	I2ADA	ADSCNA	−	ADCHA3	ADCHA2	ADCHA1	ADCHA0
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: Setting prohibited	Write "0."	Select analog input channel			

Select analog input channel

<SCAN> <ADCHA3.2, 1, 0>	0 Fixed channel	1 Channel scan (ADSCN=0)	1 Channel scan (ADSCN=1)
0000	ANn0	AN0	ANn0
0001	ANn1	AN0 to AN1	ANn0 to ANn1
0010	ANn2	AN0 to AN2	ANn0 to ANn2
0011	ANn3	AN0 to AN3	ANn0 to ANn3
0100~1111	Setting prohibited		

(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. 17.3 Registers related to the A/D Converter

A/D Mode Control Register 2

ADMOD2
(0xFF00_ 4D0C)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFHPA	ADBFHPA	HPADCEA	−	HPADCHA3	HPADCHA2	HPADCHA1	HPADCHA0
Read/Write	R	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 conversion 0: Don't care 1: Start conversion. "0" is always read.	Write "0."	Select analog input channel when activating top-priority conversion			

<HPADCHA4,3.2, 1, 0>	Analog input channel when executing top-priority conversion
0000	AIN0
0001	AIN1
0010	AIN2
0011	AIN3
0100~1111	Setting prohibited

A/D Mode Control Register 3

ADMOD3
(0xFF00 _4D10)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBICA	REGSA3	REGSA2	REGSA1	REGSA0	ADOBSVA
Read/Write	R/W	R	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 register 1: Larger than comparison register	BIT for selecting the AD conversion result storage register that is to be compared with the comparison register if the AD monitor function is enabled				AD monitor function 0: Disable 1: Enable

A/D Mode Control Register 5

ADMOD5
(0xFF00 _4D18)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBICA	REGSA3	REGSA2	REGSA1	REGSA0	ADOBSVA
Read/Write	R/W	R	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 register 1: Larger than comparison register	BIT for selecting the AD conversion result storage register that is to be compared with the comparison register if the AD monitor function is enabled				AD monitor function 0: Disable 1: Enable

<REGSA2, 1, 0>	AD conversion result storage register to be compared
0000	ADAREG0
0001	ADAREG1
0010	ADAREG2
0011	ADAREG3
1XXX	ADAREGSP

A/D Mode Control Register 4

ADAMOD4
(0xFF00_4D14)

	7	6	5	4	3	2	1	0
bit Symbol	HADHSA	HADHTGA	ADHSA	ADHTGA			ADRSTA1	ADRSTA0
Read/Write	R/W				R		W	W
After reset	0	0	0	0	0		–	–
Function	HW source for activating top-priority A/D conversion 0: External TRG 1: TB9RG0	HW for activating top-priority A/D conversion 0: Disable 1: Enable	HW source for activating normal A/D conversion 0: External TRG 1: TB1RG0	HW for activating normal A/D conversion 0: Disable 1: Enable	"0" is read.		Overwriting 10 with 01 allows ADC to be software reset.	

(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 top-priority AD conversion setting and a normal AD conversion setting simultaneously.

(Note 3) The external trigger cannot be used for H/W activation of AD conversion when it is used for H/W activation of top priority AD conversion.

Fig. 17.4 Registers related to the A/D Converter

17.1.2 Unit C

The A/D converter is controlled by A/D mode control registers (ADCMODC0, ADCMODC1, ADCMODC2, ADCMODC3 and ADCMODC4). Results of A/D conversion are stored in A/D conversion result registers ADCREG0 through ADCREG7. Results of top-priority conversion are stored in ADnREGSP.

A/D Mode Control Register 0

ADCMOD0
(0x FF00_4E04)

	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

A/D Mode Control Register 1

ADCMOD1
(0xFF00_ 4E08)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON	I2AD	ADSCN	−	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: Setting prohibited	Write "0."	Select analog input channel			

Select analog input channel

<SCAN> <ADCH3.2, 1, 0>	0 Fixed channel	1 Channel scan (ADSCN=0)	1 Channel scan (ADSCN=1)
0000	AINC0	AIN0	AINC0
0001	AINC1	AIN 0 to AIN 1	AINC 0 to AINC 1
0010	AINC2	AIN 0 to AIN 2	AINC 0 to AINC 2
0011	AINC3	AIN 0 to AIN 3	AINC 0 to AINC 3
0100	AINC4	AINC4	AINC 0 to AINC 4
0101	AINC5	AINC4 to AINC5	AINC 0 to AINC 5
0110	AINC6	AINC4 to AINC6	AINC 0 to AINC 6
0111	AINC7	AINC4 to AINC7	AINC 0 to AINC 3
1000~1111	Setting prohibited		

(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."

A/D Mode Control Register 2

ADCMOD2
(0xFF00_ 4E0C)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFHP	ADBFHP	HPADCE	−	HPADCH3	HPADCH2	HPADCH1	HPADCH0
Read/Write	R	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 conversion 0: Don't care 1: Start conversion. "0" is always read.	Write "0."	Select analog input channel when activating top-priority conversion			

<HPADCH4,3.2, 1, 0>	Analog input channel when executing top-priority conversion
0000	AIN0
0001	AIN1
0010	AIN2
0011	AIN3
0100	AIN4
0101	AIN5
0110	AIN6
0111	AIN7
1000~1111	Setting prohibited

A/D Mode Control Register 3

ADCMOD3
(0xFF00 _4E40)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC	REGS3	REGS2	REGS1	REGS0	ADOBSV
Read/Write	R/W	R	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 register 1: Larger than comparison register	BIT for selecting the AD conversion result storage register that is to be compared with the comparison register if the AD monitor function is enabled				AD monitor function 0: Disable 1: Enable

ADCMOD5
(0xFF00 _4E48)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC	REGS3	REGS2	REGS1	REGS0	ADOBSV
Read/Write	R/W	R	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 register 1: Larger than comparison register	BIT for selecting the AD conversion result storage register that is to be compared with the comparison register if the AD monitor function is enabled				AD monitor function 0: Disable 1: Enable

Fig. 17.5 Registers related to the A/D Converter

A/D Mode Control Register 3

<REGS2, 1, 0>	AD conversion result storage register to be compared
0000	ADCREG0
0001	ADCREG1
0010	ADCREG2
0011	ADCREG3
0100	ADCREG4
0101	ADCREG5
0110	ADCREG6
0111	ADCREG7
1XXX	ADAREGSP

A/D Mode Control Register 4

ADCMOD4
(0xFF00_4E44)

	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		–	–
Function	HW source for activating top-priority A/D conversion 0: External TRG 1: TB9RG0	HW for activating top-priority A/D conversion 0: Disable 1: Enable	HW source for activating normal A/D conversion 0: External TRG 1: TB1RG0	HW for activating normal A/D conversion 0: Disable 1: Enable	"0" is read.		Overwriting 10 with 01 allows ADC to be software reset.	

Overwriting 10 with 01 allows ADC registers excluding the ADCLK<ADCLK2:0>bit to be reset by software.

(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 top-priority AD conversion setting and a normal AD conversion setting simultaneously.

(Note 3) The external trigger cannot be used for H/W activation of AD conversion when it is used for H/W activation of top priority AD conversion.

A/D Conversion Result Register 0 (Unit A, B and C)

ADAREG0
(0xFF00_4D34)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00					OVR0	ADR0RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 1 (Unit A, B and C)

ADAREG1
(0xFF00_4D38)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10					OVR1	ADR1RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" is read.				Over RUNflag 0: Not generate 1: Generate	1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 2 (Unit A, B and C)

ADAREG2
(0xFF00_4D3C)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20					OVR2	ADR2RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 3 (Unit A, B and C)

ADAREG3
(0xFF00_4D40)

	7	6	5	4	3	2	1	0
bit Symbol	ADR31	ADR30					OVR3	ADR3RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result		"0" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 4 (Unit C)

ADCREG4
(0xFF00_4E40)

	7	6	5	4	3	2	1	0
bit Symbol	ADR41	ADR40					OVR4	ADR4RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result.		“0” is read.				Over RUN flag 0 : Not generate 1 : Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 5 (Unit C)

ADCREG5
(0xFF00_4E44)

	7	6	5	4	3	2	1	0
bit Symbol	ADR51	ADR50					OVR5	ADR5RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result.		“0” is read.				Over RUN flag 0 : Not generate 1 : Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 6 (Unit C)

ADCREG6
(0xFF00_4E48)

	7	6	5	4	3	2	1	0
bit Symbol	ADR61	ADR60					OVR6	ADR6RF
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result.		“0” is read.				Over RUN flag 0 : Not generate 1 : Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register 7 (Unit C)

ADCREG7
(0xFF00_4E4C)

	7	6	5	4	3	2	1	0
bit Symbol	ADR71	ADR70					OVR7	ADR7RF
Read/Write	R		R				R	R
After reset	O		0				0	0
Function	Store lower 2 bits of A/D conversion result.		“0” is read.				Over RUN flag 0 : Not generate 1 : Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Register SP (Unit A, B and C)

ADAREGSP
(0xFF00_4D50)

	7	6	5	4	3	2	1	0
bit Symbol	ADRSPA1	ADRSPA0					OVRSPA	ADRSPRFA
Read/Write	R		R				R	R
After reset	0		0				0	0
Function	Store lower 2 bits of A/D conversion result.		“0” is read.				Over RUN flag 0:Not generate 1:Generate	A/D conversion result storage flag 1: Presence of conversion result
	15	14	13	12	11	10	9	8
bit Symbol	ADRSP9	ADRSP8	ADRSP7	ADRSP6	ADRSP5	ADRSP4	ADRSP3	ADRSP2
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Comparison Register 0 (Unit A, B and C)

ADACOMREG0
(0xFF00_4D54)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20						
Read/Write	R/W		R					
After reset	0		0					
Function	Store lower 2 bits of A/D conversion result.		"0" is read.					
	15	14	13	12	11	10	9	8
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R/W							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

A/D Conversion Result Comparison Register 1 (Unit A, B and C)

ADACOMREG1
(0xFF00_4D58)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20						
Read/Write	R/W		R					
After reset	0		0					
Function	Store lower 2 bits of A/D conversion result.		"0" is read.					
	15	14	13	12	11	10	9	8
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R/W							
After reset	0							
Function	Store upper 8 bits of A/D conversion result.							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							

(Note) To set or change a value in this register, the AD monitor function must be disabled (ADnMOD3<ADOBSV>="0").

17.2 Conversion Clock

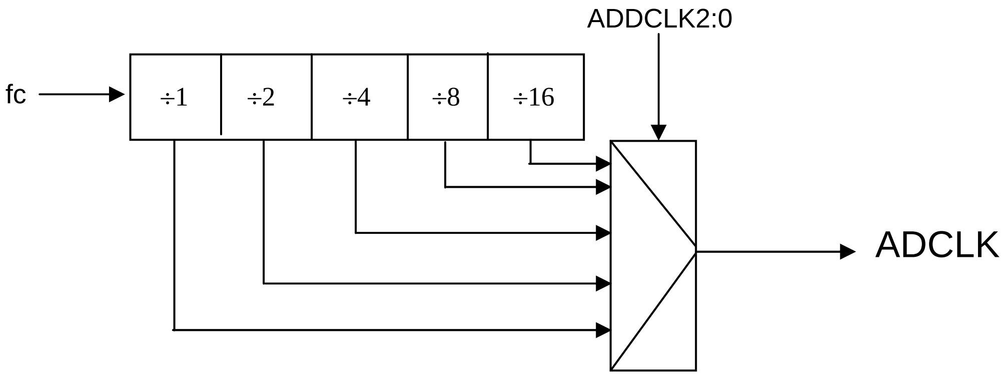
- The conversion time is calculated by the 46 conversion clock.

A/D Conversion Clock Setting Register

ADACLK
(0xFF00_4D00)

	7	6	5	4	3	2	1	0
bit Symbol	TSH3	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	1	0	0	0	0	0	0	0
Function	Select the A/D sample hold time 1000: 8 conversion clock 1001:16 conversion clock 1010: 24 conversion clock 1011: 32 conversion clock 0011: 64 conversion clock 1100: 128 conversion clock 1101: 512 conversion clock Other than above: reserved				"0" is read.	Select the A/D prescaler output 000: fc (note1) 001: fc/2 010: fc/4 011: fc/8 100: fc/16 111:reserved		

note1.Please change the setting from an initial value when 80MHz operates.
note2 ADCLK<2:0 > is not initialized in software reset.



Example: If fsys = fc = 80 MHz

Prescaler [ADDCLK2:0]	tconv. (conversion time) 40MHz	tconv. (conversion time) 80MHz
1	1.15μs	Setting prohibited
1/2	2.3μs	1.15μs
1/4	4.6μs	2.3μs

Variable S/H time

fc=fsys →Conversion clock	S/H time	tconv. (conversion time)

fc=40MHz → ADCLK=40MHz (Prescalar 1/1)
fc=80MHz → ADCLK=40MHz (Prescalar 1/2)
	Conversion clk*8 (0.2us)	1.15μs

	Conversion clk*16 (0.4us)	1.35μs
	Conversion clk*24 (0.6us)	1.55μs
	Conversion clk*32 (0.8us)	1.75μs
	Conversion clk*64 (1.6us)	2.55μs
	Conversion clk*128 (3.2us)	4.15μs
	Conversion clk*512 (12.8us)	13.75μs

(Note) "Please do not change the analog to digital conversion clock setting in the analog to digital translation. "

17.3 Description of Operations

17.3.1 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 ADnMOD0<ADS> bit.

17.3.2 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 (ADnMOD0<SCAN>="0"):

One channel is selected from analog input pins AINn0 through AINn3 and AINC0 through AINC7 by setting ADnMOD1<ADCH3 to 0> to an appropriate setting.

If the analog input channel is used in a scan state (ADnMOD0<SCAN>="1"):

One scan mode is selected from 16 scan modes by setting ADnMOD1<ADCH3 to 0> and ADSCN to appropriate settings.

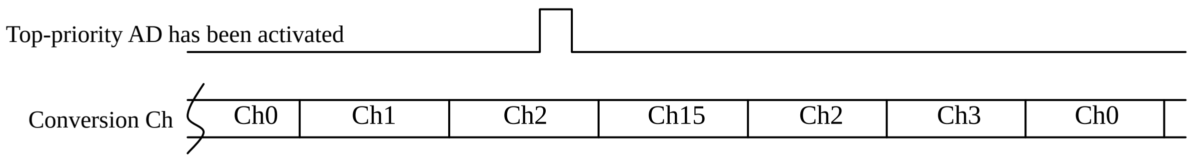
(2) Top-priority AD conversion mode

One channel is selected from analog input pins AINn0 through AINn3 and AINC0 through AINC7 by setting ADnMOD2<HPADCH3 to 0> to an appropriate setting.

After a reset, ADnMOD0<SCAN> is initialized to "0" and ADnMOD1<ADCH3:0> is initialized to "0000." This initialization works as a trigger to select a fixed channel input through the ANn0 pin. The pins that are not used as analog input channels can be used as ordinary input ports.

If top-priority AD conversion is activated during normal AD conversion, normal AD conversion is discontinued, top-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 AINC0 through AINC3 with ADCMOD0 <REPEAT:SCAN> set to "11" and ADCMOD1<ADCH3:0> set to 0011, and top-priority AD conversion has been activated at AINC7 with ADCMOD2<HPADCH3:0>=0111:



17.3.3 Starting A/D Conversion

Two types of A/D conversion are supported: normal AD conversion and top-priority AD conversion. Normal AD conversion is software activated by setting ADnMOD0<ADS> to "1." Top-priority AD conversion is software activated by setting ADnMOD2<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 ADnMOD0<2:1> to an appropriate setting. For top-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 ADnMOD4<ADHS>, and top-priority AD conversion can be activated using the HW activation source selected by ADnMOD4<HADHS>. If this bit is "0," normal and top-priority AD conversions are activated in response to the input of a falling edge through the $\overline{\text{ADTRGn}}$

pin. If this bit is "1," normal AD conversion is activated in response to TB1TRG generated by the 16-bit timer 1, and top-priority AD conversion is activated in response to TB9TRG generated by the 16-bit timer 9. Software activation is still valid even after H/W activation has been authorized.

(Note) When an external trigger is used for the HW start source of a top priority analog to digital translation, an external trigger cannot usually be set as analog to digital translation HW start.

Each unit has a pin (unit A: ADTRGA, unit B: ADTRGB, unit C: ADTRGC) that is used exclusively for an external trigger.

Inputting an external trigger from ADTRGSNC pin allows unit A and unit B to start simultaneously.

When normal A/D conversion starts, the A/D conversion Busy flag (ADnMOD0<ADBF>) showing that A/D conversion is under way is set to "1." When top-priority A/D conversion starts, the A/D conversion Busy flag (ADnMOD2<ADBFHP>) showing that A/D conversion is under way is set to "1." If normal A/D conversion is interrupted by top-priority A/D conversion, the value of the Busy flag for normal A/D conversion before the start of top-priority A/D conversion is retained. The value of the conversion completion flag EOCFN for normal A/D conversion before the start of top-priority A/D conversion can also be retained.

(Note) Normal A/D conversion must not be activated when top-priority A/D conversion is under way. If activated when top-priority A/D conversion is under way, the top-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 (ADnMOD4<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 ADnMOD2<HPADCE> is set to "1" during normal A/D conversion, ongoing A/D conversion is discontinued and top-priority A/D conversion starts; specifically, A/D conversion (fixed channel single conversion) is executed for a channel designated by ADnMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADnREGSP, normal A/D conversion is resumed.

If HW activation of top-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 top-priority A/D conversion (fixed channel single conversion) starts for a channel designated by ADnMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADnREGSP, normal A/D conversion is resumed.

17.3.4 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 ADnMOD0<2:1> to an appropriate setting. For top-priority A/D conversion, the fixed channel single conversion mode is automatically selected, irrespective of the ADnMOD0<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 ADnMOD0<REPEAT, SCAN>. As A/D conversion starts, ADnMOD0<ADBFN> is set to "1." When specified A/D conversion is completed, the A/D conversion completion interrupt (INTADn) is generated, and ADnMOD0<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 ADnMOD0 <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, ADnMOD0<EOCF> is set to "1," ADnMOD0<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 ADnMOD0 <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, ADnMOD0<EOCF> is set to "1," ADnMOD0<ADBF> is cleared to "0," and the interrupt request INTADn is generated. <EOCF> is cleared to "0" upon read.

③ Fixed channel repeat conversion mode

If ADnMOD0<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, ADnMOD <EOCF> is set to "1." ADnMOD0 <ADBF> is not cleared to "0." It remains at "1." The timing with which the interrupt request INTADn 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 ADnREG0. 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 ADREG0 through ADREG3. After the conversion results are stored in ADREG3, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADnREG0. <EOCF> is cleared to "0" upon read.

With <ITM1:0> set to "10" (applicable only to unit C), 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 ADCREG0 through ADCREG7. After the conversion results are stored in ADCREG7, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADCREG0.

<EOCF> is cleared to "0" upon read.

④ Channel scan repeat conversion mode

If ADnMOD0 <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, ADnMOD0 <EOCF> is set to "1," and the interrupt request INTADn is generated. ADnMOD0 <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 ADnMOD0 <REPEAT>. When ongoing A/D conversion is completed, the repeat conversion mode terminates, and ADnMOD0 <ADBF> is set to "0."

Before switching from one mode to standby mode (such standby modes as IDLE, STOP and BUCKUP etc.), check that A/D conversion is not being executed. If A/D conversion is under way, you must stop it or wait until it is completed.

(2) Top-priority A/D conversion

Top-priority A/D conversion is performed only in fixed channel single conversion mode. The ADnMOD0<REPEAT, SCAN> setting has no relevance to the top-priority A/D conversion operations or preparations. As activation requirements are met, A/D conversion is performed only once for a channel designated by ADnMOD2<HPADCH3:0>. After the A/D conversion is completed, the top-priority A/D conversion completion interrupt is generated, ADnMOD2<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)	ADnMOD0		
				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 (unit C only)	After eight conversions are completed	1	10		
Channel scan single conversion	After scan conversion is completed	After scan conversion is completed	0	—	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.

17.3.5 High-priority Conversion Mode

By interrupting ongoing normal A/D conversion, top-priority A/D conversion can be performed. Top-priority A/D conversion can be software activated by setting ADnMOD2<HPADCE> to "1" or it can be activated using the HW resource by setting ADnMOD4<7:6> to an appropriate setting. If top-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 ADnMOD2<3:0>. The result of single conversion is stored in ADnREGSP, and the top-priority A/D conversion interrupt is generated. After top-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. Top-priority A/D conversion activated while top-priority A/D conversion is under way is ignored.

For example, if channel repeat conversion is activated for channels ANC0 through ANC7 and if <HPADCE> is set to "1" during ANC3 conversion, AN3 conversion is suspended, and conversion is performed for a channel designated by <HPADC3:0>. After the result of conversion is stored in ADCREGSP, channel repeat conversion is resumed, starting from ANC3.

17.3.6 A/D Monitor Function

If ADnMOD3<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 <ADnRRF> is also set. To use the A/D monitor function, therefore, a flag of a corresponding conversion result storage register must not be used.

Two values can be specified in each unit at a time for the comparison.

17.3.7 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 (ADAREG0 through ADARG3, ADBREG0 through ADBRG3, ADCREG0 through ADCRG7).

In fixed channel repeat conversion mode, A/D conversion results are sequentially stored in ADnREG0 through ADnREG3 and ADnREG7. 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 ADnREG0. 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 ADnREG0 through ADnREG3.

Table 17.1 shows analog input channels and related A/D conversion result registers.

Table 17.1 Analog Input Channels and Related A/D Conversion Result Registers

Analog input channel (port 7)	**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)**
AINA0/AINB0	ADAREG0/ ADBREG0	ADAREG0 fixed/ ADBREG0 fixed	ADnREG0 ← ↓ ↓ ADnREG3	
AINA1/AINB1	ADAREG1/ ADBREG1			
AINA2/AINB2	ADRAEG2/ ADBREG2			
AINA3/AINB3	ADAREG3/ ADBREG3			

Analog input channel (port 8)	**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)**
AINC0	ADCREG0	ADCREG0 fixed	ADCREG0 ← ↓ ↓ ADCREG3	ADCREG0 ← ↓ ↓ ↓ ADCREG7
AINC1	ADCREG1			
AINC2	ADCREG2			
AINC3	ADCREG3			
AINC4	ADCREG4			
AINC5	ADCREG5			
AINC6	ADCREG6			
AINC7	ADCREG7			

17.3.8 Data Polling

To process A/D conversion results without using interrupts, ADnMOD0<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.

18. Watchdog Timer (Runaway Detection Timer)

The TMP19A44 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.

18.1 Configuration

Fig. 18.1 shows the block diagram of the watchdog timer.

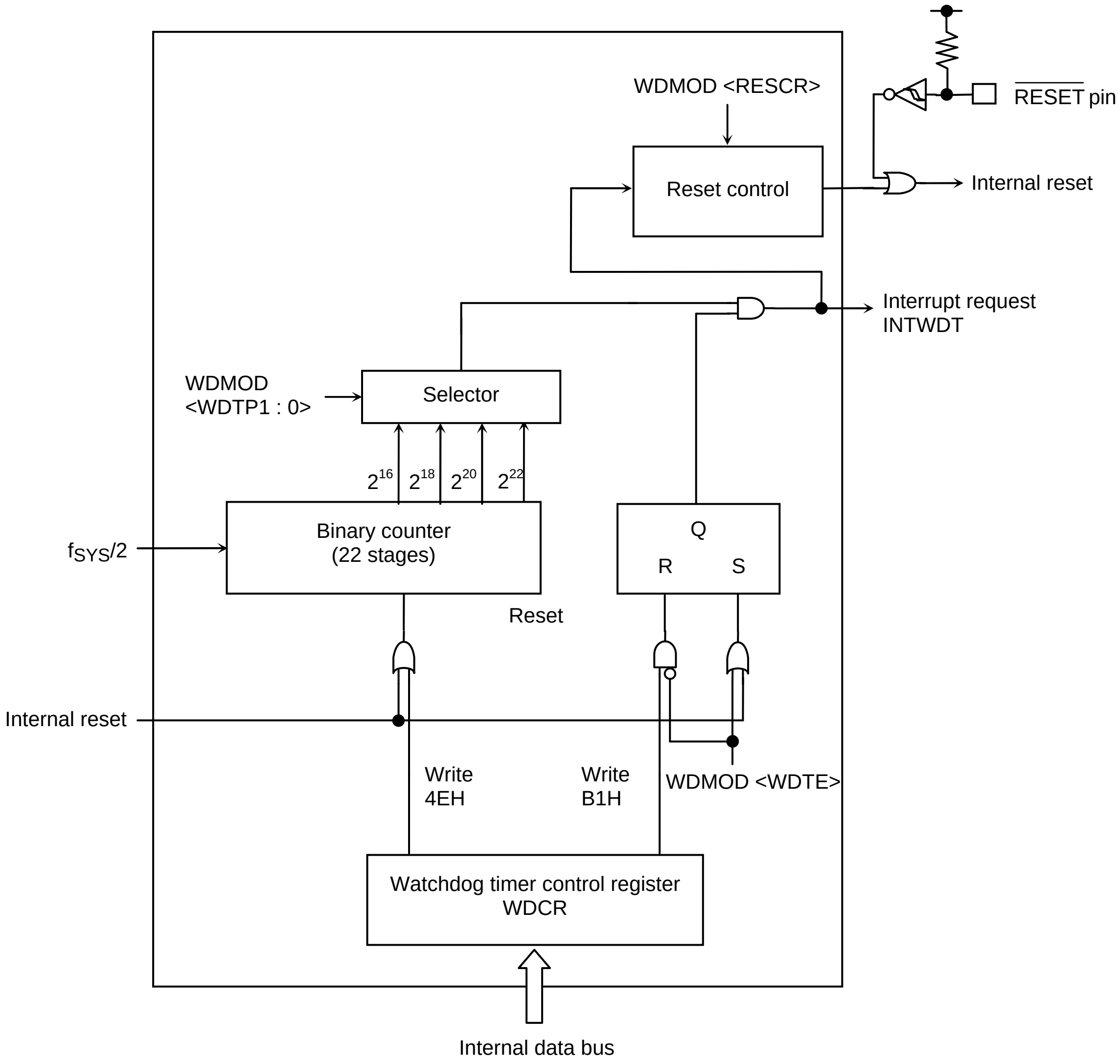


Fig. 18.1 Block Diagram of the Watchdog Timer

18.2 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^{16}, 2^{18}, 2^{20}, 2^{22}, 2^{24} and 2^{26}. 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. 18.2.

Because the watchdog timer interrupt is a non-maskable interrupt factor, NMIFLG <WDT> at the INTC performs a task of identifying it.

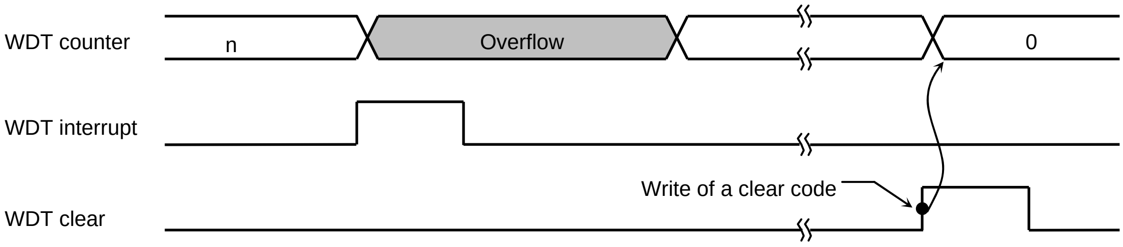


Fig. 18.2 Normal Mode

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is affected for a 32-state time, as shown in Fig. 18.3. If this reset is affected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 16 is used as an input clock $f_{SYS/2}$.

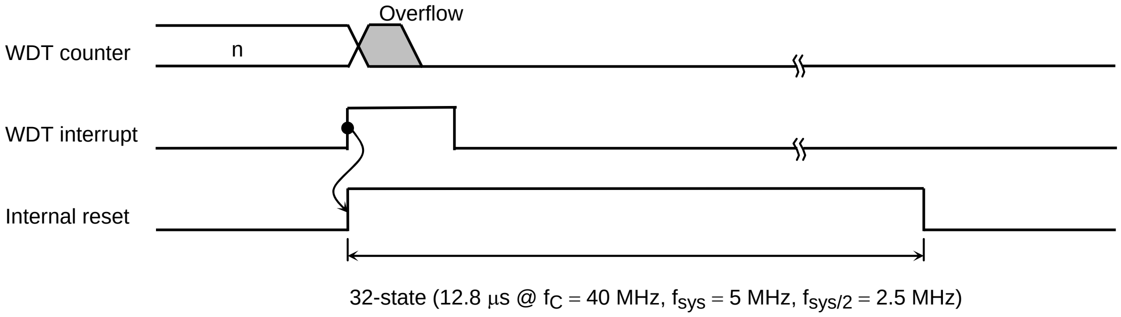

Overflow
WDT counter
n
WDT interrupt
Internal reset
32-state (12.8 μs @ fC = 40 MHz, fsys = 5 MHz, fsys/2 = 2.5 MHz)

Fig. 18.3 Reset Mode

18.3 Control Registers

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

18.3.1 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. 18.4 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 is a register for specifying whether or not to reset the watchdog timer itself after a runaway is detected. As a reset initializes this setting to WDMOD <RESCR>="0," a reset initiated the output of the watchdog timer is not performed.

18.3.2 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.

WDMOD
0xFF00_4F00

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP2	WDTP1	WDTP0		I2WDT	RESCR	
Read/Write	R/W	R/W			R	R/W		R/W
After reset	1	0	0	0		0	1	0
Function	WDT control 1: Enable	Selects WDT detection time 000: $2^{16}/f_{SYS}$ 001: $2^{18}/f_{SYS}$ 010: $2^{20}/f_{SYS}$ 011: $2^{22}/f_{SYS}$ 100: $2^{24}/f_{SYS}$ 101: $2^{26}/f_{SYS}$ 110: Setting prohibited 111: Setting prohibited			"0" is read.	IDLE 0: Stop 1: Start	0: Generate NMI interrupt 1: Internally connect WDT output to reset pin	Write "0."

Watchdog timer out control

0	Generates NMI interrupt
1	Connects WDTOUT to reset

Detection time of watchdog timer @ fc = 80 MHz

SYSCR1 clock gear value <GEAR2:0>	Detection Time of Watchdog Timer					
	WDMOD<WDTP2:0>					
	000	001	010	011	100	101
000 （fc）	0.82 ms	3.28 ms	13.11 ms	52.43 ms	209.72 ms	0.84 s
100 (fc/2)	1.64 ms	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s
101 (fc/4)	3.28 ms	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s
110 (fc/8)	6.55 ms	26.21 ms	104.86 ms	419.43 ms	1.68 s	6.71 s
111 (fc/16)	13.11 ms	52.43 ms	209.72 ms	838.86 ms	3.36 s	13.42 s

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Fig. 18.4 Watchdog Timer Mode Register

WDCR
(0xFF00_4F04)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	—							
Function	B1H : WDT disable code 4EH : WDT clear code							

Disable & clear of WDT

B1H	Disable code
4EH	Clear code
Others	—

Fig. 18.5 Watchdog Timer Control Register

18.4 Operation Description

The watchdog timer generates the INTWDT interrupt after a lapse of the detection time specified by the WDMOD <WDTP2, 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.

19. Real Time Clock (RTC)

19.1 Functions

1) Clock (hour, minute and second)
2) Calendar (month, week, date and leap year)
3) Selectable 12 (am/ pm) and 24 hour display
4) Time adjustment ＋ or － 30 seconds (by software)
5) Alarm interrupt (selectable from 1sec/ 500msec/ 250msec/ 125msec/ 62.5msec or calendar)

19.2 Block Diagram

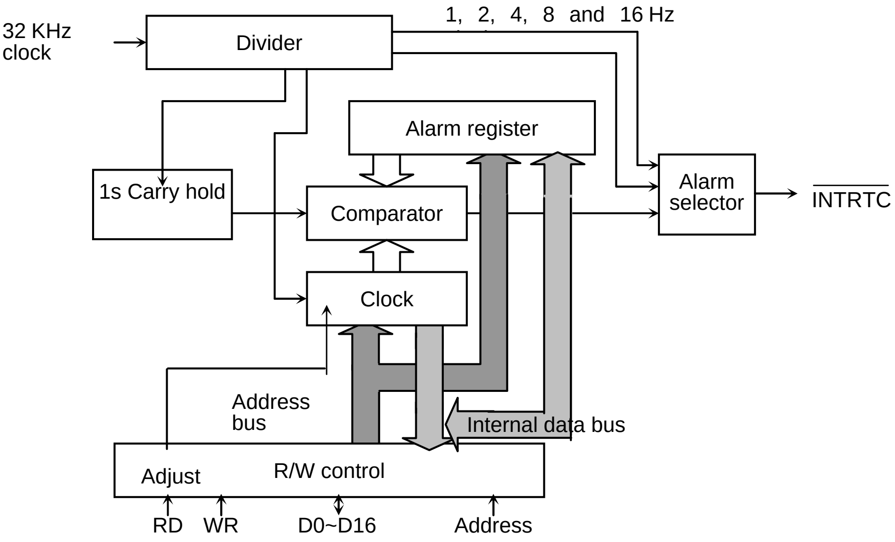


Fig. 19.1 Block Diagram

(Note 1) Western calendar year column:
This product uses only the final two digits of the year. The year following 99 is 00 years. Please take into account the first two digits when handling years in the western calendar.

(Note 2) Leap year:
A leap year is divisible by 4 excluding a year divisible by 100; the year divisible by 100 is not considered to be a leap year. Any year divisible by 400 is a leap year. This product is considered the year divisible by 4 to be a leap year and does not take into account the above exceptions. It needs adjustments for the exceptions.

19.3 Registers

19.3.1 Control Register

Table 19.1 PAGE0 (clock function) register

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
SECR	0xFF00_1500		40 sec	20 sec	10 sec	8 sec	4 sec	2 sec	1 sec	Second column	R/W
MINR	0xFF00_1501		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
HOURR	0xFF00_1502			20 hours /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hours	Hour column	R/W
DAYR	0xFF00_1504						W2	W1	W0	Day of the week column	R/W
DATER	0xFF00_1505			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
MONTHR	0xFF00_1506				Oct.	Aug.	Apr.	Feb.	Jan.	Month column	R/W
YEARR	0xFF00_1507	Year 80	Year 40	Year 20	Year 10	Year 8	Year 4	Year 2	Year 1	Year column (lower two columns)	R/W
PAGER	0xFF00_1508	Interrupt enable			Adjustment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RESTR	0xFF00_150C	1Hz enable	16Hz enable	Clock reset	Alarm reset	Always write "0".				Reset register	W only

(Note) Reading SECR, MINR, HOURR, DAYR, MONTHR, YEARR of PAGE0 captures the current state.

Table 19.2 PAGE1 (alarm function) registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
SECR	0xFF00_1500										
MINR	0xFF00_1501		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
HOURR	0xFF00_1502			20 hours /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hours	Hour column	R/W
DAYR	0xFF00_1504						W2	W1	W0	Day of the week column	R/W
DATER	0xFF00_1505			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
MONTHR	0xFF00_1506								24/12	24-hour clock mode	R/W
YEARR	0xFF00_1507							Leap-year setting		Leap-year mode	R/W
PAGER	0xFF00_1508	Interrupt enable			Adjustment function	Clock enable	Alarm enable		PAGE setting	PAGE register	W,R/W
RESTR	0xFF00_150C	1Hz enable	16Hz enable	Clock reset	Alarm reset	Always write "0".				Reset register	W only

(Note 1) Reading SECR, MINR, HOURR, DAYR, MONTHR, YEARR of PAGE1 captures the current state.

(Note 2) SECR, MINR, HOURR, DAYR, MONTHR, YEARR of PAGE0 and YEARR of PAGE1 (for leap year) must be read twice and compare the data captured.

19.3.2 Detailed Description of Control Register

The RTC is not initialized by system reset. All registers must be initialized at the beginning of the program.

(1) Second column register (for PAGE0 only)

SECR

	7	6	5	4	3	2	1	0
Bit symbol		SE6	SE5	SE4	SE3	SE2	SE1	SE0
Read/Write		R/W						
After reset		Undefined						
Function	“0” is read.	40 sec. column	20 sec. column	10 sec. column	8 sec. column	4 sec. column	2 sec. column	1 sec. column

0	0	0	0	0	0	0	0 sec
0	0	0	0	0	0	1	1 sec
0	0	0	0	0	1	0	2 sec
0	0	0	0	0	1	1	3 sec
0	0	0	0	1	0	0	4 sec
0	0	0	0	1	0	1	5 sec
0	0	0	0	1	1	0	6 sec
0	0	0	0	1	1	1	7 sec
0	0	0	1	0	0	0	8 sec
0	0	0	1	0	0	1	9 sec
0	0	1	0	0	0	0	10 sec
			:				
0	0	1	1	0	0	1	19 sec
0	1	0	0	0	0	0	20 sec
			:				
0	1	0	1	0	0	1	29 sec
0	1	1	0	0	0	0	30 sec
			:				
0	1	1	1	0	0	1	39 sec
1	0	0	0	0	0	0	40 sec
			:				
1	0	0	1	0	0	1	49 sec
1	0	1	0	0	0	0	50 sec
			:				
1	0	1	1	0	0	1	59 sec

Note) Do not set data other than as shown above.

(2) Minute column register (for PAGE0/1)

MINR

	7	6	5	4	3	2	1	0
Bit symbol		MI6	MI5	MI4	MI3	MI2	MI1	MI0
Read/Write		R/W						
After reset		Undefined						
Function	"0" is read	40 min. column	20 min. column	10 min. column	8 min. column	4 min. column	2 min. column	1 min. column

0	0	0	0	0	0	0	0 min
0	0	0	0	0	0	1	1 min
0	0	0	0	0	1	0	2 min
0	0	0	0	0	1	1	3 min
0	0	0	0	1	0	0	4 min
0	0	0	0	1	0	1	5 min
0	0	0	0	1	1	0	6 min
0	0	0	0	1	1	1	7 min
0	0	0	1	0	0	0	8 min
0	0	0	1	0	0	1	9 min
0	0	1	0	0	0	0	10 min

:

0	0	1	1	0	0	1	19 min
0	1	0	0	0	0	0	20 min

:

0	1	0	1	0	0	1	29 min
0	1	1	0	0	0	0	30 min

:

0	1	1	1	0	0	1	39 min
1	0	0	0	0	0	0	40 min

:

1	0	0	1	0	0	1	49 min
1	0	1	0	0	0	0	50 min

:

1	0	1	1	0	0	1	59 min

Note) Do not set data other than as shown above.

1	1	1	1	1	1	1	don't care

(3) Hour column register (for PAGE0/1)

1. 24-hour clock mode (MONTHR<MO0>="1")

HOURR

	7	6	5	4	3	2	1	0
Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
Read/Write			R/W					
After reset			Undefined					
Function	"0"is read.		20 hour column	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o' clock
0	0	0	0	0	1	1 o' clock
0	0	0	0	1	0	2 o' clock

:

0	0	1	0	0	0	8 o' clock
0	0	1	0	0	1	9 o' clock
0	1	0	0	0	0	10 o' clock

:

0	1	1	0	0	1	19 o' clock
1	0	0	0	0	0	20 o' clock

:

1	0	0	0	1	1	23 o' clock

Note) Do not set data other than as shown above.

1	1	1	1	1	1	don't care

2. 12-hour clock mode (MONTHR<MO0>="0")

HOURR

	7	6	5	4	3	2	1	0
Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
Read/Write			R/W					
After reset			Undefined					
Function	"0"is read.		PM/AM	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o' clock (AM)
0	0	0	0	0	1	1 o' clock
0	0	0	0	1	0	2 o' clock

:

0	0	1	0	0	1	9 o' clock
0	1	0	0	0	0	10 o' clock
0	1	0	0	0	1	11 o' clock
1	0	0	0	0	0	0 o' clock (PM)
1	0	0	0	0	1	1 o' clock

Note) Do not set data other than as shown above.

1	1	1	1	1	1	don't care

(4) Day of the week column register (for PAGE0/1)

DAYR

	7	6	5	4	3	2	1	0
Bit symbol						WE2	WE1	WE0
Read/Write						R/W		
After reset						Undefined		
Function	"0" is read.					W2	W1	W0

0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

Note) Do not set data other than as shown above.

1	1	1	don't care

(5) Day column register (PAGE0/1)

DATER

	7	6	5	4	3	2	1	0
Bit symbol			DA5	DA4	DA3	DA2	DA1	DA0
Read/Write			R/W					
After reset			Undefined					
Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

0	0	0	0	0	0	0
0	0	0	0	0	1	1st day
0	0	0	0	1	0	2nd day
0	0	0	0	1	1	3rd day
0	0	0	1	0	0	4th day

:

0	0	1	0	0	1	9th day
0	1	0	0	0	0	10th day
0	1	0	0	0	1	11th day

:

0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day

:

1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

Note 1) Do not set data other than as shown above.

Note 2) Do not set for non-existent days (e.g.: 30th Feb)

1	1	1	1	1	1	don't care

(6) Month column register (for PAGE0 only)

MONTHR

	7	6	5	4	3	2	1	0
Bit symbol				MO4	MO4	MO2	MO1	MO0
Read/Write				R/W				
After reset				Undefined				
Function	“0” is read.			10 months	8 months	4 months	2 months	1 month

0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May
0	0	1	1	0	June
0	0	1	1	1	July
0	1	0	0	0	August
0	1	0	0	1	September
1	0	0	0	0	October
1	0	0	0	1	November
1	0	0	1	0	December

Note) Do not set data other than as shown above.

(7) Selection of 24-hour clock or 12-hour clock (for PAGE1 only)

MONTHR

	7	6	5	4	3	2	1	0
Bit symbol								MO0
Read/Write								R/W
After reset								Undefined
Function	“0” is read.							1: 24-hour 0: 12-hour

(Note) Do not change the MONTHR<MO0> bit while the RTC is in operation (PAGER<RNATMR>="1").

(8) Year column register (for PAGE0 only)

YEARR

	7	6	5	4	3	2	1	0
Bit symbol	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
Read/Write	R/W							
After reset	Undefined							
Function	80 years	40 years	20 years	10 years	8 years	4 years	2 years	1 years

0	0	0	0	0	0	0	0	00 years
0	0	0	0	0	0	0	1	01 years
0	0	0	0	0	0	1	0	02 years
0	0	0	0	0	0	1	1	03 years
0	0	0	0	0	1	0	0	04 years
0	0	0	0	0	1	0	1	05 years
				:				
1	0	0	1	1	0	0	1	99 years

Note) Do not set data other than as shown above.

(9) Leap year register (for PAGE1 only)

YEARR

	7	6	5	4	3	2	1	0
Bit symbol							LEAP1	LEAP0
Read/Write							R/W	
After reset							Undefined	
Function	"0" is read.						00: leap year 01: one year after leap year 10: two years after leap year 11: three years after leap year	

0	0	Current year is a leap-year.
0	1	Current year is the year following a leap-year.
1	0	Current year is two years after a leap year.
1	1	Current year is three years after a leap year

(10) PAGE register (for PAGE0/1)

PAGER

a read-modify-write operation cannot be performed.

	7	6	5	4	3	2	1	0
Bit symbol	INTENA			ADJUST	ENATMR	ENAALM		PAGE
Read/Write	R/W			W	R/W			R/W
After reset	0			Undefined	Undefined			Undefined
Function	INTRTC 0: Disabled 1: Enabled	"0" is read.		0: Don't care 1: Adjust	Clock 0: Disabled 1: Enabled	ALARM 0: Disabled 1: Enabled	"0" is read.	PAGE selection

(Note) Keep the setting order of <ENATMR>, <ENAAML> and <INTENA> as shown in the example below. Ensure an interval of time between Clock/Alarm and interrupt.

PAGE	0	Selects Page0
	1	Selects Page1

ADJUST	0	Don't care
	1	Adjusts sec. counter by setting this bit to "1". If it is set when the time elapsed is within 0 and 29, the sec. counter is cleared to "0". If the time elapsed is within 30 and 59, the min. counter is carried and sec. counter is cleared to "0". The ADJUST signal is output during 1 cycle of $f_{SYS.}$ After being adjusted once, the ADJUST state is released automatically (PAGE0 only).

(11) Reset register (for PAGE0/1)

RESTR

(0xFF00_150C)

A read-modify-write operation cannot be performed.

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	–	DIS2HZ	DIS4HZ	DIS8HZ
Read/Write	R/W		R/W		R	R/W		
After reset	1		0		0	1	1	1
Function	1 Hz 0: Enabled 1: Disabled	16 Hz 0: Enabled 1: Disabled	1: Clock reset	1: Alarm reset	Always write "0".	1 Hz 0: Enabled 1: Disabled	1 Hz 0: Enabled 1: Disabled	1 Hz 0: Enabled 1: Disabled

RSTALM	0	Unused
	1	Reset alarm register.

RSTTMR	0	Unused
	1	Reset clock register.

<DIS1HZ> <DIS2HZ> <DIS4HZ> <DIS8HZ> <DIS16HZ>	0	Enabled
	1	Disabled

19.4 Operational description

The RTC incorporates a sec. counter that generates an 1Hz signal from a 32.768 KHz signal. The sec. counter operation must be taken into account when using the RTC.

19.4.1 Clock Operation

(1) Reading clock data

1. Using 1Hz interrupt

The count-up of the internal data synchronizes with 1Hz interrupt. Data can be read correctly if reading data after 1Hz interrupt occurred.

2. Using pair reading

There is a possibility that the clock data may be read incorrectly if the internal counter operates carry during reading. To ensure correct data reading, read the clock data twice as shown below. A pair of data read successively needs to match.

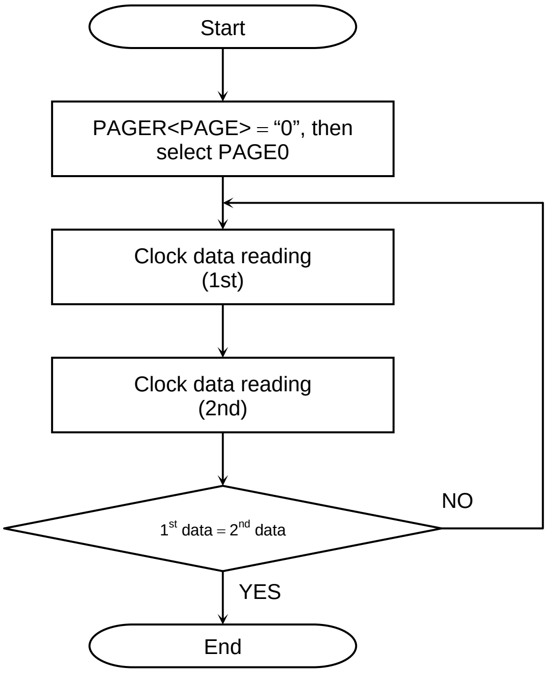


Fig. 19.2 Flowchart of the clock data reading

(2) Writing clock data

A carry during writing ruins correct data writing. The following procedure ensures the correct data writing.

1. Using 1Hz interrupt

The count-up of the internal data synchronizes with 1Hz interrupt. Data can be written correctly if writing data after 1Hz interrupt occurred.

2. Resetting counter

The RTC incorporates 15-stage counter that generates a 1Hz clock from 32,768 KHz. After resetting the counter, the data is written.

If clearing the counter, an interrupt is output only first writing at half of the setting time. To ensure the correct clock counting, enable the 1Hz-interrupt after clearing the counter. And then set the time after the first interrupt (occurs at 0.5Hz) occurs.

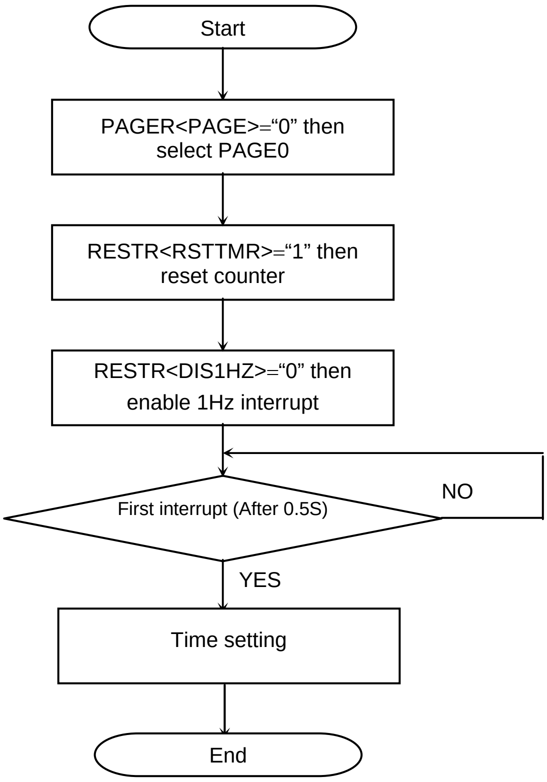

Start
PAGER<PAGE>="0" then select PAGE0
RESTR<RSTTMR>="1" then reset counter
RESTR<DIS1HZ>="0" then enable 1Hz interrupt
First interrupt (After 0.5S)
NO
YES
Time setting
End

Fig. 19.3 Flowchart of the clock data writing

3. Disabling the clock

Writing “0” to PAGER<ENATMR> disables clock operation including a carry. In the meantime, the 1 second carry hold circuit executes time adjustment instead.

While the clock is disabled, the carry hold circuit holds a 1 second carry signal from a divider. When the clock becomes enabled, the carry signal is output to the clock, the time is revised and operation continues. The disabled duration for one second and more causes slowing of clocks.

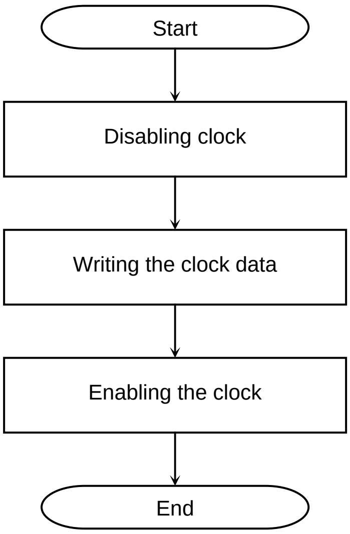


Fig. 19.4 Flowchart of the disabling clock

19.4.2 Alarm function

By writing “1” to PAGER<PAGE>, the alarm function of the PAGE1 registers is enabled. The INTRTC outputs a 1-shot pulse by detecting the falling edge.

The RTC is not initialized by reset. Clear the interrupt request flag in the interrupt controller when the clock or alarm function is used.

(1) INTRTC interrupt is generated when the alarm register corresponds with the clock

(2) 1Hz ,2Hz,4Hz,8Hz, 16Hz clock

(1) How to use alarm

To initialize the alarm, write “1” to RESTR<RSTALM>. It makes all alarm settings “don’t care”. In this case, the alarm always corresponds with the value of the clock. The INTRTC interrupt request is generated if PAGER <INENA> <ENAALM>=”1”.

Setting alarm for min., hour, date and day is done by writing data to the relevant PAGE1 register. Each writing releases the “don’t care” state respectively.

When all setting contents correspond with the setting of PAGER<INTENA> <ENAALM>=1”, the RTC generates an INTRTC interrupt. However, contents which have not been set up (“don't care” state) are always considered to be corresponding.

Contents which have already been set up cannot be returned independently but all together to the “don’t care” state by initializing the alarm.

The following is an example program for outputting an alarm from the $\overline{\text{ALARM}}$ pin at noon (PM12:00) every day.

```
LD      (PAGER), 09H        ;   Disables alarm, sets PAGE1
LD      (RESTR), D7H        ;   Initializes alarm
LD      (DAYR), FFH         ;
LD      (DATAR),FFH
LD      (HOURR), FFH        ;
LD      (MINR), FFH
LD      (HOURR), 12H        ;   Sets 12 o’clock
LD      (MINR), 00H         ;   Sets 00 min.
                            ;   Set up time 31   μs (Note)
LD      (PAGER), 0CH        ;   Enables alarm
LD      (PAGER), 8CH        ;   Enables interrupt
```

When the CPU is operating at high frequency oscillation, it may take a maximum of one clock at 32 kHz (about 30us) for the time register setting to become valid. In the above example, it is necessary to set 31μs of set up time between setting the time register and enabling the alarm register.

(Note) This set up time is unnecessary when you use only internal interruption.

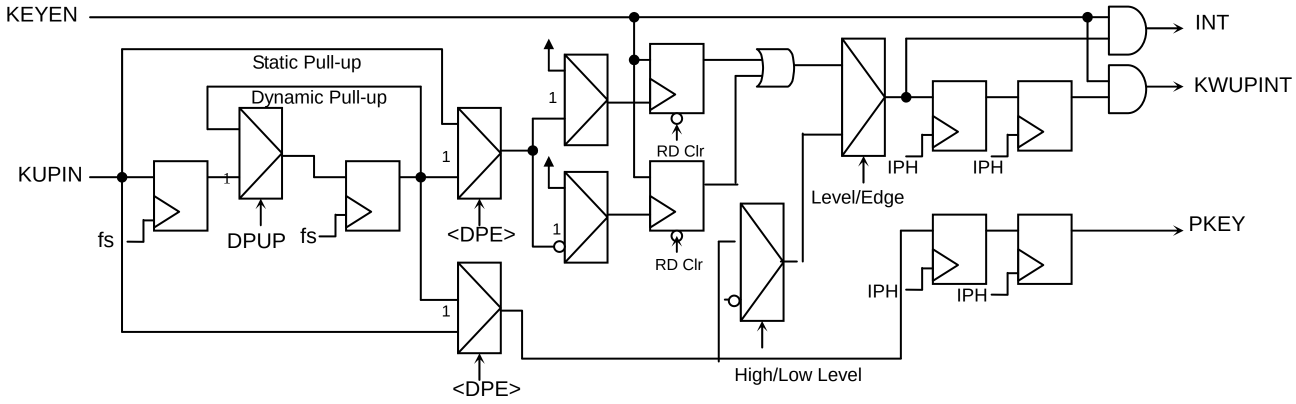
(2) 1Hz clock (2,4,8,16Hz clock)

Setting PAGER<ENAALM>= “0”, RESTR<DIS1HZ>= “0” and <DIS16HZ>= “1” generates an INTRC interrupt per second.

20. Key-on Wakeup Circuit

20.1 Outline

- The TMP19A44 has 32 key inputs, KEY00 to KEY31, which can be used for releasing the Standby mode or for external interrupts. Note that interrupt processing is executed with one interrupt factor for the 32 inputs. (This is programmed in the CG block.) Each key input can be configured to be used or not, by programming (KWUPSTn)<KEYnEN>.
- The active state of each input can be configured to the rising edge, the falling edge, both edges, the high level or the low level, by programming (KWUPSTn)<KEYn>.
- An interrupt request is cleared by programming the key interrupt request clear register KWUPCLR in the interrupt processing.
- The key input pins have pull-up functions, which can be switched between static pull-up and dynamic pull-up by programming the (KWUPSTn)<DPEn> bit. This programming is needed for each of 32 inputs.



20.2 Key-on Wakeup Operation

The TMP19A44 has 32 key input pins, KEY00 to KEY31. Program the IMCGD<KWUPEN> register in the CG to determine whether to use the key inputs for releasing the Standby mode or for normal interrupts. Setting <KWUPEN> to “1” causes all the key inputs, KEY00 to KEY31, to be used for interrupts for releasing the STOP mode. Program KWUPSTn<KEYnEN> to enable or disable interrupt inputs for each key input pin. Also, program KWUPSTn<KEYn2: 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<EMCGD2:0> to “001” 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, KEY00 to KEY31 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. Writing “1010” to KWUPCLR during interrupt processing clears all the key interrupt requests.

(Note) If two or more key inputs are generated, all the key input requests will be cleared by clearing interrupt requests.

20.3 Pull-up Function

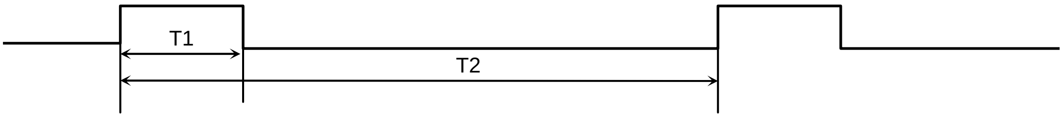
Each key input has the pull-up function and can be programmed by setting the register in the port. When a static pull-up is set, the pull-up function can be used regardless of what is set in KWUPSTn<KEYnEN > (it is controlled by the PxPUP<PExx> bit of each port).

Key-on Wakeup Control

KWUPCNT
(0xFF00_1A84)

	7	6	5	4	3	2	1	0
Bit Symbol			T2S1	T2S0	T1S1	T1S0		
Read/Write	R/W	R	R/W				R	
After reset	0	0	0	0	0	0	0	
Function	Make sure that you write “0.”	This can be read as “0.”	Dynamic pull-up duration 00: 256/fs 10: 1024/fs 01: 512/fs 11: 2048/fs		Dynamic pull-up duration 00: 2/fs 10: 8/fs 01: 4/fs 11: 16/fs		This can be read as “0.”	
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

Dynamic pull-up operation is executed as shown below.



Pull-up is executed only in the T1 period determined by <T1S1:0>. Pull-up is not executed in the remaining period.

00: 2/fs (62.5 µs @fs = 32 kHz)

01: 4/fs (125 µs @fs = 32 kHz)

10: 8/fs (250 µs @fs = 32 kHz)

11: 16/fs (500 µs @fs = 32 kHz)

Dynamic pull-up operation is repeated in the T2 cycle determined by <T2S1:0>.

00: 256/fs (8 ms @fs = 32 kHz)

01: 512/fs (16 ms @fs = 32 kHz)

10: 1024/fs (32 ms @fs = 32 kHz)

11: 2048/fs (64 ms @fs = 32 kHz)

- **fs must be operated while dynamic pull-up is used.**
- **Key input must be started during the second T1 period after enabling dynamic pull-up.**

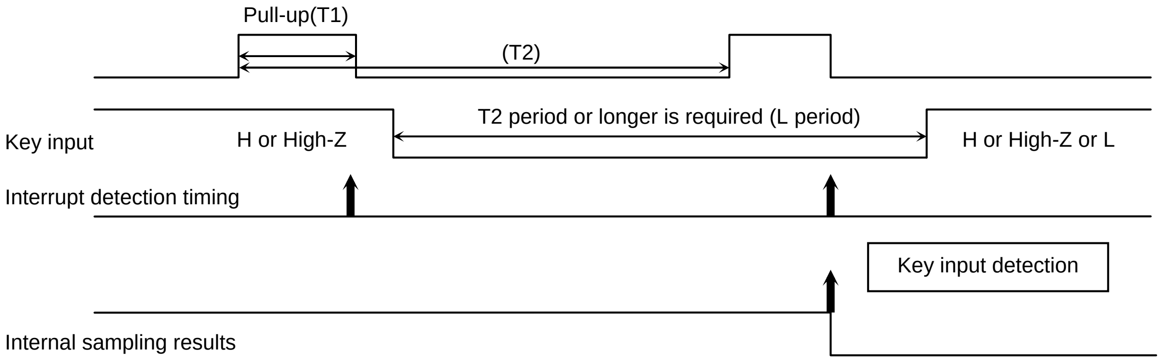
20.4 Key Input Detection Timing

1) When the static pull-up is selected by setting PnPE<PEn> to 1 and KWUPSTn<DPEn> to 0:

   The active state of each key input can be defined to the high or low level or to the rising and/or falling edges by setting KWUPSTn<KEYn2:0>. The active states of key inputs are continuously detected.

2) When the dynamic pull-up is selected by setting PnPE<PEn> to 1 and KWUPSTn<DPEn> to 1:

   Detection of the active state of each key input (interrupt detection) is carried out only at the edge one-clock before fs at the end of the T1 period. Therefore, a key input not shorter than the T2 period is needed. In this case, do not define the active state to the high or low level. There is a delay up to the T2 period before key input detection. The figure below shows an example of defining the active state to the falling edge.


Pull-up(T1)
(T2)
Key input
H or High-Z
T2 period or longer is required (L period)
H or High-Z or L
Interrupt detection timing
Key input detection
Internal sampling results

The external state of port value can be monitored during dynamic pull-up operation by referring to the PKEYn <PKEYn> register.

Sampling is executed in the dynamic pull-up cycle.

PKEY0
(0xFF00_1A80)

	7	6	5	4	3	2	1	0
Bit Symbol	PKEY07	PKEY06	PKEY05	PKEY04	PKEY03	PKEY02	PKEY01	PKEY00
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”
	15	14	13	12	11	10	9	8
Bit Symbol	PKEY15	PKEY14	PKEY13	PKEY12	PKEY11	PKEY10	PKEY09	PKEY08
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”
	23	22	21	20	19	18	17	16
Bit Symbol	PKEY23	PKEY22	PKEY21	PKEY20	PKEY19	PKEY18	PKEY17	PKEY16
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”
	31	30	29	28	27	26	25	24
Bit Symbol	PKEY31	PKEY30	PKEY29	PKEY28	PKEY27	PKEY26	PKEY25	PKEY24
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”	PORT STATE 0:”Lo” 1:”Hi”

KWUPST00
(0xFF00_1A00)

	7	6	5	4	3	2	1	0
bit Symbol	DPE00	KEY002	KEY001	KEY000				KEY00EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY00 active state 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as “0.”			KEY00 interrupt input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

KWUPST31
(0xFF00_1A7C)

	7	6	5	4	3	2	1	0
bit Symbol	DPE31	KEY312	KEY311	KEY310				KEY31EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY31 active state 000: “L” level 001: “H” level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as “0.”			KEY31 interrupt input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

20.5 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 KWUPINTn is set to "1," indicating that an interrupt is generated. The KWUPINTn is the read-only register. Reading this register clears the corresponding bit that has been set to "1" and the interrupt request.

(A clear by KWUPCLR is also possible.)

If the active state is set to the high or low level, the corresponding bit of the KWUPINTn register remains "1" after it is read, unless the external input is withdrawn.

KWUPINT
(0xFF00_1A8C)

	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	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated
	15	14	13	12	11	10	9	8
bit Symbol	KEYINT15	KEYINT14	KEYINT13	KEYINT12	KEYINT11	KEYINT10	KEYINT9	KEYINT8
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated
	23	22	21	20	19	18	17	16
bit Symbol	KEYINT23	KEYINT22	KEYINT21	KEYINT20	KEYINT19	KEYINT18	KEYINT17	KEYINT16
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated
	31	30	29	28	27	26	25	24
bit Symbol	KEYINT31	KEYINT30	KEYINT29	KEYINT28	KEYINT27	KEYINT26	KEYINT25	KEYINT24
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated	Interrupt 0:Not generated 1:Generated

KWUPCLR
(0xFF00_1A88)

	7	6	5	4	3	2	1	0
bit Symbol					KEYCLR3	KEYCLR2	KEYCLR1	KEYCLR0
Read/Write	R				W			
After reset	0							
Function	This can be read as “0.”				Writing “1010” clears all the key factors. This can be read as “0.”			
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0							
Function	This can be read as “0.”							

20.6 Setting example

Cautions on Use of Key Inputs with Pull-up Enabled

A) When you make the first setting after turning the power ON (Example: port E0 with interrupts at both edges)

1) Make a setting of the port.

PECR<PE0C> = “0”	The setting for input pin.
PEFC1<PE0F> = “1”	The function is set to the key.
PEPUP<PEE0> = “1”	Pull-up ON control
PEIE<PIEE0> = “1”	Input enabled

2) Set KWUPST08<KEY08EN> to “0” for the key input to be used.

3) Set KWUPST08<KEY082:KEY080> to “100” to define the active state of the key input to be used.

4) Set KWUPST08<KEY08EN> to “1” for the key input to be used.

5) Wait until the pull-up operation is completed.

6) Set KWUPCLR to “1010” to clear interrupt requests.

7) Program the CG and the INTC by setting IMCGD3<EMCGC2:0> to “001” and IMCGD3<KWUPEN> to “1.”

(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 IMC04<IL122:120> to “000” at the INTC.

2) Set KWUPST08<KEY08EN> to “0” for the key input to be used.

3) Change the active state by setting KWUPST08<KEY082:KEY080> to “000” for the key input to be changed. (Example: Lo level interrupt)

4) Set KWUPST08<KEY08EN> to “1” for the key input to be used.

5) Clear interrupt requests by setting KWUPCLR to “1010.”

6) Enable the key interrupt at the INTC. Set IMC04<IL122:120> to a desired level “xxx.”

C) To enable a key input during operation

1) Disable key interrupts by setting IMC04<IL122:120> to “000” at the INTC.

2) Set KWUPST08＜KEY08EN＞ to “0” for the key input to be used.

3) Define the active state of the key input to be used at the corresponding KWUPST08.

4) Set KWUPST08＜KEY08EN＞ to “1” for the key input to be used.

5) Wait until the pull-up operation is completed.

6) Clear interrupt requests by setting KWUPCLR.

7) Enable key interrupts at the INTC. (Set IMC04<IL122:120> 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) PECR<PE0C> = “0” The setting for input pin.

   PEFC1<PE0F> = “1” The function is set to the key.

   PEPUP<PEE0> = “0” Pull-up OFFcontrol

   PEIE<PIEE0> = “1” Input enabled

2) Set KWUPST08＜KEY08EN＞ to “0” for the key input to be used.

3) Set KWUPST08<KEY082:KEY080> to “000” to define the active state of the key input to be used.

4) Set KWUPST08＜KEY08EN＞ to “1” for the key input to be used.

5) Set KWUPCLR to “1010” to clear interrupt requests.

6) Set KWUPST08＜KEY08EN＞ to “1” for the key input to be used.

7) Program the 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 IMC04<IL122:120> to “000” at the INTC.

2) Set KWUPST08＜KEY08EN＞ to “0” for the key input to be used.

3) Change the active state by setting KWUPSTn for the key input to be changed.

4) Set KWUPST08＜KEY08EN＞ to “1” for the key input to be used.

5) Clear interrupt requests by setting KWUPCLR.

6) Enable key interrupts at the INTC. (Set IMC04<IL122:120> to a desired level.)

C) To enable a key input during operation

1) Disable key interrupts by setting IMC04<IL122:120> to “000” at the INTC.

2) Set KWUPST08＜KEY08EN＞ to “0” for the key input to be used.

3) Define the active state by setting KWUPSTn for the key input to be used.

4) Set KWUPST08＜KEY08EN＞ to “1” for the key input to be used.

5) Clear interrupt requests by setting KWUPCLR.

6) Set KWUPSTn＜KEYnEN＞ to “1” for the key input to be used.

7) Enable key interrupts at the INTC. (Set IMC04<IL122:120> to a desired level.)

21. ROM Correction Function

This chapter describes the ROM correction function built into the TMP19A44.

21.1 Features

- Using this function, twelve pieces of eight-word data can be replaced.
- If an address (lower 5 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.
- If ROM correction cannot be executed using eight-word data due to a program modification or for other reasons, it is possible to place a "jump-to-RAM" instruction in a data register in a RAM area and to correct ROM data in that RAM area.

21.2 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 initialization 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 authority, ROM data is replaced when the value generated by the PC matches that of the address register. If the DMAC has the bus authority, 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<19:5> for reasons of circuitry simplification. Internal processing is that data replacement is executed when the calculation of a logical product is completed 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	0xff00_0000	0xFFFF_FE80 - 0xFFFF_FE9F	8
ADDREG1	0xff00_0004	0xFFFF_FEA0 - 0xFFFF_FEBF	8
ADDREG2	0xff00_0008	0xFFFF_FEC0 - 0xFFFF_FEDF	8
ADDREG3	0xff00_000C	0xFFFF_FEE0 - 0xFFFF_FEFF	8
ADDREG4	0xff00_0010	0xFFFF_FF00 - 0xFFFF_FF1F	8
ADDREG5	0xff00_0014	0xFFFF_FF20 - 0xFFFF_FF3F	8
ADDREG6	0xff00_0018	0xFFFF_FF40 - 0xFFFF_FF5F	8
ADDREG7	0xff00_001C	0xFFFF_FF60 - 0xFFFF_FF7F	8
ADDREG8	0xff00_0020	0xFFFF_FF80 - 0xFFFF_FF9F	8
ADDREG9	0xff00_0024	0xFFFF_FFA0 - 0xFFFF_FFBF	8
ADDREGA	0xff00_0028	0xFFFF_FFC0 - 0xFFFF_FFDF	8
ADDREGB	0xff00_002C	0xFFFF_FFE0 - 0xFFFF_FFFF	8

(Note 1) To use the ROM correction function, the ROM must be unprotected. An instruction to be corrected under ROM protection is replaced by an instruction in RAM. Neither ROM read nor DMAC setting can be executed by the instruction that the ROM correction is applied.

(Note 2) When executing ROM correction to the ROM area, upper address specified in the address register is ignored and the address [19:5] is decoded.

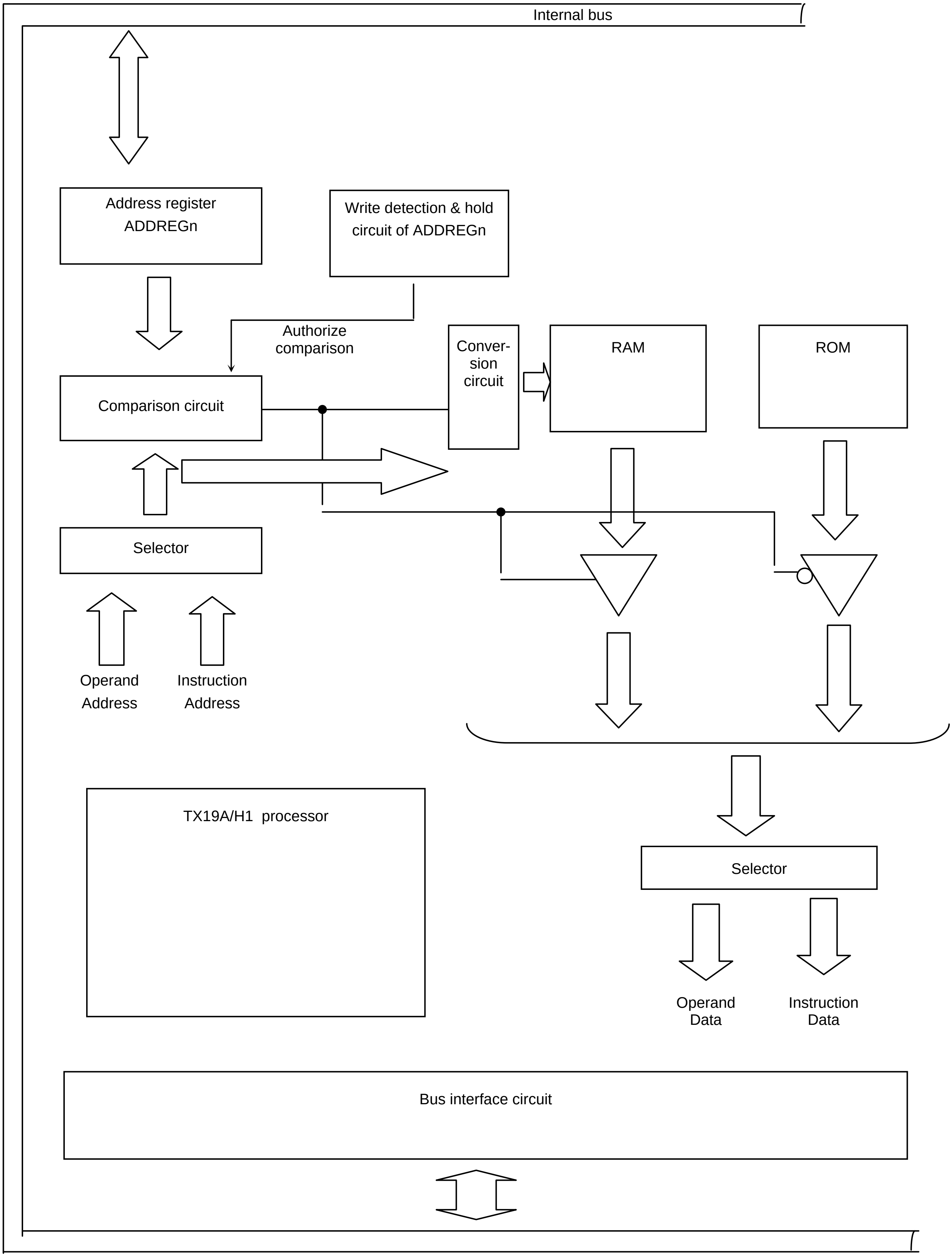


Fig. 21.1 ROM Correction System Diagram

21.3 Registers

(1) Address registers

		7	6	5	4	3	2	1	0
ADDREG0	bit Symbol	ADD07	ADD06	ADD05					ADD00
(0xFF00_0000)	Read/Write	R/W			R				
	After reset	0			0				0
	Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
		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							
	Function	Set the physical address of the ROM area to correct.							
		23	22	21	20	19	18	17	16
	bit Symbol					ADD019	ADD018	ADD017	ADD016
	Read/Write	R				R/W			
	After reset	1		0					
	Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0			1				
	Function	This can be read as “0”.			This can be read as “1”.				

		7	6	5	4	3	2	1	0
ADDREG1	bit Symbol	ADD17	ADD16	ADD15					ADD10
(0xFF00_0004)	Read/Write	R/W			R				R
	After reset	0			0				0
	Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
		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							
	Function	Set the physical address of the ROM area to correct.							
		23	22	21	20	19	18	17	16
	bit Symbol					ADD119	ADD118	ADD117	ADD116
	Read/Write	R				R/W			
	After reset	1		0					
	Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
		31	30	29	28	27	26	25	24
	bit Symbol								
	Read/Write	R							
	After reset	0			1				
	Function	This can be read as “0”.			This can be read as “1”.				

Fig. 21.2 ROM correction registers

ADDREG2
(0xFF00_0008)

	7	6	5	4	3	2	1	0
bit Symbol	ADD27	ADD26	ADD25					ADD20
Read/Write	R/W			R				
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0 : Disable 1 : Enable

	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							
Function	Set the physical address of the ROM area to correct.							

	23	22	21	20	19	18	17	16
bit Symbol					ADD219	ADD218	ADD217	ADD216
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

ADDREG3
(0xFF00_000C)

	7	6	5	4	3	2	1	0
bit Symbol	ADD37	ADD36	ADD35					ADD30
Read/Write	R/W			R				R
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0 : Disable 1 : Enable

	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							
Function	Set the physical address of the ROM area to correct.							

	23	22	21	20	19	18	17	16
bit Symbol					ADD319	ADD318	ADD317	ADD316
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			

	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

Fig. 21.3 ROM correction registers

ADDREG4
(0xFF00_0010)

	7	6	5	4	3	2	1	0
bit Symbol	ADD47	ADD46	ADD45					ADD40
Read/Write	R/W			R				
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADD419	ADD418	ADD417	ADD416
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

ADDREG5
(0xFF00_0014)

	7	6	5	4	3	2	1	0
bit Symbol	ADD57	ADD56	ADD55					ADD50
Read/Write	R/W			R				R
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADD519	ADD518	ADD517	ADD516
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

Fig. 21.4 ROM correction registers

ADDREG6
(0xFF00_0018)

	7	6	5	4	3	2	1	0
bit Symbol	ADD67	ADD66	ADD65					ADD60
Read/Write	R/W			R				
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0 : Disable 1 : Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADD619	ADD618	ADD617	ADD616
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

ADDREG7
(0xFF00_001C)

	7	6	5	4	3	2	1	0
bit Symbol	ADD77	ADD76	ADD75					ADD70
Read/Write	R/W			R				R
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0 : Disable 1 : Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADD719	ADD718	ADD717	ADD716
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

Fig. 21.5 ROM correction registers

ADDREG8
(0xFF00_0020)

	7	6	5	4	3	2	1	0
bit Symbol	ADD87	ADD86	ADD85					ADD80
Read/Write	R/W			R				
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADD819	ADD818	ADD817	ADD816
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

ADDREG9
(0xFF00_0024)

	7	6	5	4	3	2	1	0
bit Symbol	ADD97	ADD96	ADD95					ADD90
Read/Write	R/W			R				R
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADD919	ADD918	ADD917	ADD916
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

Fig. 21.6 ROM correction registers

ADDREGA
(0xFF00_0028)

	7	6	5	4	3	2	1	0
bit Symbol	ADDA7	ADDA6	ADDA5					ADDA0
Read/Write	R/W			R				
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADDA19	ADDA18	ADDA17	ADDA16
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

ADDREGB
(0xFF00_002C)

	7	6	5	4	3	2	1	0
bit Symbol	ADDB7	ADDB6	ADDB5					ADDB0
Read/Write	R/W			R				R
After reset	0			0				0
Function	Set the physical address of the ROM area to correct.			This can be read as “0”.				0：Disable 1：Enable
	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							
Function	Set the physical address of the ROM area to correct.							
	23	22	21	20	19	18	17	16
bit Symbol					ADDB19	ADDB18	ADDB17	ADDB16
Read/Write	R				R/W			
After reset	1		0					
Function	This can be read as “1”.		This can be read as “0”.		Set the physical address of the ROM area to correct.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0			1				
Function	This can be read as “0”.			This can be read as “1”.				

(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.

Fig. 21.7 ROM correction registers

22. Table of Special Function Registers

[1] ROM correction

[2] FLASH control

[3] Protect control

[4] Interrupt controller

[5] DMA controller

[6] Chip select/wait controller

[7] Real time clock

[8] Two-phase pulse input counter

[9] High speed serial channel

[10] Clock generator

[11] Key-on wake-up

[12] Port registers

[13] 16-bit timer

[14] 32-bit timer

[15] I^{2}CBUS/serial channel

[16] UART/serial channel

[17] 10-bit A/D converter

[18] Watchdog timer

Little

[1] ROM correction

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF000000H	ADDREG0	FF000010H	ADDREG4	FF000020H	ADDREG8	FF000030H	
1H	“	1H	“	1H	“	1H	
2H	“	2H	“	2H	“	2H	
3H	“	3H	“	3H	“	3H	
4H	ADDREG1	4H	ADDREG5	4H	ADDREG9	4H	
5H	“	5H	“	5H	“	5H	
6H	“	6H	“	6H	“	6H	
7H	“	7H	“	7H	“	7H	
8H	ADDREG2	8H	ADDREG6	8H	ADDREGA	8H	
9H	“	9H	“	9H	“	9H	
AH	“	AH	“	AH	“	AH	
BH	“	BH	“	BH	“	BH	
CH	ADDREG3	CH	ADDREG7	CH	ADDREGB	CH	
DH	“	DH	“	DH	“	DH	
EH	“	EH	“	EH	“	EH	
FH	“	FH	“	FH	“	FH	

[2] FLASH control [3] Protect control

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF000100H	FLCS	FF000200H	SECBIT	FF000210H		FF000220H	
1H	“	1H	“	1H		1H	
2H	“	2H	“	2H		2H	
3H	“	3H	“	3H		3H	
4H	Reserved	4H	DSUSECBIT	4H		4H	
5H	“	5H	“	5H		5H	
6H	“	6H	“	6H		6H	
7H	“	7H	“	7H		7H	
8H	Reserved	8H	SECCODE	8H		8H	
9H	“	9H	“	9H		9H	
AH	“	AH	“	AH		AH	
BH	“	BH	“	BH		BH	
CH	Reserved	CH	DSUSECCODE	CH		CH	
DH	“	DH	“	DH		DH	
EH	“	EH	“	EH		EH	
FH	“	FH	“	FH		FH	

[4] Interrupt controller

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF001000H	IMC0	FF001010H	IMC4	FF001020H	IMC8	FF001030H	IMCC
1H	“	1H	“	1H	“	1H	“
2H	“	2H	“	2H	“	2H	“
3H	“	3H	“	3H	“	3H	“
4H	IMC1	4H	IMC5	4H	IMC9	4H	IMCD
5H	“	5H	“	5H	“	5H	“
6H	“	6H	“	6H	“	6H	“
7H	“	7H	“	7H	“	7H	“
8H	IMC2	8H	IMC6	8H	IMCA	8H	IMCE
9H	“	9H	“	9H	“	9H	“
AH	“	AH	“	AH	“	AH	“
BH	“	BH	“	BH	“	BH	“
CH	IMC3	CH	IMC7	CH	IMCB	CH	IMCF
DH	“	DH	“	DH	“	DH	“
EH	“	EH	“	EH	“	EH	“
FH	“	FH	“	FH	“	FH	“

ADR	Register name
FF001040H	IMC10
1H	“
2H	“
3H	“
4H	IMC11
5H	“
6H	“
7H	“
8H	IMC12
9H	“
AH	“
BH	“
CH	IMC13
DH	“
EH	“
FH	“

ADR	Register name
FF001050H	IMC14
1H	“
2H	“
3H	“
4H	IMC15
5H	“
6H	“
7H	“
8H	IMC16
9H	“
AH	“
BH	“
CH	IMC17
DH	“
EH	“
FH	“

ADR	Register name
FF001060H	IMC18
1H	“
2H	“
3H	“
4H	IMC19
5H	“
6H	“
7H	“
8H	Reservd
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF001070H	Reservd
1H	“
2H	“
3H	“
4H	Reservd
5H	“
6H	“
7H	“
8H	Reservd
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF001080H	IVR
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001090H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010C0H	INTCLR
1H	“
2H	“
3H	“
4H	DREQFLG
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001100H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	ILEV
DH	“
EH	“
FH	“

ADR	Register name
FF001110H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001120H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001130H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] DMA controller

ADR	Register name
FF001200H	CCR0
1H	“
2H	“
3H	“
4H	CSR0
5H	“
6H	“
7H	“
8H	SAR0
9H	“
AH	“
BH	“
CH	DAR0
DH	“
EH	“
FH	“

ADR	Register name
FF001210H	BCR0
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR0
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001220H	CCR1
1H	“
2H	“
3H	“
4H	CSR1
5H	“
6H	“
7H	“
8H	SAR1
9H	“
AH	“
BH	“
CH	DAR1
DH	“
EH	“
FH	“

ADR	Register name
FF001230H	BCR1
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001240H	CCR2
1H	“
2H	“
3H	“
4H	CSR2
5H	“
6H	“
7H	“
8H	SAR2
9H	“
AH	“
BH	“
CH	DAR2
DH	“
EH	“
FH	“

ADR	Register name
FF001250H	BCR2
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR2
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001260H	CCR3
1H	“
2H	“
3H	“
4H	CSR3
5H	“
6H	“
7H	“
8H	SAR3
9H	“
AH	“
BH	“
CH	DAR3
DH	“
EH	“
FH	“

ADR	Register name
FF001270H	BCR3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR3
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001280H	CCR4
1H	“
2H	“
3H	“
4H	CSR4
5H	“
6H	“
7H	“
8H	SAR4
9H	“
AH	“
BH	“
CH	DAR4
DH	“
EH	“
FH	“

ADR	Register name
FF001290H	BCR4
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR4
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0012A0H	CCR5
1H	“
2H	“
3H	“
4H	CSR5
5H	“
6H	“
7H	“
8H	SAR5
9H	“
AH	“
BH	“
CH	DAR5
DH	“
EH	“
FH	“

ADR	Register name
FF0012B0H	BCR5
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0012C0H	CCR6
1H	“
2H	“
3H	“
4H	CSR6
5H	“
6H	“
7H	“
8H	SAR6
9H	“
AH	“
BH	“
CH	DAR6
DH	“
EH	“
FH	“

ADR	Register name
FF0012D0H	BCR6
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR6
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0012E0H	CCR7
1H	“
2H	“
3H	“
4H	CSR7
5H	“
6H	“
7H	“
8H	SAR7
9H	“
AH	“
BH	“
CH	DAR7
DH	“
EH	“
FH	“

ADR	Register name
FF0012F0H	BCR7
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR7
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

[6] Chip select/wait controller

ADR	Register name
FF001300H	DCR
1H	“
2H	“
3H	“
4H	RSR
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	DHR
DH	“
EH	“
FH	“

ADR	Register name
FF001400H	BMA0
1H	“
2H	“
3H	“
4H	BMA1
5H	“
6H	“
7H	“
8H	BMA2
9H	“
AH	“
BH	“
CH	BMA3
DH	“
EH	“
FH	“

ADR	Register name
FF001410H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001420H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001480H	B01CS
1H	“
2H	“
3H	“
4H	B23CS
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	BUSCR
DH	“
EH	“
FH	“

[7] Real time clock

ADR	Register name
FF001500H	SECR
1H	MINR
2H	HOURR
3H	“
4H	DAYR
5H	DATER
6H	MONTHR
7H	YEARR
8H	PAGER
9H	“
AH	“
BH	“
CH	RESTR
DH	“
EH	“
FH	“

ADR	Register name
FF001510H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001520H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[8] Two-phase pulse input counter

ADR	Register name
FF001600H	PHC0RUN
1H	“
2H	“
3H	“
4H	PHC0CR
5H	“
6H	“
7H	“
8H	PHC0EN
9H	“
AH	“
BH	“
CH	PHC0FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001610H	PHC0CMP0
1H	“
2H	“
3H	“
4H	PHC0CMP1
5H	“
6H	“
7H	“
8H	PHC0CNT
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF001620H	Reserved
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001630H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001640H	PHC1RUN
1H	“
2H	“
3H	“
4H	PHC1CR
5H	“
6H	“
7H	“
8H	PHC1EN
9H	“
AH	“
BH	“
CH	PHC1FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001650H	PHC1CMP0
1H	“
2H	“
3H	“
4H	PHC1CMP1
5H	“
6H	“
7H	“
8H	PHC1CNT
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF001660H	Reservd
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001670H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001680H	PHC2RUN
1H	“
2H	“
3H	“
4H	PHC2CR
5H	“
6H	“
7H	“
8H	PHC2EN
9H	“
AH	“
BH	“
CH	PHC2FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001690H	PHC2CMP0
1H	“
2H	“
3H	“
4H	PHC2CMP1
5H	“
6H	“
7H	“
8H	PHC2CNT
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF0016A0H	Reservd
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0016B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0016C0H	PHC3RUN
1H	“
2H	“
3H	“
4H	PHC3CR
5H	“
6H	“
7H	“
8H	PHC3EN
9H	“
AH	“
BH	“
CH	PHC3FLG
DH	“
EH	“
FH	“

ADR	Register name
FF0016D0H	PHC3CMP0
1H	“
2H	“
3H	“
4H	PHC3CMP1
5H	“
6H	“
7H	“
8H	PHC3CNT
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF0016E0H	Reservd
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0016F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001700H	PHC4RUN
1H	“
2H	“
3H	“
4H	PHC4CR
5H	“
6H	“
7H	“
8H	PHC4EN
9H	“
AH	“
BH	“
CH	PHC4FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001710H	PHC4CMP0
1H	“
2H	“
3H	“
4H	PHC4CMP1
5H	“
6H	“
7H	“
8H	PHC4CNT
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF001720H	Reservd
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001730H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001740H	PHC5RUN
1H	“
2H	“
3H	“
4H	PHC5CR
5H	“
6H	“
7H	“
8H	PHC5EN
9H	“
AH	“
BH	“
CH	PHC5FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001750H	PHC5CMP0
1H	“
2H	“
3H	“
4H	PHC5CMP1
5H	“
6H	“
7H	“
8H	PHC5CNT
9H	“
AH	“
BH	“
CH	Reservd
DH	“
EH	“
FH	“

ADR	Register name
FF001760H	Reservd
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001770H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] High speed serial channel

ADR	Register name
FF001800H	HSC0BUF
1H	
2H	
3H	
4H	HBR0ADD
5H	HSC0MOD1
6H	HSC0MOD2
7H	HSC0EN
8H	HSC0RFC
9H	HSC0TFC
AH	HSC0RST
BH	HSC0TST
CH	HSC0FCNF
DH	HSC0CR
EH	HSC0MOD0
FH	HBR0CR

ADR	Register name
FF001810H	HSC1BUF
1H	
2H	
3H	
4H	HBR1ADD
5H	HSC1MOD1
6H	HSC1MOD2
7H	HSC1EN
8H	HSC1RFC
9H	HSC1TFC
AH	HSC1RST
BH	HSC1TST
CH	HSC1FCNF
DH	HSC1CR
EH	HSC1MOD0
FH	HBR1CR

ADR	Register name
FF001820H	HSC2BUF
1H	
2H	
3H	
4H	HBR2ADD
5H	HSC2MOD1
6H	HSC2MOD2
7H	HSC2EN
8H	HSC2RFC
9H	HSC2TFC
AH	HSC2RST
BH	HSC2TST
CH	HSC2FCNF
DH	HSC2CR
EH	HSC2MOD0
FH	HBR2CR

ADR	Register name
FF001830H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] Clock generator

ADR	Register name
FF001900H	SYSCR
1H	“
2H	“
3H	“
4H	OSCCR
5H	“
6H	“
7H	“
8H	STBYCR
9H	“
AH	“
BH	“
CH	PLLSEL
DH	“
EH	“
FH	“

ADR	Register name
FF001910H	SCKSEL
1H	“
2H	“
3H	“
4H	ICRCG
5H	“
6H	“
7H	“
8H	NMIFLG
9H	“
AH	“
BH	“
CH	RSTFLG
DH	“
EH	“
FH	“

ADR	Register name
FF001920H	IMCGA
1H	“
2H	“
3H	“
4H	IMCGB
5H	“
6H	“
7H	“
8H	IMCGC
9H	“
AH	“
BH	“
CH	IMCGD
DH	“
EH	“
FH	“

ADR	Register name
FF001930H	IMCGE
1H	“
2H	“
3H	“
4H	IMCGF
5H	“
6H	“
7H	“
8H	IMCG10
9H	“
AH	“
BH	“
CH	IMCG11
DH	“
EH	“
FH	“

[11] Key-on wake-up

ADR	Register name
FF001A00H	KWUPST00
1H	“
2H	“
3H	“
4H	KWUPST01
5H	“
6H	“
7H	“
8H	KWUPST02
9H	“
AH	“
BH	“
CH	KWUPST03
DH	“
EH	“
FH	“

ADR	Register name
FF001A10H	KWUPST04
1H	“
2H	“
3H	“
4H	KWUPST05
5H	“
6H	“
7H	“
8H	KWUPST06
9H	“
AH	“
BH	“
CH	KWUPST07
DH	“
EH	“
FH	“

ADR	Register name
FF001A20H	KWUPST08
1H	“
2H	“
3H	“
4H	KWUPST09
5H	“
6H	“
7H	“
8H	KWUPST10
9H	“
AH	“
BH	“
CH	KWUPST11
DH	“
EH	“
FH	“

ADR	Register name
FF001A30H	KWUPST12
1H	“
2H	“
3H	“
4H	KWUPST13
5H	“
6H	“
7H	“
8H	KWUPST14
9H	“
AH	“
BH	“
CH	KWUPST15
DH	“
EH	“
FH	“

ADR	Register name
FF001A40H	KWUPST 16
1H	“
2H	“
3H	“
4H	KWUPST 17
5H	“
6H	“
7H	“
8H	KWUPST 18
9H	“
AH	“
BH	“
CH	KWUPST 19
DH	“
EH	“
FH	“

ADR	Register name
FF001A50H	KWUPST 20
1H	“
2H	“
3H	“
4H	KWUPST 21
5H	“
6H	“
7H	“
8H	KWUPST 22
9H	“
AH	“
BH	“
CH	KWUPST 23
DH	“
EH	“
FH	“

ADR	Register name
FF001A60H	KWUPST 24
1H	“
2H	“
3H	“
4H	KWUPST 25
5H	“
6H	“
7H	“
8H	KWUPST 26
9H	“
AH	“
BH	“
CH	KWUPST 27
DH	“
EH	“
FH	“

ADR	Register name
FF001A70H	KWUPST 28
1H	“
2H	“
3H	“
4H	KWUPST 29
5H	“
6H	“
7H	“
8H	KWUPST 30
9H	“
AH	“
BH	“
CH	KWUPST 31
DH	“
EH	“
FH	“

ADR	Register name
FF001A80H	PKEY
1H	“
2H	“
3H	“
4H	KWUPCNT
5H	“
6H	“
7H	“
8H	KWUPCLR
9H	“
AH	“
BH	“
CH	KWUPINT
DH	“
EH	“
FH	“

ADR	Register name
FF001A90H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001AA0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001AB0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] Port registers

ADR	Register name
FF004000H	P0
1H	“
2H	“
3H	“
4H	P0CR
5H	“
6H	“
7H	“
8H	P0FC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004010H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004020H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P0PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004030H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004040H	P1
1H	“
2H	“
3H	“
4H	P1CR
5H	“
6H	“
7H	“
8H	P1FC1
9H	“
AH	“
BH	“
CH	P1FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004050H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004060H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P1PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004080H	P2
1H	“
2H	“
3H	“
4H	P2CR
5H	“
6H	“
7H	“
8H	P2FC1
9H	“
AH	“
BH	“
CH	P2FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004090H	P2FC3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0040A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P2PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0040B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P2IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0040C0H	P3
1H	“
2H	“
3H	“
4H	P3CR
5H	“
6H	“
7H	“
8H	P3FC1
9H	“
AH	“
BH	“
CH	P3FC2
DH	“
EH	“
FH	“

ADR	Register name
FF0040D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0040E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P3PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0040F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P3IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004100H	P4
1H	“
2H	“
3H	“
4H	P4CR
5H	“
6H	“
7H	“
8H	P4FC1
9H	“
AH	“
BH	“
CH	P4FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004110H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004120H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P4PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004130H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P4IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004140H	P5
1H	“
2H	“
3H	“
4H	P5CR
5H	“
6H	“
7H	“
8H	P5FC1
9H	“
AH	“
BH	“
CH	P5FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004150H	P5FC3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004160H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P5PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004170H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P5IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004180H	P6
1H	“
2H	“
3H	“
4H	P6CR
5H	“
6H	“
7H	“
8H	P6FC1
9H	“
AH	“
BH	“
CH	P6FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004190H	P6FC3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0041A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P6ODE
9H	“
AH	“
BH	“
CH	P6PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0041B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P6IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0041C0H	P7
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P7FC2
DH	“
EH	“
FH	“

ADR	Register name
FF0041D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0041E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P7PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0041F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P7IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004200H	P8
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P8FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004210H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004220H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P8PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004230H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P8IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004240H	P9
1H	“
2H	“
3H	“
4H	P9CR
5H	“
6H	“
7H	“
8H	P9FC1
9H	“
AH	“
BH	“
CH	P9FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004250H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004260H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P9ODE
9H	“
AH	“
BH	“
CH	P9PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004270H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P9IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004280H	PA
1H	“
2H	“
3H	“
4H	PACR
5H	“
6H	“
7H	“
8H	PAFC1
9H	“
AH	“
BH	“
CH	PAFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004290H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0042A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PAPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0042B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PAIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0042C0H	PB
1H	“
2H	“
3H	“
4H	PBCR
5H	“
6H	“
7H	“
8H	PBFC1
9H	“
AH	“
BH	“
CH	PBFC2
DH	“
EH	“
FH	“

ADR	Register name
FF0042D0H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0042E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PBODE
9H	“
AH	“
BH	“
CH	PBPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0042F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PBIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004300H	PC
1H	“
2H	“
3H	“
4H	PCCR
5H	“
6H	“
7H	“
8H	PCFC1
9H	“
AH	“
BH	“
CH	PCFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004310H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004320H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PCODE
9H	“
AH	“
BH	“
CH	PCPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004330H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PCIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004340H	PD
1H	“
2H	“
3H	“
4H	PDCR
5H	“
6H	“
7H	“
8H	PDFC1
9H	“
AH	“
BH	“
CH	PDFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004350H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004360H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PDODE
9H	“
AH	“
BH	“
CH	PDPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004370H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PDIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004380H	PE
1H	“
2H	“
3H	“
4H	PECR
5H	“
6H	“
7H	“
8H	PEFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004390H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0043A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PEPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0043B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PEIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0043C0H	PF
1H	“
2H	“
3H	“
4H	PFCR
5H	“
6H	“
7H	“
8H	PFFC1
9H	“
AH	“
BH	“
CH	PFFC2
DH	“
EH	“
FH	“

ADR	Register name
FF0043D0H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0043E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PFPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0043F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PFIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004400H	PG
1H	“
2H	“
3H	“
4H	PGCR
5H	“
6H	“
7H	“
8H	PGFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004410H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004420H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PGPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004430H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PGIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004440H	PH
1H	“
2H	“
3H	“
4H	PHCR
5H	“
6H	“
7H	“
8H	PHFC1
9H	“
AH	“
BH	“
CH	PHFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004450H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004460H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PHPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004470H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PHIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004480H	PI
1H	“
2H	“
3H	“
4H	PICR
5H	“
6H	“
7H	“
8H	PIFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004490H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0044A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PIPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0044B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PIIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0044C0H	PJ
1H	“
2H	“
3H	“
4H	PJCR
5H	“
6H	“
7H	“
8H	PJFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0044D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0044E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PJPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0044F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PJIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

[13] 16-bit timer

ADR	Register name
FF004500H	TB0EN
1H	“
2H	“
3H	“
4H	TB0RUN
5H	“
6H	“
7H	“
8H	TB0CR
9H	“
AH	“
BH	“
CH	TB0MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004510H	TB0FFCR
1H	“
2H	“
3H	“
4H	TB0ST
5H	“
6H	“
7H	“
8H	TB0IM
9H	“
AH	“
BH	“
CH	TM0UC
DH	“
EH	“
FH	“

ADR	Register name
FF004520H	TB0RG0
1H	“
2H	“
3H	“
4H	TB0RG1
5H	“
6H	“
7H	“
8H	TB0CP0
9H	“
AH	“
BH	“
CH	TB0CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004530H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004540H	TB1EN
1H	“
2H	“
3H	“
4H	TB1RUN
5H	“
6H	“
7H	“
8H	TB1CR
9H	“
AH	“
BH	“
CH	TB1MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004550H	TB1FFCR
1H	“
2H	“
3H	“
4H	TB1ST
5H	“
6H	“
7H	“
8H	TB1IM
9H	“
AH	“
BH	“
CH	TM1UC
DH	“
EH	“
FH	“

ADR	Register name
FF004560H	TB1RG0
1H	“
2H	“
3H	“
4H	TB1RG1
5H	“
6H	“
7H	“
8H	TB1CP0
9H	“
AH	“
BH	“
CH	TB1CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004570H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004580H	TB2EN
1H	“
2H	“
3H	“
4H	TB2RUN
5H	“
6H	“
7H	“
8H	TB2CR
9H	“
AH	“
BH	“
CH	TB2MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004590H	TB2FFCR
1H	“
2H	“
3H	“
4H	TB2ST
5H	“
6H	“
7H	“
8H	TB2IM
9H	“
AH	“
BH	“
CH	TM2UC
DH	“
EH	“
FH	“

ADR	Register name
FF0045A0H	TB2RG0
1H	“
2H	“
3H	“
4H	TB2RG1
5H	“
6H	“
7H	“
8H	TB2CP0
9H	“
AH	“
BH	“
CH	TB2CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0045B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0045C0H	TB3EN
1H	“
2H	“
3H	“
4H	TB3RUN
5H	“
6H	“
7H	“
8H	TB3CR
9H	“
AH	“
BH	“
CH	TB3MOD
DH	“
EH	“
FH	“

ADR	Register name
FF0045D0H	TB3FFCR
1H	“
2H	“
3H	“
4H	TB3ST
5H	“
6H	“
7H	“
8H	TB3IM
9H	“
AH	“
BH	“
CH	TM3UC
DH	“
EH	“
FH	“

ADR	Register name
FF0045E0H	TB3RG0
1H	“
2H	“
3H	“
4H	TB3RG1
5H	“
6H	“
7H	“
8H	TB3CP0
9H	“
AH	“
BH	“
CH	TB3CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0045F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004600H	TB4EN
1H	“
2H	“
3H	“
4H	TB4RUN
5H	“
6H	“
7H	“
8H	TB4CR
9H	“
AH	“
BH	“
CH	TB4MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004610H	TB4FFCR
1H	“
2H	“
3H	“
4H	TB4ST
5H	“
6H	“
7H	“
8H	TB4IM
9H	“
AH	“
BH	“
CH	TM4UC
DH	“
EH	“
FH	“

ADR	Register name
FF004620H	TB4RG0
1H	“
2H	“
3H	“
4H	TB4RG1
5H	“
6H	“
7H	“
8H	TB4CP0
9H	“
AH	“
BH	“
CH	TB4CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004630H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004640H	TB5EN
1H	“
2H	“
3H	“
4H	TB5RUN
5H	“
6H	“
7H	“
8H	TB5CR
9H	“
AH	“
BH	“
CH	TB5MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004650H	TB5FFCR
1H	“
2H	“
3H	“
4H	TB5ST
5H	“
6H	“
7H	“
8H	TB5IM
9H	“
AH	“
BH	“
CH	TM5UC
DH	“
EH	“
FH	“

ADR	Register name
FF004660H	TB5RG0
1H	“
2H	“
3H	“
4H	TB5RG1
5H	“
6H	“
7H	“
8H	TB5CP0
9H	“
AH	“
BH	“
CH	TB5CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004670H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004680H	TB6EN
1H	“
2H	“
3H	“
4H	TB6RUN
5H	“
6H	“
7H	“
8H	TB6CR
9H	“
AH	“
BH	“
CH	TB6MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004690H	TB6FFCR
1H	“
2H	“
3H	“
4H	TB6ST
5H	“
6H	“
7H	“
8H	TB6IM
9H	“
AH	“
BH	“
CH	TM6UC
DH	“
EH	“
FH	“

ADR	Register name
FF0046A0H	TB6RG0
1H	“
2H	“
3H	“
4H	TB6RG1
5H	“
6H	“
7H	“
8H	TB6CP0
9H	“
AH	“
BH	“
CH	TB6CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0046B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0046C0H	TB7EN
1H	“
2H	“
3H	“
4H	TB7RUN
5H	“
6H	“
7H	“
8H	TB7CR
9H	“
AH	“
BH	“
CH	TB7MOD
DH	“
EH	“
FH	“

ADR	Register name
FF0046D0H	TB7FFCR
1H	“
2H	“
3H	“
4H	TB7ST
5H	“
6H	“
7H	“
8H	TB7IM
9H	“
AH	“
BH	“
CH	TM7UC
DH	“
EH	“
FH	“

ADR	Register name
FF0046E0H	TB7RG0
1H	“
2H	“
3H	“
4H	TB7RG1
5H	“
6H	“
7H	“
8H	TB7CP0
9H	“
AH	“
BH	“
CH	TB7CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0046F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004700H	TB8EN
1H	“
2H	“
3H	“
4H	TB8RUN
5H	“
6H	“
7H	“
8H	TB8CR
9H	“
AH	“
BH	“
CH	TB8MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004710H	TB8FFCR
1H	“
2H	“
3H	“
4H	TB8ST
5H	“
6H	“
7H	“
8H	TB8IM
9H	“
AH	“
BH	“
CH	TM8UC
DH	“
EH	“
FH	“

ADR	Register name
FF004720H	TB8RG0
1H	“
2H	“
3H	“
4H	TB8RG1
5H	“
6H	“
7H	“
8H	TB8CP0
9H	“
AH	“
BH	“
CH	TB8CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004730H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004740H	TB9EN
1H	“
2H	“
3H	“
4H	TB9RUN
5H	“
6H	“
7H	“
8H	TB9CR
9H	“
AH	“
BH	“
CH	TB9MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004785H	TB9FFCR
1H	“
2H	“
3H	“
4H	TB9ST
5H	“
6H	“
7H	“
8H	TB9IM
9H	“
AH	“
BH	“
CH	TM9UC
DH	“
EH	“
FH	“

ADR	Register name
FF004760H	TB9RG0
1H	“
2H	“
3H	“
4H	TB9RG1
5H	“
6H	“
7H	“
8H	TB9CP0
9H	“
AH	“
BH	“
CH	TB9CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004770H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004780H	TBAEN
1H	“
2H	“
3H	“
4H	TBARUN
5H	“
6H	“
7H	“
8H	TBACR
9H	“
AH	“
BH	“
CH	TBAMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004790H	TBAFFCR
1H	“
2H	“
3H	“
4H	TBAST
5H	“
6H	“
7H	“
8H	TBAIM
9H	“
AH	“
BH	“
CH	TMAUC
DH	“
EH	“
FH	“

ADR	Register name
FF0047A0H	TBARG0
1H	“
2H	“
3H	“
4H	TBARG1
5H	“
6H	“
7H	“
8H	TBACP0
9H	“
AH	“
BH	“
CH	TBACP1
DH	“
EH	“
FH	“

ADR	Register name
FF0047B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0047C0H	TBBEN
1H	“
2H	“
3H	“
4H	TBBRUN
5H	“
6H	“
7H	“
8H	TBBCR
9H	“
AH	“
BH	“
CH	TBBMOD
DH	“
EH	“
FH	“

ADR	Register name
FF0047D0H	TBBFFCR
1H	“
2H	“
3H	“
4H	TBBST
5H	“
6H	“
7H	“
8H	TBBIM
9H	“
AH	“
BH	“
CH	TMBUC
DH	“
EH	“
FH	“

ADR	Register name
FF0047E0H	TBBRG0
1H	“
2H	“
3H	“
4H	TBBRG1
5H	“
6H	“
7H	“
8H	TBBCP0
9H	“
AH	“
BH	“
CH	TBBCP1
DH	“
EH	“
FH	“

ADR	Register name
FF0047F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004800H	TBCEN
1H	“
2H	“
3H	“
4H	TBCRUN
5H	“
6H	“
7H	“
8H	TBCCR
9H	“
AH	“
BH	“
CH	TBCMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004810H	TBCFFCR
1H	“
2H	“
3H	“
4H	TBCST
5H	“
6H	“
7H	“
8H	TBCIM
9H	“
AH	“
BH	“
CH	TMCUC
DH	“
EH	“
FH	“

ADR	Register name
FF004820H	TBCRG0
1H	“
2H	“
3H	“
4H	TBCRG1
5H	“
6H	“
7H	“
8H	TBCCP0
9H	“
AH	“
BH	“
CH	TBCCP1
DH	“
EH	“
FH	“

ADR	Register name
FF004830H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004840H	TBDEN
1H	“
2H	“
3H	“
4H	TBDRUN
5H	“
6H	“
7H	“
8H	TBDCR
9H	“
AH	“
BH	“
CH	TBDMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004850H	TBDFFCR
1H	“
2H	“
3H	“
4H	TBDST
5H	“
6H	“
7H	“
8H	TBDIM
9H	“
AH	“
BH	“
CH	TMDUC
DH	“
EH	“
FH	“

ADR	Register name
FF004860H	TBDRG0
1H	“
2H	“
3H	“
4H	TBDRG1
5H	“
6H	“
7H	“
8H	TBDCP0
9H	“
AH	“
BH	“
CH	TBDCP1
DH	“
EH	“
FH	“

ADR	Register name
FF004870H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004880H	TBEEN
1H	“
2H	“
3H	“
4H	TBERUN
5H	“
6H	“
7H	“
8H	TBECR
9H	“
AH	“
BH	“
CH	TBEMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004890H	TBEFFCR
1H	“
2H	“
3H	“
4H	TBEST
5H	“
6H	“
7H	“
8H	TBEIM
9H	“
AH	“
BH	“
CH	TMEUC
DH	“
EH	“
FH	“

ADR	Register name
FF0048A0H	TBERG0
1H	“
2H	“
3H	“
4H	TBERG1
5H	“
6H	“
7H	“
8H	TBECP0
9H	“
AH	“
BH	“
CH	TBECP1
DH	“
EH	“
FH	“

ADR	Register name
FF0048B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0048C0H	TBFEN
1H	“
2H	“
3H	“
4H	TBFRUN
5H	“
6H	“
7H	“
8H	TBFCR
9H	“
AH	“
BH	“
CH	TBFMOD
DH	“
EH	“
FH	“

ADR	Register name
FF0048D0H	TBFFFCR
1H	“
2H	“
3H	“
4H	TBFST
5H	“
6H	“
7H	“
8H	TBFIM
9H	“
AH	“
BH	“
CH	TMFUC
DH	“
EH	“
FH	“

ADR	Register name
FF0048E0H	TBFRG0
1H	“
2H	“
3H	“
4H	TBFRG1
5H	“
6H	“
7H	“
8H	TBFCP0
9H	“
AH	“
BH	“
CH	TBFCP1
DH	“
EH	“
FH	“

ADR	Register name
FF0048F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004900H	TB10EN
1H	“
2H	“
3H	“
4H	TB10RUN
5H	“
6H	“
7H	“
8H	TB10CR
9H	“
AH	“
BH	“
CH	TB10MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004910H	TB10FFCR
1H	“
2H	“
3H	“
4H	TB10ST
5H	“
6H	“
7H	“
8H	TB10IM
9H	“
AH	“
BH	“
CH	TM10UC
DH	“
EH	“
FH	“

ADR	Register name
FF004920H	TB10RG0
1H	“
2H	“
3H	“
4H	TB10RG1
5H	“
6H	“
7H	“
8H	TB10CP0
9H	“
AH	“
BH	“
CH	TB10CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004930H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004940H	TB11EN
1H	“
2H	“
3H	“
4H	TB11RUN
5H	“
6H	“
7H	“
8H	TB11CR
9H	“
AH	“
BH	“
CH	TB11MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004950H	TB11FFCR
1H	“
2H	“
3H	“
4H	TB11ST
5H	“
6H	“
7H	“
8H	TB11IM
9H	“
AH	“
BH	“
CH	TM11UC
DH	“
EH	“
FH	“

ADR	Register name
FF004960H	TB11RG0
1H	“
2H	“
3H	“
4H	TB11RG1
5H	“
6H	“
7H	“
8H	TB11CP0
9H	“
AH	“
BH	“
CH	TB11CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004970H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] 32-bit timer

ADR	Register name
FF004A00H	TCEN
1H	“
2H	“
3H	“
4H	TBTRUN
5H	“
6H	“
7H	“
8H	TBTCR
9H	“
AH	“
BH	“
CH	TBTCAP
DH	“
EH	“
FH	“

ADR	Register name
FF004A10H	TBTRDCAP
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A20H	CMPCTL0
1H	“
2H	“
3H	“
4H	TCCMP0
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A30H	CMPCTL1
1H	“
2H	“
3H	“
4H	TCCMP1
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A40H	CMPCTL2
1H	“
2H	“
3H	“
4H	TCCMP2
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A50H	CMPCTL3
1H	“
2H	“
3H	“
4H	TCCMP3
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A60H	CMPCTL4
1H	“
2H	“
3H	“
4H	TCCMP4
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A70H	CMPCTL5
1H	“
2H	“
3H	“
4H	TCCMP5
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A80H	CMPCTL6
1H	“
2H	“
3H	“
4H	TCCMP6
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A90H	CMPCTL7
1H	“
2H	“
3H	“
4H	TCCMP7
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AA0H	CAPCR0
1H	“
2H	“
3H	“
4H	TCCAP0
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AB0H	CAPCR1
1H	“
2H	“
3H	“
4H	TCCAP1
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AC0H	CAPCR2
1H	“
2H	“
3H	“
4H	TCCAP2
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AD0H	CAPCR3
1H	“
2H	“
3H	“
4H	TCCAP3
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AE0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AF0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] I^{2}CBUS/serial channel

ADR	Register name
FF004B00H	SBICR0
1H	“
2H	“
3H	“
4H	SBICR1
5H	“
6H	“
7H	“
8H	SBIDBR
9H	“
AH	“
BH	“
CH	I2CAR
DH	“
EH	“
FH	“

ADR	Register name
FF004B10H	SBICR2/SBISR
1H	“
2H	“
3H	“
4H	SBIBR
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004B20H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004B30H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] UART/serial channel

ADR	Register name
FF004C00H	SC0EN
1H	“
2H	“
3H	“
4H	SC0BUF
5H	“
6H	“
7H	“
8H	SC0CR
9H	“
AH	“
BH	“
CH	SC0MOD0
DH	“
EH	“
FH	“

ADR	Register name
FF004C10H	BR0CR
1H	“
2H	“
3H	“
4H	BR0ADD
5H	“
6H	“
7H	“
8H	SC0MOD1
9H	“
AH	“
BH	“
CH	SC0MOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004C20H	SC0RFC
1H	“
2H	“
3H	“
4H	SC0TFC
5H	“
6H	“
7H	“
8H	SC0RST
9H	“
AH	“
BH	“
CH	SC0TST
DH	“
EH	“
FH	“

ADR	Register name
FF004C30H	SC0FCNF
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004C40H	SC1EN
1H	“
2H	“
3H	“
4H	SC1BUF
5H	“
6H	“
7H	“
8H	SC1CR
9H	“
AH	“
BH	“
CH	SC1MOD0
DH	“
EH	“
FH	“

ADR	Register name
FF004C50H	BR1CR
1H	“
2H	“
3H	“
4H	BR1ADD
5H	“
6H	“
7H	“
8H	SC1MOD1
9H	“
AH	“
BH	“
CH	SC1MOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004C60H	SC1RFC
1H	“
2H	“
3H	“
4H	SC1TFC
5H	“
6H	“
7H	“
8H	SC1RST
9H	“
AH	“
BH	“
CH	SC1TST
DH	“
EH	“
FH	“

ADR	Register name
FF004C70H	SC1FCNF
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004C80H	SC2EN
1H	“
2H	“
3H	“
4H	SC2BUF
5H	“
6H	“
7H	“
8H	SC2CR
9H	“
AH	“
BH	“
CH	SC2MOD0
DH	“
EH	“
FH	“

ADR	Register name
FF004C90H	BR2CR
1H	“
2H	“
3H	“
4H	BR2ADD
5H	“
6H	“
7H	“
8H	SC2MOD1
9H	“
AH	“
BH	“
CH	SC2MOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004CA0H	SC2RFC
1H	“
2H	“
3H	“
4H	SC2TFC
5H	“
6H	“
7H	“
8H	SC2RST
9H	“
AH	“
BH	“
CH	SC2TST
DH	“
EH	“
FH	“

ADR	Register name
FF004CB0H	SC2FCNF
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] 10-bit A/D converter

ADR	Register name
FF004D00H	ADACLK
1H	“
2H	“
3H	“
4H	ADAMOD0
5H	“
6H	“
7H	“
8H	ADAMOD1
9H	“
AH	“
BH	“
CH	ADAMOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004D10H	ADAMOD3
1H	“
2H	“
3H	“
4H	ADAMOD4
5H	“
6H	“
7H	“
8H	ADAMOD5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004D20H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004D30H	ADAREG0
1H	“
2H	“
3H	“
4H	ADAREG1
5H	“
6H	“
7H	“
8H	ADAREG2
9H	“
AH	“
BH	“
CH	ADAREG3
DH	“
EH	“
FH	“

ADR	Register name
FF004D40H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	reserved
DH	“
EH	“
FH	“

ADR	Register name
FF004D50H	ADAREGSP
1H	“
2H	“
3H	“
4H	ADACOMREG0
5H	“
6H	“
7H	“
8H	ADACOMREG1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004D60H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004D70H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004D80H	ADBCLK
1H	“
2H	“
3H	“
4H	ADBMOD0
5H	“
6H	“
7H	“
8H	ADBMOD1
9H	“
AH	“
BH	“
CH	ADBMOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004D90H	ADBMOD3
1H	“
2H	“
3H	“
4H	ADBMOD4
5H	“
6H	“
7H	“
8H	ADBMOD5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004DA0H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004DB0H	ADBREG0
1H	“
2H	“
3H	“
4H	ADBREG1
5H	“
6H	“
7H	“
8H	ADBREG2
9H	“
AH	“
BH	“
CH	ADBREG3
DH	“
EH	“
FH	“

ADR	Register name
FF004DC0H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	reserved
DH	“
EH	“
FH	“

ADR	Register name
FF004DD0H	ADBREGSP
1H	“
2H	“
3H	“
4H	ADBCOMREG0
5H	“
6H	“
7H	“
8H	ADBCOMREG1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004DE0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004DF0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004E00H	ADCCLK
1H	“
2H	“
3H	“
4H	ADCMOD0
5H	“
6H	“
7H	“
8H	ADCMOD1
9H	“
AH	“
BH	“
CH	ADCMOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004E10H	ADCMOD3
1H	“
2H	“
3H	“
4H	ADCMOD4
5H	“
6H	“
7H	“
8H	ADCMOD5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004E20H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004E30H	ADCREG0
1H	“
2H	“
3H	“
4H	ADCREG1
5H	“
6H	“
7H	“
8H	ADCREG2
9H	“
AH	“
BH	“
CH	ADCREG3
DH	“
EH	“
FH	“

ADR	Register name
FF004E40H	ADCREG4
1H	“
2H	“
3H	“
4H	ADCREG5
5H	“
6H	“
7H	“
8H	ADCREG6
9H	“
AH	“
BH	“
CH	ADCREG7
DH	“
EH	“
FH	“

ADR	Register name
FF004E50H	ADREGSP
1H	“
2H	“
3H	“
4H	ADCOMREG0
5H	“
6H	“
7H	“
8H	ADCOMREG1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004E60H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004E70H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[18] Watchdog timer

ADR	Register name
FF004F00H	WDMOD
1H	“
2H	“
3H	“
4H	WDCR
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	reserved
DH	“
EH	“
FH	“

ADR	Register name
FF004F10H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004F20H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004F30H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

[1] ROM correction

ADR	Register name
FF000000H	ADDREG0
1H	“
2H	“
3H	“
4H	ADDREG1
5H	“
6H	“
7H	“
8H	ADDREG2
9H	“
AH	“
BH	“
CH	ADDREG3
DH	“
EH	“
FH	“

ADR	Register name
FF000010H	ADDREG4
1H	“
2H	“
3H	“
4H	ADDREG5
5H	“
6H	“
7H	“
8H	ADDREG6
9H	“
AH	“
BH	“
CH	ADDREG7
DH	“
EH	“
FH	“

ADR	Register name
FF000020H	ADDREG8
1H	“
2H	“
3H	“
4H	ADDREG9
5H	“
6H	“
7H	“
8H	ADDREGA
9H	“
AH	“
BH	“
CH	ADDREGB
DH	“
EH	“
FH	“

ADR	Register name
FF000030H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[2] FLASH control

ADR	Register name
FF000100H	FLCS
1H	“
2H	“
3H	“
4H	Reserved
5H	“
6H	“
7H	“
8H	Reserved
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

[3] Protect control

ADR	Register name
FF000200H	SECBIT
1H	“
2H	“
3H	“
4H	DSUSECBIT
5H	“
6H	“
7H	“
8H	SECCODE
9H	“
AH	“
BH	“
CH	DSUSECCODE
DH	“
EH	“
FH	“

ADR	Register name
FF000210H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF000220H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[4] Interrupt controller

ADR	Register name
FF001000H	IMC0
1H	“
2H	“
3H	“
4H	IMC1
5H	“
6H	“
7H	“
8H	IMC2
9H	“
AH	“
BH	“
CH	IMC3
DH	“
EH	“
FH	“

ADR	Register name
FF001010H	IMC4
1H	“
2H	“
3H	“
4H	IMC5
5H	“
6H	“
7H	“
8H	IMC6
9H	“
AH	“
BH	“
CH	IMC7
DH	“
EH	“
FH	“

ADR	Register name
FF001020H	IMC8
1H	“
2H	“
3H	“
4H	IMC9
5H	“
6H	“
7H	“
8H	IMCA
9H	“
AH	“
BH	“
CH	IMCB
DH	“
EH	“
FH	“

ADR	Register name
FF001030H	IMCC
1H	“
2H	“
3H	“
4H	IMCD
5H	“
6H	“
7H	“
8H	IMCE
9H	“
AH	“
BH	“
CH	IMCF
DH	“
EH	“
FH	“

Big

ADR	Register name
FF001040H	IMC10
1H	“
2H	“
3H	“
4H	IMC11
5H	“
6H	“
7H	“
8H	IMC12
9H	“
AH	“
BH	“
CH	IMC13
DH	“
EH	“
FH	“

ADR	Register name
FF001050H	IMC14
1H	“
2H	“
3H	“
4H	IMC15
5H	“
6H	“
7H	“
8H	IMC16
9H	“
AH	“
BH	“
CH	IMC17
DH	“
EH	“
FH	“

ADR	Register name
FF001060H	IMC18
1H	“
2H	“
3H	“
4H	IMC19
5H	“
6H	“
7H	“
8H	Reserved
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF001070H	Reserved
1H	“
2H	“
3H	“
4H	Reserved
5H	“
6H	“
7H	“
8H	Reserved
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF001080H	IVR
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001090H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010C0H	INTCLR
1H	“
2H	“
3H	“
4H	DREQFLG
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0010F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF001100H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	ILEV
DH	“
EH	“
FH	“

ADR	Register name
FF001110H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001120H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001130H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] DMA controller

ADR	Register name
FF001200H	CCR0
1H	“
2H	“
3H	“
4H	CSR0
5H	“
6H	“
7H	“
8H	SAR0
9H	“
AH	“
BH	“
CH	DAR0
DH	“
EH	“
FH	“

ADR	Register name
FF001210H	BCR0
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR0
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001220H	CCR1
1H	“
2H	“
3H	“
4H	CSR1
5H	“
6H	“
7H	“
8H	SAR1
9H	“
AH	“
BH	“
CH	DAR1
DH	“
EH	“
FH	“

ADR	Register name
FF001230H	BCR1
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001240H	CCR2
1H	“
2H	“
3H	“
4H	CSR2
5H	“
6H	“
7H	“
8H	SAR2
9H	“
AH	“
BH	“
CH	DAR2
DH	“
EH	“
FH	“

ADR	Register name
FF001250H	BCR2
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR2
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF001260H	CCR3
1H	“
2H	“
3H	“
4H	CSR3
5H	“
6H	“
7H	“
8H	SAR3
9H	“
AH	“
BH	“
CH	DAR3
DH	“
EH	“
FH	“

ADR	Register name
FF001270H	BCR3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR3
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF001280H	CCR4
1H	“
2H	“
3H	“
4H	CSR4
5H	“
6H	“
7H	“
8H	SAR4
9H	“
AH	“
BH	“
CH	DAR4
DH	“
EH	“
FH	“

ADR	Register name
FF001290H	BCR4
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR4
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0012A0H	CCR5
1H	“
2H	“
3H	“
4H	CSR5
5H	“
6H	“
7H	“
8H	SAR5
9H	“
AH	“
BH	“
CH	DAR5
DH	“
EH	“
FH	“

ADR	Register name
FF0012B0H	BCR5
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0012C0H	CCR6
1H	“
2H	“
3H	“
4H	CSR6
5H	“
6H	“
7H	“
8H	SAR6
9H	“
AH	“
BH	“
CH	DAR6
DH	“
EH	“
FH	“

ADR	Register name
FF0012D0H	BCR6
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR6
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0012E0H	CCR7
1H	“
2H	“
3H	“
4H	CSR7
5H	“
6H	“
7H	“
8H	SAR7
9H	“
AH	“
BH	“
CH	DAR7
DH	“
EH	“
FH	“

ADR	Register name
FF0012F0H	BCR7
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	DTCR7
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

[6] Chip select/wait controller

ADR	Register name
FF001300H	DCR
1H	“
2H	“
3H	“
4H	RSR
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	DHR
DH	“
EH	“
FH	“

ADR	Register name
FF001400H	BMA0
1H	“
2H	“
3H	“
4H	BMA1
5H	“
6H	“
7H	“
8H	BMA2
9H	“
AH	“
BH	“
CH	BMA3
DH	“
EH	“
FH	“

ADR	Register name
FF001410H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001420H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

[7] Real time clock

ADR	Register name
FF001480H	B01CS
1H	“
2H	“
3H	“
4H	B23CS
5H	“
6H	“
7H	“
8H	BEXCS
9H	“
AH	“
BH	“
F0014C0H	BUSCR
1H	
2H	
3H	

ADR	Register name
FF001500H	HOURR
1H	“
2H	MINR
3H	SECR
4H	YEARR
5H	MONTHR
6H	DATER
7H	DAYR
8H	PAGER
9H	“
AH	“
BH	“
CH	RESTR
DH	“
EH	“
FH	“

ADR	Register name
FF001510H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001520H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[8] Two-phase pulse input counter

ADR	Register name
FF001600H	PHC0RUN
1H	“
2H	“
3H	“
4H	PHC0CR
5H	“
6H	“
7H	“
8H	PHC0EN
9H	“
AH	“
BH	“
CH	PHC0FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001610H	PHC0CMP0
1H	“
2H	“
3H	“
4H	PHC0CMP1
5H	“
6H	“
7H	“
8H	PHC0CNT
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF001620H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001630H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001640H	PHC1RUN
1H	“
2H	“
3H	“
4H	PHC1CR
5H	“
6H	“
7H	“
8H	PHC1EN
9H	“
AH	“
BH	“
CH	PHC1FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001650H	PHC1CMP0
1H	“
2H	“
3H	“
4H	PHC1CMP1
5H	“
6H	“
7H	“
8H	PHC1CNT
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF001660H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001670H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF001680H	PHC2RUN
1H	“
2H	“
3H	“
4H	PHC2CR
5H	“
6H	“
7H	“
8H	PHC2EN
9H	“
AH	“
BH	“
CH	PHC2FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001690H	PHC2CMP0
1H	“
2H	“
3H	“
4H	PHC2CMP1
5H	“
6H	“
7H	“
8H	PHC2CNT
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF0016A0H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0016B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0016C0H	PHC3RUN
1H	“
2H	“
3H	“
4H	PHC3CR
5H	“
6H	“
7H	“
8H	PHC3EN
9H	“
AH	“
BH	“
CH	PHC3FLG
DH	“
EH	“
FH	“

ADR	Register name
FF0016D0H	PHC3CMP0
1H	“
2H	“
3H	“
4H	PHC3CMP1
5H	“
6H	“
7H	“
8H	PHC3CNT
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF0016E0H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0016F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001700H	PHC4RUN
1H	“
2H	“
3H	“
4H	PHC4CR
5H	“
6H	“
7H	“
8H	PHC4EN
9H	“
AH	“
BH	“
CH	PHC4FLG
DH	“
EH	“
FH	“

ADR	Register name
FF001710H	PHC4CMP0
1H	“
2H	“
3H	“
4H	PHC4CMP1
5H	“
6H	“
7H	“
8H	PHC4CNT
9H	“
AH	“
BH	“
CH	Reserved
DH	“
EH	“
FH	“

ADR	Register name
FF001720H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001730H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF001740H	PHC5RUN	FF001750H	PHC5CMP0	FF001760H	Reserved	FF001770H	
1H	“	1H	“	1H	“	1H	
2H	“	2H	“	2H	“	2H	
3H	“	3H	“	3H	“	3H	
4H	PHC5CR	4H	PHC5CMP1	4H		4H	
5H	“	5H	“	5H		5H	
6H	“	6H	“	6H		6H	
7H	“	7H	“	7H		7H	
8H	PHC5EN	8H	PHC5CNT	8H		8H	
9H	“	9H	“	9H		9H	
AH	“	AH	“	AH		AH	
BH	“	BH	“	BH		BH	
CH	PHC5FLG	CH	Reserved	CH		CH	
DH	“	DH	“	DH		DH	
EH	“	EH	“	EH		EH	
FH	“	FH	“	FH		FH	

[9] High speed serial channel

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF001800H		FF001810H		FF001820H		FF001830H	
1H		1H		1H		1H	
2H		2H		2H		2H	
3H	HSC0BUF	3H	HSC1BUF	3H	HSC2BUF	3H	
4H	HSC0EN	4H	HSC1EN	4H	HSC3EN	4H	
5H	HSC0MOD2	5H	HSC1MOD2	5H	HSC3MOD2	5H	
6H	HSC0MOD1	6H	HSC1MOD1	6H	HSC3MOD1	6H	
7H	HBR0ADD	7H	HBR1ADD	7H	HBR3ADD	7H	
8H	HSC0TST	8H	HSC1TST	8H	HSC3TST	8H	
9H	HSC0RST	9H	HSC1RST	9H	HSC3RST	9H	
AH	HSC0TFC	AH	HSC1TFC	AH	HSC3TFC	AH	
BH	HSC0RFC	BH	HSC1RFC	BH	HSC3RFC	BH	
CH	HBR0CR	CH	HBR1CR	CH	HBR3CR	CH	
DH	HSC0MOD0	DH	HSC1MOD0	DH	HSC3MOD0	DH	
EH	HSC0CR	EH	HSC1CR	EH	HSC3CR	EH	
FH	HSC0FCNF	FH	HSC1FCNF	FH	HSC3FCNF	FH	

[10] Clock generator

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF001900H	SYSCR	FF001910H	SCKSEL	FF001920H	IMCGA	FF001930H	IMCGE
1H	“	1H	“	1H	“	1H	“
2H	“	2H	“	2H	“	2H	“
3H	“	3H	“	3H	“	3H	“
4H	OSCCR	4H	ICRCG	4H	IMCGB	4H	IMCGF
5H	“	5H	“	5H	“	5H	“
6H	“	6H	“	6H	“	6H	“
7H	“	7H	“	7H	“	7H	“
8H	STBYCR	8H	NMIFLG	8H	IMCGC	8H	IMCG10
9H	“	9H	“	9H	“	9H	“
AH	“	AH	“	AH	“	AH	“
BH	“	BH	“	BH	“	BH	“
CH	PLLSEL	CH	RSTFLG	CH	IMCGD	CH	IMCG11
DH	“	DH	“	DH	“	DH	“
EH	“	EH	“	EH	“	EH	“
FH	“	FH	“	FH	“	FH	“

Big

[11] Key-on wake-up

ADR	Register name
FF001A00H	KWUPST00
1H	“
2H	“
3H	“
4H	KWUPST01
5H	“
6H	“
7H	“
8H	KWUPST02
9H	“
AH	“
BH	“
CH	KWUPST03
DH	“
EH	“
FH	“

ADR	Register name
FF001A10H	KWUPST04
1H	“
2H	“
3H	“
4H	KWUPST05
5H	“
6H	“
7H	“
8H	KWUPST06
9H	“
AH	“
BH	“
CH	KWUPST07
DH	“
EH	“
FH	“

ADR	Register name
FF001A20H	KWUPST08
1H	“
2H	“
3H	“
4H	KWUPST09
5H	“
6H	“
7H	“
8H	KWUPST10
9H	“
AH	“
BH	“
CH	KWUPST11
DH	“
EH	“
FH	“

ADR	Register name
FF001A30H	KWUPST12
1H	“
2H	“
3H	“
4H	KWUPST13
5H	“
6H	“
7H	“
8H	KWUPST14
9H	“
AH	“
BH	“
CH	KWUPST15
DH	“
EH	“
FH	“

ADR	Register name
FF001A40H	KWUPST 16
1H	“
2H	“
3H	“
4H	KWUPST 17
5H	“
6H	“
7H	“
8H	KWUPST 18
9H	“
AH	“
BH	“
CH	KWUPST 19
DH	“
EH	“
FH	“

ADR	Register name
FF001A50H	KWUPST 20
1H	“
2H	“
3H	“
4H	KWUPST 21
5H	“
6H	“
7H	“
8H	KWUPST 22
9H	“
AH	“
BH	“
CH	KWUPST 23
DH	“
EH	“
FH	“

ADR	Register name
FF001A60H	KWUPST 24
1H	“
2H	“
3H	“
4H	KWUPST 25
5H	“
6H	“
7H	“
8H	KWUPST 26
9H	“
AH	“
BH	“
CH	KWUPST 27
DH	“
EH	“
FH	“

ADR	Register name
FF001A70H	KWUPST 28
1H	“
2H	“
3H	“
4H	KWUPST 29
5H	“
6H	“
7H	“
8H	KWUPST 30
9H	“
AH	“
BH	“
CH	KWUPST 31
DH	“
EH	“
FH	“

ADR	Register name
FF001A80H	PKEY
1H	“
2H	“
3H	“
4H	KWUPCNT
5H	“
6H	“
7H	“
8H	KWUPCLR
9H	“
AH	“
BH	“
CH	KWUPINT
DH	“
EH	“
FH	“

ADR	Register name
FF001A90H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001AA0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF001AB0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big
[12] Port registers

ADR	Register name
FF004000H	P0
1H	“
2H	“
3H	“
4H	P0CR
5H	“
6H	“
7H	“
8H	P0FC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004010H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004020H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P0PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004030H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004040H	P1
1H	“
2H	“
3H	“
4H	P1CR
5H	“
6H	“
7H	“
8H	P1FC1
9H	“
AH	“
BH	“
CH	P1FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004050H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004060H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P1PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004080H	P2
1H	“
2H	“
3H	“
4H	P2CR
5H	“
6H	“
7H	“
8H	P2FC1
9H	“
AH	“
BH	“
CH	P2FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004090H	P2FC3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0040A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P2PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0040B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P2IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF0040C0H	P3
1H	“
2H	“
3H	“
4H	P3CR
5H	“
6H	“
7H	“
8H	P3FC1
9H	“
AH	“
BH	“
CH	P3FC2
DH	“
EH	“
FH	“

ADR	Register name
FF0040D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0040E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P3PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0040F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P3IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004100H	P4
1H	“
2H	“
3H	“
4H	P4CR
5H	“
6H	“
7H	“
8H	P4FC1
9H	“
AH	“
BH	“
CH	P4FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004110H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004120H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P4PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004130H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P4IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004140H	P5
1H	“
2H	“
3H	“
4H	P5CR
5H	“
6H	“
7H	“
8H	P5FC1
9H	“
AH	“
BH	“
CH	P5FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004150H	P5FC3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004160H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P5PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004170H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P5IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004180H	P6
1H	“
2H	“
3H	“
4H	P6CR
5H	“
6H	“
7H	“
8H	P6FC1
9H	“
AH	“
BH	“
CH	P6FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004190H	P6FC3
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0041A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P6ODE
9H	“
AH	“
BH	“
CH	P6PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0041B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P6IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0041C0H	P7
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P7FC2
DH	“
EH	“
FH	“

ADR	Register name
FF0041D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0041E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P7PUP
DH	“
EH	“
FH	“

ADR	Register name
FF0041F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P7IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004200H	P8
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P8FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004210H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004220H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	P8PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004230H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P8IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004240H	P9
1H	“
2H	“
3H	“
4H	P9CR
5H	“
6H	“
7H	“
8H	P9FC1
9H	“
AH	“
BH	“
CH	P9FC2
DH	“
EH	“
FH	“

ADR	Register name
FF004250H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004260H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P9ODE
9H	“
AH	“
BH	“
CH	P9PUP
DH	“
EH	“
FH	“

ADR	Register name
FF004270H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	P9IE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004280H	PA
1H	“
2H	“
3H	“
4H	PACR
5H	“
6H	“
7H	“
8H	PAFC1
9H	“
AH	“
BH	“
CH	PAFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004290H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0042A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PAPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0042B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PAIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0042C0H	PB
1H	“
2H	“
3H	“
4H	PBCR
5H	“
6H	“
7H	“
8H	PBFC1
9H	“
AH	“
BH	“
CH	PBFC2
DH	“
EH	“
FH	“

ADR	Register name
FF0042D0H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0042E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PBODE
9H	“
AH	“
BH	“
CH	PBPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0042F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PBIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004300H	PC
1H	“
2H	“
3H	“
4H	PCCR
5H	“
6H	“
7H	“
8H	PCFC1
9H	“
AH	“
BH	“
CH	PCFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004310H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004320H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PCODE
9H	“
AH	“
BH	“
CH	PCPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004330H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PCIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004340H	PD
1H	“
2H	“
3H	“
4H	PDCR
5H	“
6H	“
7H	“
8H	PDFC1
9H	“
AH	“
BH	“
CH	PDFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004350H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004360H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PDODE
9H	“
AH	“
BH	“
CH	PDPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004370H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PDIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004380H	PE
1H	“
2H	“
3H	“
4H	PECR
5H	“
6H	“
7H	“
8H	PEFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004390H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0043A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PEPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0043B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PEIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF0043C0H	PF
1H	“
2H	“
3H	“
4H	PFCR
5H	“
6H	“
7H	“
8H	PFFC1
9H	“
AH	“
BH	“
CH	PFFC2
DH	“
EH	“
FH	“

ADR	Register name
FF0043D0H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0043E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PFPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0043F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PFIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004400H	PG
1H	“
2H	“
3H	“
4H	PGCR
5H	“
6H	“
7H	“
8H	PGFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004410H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004420H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PGPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004430H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PGIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004440H	PH
1H	“
2H	“
3H	“
4H	PHCR
5H	“
6H	“
7H	“
8H	PHFC1
9H	“
AH	“
BH	“
CH	PHFC2
DH	“
EH	“
FH	“

ADR	Register name
FF004450H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004460H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PHPUP
DH	“
EH	“
FH	“

ADR	Register name
FF004470H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PHIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004480H	PI
1H	“
2H	“
3H	“
4H	PICR
5H	“
6H	“
7H	“
8H	PIFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004490H	Reserved
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0044A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PIPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0044B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PIIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0044C0H	PJ
1H	“
2H	“
3H	“
4H	PJCR
5H	“
6H	“
7H	“
8H	PJFC1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF0044D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0044E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PJPUP
DH	“
EH	“
FH	“

ADR	Register name
FF0044F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	PJIE
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

[13] 16-bit timer

ADR	Register name
FF004500H	TB0EN
1H	“
2H	“
3H	“
4H	TB0RUN
5H	“
6H	“
7H	“
8H	TB0CR
9H	“
AH	“
BH	“
CH	TB0MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004510H	TB0FFCR
1H	“
2H	“
3H	“
4H	TB0ST
5H	“
6H	“
7H	“
8H	TB0IM
9H	“
AH	“
BH	“
CH	TM0UC
DH	“
EH	“
FH	“

ADR	Register name
FF004520H	TB0RG0
1H	“
2H	“
3H	“
4H	TB0RG1
5H	“
6H	“
7H	“
8H	TB0CP0
9H	“
AH	“
BH	“
CH	TB0CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004530H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004540H	TB1EN
1H	“
2H	“
3H	“
4H	TB1RUN
5H	“
6H	“
7H	“
8H	TB1CR
9H	“
AH	“
BH	“
CH	TB1MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004550H	TB1FFCR
1H	“
2H	“
3H	“
4H	TB1ST
5H	“
6H	“
7H	“
8H	TB1IM
9H	“
AH	“
BH	“
CH	TM1UC
DH	“
EH	“
FH	“

ADR	Register name
FF004560H	TB1RG0
1H	“
2H	“
3H	“
4H	TB1RG1
5H	“
6H	“
7H	“
8H	TB1CP0
9H	“
AH	“
BH	“
CH	TB1CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004570H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004580H	TB2EN
1H	“
2H	“
3H	“
4H	TB2RUN
5H	“
6H	“
7H	“
8H	TB2CR
9H	“
AH	“
BH	“
CH	TB2MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004590H	TB2FFCR
1H	“
2H	“
3H	“
4H	TB2ST
5H	“
6H	“
7H	“
8H	TB2IM
9H	“
AH	“
BH	“
CH	TM2UC
DH	“
EH	“
FH	“

ADR	Register name
FF0045A0H	TB2RG0
1H	“
2H	“
3H	“
4H	TB2RG1
5H	“
6H	“
7H	“
8H	TB2CP0
9H	“
AH	“
BH	“
CH	TB2CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0045B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0045C0H	TB3EN
1H	“
2H	“
3H	“
4H	TB3RUN
5H	“
6H	“
7H	“
8H	TB3CR
9H	“
AH	“
BH	“
CH	TB3MOD
DH	“
EH	“
FH	“

ADR	Register name
FF0045D0H	TB3FFCR
1H	“
2H	“
3H	“
4H	TB3ST
5H	“
6H	“
7H	“
8H	TB3IM
9H	“
AH	“
BH	“
CH	TM3UC
DH	“
EH	“
FH	“

ADR	Register name
FF0045E0H	TB3RG0
1H	“
2H	“
3H	“
4H	TB3RG1
5H	“
6H	“
7H	“
8H	TB3CP0
9H	“
AH	“
BH	“
CH	TB3CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0045F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004600H	TB4EN
1H	“
2H	“
3H	“
4H	TB4RUN
5H	“
6H	“
7H	“
8H	TB4CR
9H	“
AH	“
BH	“
CH	TB4MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004610H	TB4FFCR
1H	“
2H	“
3H	“
4H	TB4ST
5H	“
6H	“
7H	“
8H	TB4IM
9H	“
AH	“
BH	“
CH	TM4UC
DH	“
EH	“
FH	“

ADR	Register name
FF004620H	TB4RG0
1H	“
2H	“
3H	“
4H	TB4RG1
5H	“
6H	“
7H	“
8H	TB4CP0
9H	“
AH	“
BH	“
CH	TB4CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004630H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004640H	TB5EN
1H	“
2H	“
3H	“
4H	TB5RUN
5H	“
6H	“
7H	“
8H	TB5CR
9H	“
AH	“
BH	“
CH	TB5MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004650H	TB5FFCR
1H	“
2H	“
3H	“
4H	TB5ST
5H	“
6H	“
7H	“
8H	TB5IM
9H	“
AH	“
BH	“
CH	TM5UC
DH	“
EH	“
FH	“

ADR	Register name
FF004660H	TB5RG0
1H	“
2H	“
3H	“
4H	TB5RG1
5H	“
6H	“
7H	“
8H	TB5CP0
9H	“
AH	“
BH	“
CH	TB5CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004670H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004680H	TB6EN
1H	“
2H	“
3H	“
4H	TB6RUN
5H	“
6H	“
7H	“
8H	TB6CR
9H	“
AH	“
BH	“
CH	TB6MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004690H	TB6FFCR
1H	“
2H	“
3H	“
4H	TB6ST
5H	“
6H	“
7H	“
8H	TB6IM
9H	“
AH	“
BH	“
CH	TM6UC
DH	“
EH	“
FH	“

ADR	Register name
FF0046A0H	TB6RG0
1H	“
2H	“
3H	“
4H	TB6RG1
5H	“
6H	“
7H	“
8H	TB6CP0
9H	“
AH	“
BH	“
CH	TB6CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0046B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF0046C0H	TB7EN
1H	“
2H	“
3H	“
4H	TB7RUN
5H	“
6H	“
7H	“
8H	TB7CR
9H	“
AH	“
BH	“
CH	TB7MOD
DH	“
EH	“
FH	“

ADR	Register name
FF0046D0H	TB7FFCR
1H	“
2H	“
3H	“
4H	TB7ST
5H	“
6H	“
7H	“
8H	TB7IM
9H	“
AH	“
BH	“
CH	TM7UC
DH	“
EH	“
FH	“

ADR	Register name
FF0046E0H	TB7RG0
1H	“
2H	“
3H	“
4H	TB7RG1
5H	“
6H	“
7H	“
8H	TB7CP0
9H	“
AH	“
BH	“
CH	TB7CP1
DH	“
EH	“
FH	“

ADR	Register name
FF0046F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004700H	TB8EN
1H	“
2H	“
3H	“
4H	TB8RUN
5H	“
6H	“
7H	“
8H	TB8CR
9H	“
AH	“
BH	“
CH	TB8MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004710H	TB8FFCR
1H	“
2H	“
3H	“
4H	TB8ST
5H	“
6H	“
7H	“
8H	TB8IM
9H	“
AH	“
BH	“
CH	TM8UC
DH	“
EH	“
FH	“

ADR	Register name
FF004720H	TB8RG0
1H	“
2H	“
3H	“
4H	TB8RG1
5H	“
6H	“
7H	“
8H	TB8CP0
9H	“
AH	“
BH	“
CH	TB8CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004730H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004740H	TB9EN
1H	“
2H	“
3H	“
4H	TB9RUN
5H	“
6H	“
7H	“
8H	TB9CR
9H	“
AH	“
BH	“
CH	TB9MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004785H	TB9FFCR
1H	“
2H	“
3H	“
4H	TB9ST
5H	“
6H	“
7H	“
8H	TB9IM
9H	“
AH	“
BH	“
CH	TM9UC
DH	“
EH	“
FH	“

ADR	Register name
FF004760H	TB9RG0
1H	“
2H	“
3H	“
4H	TB9RG1
5H	“
6H	“
7H	“
8H	TB9CP0
9H	“
AH	“
BH	“
CH	TB9CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004770H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004780H	TBAEN
1H	“
2H	“
3H	“
4H	TBARUN
5H	“
6H	“
7H	“
8H	TBACR
9H	“
AH	“
BH	“
CH	TBAMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004790H	TBAFFCR
1H	“
2H	“
3H	“
4H	TBAST
5H	“
6H	“
7H	“
8H	TBAIM
9H	“
AH	“
BH	“
CH	TMAUC
DH	“
EH	“
FH	“

ADR	Register name
FF0047A0H	TBARG0
1H	“
2H	“
3H	“
4H	TBARG1
5H	“
6H	“
7H	“
8H	TBACP0
9H	“
AH	“
BH	“
CH	TBACP1
DH	“
EH	“
FH	“

ADR	Register name
FF0047B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0047C0H	TBBEN
1H	“
2H	“
3H	“
4H	TBBRUN
5H	“
6H	“
7H	“
8H	TBBCR
9H	“
AH	“
BH	“
CH	TBBMOD
DH	“
EH	“
FH	“

ADR	Register name
FF0047D0H	TBBFFCR
1H	“
2H	“
3H	“
4H	TBBST
5H	“
6H	“
7H	“
8H	TBBIM
9H	“
AH	“
BH	“
CH	TMBUC
DH	“
EH	“
FH	“

ADR	Register name
FF0047E0H	TBBRG0
1H	“
2H	“
3H	“
4H	TBBRG1
5H	“
6H	“
7H	“
8H	TBBCP0
9H	“
AH	“
BH	“
CH	TBBCP1
DH	“
EH	“
FH	“

ADR	Register name
FF0047F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004800H	TBCEN
1H	“
2H	“
3H	“
4H	TBCRUN
5H	“
6H	“
7H	“
8H	TBCCR
9H	“
AH	“
BH	“
CH	TBCMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004810H	TBCFFCR
1H	“
2H	“
3H	“
4H	TBCST
5H	“
6H	“
7H	“
8H	TBCIM
9H	“
AH	“
BH	“
CH	TMCUC
DH	“
EH	“
FH	“

ADR	Register name
FF004820H	TBCRG0
1H	“
2H	“
3H	“
4H	TBCRG1
5H	“
6H	“
7H	“
8H	TBCCP0
9H	“
AH	“
BH	“
CH	TBCCP1
DH	“
EH	“
FH	“

ADR	Register name
FF004830H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004840H	TBDEN
1H	“
2H	“
3H	“
4H	TBDRUN
5H	“
6H	“
7H	“
8H	TBDCR
9H	“
AH	“
BH	“
CH	TBDMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004850H	TBDFFCR
1H	“
2H	“
3H	“
4H	TBDST
5H	“
6H	“
7H	“
8H	TBDIM
9H	“
AH	“
BH	“
CH	TMDUC
DH	“
EH	“
FH	“

ADR	Register name
FF004860H	TBDRG0
1H	“
2H	“
3H	“
4H	TBDRG1
5H	“
6H	“
7H	“
8H	TBDCP0
9H	“
AH	“
BH	“
CH	TBDCP1
DH	“
EH	“
FH	“

ADR	Register name
FF004870H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004880H	TBEEN
1H	“
2H	“
3H	“
4H	TBERUN
5H	“
6H	“
7H	“
8H	TBECR
9H	“
AH	“
BH	“
CH	TBEMOD
DH	“
EH	“
FH	“

ADR	Register name
FF004890H	TBEFFCR
1H	“
2H	“
3H	“
4H	TBEST
5H	“
6H	“
7H	“
8H	TBEIM
9H	“
AH	“
BH	“
CH	TMEUC
DH	“
EH	“
FH	“

ADR	Register name
FF0048A0H	TBERG0
1H	“
2H	“
3H	“
4H	TBERG1
5H	“
6H	“
7H	“
8H	TBECP0
9H	“
AH	“
BH	“
CH	TBECP1
DH	“
EH	“
FH	“

ADR	Register name
FF0048B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF0048C0H	TBFEN
1H	“
2H	“
3H	“
4H	TBFRUN
5H	“
6H	“
7H	“
8H	TBFCR
9H	“
AH	“
BH	“
CH	TBFMOD
DH	“
EH	“
FH	“

ADR	Register name
FF0048D0H	TBFFFCR
1H	“
2H	“
3H	“
4H	TBFST
5H	“
6H	“
7H	“
8H	TBFIM
9H	“
AH	“
BH	“
CH	TMFUC
DH	“
EH	“
FH	“

ADR	Register name
FF0048E0H	TBFRG0
1H	“
2H	“
3H	“
4H	TBFRG1
5H	“
6H	“
7H	“
8H	TBFCP0
9H	“
AH	“
BH	“
CH	TBFCP1
DH	“
EH	“
FH	“

ADR	Register name
FF0048F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004900H	TB10EN
1H	“
2H	“
3H	“
4H	TB10RUN
5H	“
6H	“
7H	“
8H	TB10CR
9H	“
AH	“
BH	“
CH	TB10MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004910H	TB10FFCR
1H	“
2H	“
3H	“
4H	TB10ST
5H	“
6H	“
7H	“
8H	TB10IM
9H	“
AH	“
BH	“
CH	TM10UC
DH	“
EH	“
FH	“

ADR	Register name
FF004920H	TB10RG0
1H	“
2H	“
3H	“
4H	TB10RG1
5H	“
6H	“
7H	“
8H	TB10CP0
9H	“
AH	“
BH	“
CH	TB10CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004930H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004940H	TB11EN
1H	“
2H	“
3H	“
4H	TB11RUN
5H	“
6H	“
7H	“
8H	TB11CR
9H	“
AH	“
BH	“
CH	TB11MOD
DH	“
EH	“
FH	“

ADR	Register name
FF004950H	TB11FFCR
1H	“
2H	“
3H	“
4H	TB11ST
5H	“
6H	“
7H	“
8H	TB11IM
9H	“
AH	“
BH	“
CH	TM11UC
DH	“
EH	“
FH	“

ADR	Register name
FF004960H	TB11RG0
1H	“
2H	“
3H	“
4H	TB11RG1
5H	“
6H	“
7H	“
8H	TB11CP0
9H	“
AH	“
BH	“
CH	TB11CP1
DH	“
EH	“
FH	“

ADR	Register name
FF004970H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] 32-bit timer

ADR	Register name
FF004A00H	TCEN
1H	“
2H	“
3H	“
4H	TBTRUN
5H	“
6H	“
7H	“
8H	TBTCR
9H	“
AH	“
BH	“
CH	TBTCAP
DH	“
EH	“
FH	“

ADR	Register name
FF004A10H	TBTRDCAP
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A20H	CMPCTL0
1H	“
2H	“
3H	“
4H	TCCMP0
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A30H	CMPCTL1
1H	“
2H	“
3H	“
4H	TCCMP1
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004A40H	CMPCTL2
1H	“
2H	“
3H	“
4H	TCCMP2
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A50H	CMPCTL3
1H	“
2H	“
3H	“
4H	TCCMP3
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A60H	CMPCTL4
1H	“
2H	“
3H	“
4H	TCCMP4
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A70H	CMPCTL5
1H	“
2H	“
3H	“
4H	TCCMP5
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A80H	CMPCTL6
1H	“
2H	“
3H	“
4H	TCCMP6
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004A90H	CMPCTL7
1H	“
2H	“
3H	“
4H	TCCMP7
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AA0H	CAPCR0
1H	“
2H	“
3H	“
4H	TCCAP0
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AB0H	CAPCR1
1H	“
2H	“
3H	“
4H	TCCAP1
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AC0H	CAPCR2
1H	“
2H	“
3H	“
4H	TCCAP2
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AD0H	CAPCR3
1H	“
2H	“
3H	“
4H	TCCAP3
5H	“
6H	“
7H	“
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AE0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004AF0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

[15] I^2CBUS/serial channel

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF004B00H	SBICR0	FF004B10H	SBICR2/SBISR	FF004B20H		FF004B30H	
1H	“	1H	“	1H		1H	
2H	“	2H	“	2H		2H	
3H	“	3H	“	3H		3H	
4H	SBICR1	4H	SBIBR	4H		4H	
5H	“	5H	“	5H		5H	
6H	“	6H	“	6H		6H	
7H	“	7H	“	7H		7H	
8H	SBIDBR	8H		8H		8H	
9H	“	9H		9H		9H	
AH	“	AH		AH		AH	
BH	“	BH		BH		BH	
CH	I2CAR	CH		CH		CH	
DH	“	DH		DH		DH	
EH	“	EH		EH		EH	
FH	“	FH		FH		FH	

[16] UART/serial channel

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF004C00H	SC0EN	FF004C10H	BR0CR	FF004C20H	SC0RFC	FF004C30H	SC0FCNF
1H	“	1H	“	1H	“	1H	“
2H	“	2H	“	2H	“	2H	“
3H	“	3H	“	3H	“	3H	“
4H	SC0BUF	4H	BR0ADD	4H	SC0TFC	4H	
5H	“	5H	“	5H	“	5H	
6H	“	6H	“	6H	“	6H	
7H	“	7H	“	7H	“	7H	
8H	SC0CR	8H	SC0MOD1	8H	SC0RST	8H	
9H	“	9H	“	9H	“	9H	
AH	“	AH	“	AH	“	AH	
BH	“	BH	“	BH	“	BH	
CH	SC0MOD0	CH	SC0MOD2	CH	SC0TST	CH	
DH	“	DH	“	DH	“	DH	
EH	“	EH	“	EH	“	EH	
FH	“	FH	“	FH	“	FH	

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FF004C40H	SC1EN	FF004C50H	BR1CR	FF004C60H	SC1RFC	FF004C70H	SC1FCNF
1H	“	1H	“	1H	“	1H	“
2H	“	2H	“	2H	“	2H	“
3H	“	3H	“	3H	“	3H	“
4H	SC1BUF	4H	BR1ADD	4H	SC1TFC	4H	
5H	“	5H	“	5H	“	5H	
6H	“	6H	“	6H	“	6H	
7H	“	7H	“	7H	“	7H	
8H	SC1CR	8H	SC1MOD1	8H	SC1RST	8H	
9H	“	9H	“	9H	“	9H	
AH	“	AH	“	AH	“	AH	
BH	“	BH	“	BH	“	BH	
CH	SC1MOD0	CH	SC1MOD2	CH	SC1TST	CH	
DH	“	DH	“	DH	“	DH	
EH	“	EH	“	EH	“	EH	
FH	“	FH	“	FH	“	FH	

Big

ADR	Register name
FF004C80H	SC2EN
1H	“
2H	“
3H	“
4H	SC2BUF
5H	“
6H	“
7H	“
8H	SC2CR
9H	“
AH	“
BH	“
CH	SC2MOD0
DH	“
EH	“
FH	“

ADR	Register name
FF004C90H	BR2CR
1H	“
2H	“
3H	“
4H	BR2ADD
5H	“
6H	“
7H	“
8H	SC2MOD1
9H	“
AH	“
BH	“
CH	SC2MOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004CA0H	SC2RFC
1H	“
2H	“
3H	“
4H	SC2TFC
5H	“
6H	“
7H	“
8H	SC2RST
9H	“
AH	“
BH	“
CH	SC2TST
DH	“
EH	“
FH	“

ADR	Register name
FF004CB0H	SC2FCNF
1H	“
2H	“
3H	“
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] 10-bit A/D converter

ADR	Register name
FF004D00H	ADACLK
1H	“
2H	“
3H	“
4H	ADAMOD0
5H	“
6H	“
7H	“
8H	ADAMOD1
9H	“
AH	“
BH	“
CH	ADAMOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004D10H	ADAMOD3
1H	“
2H	“
3H	“
4H	ADAMOD4
5H	“
6H	“
7H	“
8H	ADAMOD5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004D20H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004D30H	ADAREG0
1H	“
2H	“
3H	“
4H	ADAREG1
5H	“
6H	“
7H	“
8H	ADAREG2
9H	“
AH	“
BH	“
CH	ADAREG3
DH	“
EH	“
FH	“

ADR	Register name
FF004D40H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	reserved
DH	“
EH	“
FH	“

ADR	Register name
FF004D50H	ADAREGSP
1H	“
2H	“
3H	“
4H	ADACOMREG0
5H	“
6H	“
7H	“
8H	ADACOMREG1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004D60H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004D70H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big

ADR	Register name
FF004D80H	ADBCLK
1H	“
2H	“
3H	“
4H	ADBMOD0
5H	“
6H	“
7H	“
8H	ADBMOD1
9H	“
AH	“
BH	“
CH	ADBMOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004D90H	ADBMOD3
1H	“
2H	“
3H	“
4H	ADBMOD4
5H	“
6H	“
7H	“
8H	ADBMOD5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004DA0H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004DB0H	ADBREG0
1H	“
2H	“
3H	“
4H	ADBREG1
5H	“
6H	“
7H	“
8H	ADBREG2
9H	“
AH	“
BH	“
CH	ADBREG3
DH	“
EH	“
FH	“

ADR	Register name
FF004DC0H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	reserved
DH	“
EH	“
FH	“

ADR	Register name
FF004DD0H	ADBREGSP
1H	“
2H	“
3H	“
4H	ADBCOMREG0
5H	“
6H	“
7H	“
8H	ADBCOMREG1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004DE0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004DF0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004E00H	ADCCLK
1H	“
2H	“
3H	“
4H	ADCMOD0
5H	“
6H	“
7H	“
8H	ADCMOD1
9H	“
AH	“
BH	“
CH	ADCMOD2
DH	“
EH	“
FH	“

ADR	Register name
FF004E10H	ADCMOD3
1H	“
2H	“
3H	“
4H	ADCMOD4
5H	“
6H	“
7H	“
8H	ADCMOD5
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004E20H	reserved
1H	“
2H	“
3H	“
4H	reserved
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004E30H	ADCREG0
1H	“
2H	“
3H	“
4H	ADCREG1
5H	“
6H	“
7H	“
8H	ADCREG2
9H	“
AH	“
BH	“
CH	ADCREG3
DH	“
EH	“
FH	“

Big

ADR	Register name
FF004E40H	ADCREG4
1H	“
2H	“
3H	“
4H	ADCREG5
5H	“
6H	“
7H	“
8H	ADCREG6
9H	“
AH	“
BH	“
CH	ADCREG7
DH	“
EH	“
FH	“

ADR	Register name
FF004E50H	ADREGSP
1H	“
2H	“
3H	“
4H	ADCOMREG0
5H	“
6H	“
7H	“
8H	ADCOMREG1
9H	“
AH	“
BH	“
CH	
DH	
EH	
FH	

ADR	Register name
FF004E60H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004E70H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[18] Watchdog timer

ADR	Register name
FF004F00H	WDMOD
1H	“
2H	“
3H	“
4H	WDCR
5H	“
6H	“
7H	“
8H	reserved
9H	“
AH	“
BH	“
CH	reserved
DH	“
EH	“
FH	“

ADR	Register name
FF004F10H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004F20H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FF004F30H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

23. JTAG Interface

The TMP19A44 is equipped with the boundary scan interface that conforms to the Joint Test Action Group (JTAG) standard. This interface uses the industry-standard JTAG protocol (IEEE Standard 1149.1/D6). This chapter describes this JTAG interface with a mention of boundary scan, interface pins, interface signals, and test access ports (TAP).

23.1 Boundary Scan Overview

IC (Integrated Circuit) density is ever increasing, SMDs (Surface Mount Devices) continue to decrease in size, components are now mounted on both sides of printed circuit boards (PCBs), and there are considerable technical developments related to embedding holes. Conventional internal circuit testing techniques are dependent on the physical contact between internal circuitry and chips and, therefore, their limitations with respect to efficiency and accuracy are manifest. With the ever-increasing IC complexity, tests conducted to perform inspections on all chips integrated into an IC are becoming larger in scale, and it is becoming more difficult to design an efficient, reliable IC testing program.

To overcome this difficulty in performing IC tests, the "boundary scan" circuit was developed. It is a group of shift registers called "boundary scan cells" established between pins and internal circuitry (see Fig. 23-1). These boundary scan cells are bypassed under normal conditions. When an IC goes into test mode, data is sent from the boundary scan cells through the shift register bus in response to the instruction given by a test program, and various diagnostic tests are executed. In IC tests, five signals TDI, TDO, TMS, TCK and $\overline{\text{TRST}}$ are used. These signals are explained in the next section.

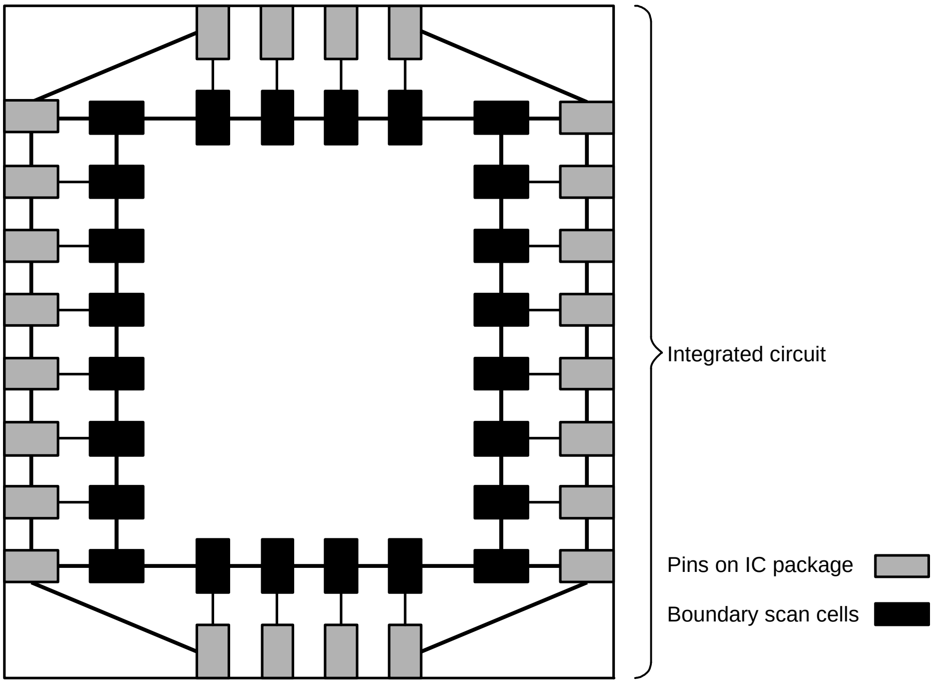


Fig. 23-1 JTAG Boundary Scan Cells

Note) The optional instructions IDCODE, USERCODE, INTEST and RUNBIST are not implemented in the TMP19A44.

23.2 JTAG Interface Signals

JTAG interface signals are as follows (see Fig. 23-2):

TDI	: To input JTAG serial data
TDO	: To output JTAG serial data
TMS	: To select JTAG test mode
TCK	: To input JTAG serial clock
$\overline{\text{TRST}}$	: To input JTAG test reset

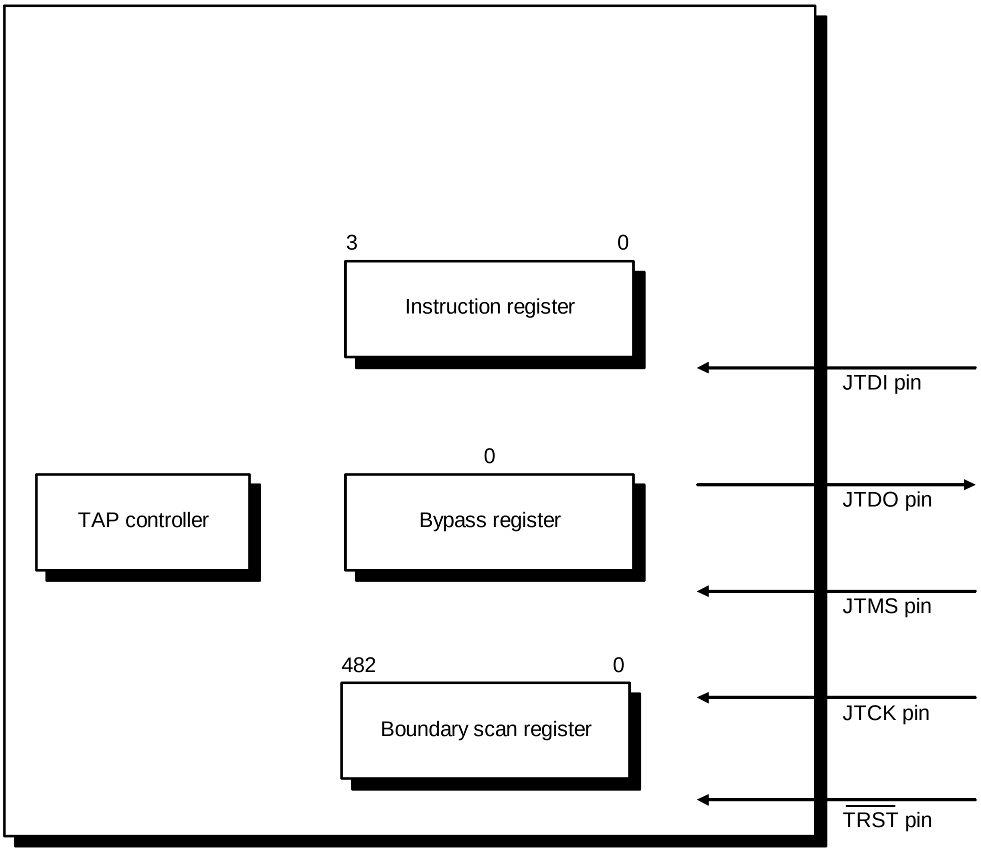


Fig. 23-2 JTAG Interface Signals and Registers

The JTAG boundary scan mechanism (hereafter called "JTAG mechanism") enables testing of the processor, printed circuit boards connected to the processor, and connections between other components on printed circuit boards.

The JTAG mechanism does not have a function of testing the processor itself.

23.3 JTAG Controller and Registers

The following JTAG controller and registers are built into the processor:

- Instruction register
- Boundary scan register
- Bypass register
- Device identification register
- Test Access Port (TAP) controller

In the JTAG basic mechanism, the TAP controller state machine monitors the signals input through the JTMS pin. As the JTAG mechanism starts operation, the TAP controller determines a test function to be executed by loading data into the JTAG instruction register (IR) and performing a serial data scan via the data register (DR), as shown in Table 23-1. When data is scanned, the state of the JTMS pin represents new specific data words and the end of data flow. The data register is selected according to data loaded into the instruction register.

23.3.1 Instruction Register

The JTAG instruction register consists of four cells, including shift registers. It is used to select either a test to be executed or a test data register to be accessed or to select both. Either the boundary scan register or the bypass register is selected according to combinations shown in Table 23-1.

Table 23-1 Bit Configurations of the JTAG Instruction Register

Instruction code Most significant to least significant bit	Instruction	Data register to be selected
0000	EXTEST	Boundary scan register
0001	SAMPLE/PRELOAD	Boundary scan register
0010 to 1110	Reserved	Reserved
1111	BYPASS	Bypass register

Fig. 23-3 shows the format of the instruction register.

3	2	1	0
MSB			LSB

Fig. 23-3 Instruction Register

The instruction code is shifted from the least significant bit to the instruction register.

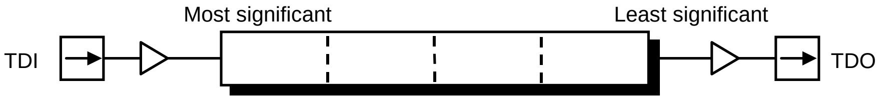


Fig. 23-4 Direction of a Shift of the Instruction Code to the Instruction Register

23.3.2 Bypass Register

The bypass register has a one-bit width. If the TAP controller is in the Shift-DR state (bypass state), data at the TDI pin is shifted into the bypass register, and the output from the bypass register is shifted out to the TDO output pin.

Simply put, the bypass register is a circuit for bypassing the devices in a serial boundary scan chain connected to the substrates that are not required for a test to be conducted. Fig. 23-5 shows the logical position of the bypass register in a boundary scan chain.

If the bypass register is used, the speed of access to boundary scan registers in an active IC in a data path used for substrate level testing can be increased.

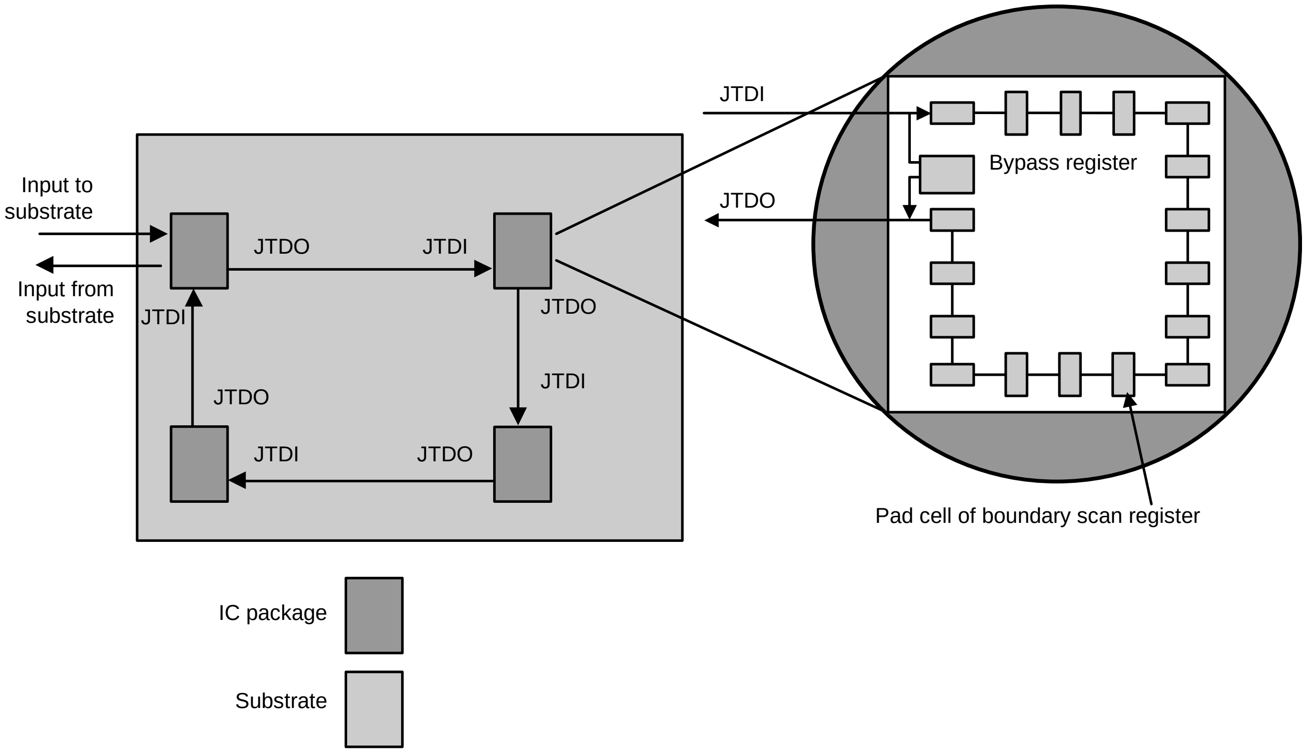


Fig. 23-5 Function of the Bypass Register

23.3.3 Boundary Scan Register

The boundary scan register has inputs and outputs for some analog output signals, as well as all signals from the TMP19A44 except control signals. Pins of the TMP19A44 can drive any test patterns by scanning data into the boundary scan register in the Shift-DR state. After the boundary scan register goes into the Capture-DR state, data enters the processor, is shifted, and inspected.

The boundary scan register forms a data path. It basically functions as a single shift register of 483-bit width. Cells in this data path are connected to all input and output pads of the TMP19A44.

The TDI input is introduced to the least significant bit (LSB) in the boundary scan register. The most significant bit in the boundary scan register is taken out of the TDO output.

23.3.4 Test Access Port (TAP)

The test access port (TAP) consists of five signal pins: $\overline{\text{TRST}}$, TDI, TDO, TMS, and TCK. Serial test data, instructions and test control signals are sent and received through these signal pins.

Data is serially scanned into one of three registers (instruction register, bypass register and boundary scan register) via the TDI pin or it is scanned out from one of these three registers into the TDO pin, as shown in Fig. 23-6.

The TMS input is used to control the state transitions of the main TAP controller state machine. The TCK input is a test clock exclusively for shifting serial JTAG data synchronously; it works independently of a chip core clock or a system clock.

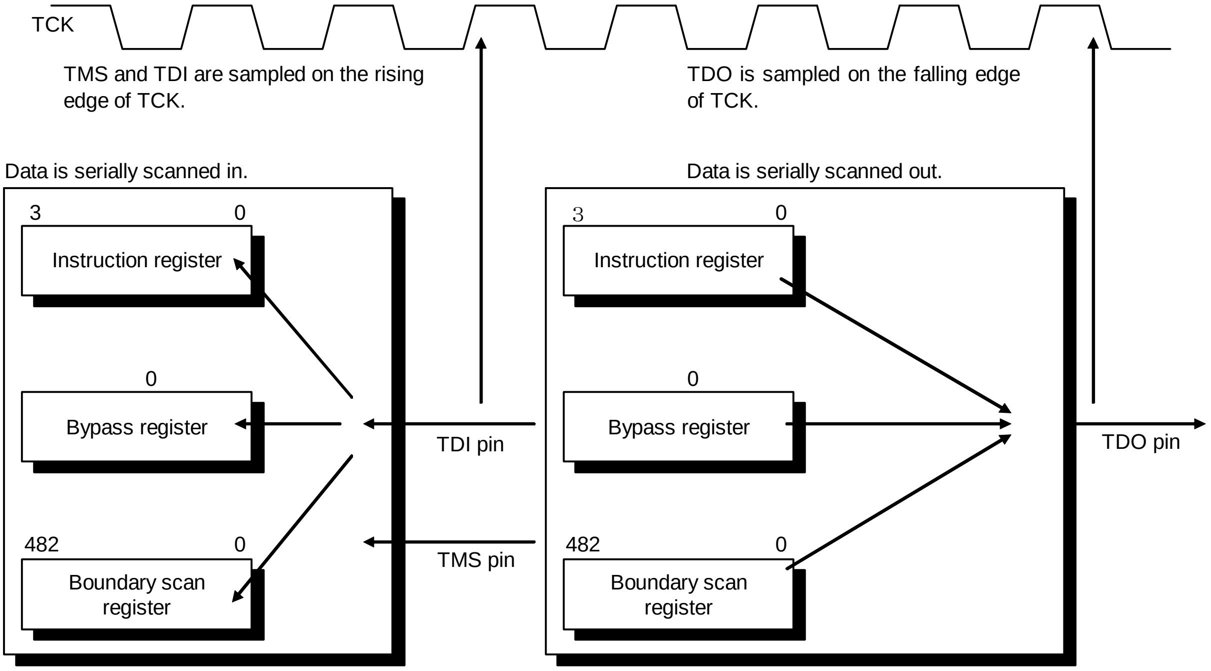


Fig. 23-6 JTAG Test Access Port

Data through the TDI and TMS pins are sampled on the rising edge of the input clock signal TCK. Data through the TDO pin changes on the falling edge of the clock signal TCK.

23.3.5 TAP Controller

In the processor, a 16-state TAP controller specified in the IEEE JTAG standard is implemented.

23.3.6 Controller Reset

To reset the state machine of the TAP controller,

- assert the $\overline{\text{TRST}}$ signal input (Low) to reset the TAP controller or
- continue to assert the input signal TMS by using the rising edge of the TCK input five times successively after clearing the reset state of the processor.

The reset state can be maintained by keeping TMS in an asserted state.

23.3.7 State Transitions of the TAP Controller

Fig. 23-7 shows the state transitions of the TAP controller. The state of the TAP controller changes depending on which value TMS will select on the rising edge of TCK, 0 or 1. In this figure, the arrow shows a state transition and the value that TMS selects to execute each state transition is shown alongside of the arrow.

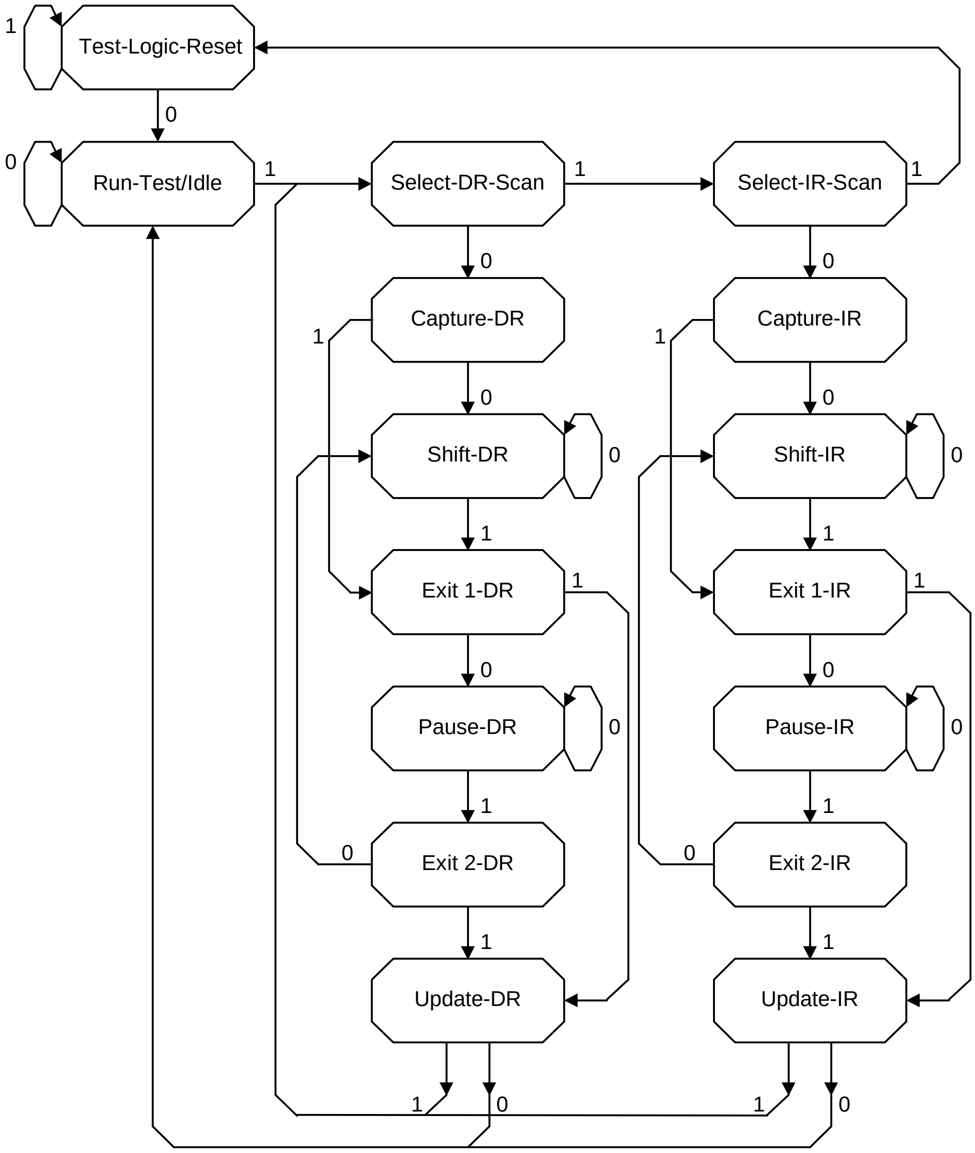


Fig. 23-7 State Transition Diagram of the TAP Controller

The TAP controller operates in each state described below. In Fig. 23-7, a column to the left is the data column and a column to the right is the instruction column. The data column represents the data register (DR), and the instruction column represents the instruction register (IR).

- Test-Logic-Reset
  If the TAP controller is in a reset state, the device identification register is selected by default. The most significant bit in the boundary scan register is cleared to "0," and the output is disabled.
  The TAP controller remains in the Test-Logic-Reset state if TMS is "1." If "0" is input into TMS in the Test-Logic-Reset state, the TAP controller goes into the Run-Test/Idle state.

- Run-Test/Idle
  In the Run-Test/Idle state, the IC goes into test mode only if a specific instruction, such as the built-in self test (BIST) instruction, is issued. If an instruction that cannot be executed in the Run-Test/Idle state has been issued, the test data register selected by the last instruction maintains the existing state.
  The TAP controller remains in the Run-Test/Idle state if TMS is "0." If "1" is input into TMS, the TAP controller goes into the Select-DR-Scan state.

- Select-DR-Scan
  The Select-DR-Scan state of the TAP controller is a transient state. In this state, the IC performs no operations. If "0" is input into TMS when the TAP controller is in the Select-DR-Scan state, the TAP controller goes into the Capture-DR state. If "1" is input into TMS, the instruction column goes into the Select-IR-Scan state.

- Select-IR-Scan
  The Select-IR-Scan state of the TAP controller is a transient state. In this state, the IC performs no operations.
  If "0" is input into TMS when the TAP controller is in the Select-IR-Scan state, the TAP controller goes into the Capture-IR state. If "1" is input into TMS, the TAP controller returns to the Test-Logic-Reset state.

- Capture-DR
  If the data register selected by the instruction register has parallel inputs when the TAP controller is in the Capture-DR state, data is loaded into the data register in a parallel fashion. If the data register does not have parallel inputs or if data does not need to be loaded into the selected test data register, the data register maintains the existing state.
  If "0" is input into TMS when the TAP controller is in the Capture-DR state, the TAP controller goes into the Shift-DR state. If "1" is input into TMS, the TAP controller goes into the Exit 1-DR state.

- Shift-DR
  If the TAP controller is in the Shift-DR state, data is serially shifted out by the data register connected between TDI and TDO.
  If the TAP controller is in the Shift-DR state, the Shift-DR state is maintained while TMS is "0." If "1" is input into TMS, the TAP controller goes into the Exit 1-DR state.

- Exit 1-DR
  The Exit 1-DR state of the TAP controller is a transient state.
  If "0" is input into TMS when the TAP controller is in the Exit 1-DR state, the TAP controller goes into the Pause-DR state. If "1" is input into TMS, it goes into the Update-DR state.

- Pause-DR
  In the Pause-DR state, the shift operation performed by the data register selected by the instruction register is temporarily suspended. The instruction register and the data register maintain their existing state.
  The TAP controller remains in the Pause-DR state while TMS is "0." If "1" is input into TMS, it goes into the Exit 2-DR state.

- Exit 2-DR
  The Exit 2-DR state of the TAP controller is a transient state.
  If "0" is input into TMS when the TAP controller is in the Exit 2-DR state, the TAP controller returns to the Shift-DR state. If "1" is input into TMS, it goes into the Update-DR state.

- Update-DR
  In the Update-DR state, data is output in a parallel fashion from the data register having a parallel output synchronously to the rising edge of TCK. The data register with a parallel output latch does not output data during the shift operation; it outputs data only in the Update-DR state.
  If "0" is input into TMS when the TAP controller is in the Update-DR state, the TAP controller goes into the Run-Test/Idle state. If "1" is input into TMS, it goes into the Select-DR-Scan state.

- Capture-IR
  In the Capture-IR state, data is loaded into the instruction register in a parallel fashion. Data loaded is 0001. The Capture-IR state is used to test the instruction register. A malfunction of the instruction register can be detected by shifting out the data loaded.
  If "0" is input into TMS when the TAP controller is in the Capture-IR state, the TAP controller goes into the Shift-IR state. If "1" is input into TMS, it goes into the Exit 1-IR state.

- Shift-IR
  In the Shift-IR state, the instruction register is connected between TDI and TDO, and data loaded synchronously to the rising edge of TCK is serially shifted out.
  The TAP controller remains in the Shift-IR state while TMS is "0." If "1" is input into TMS, the TAP controller goes into the Exit 1-IR state.

- Exit 1-IR
  The Exit 1-IR state of the TAP controller is a transient state.

  If "0" is input into TMS when the TAP controller is in the Exit 1-IR state, the TAP controller goes into the Pause-IR state. If "1" is input into TMS, it goes into the Update-IR state.

- Pause-IR

  In the Pause-IR state, the shift operation performed by the instruction register is temporarily suspended. The existing state of the instruction register and that of the data register are maintained.

  The TAP controller remains in the Pause-IR state while TMS is "0." If "1" is input into TMS, it goes into the Exit 2-IR state.

- Exit 2-IR

  The Exit 2-IR state of the TAP controller is a transient state.

  If "0" is input into TMS when the TAP controller is in the Exit 2-IR state, the TAP controller goes into the Shift-IR state. If "1" is input into TMS, it goes into the Update-IR state.

- Update-IR

  In the Update-IR state, instructions shifted into the instruction register are updated by outputting them in a parallel fashion synchronously to the rising edge of TCK.

  If "0" is input into TMS when the TAP controller is in the Update-IR state, the TAP controller goes into the Run-Test/Idle state. If "1" is input into TMS, it goes into the Select-DR-Scan state.

Table 23-2 shows the boundary scan sequence relative to processor signals.

Table 23-2 JTAG Scan Sequence Relative to the TMP19A44 Processor Pins

1: P86	2: P87	3: P70	4: P71	5: P72	6: P73	7: P74
8: P75	9: P76	10: P77	11: P92	12: P90	13: P91	14: P93
15: P96	16: P94	17: P95	18:P97	19: PB3	20: PB1	21: PB2
22: PB4	23:PB6	24: PB5	25: PB7	26: PB0	27: PC1	28: PC2
29: PC0	30: PC3	31: PC4	32: PC5	33: P31	34: PC6	35: P30
36: P32	37: P33	38: PC7	39: P34	40:P35	41: P36	42: P02
43: P37	44: P01	45: P05	46: P00	47: P04	48: P03	49: P10
50: P07	51:P11	52:P06	53:P13	54:P12	55: P14	56: P15
57: P21	58: P17	59: P20	60: P16	61: P22	62: P24	63: P25
64: P23	65: P27	66: P40	67: P26	68: P42	69: BOOT	70: P41
71: P43	72: P44	73: P45	74: P46	75: P47	76: P50	77: P51
78: P52	79: P54	80: P56	81: P53	82: P55	83: PJ0	84: P57
85: P61	86: P64	87: P60	88: P63	89: P62	90: P67	91: P66
92: PJ3	93: PJ4	94: P65	95: PJ1	96: PJ2	97: PJ6	98: PJ7
99: PJ5	100: PG0	101: PG2	102: PG1	103: PG4	104: PG3	105:PG6
106:PG7	107:PG5	108: PI1	109: PI0	110: PI4	111: PI3	112:DINT
113:PI2	114:PI6	115:PI7	116:PI5	117:PH2	118:PH0	119:PH1
120:BW0	121:PH5	122:PH4	123:PH3	124:PH6	125:PA0	126:PH7
127:PA3	128:PA2	129:PA1	130:PA7	131:PA5	132:PD3	133:PA6
134:PA4	135:PD7	136:PD0	137:PD1	138:PD2	139:PD6	140:PD4
141:PD5	142:PE0	143:PE3	144:PE4	145:PE5	146:PE2	147:PF1
148:PF0	149:PE1	150:PF2	151:PF5	152:PE6	153:PE7	154:PF4
155:PF6	156:PF7	157:PF3	158:P80	159:P81	160:P82	161:P83
162:P84	163:P85					

Note: Terminal list to which JTAG can be scanned.

23.4 Instructions Supported by the JTAG Controller Cells

This section describes the instructions supported by the JTAG controller cells of the TMP19A44.

23.4.1 EXTEST Instruction

The EXTEST instruction is used for external interconnect test. If this instruction is issued, the BSR cells at output pins output test patterns in the Update-DR state, and the BSR cells at input pins capture test results in the Capture-DR state.

Before the EXTEST instruction is selected, the boundary scan register is usually initialized using the SAMPLE/PRELOAD instruction. If the boundary scan register has not been initialized, there is the possibility that indeterminate data will be transmitted in the Update-DR state and bus conflicts may occur between ICs. Fig. 23-8 shows the flow of data while the EXTEST instruction is selected.

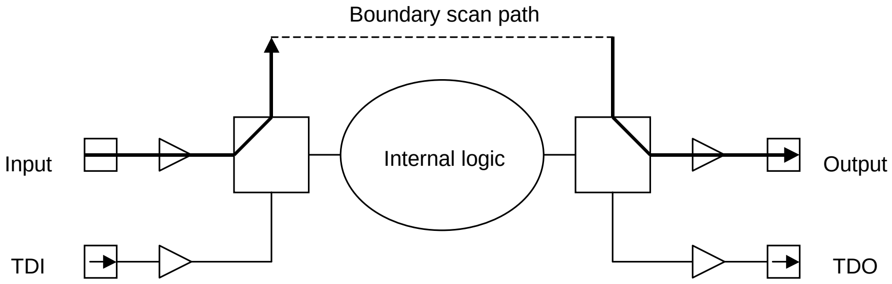


Fig. 23-8 Flow of Data While the EXTEST Instruction Is Selected

The basic external interconnect test procedure is as follows:

1. Initialize the TAP controller to put it in the Test-Logic-Reset state.
2. Load the SAMPLE/PRELOAD instruction into the instruction register. This allows the boundary scan register to be connected between TDI and TDO.
3. Initialize the boundary scan register by shifting in determinate data.
4. Load the initial test data into the boundary scan register.
5. Load the EXTEST instruction into the instruction register.
6. Capture the data applied to the input pin and input it into the boundary scan register.
7. Shift out the captured data while simultaneously shifting in the next test pattern.
8. Output to the output pin the test pattern that was shifted into the boundary scan register for output.

Repeat steps 6 through 8 for each test pattern.

「EXTEST Instruction ：CPU is working and note the terminal input, please when using it.」
「EXTEST Instruction：Please test after releasing system reset when using it.」

23.4.2 SAMPLE and PRELOAD Instructions

The SAMPLE and PRELOAD instructions are used to connect TDI and TDO by way of the boundary scan register.

Each instruction performs the function described below:

- The SAMPLE instruction is used to monitor the I/O pad of an IC. While SAMPLE is monitoring the I/O pads, the internal logic is not disconnected from the I/O pins of an IC. This instruction is executed in the Capture-DR state. A main function of SAMPLE is to read values of the I/O pins of an IC at the rising edge of TCK during normal functional operation. Fig. 23-9 shows the flow of data while the SAMPLE instruction is selected.

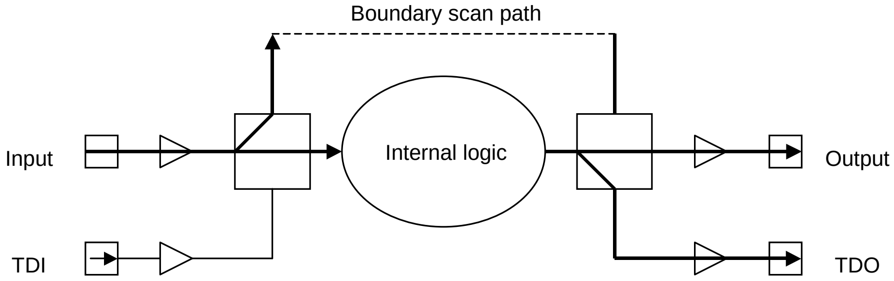


Fig. 23-9 Flow of Data While SAMPLE Is Selected

- The PRELOAD instruction is used to initialize the boundary scan register before selecting other instructions. For example, the boundary scan register is initialized using PRELOAD before selecting the EXTEST instruction, as previously explained. PRELOAD shifts data into the boundary scan register without affecting the normal operation of the system logic. Fig. 23-10 shows the flow of data while the PRELOAD instruction is selected.

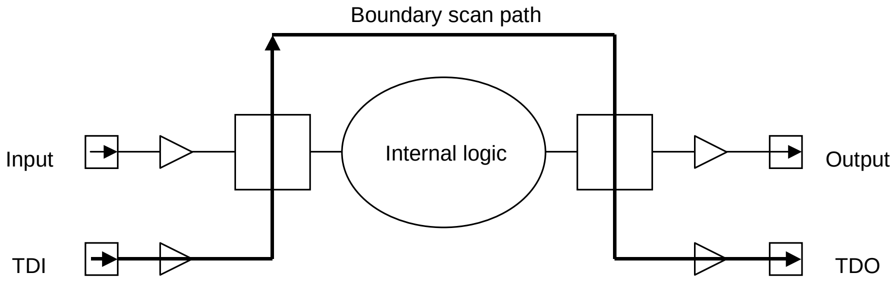


Fig. 23-10 Flow of Test Data While PRELOAD Is Selected

23.4.3 BYPASS Instruction

When conducting the type of test in which an IC does not need to be controlled or monitored, the BYPASS instruction is used to form the shortest serial path bypassing an IC by connecting the bypass register between JTDI and JTDO. The BYPASS instruction does not affect the normal operation of the system logic implemented on a chip. Data passes through the bypass register while the BYPASS instruction is selected, as shown in Fig. 23-11.

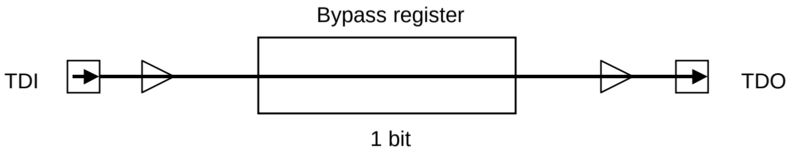


Fig. 23-11 Flow of Data While the Bypass Register Is Selected

23.5 Points to Note

This section describes the points to note regarding JTAG boundary scan operations implemented in this processor.

- The X2 and X1 signal pads do not comply with JTAG.

- To reset the JTAG circuit, execute either of the following:

  ① Initialize the JTAG circuit by asserting $\overline{\text{TRST}}$, and then deassert $\overline{\text{TRST}}$.

  ② Set the TMS pin to "1," and supply TCK with more than 5 clocks.

24. Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

24.1 Flash Memory

24.1.1 Features

1) Memory capacity

- The TMP19A44F10XBG device contains 8M bits (1024 kB) of flash memory capacity. The memory area consists of 10 independent memory blocks (128 kB × 7, 64 kB × 1 and 32 kB × 2) to enable independent write access to each block. When the CPU is to access the internal flash memory, 32-bit data bus width is used.
- The TMP19A44FEXBG device contains 6M bits (768 kB) of flash memory capacity. The memory area consists of 8 independent memory blocks (128 kB × 5, 64 kB × 1 and 32 kB × 2).
- The TMP19A44FDAXBG device contains 4M bits (512 kB) of flash memory capacity. The memory area consists of 6 independent memory blocks (128 kB × 3, 64 kB × 1 and 32 kB × 2).
- Flash memory access: Interleave access is used in this device.

2) Write/erase time

Write time: 0.5 sec/128 Kbyte (Typ.)
TMP19A44F10XBG: 4sec (Typ)/ TMP19A44FDAXBG: 2 sec (Typ）
Erase: 100 msec/1 block (Typ.)
TMP19A44F10XBG: 1sec (Typ)/ TMP19A44FDAXBG: 0.6 sec (Typ)

(Note) The above values are theoretical values not including data transfer time.
The write time per chip depends on the write method to be used by the user.

3) Programming method

The onboard programming mode is available for the user to program (rewrite) the device while it is mounted on the user's board.

4-1) User boot mode
The user's original rewriting method can be supported.

4-2) Single boot mode
The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

Rewriting method

The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if the user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.

This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. The above described protection function is enabled for each area respectively. When the user removes protection, the internal data is automatically erased before the protection is actually removed.

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function Automatic multiple block erase (supported to the chip level)

24.1.2 Block Diagram of the Flash Memory Section

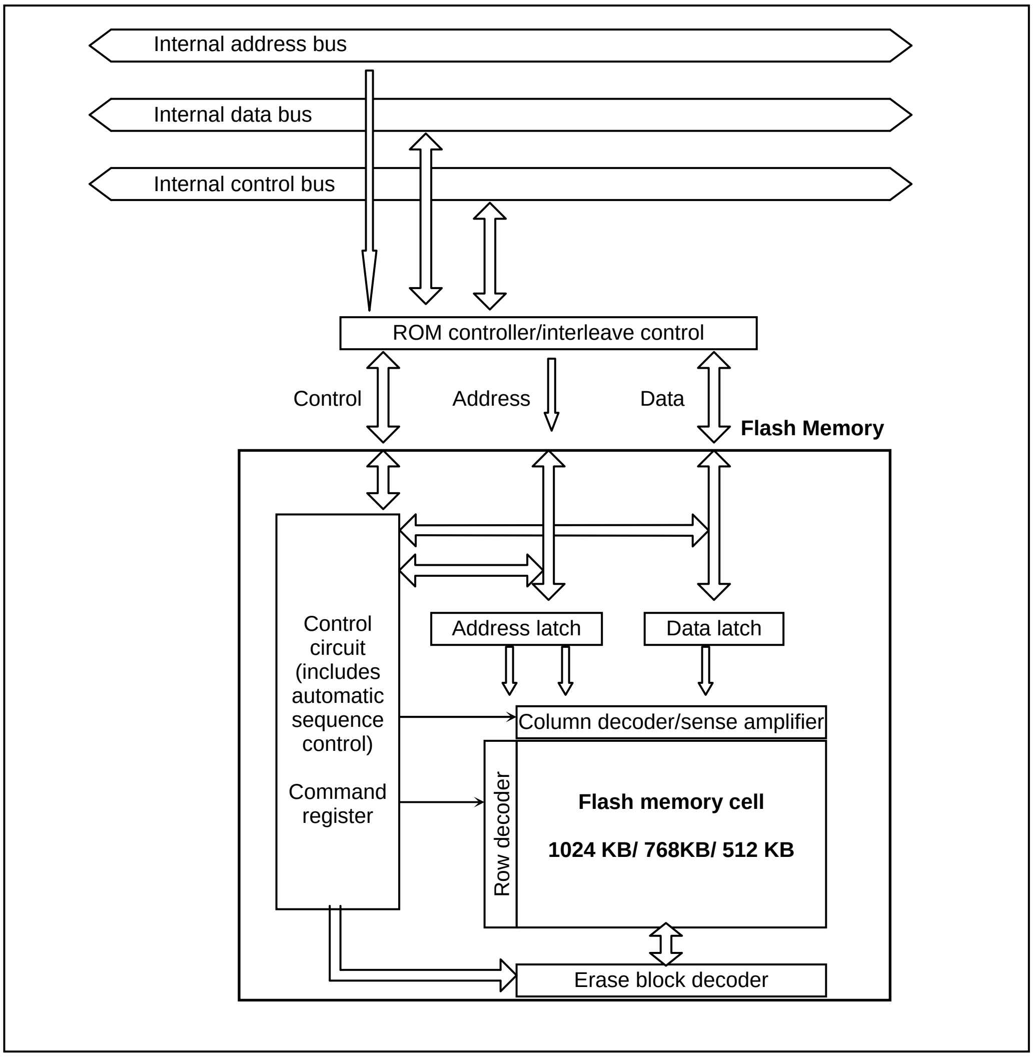


Fig. 24.1 Block Diagram of the Flash Memory Section

24.2 Operation Mode

This device has three operation modes including the mode not to use the internal flash memory.

Table 24.1 Operation Modes

Operation mode		Operation details
Single chip mode		After reset is cleared, it starts up from the internal flash memory.
	Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user’s card, are defined. The former is referred to as "normal mode" and the latter "user boot mode." The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "00" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.
	User boot mode	
Single boot mode		After reset is cleared, it starts up from the internal Boot ROM (Mask ROM). In the Boot ROM, an algorithm to enable flash memory rewriting on the user’s set through the serial port of this device is programmed. By connecting to an external host computer through the serial port, the internal flash memory can be programmed by transferring data in accordance with predefined protocols.

Among the flash memory operation modes listed in the above table, the User Boot mode and the Single Boot mode are the programmable modes. These two modes, the User Boot mode and the Single Boot mode, are referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ input pin while the device is in reset status.

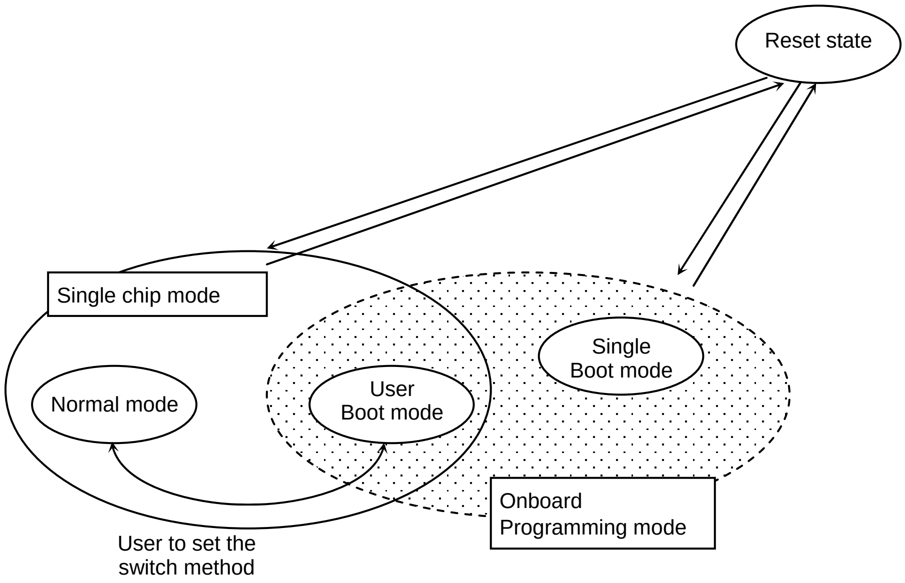
After the level is set, the CPU starts operation in the selected operation mode when the reset condition is removed. Regarding the TEST0, TEST1, and BOOT pins, be sure not to change the levels during operation once the mode is selected.

The mode setting method and the mode transition diagram are shown below:

Table 24.2 Operation Mode Setting

Operation mode	Input pin	
	$\overline{\text{RESET}}$	$\overline{\text{BOOT}}$
Single chip mode	0 to 1	1
Single boot mode	0 to 1	0

Fig. 24.2 Mode Transition Diagram



24.2.1 Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the $\overline{\text{RESET}}$ input is held at "0" for a minimum duration of 12 system clocks (1.2 μs with 10MHz oscillator operation; the "1/8" clock gear mode is applied after reset).

(Note 1) Regarding power-on reset of devices with internal flash memory;

For devices with internal flash memory, it is necessary to apply "0" to the $\overline{\text{RESET}}$ inputs upon power on for a minimum duration of 500 microseconds regardless of the operating frequency.

(Note 2) While flash programming or deletion is in progress, at least 0.5 microseconds of reset period is required regardless of the system clock frequency.

24.2.2 User Boot mode(Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of 19A44 in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete/ writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental writing/ erasing during subsequent Single-Chip (Normal mode) operations. All the interruption including a non-maskable are inhibited at User Boot Mode.

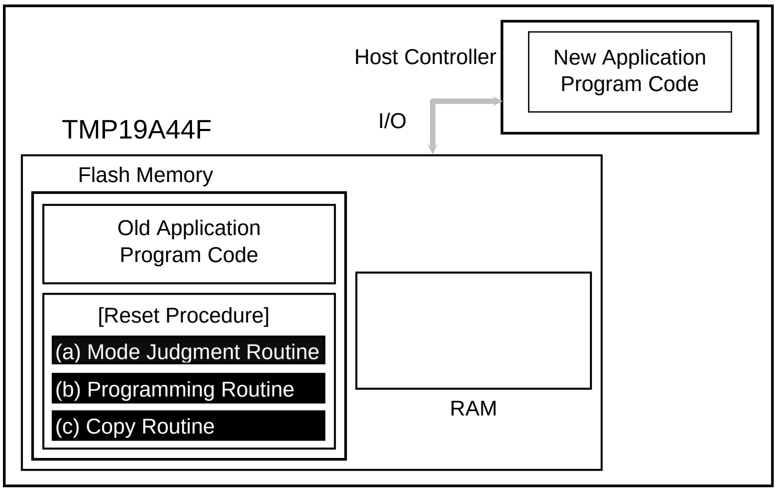
(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to "On-board Programming of Flash Memory (Rewrite/Erase)".

User Boot Mode

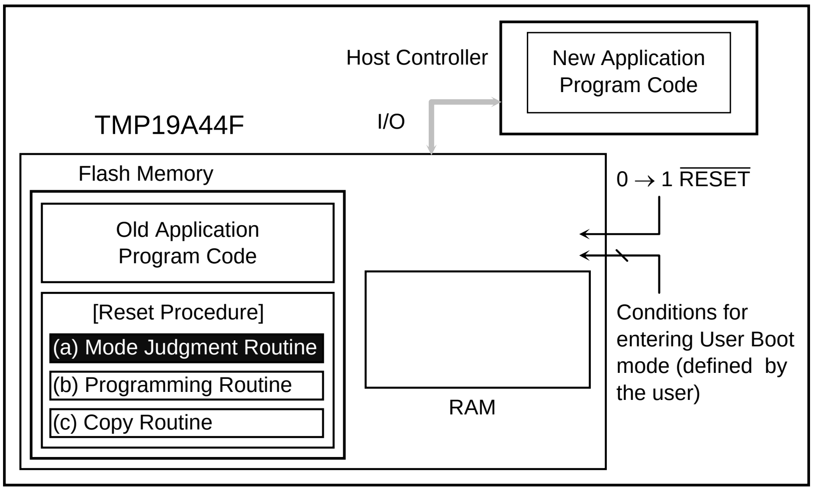
(1-A) Method 1: Storing a Programming Routine in the Flash Memory

(1) Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP19A44F on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

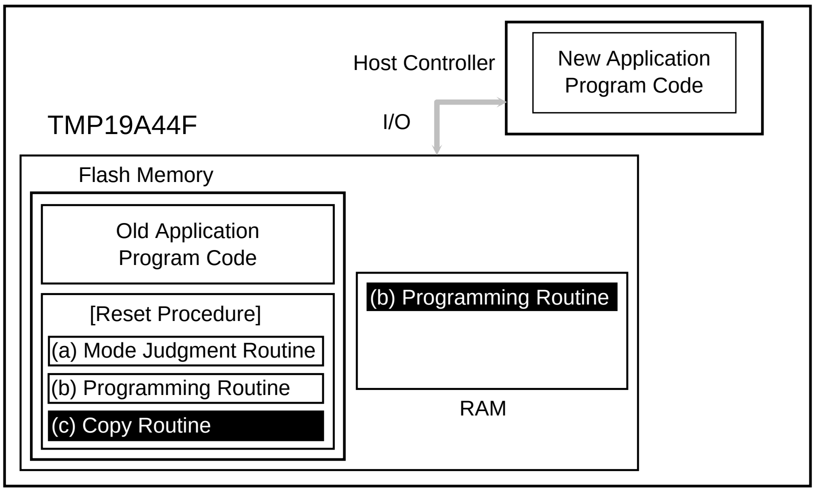
- Mode judgment routine: Code to determine whether or not to switch to User Boot mode
- Programming routine: Code to download new program code from a host controller and re-program the flash memory
- Copy routine: Code to copy the flash programming routine from the TMP19A44F flash memory to either the TMP19A44F on-chip RAM or external memory device.



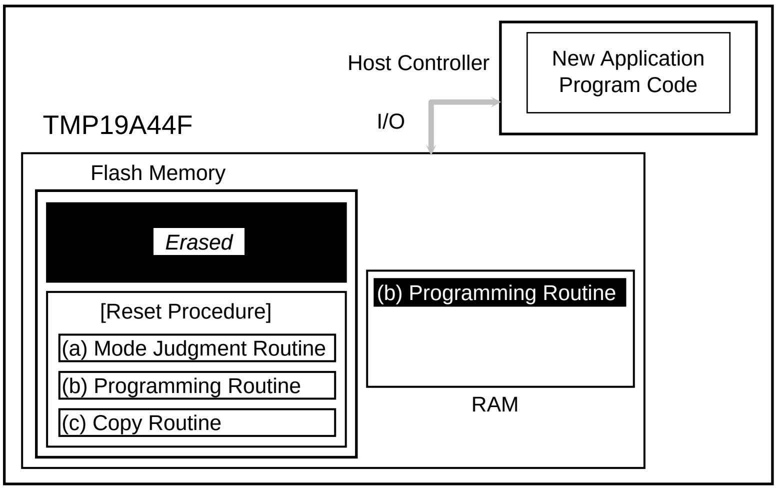
(2) After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP19A44F flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be globally disabled while in User Boot mode.)


Host Controller
New Application Program Code
I/O
TMP19A44F
Flash Memory
Old Application Program Code
[Reset Procedure]
(a) Mode Judgment Routine
(b) Programming Routine
(c) Copy Routine
RAM
0 → 1 RESET
Conditions for entering User Boot mode (defined by the user)

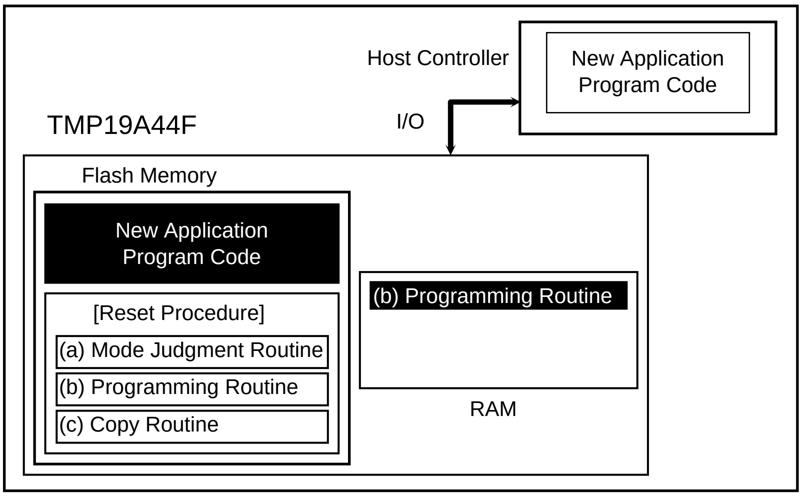
(3) Once User Boot mode is entered, execute the copy routine to copy the flash programming routine to either the TMP19A44F on-chip RAM.



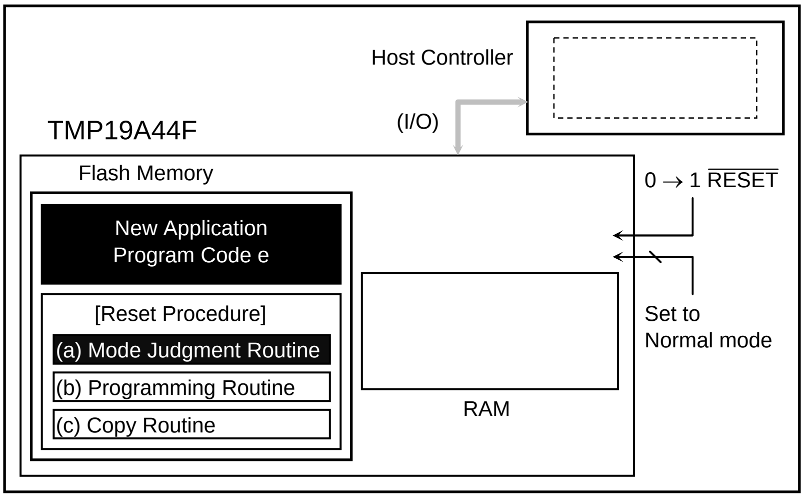
(4) Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block after releasing the protection for accidental writing/ erasing of the old application program code.



(5) Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection for writing/ erasing of that flash block.



(6) Drive $\overline{\text{RESET}}$ low to reset the TMP19A44F. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



(1-B) Method 2: Transferring a Programming Routine from an External Host

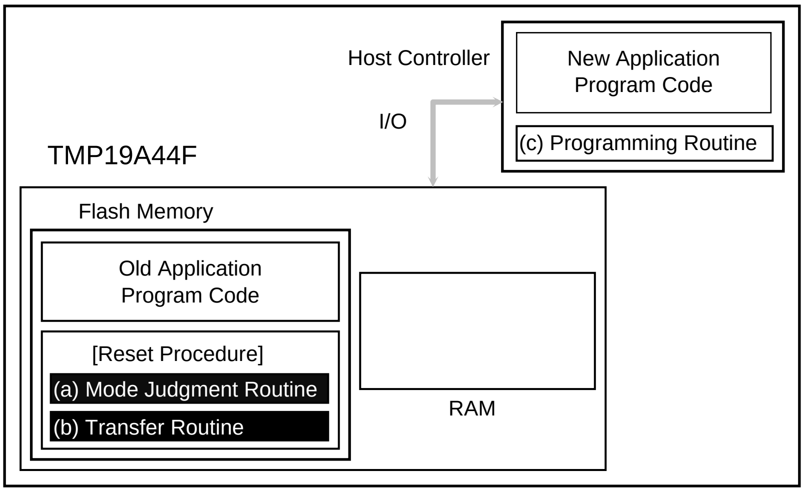
(1) Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP19A44F on a printed circuit board, write the following program routines into an arbitrary flash block using programming equipment.

Mode judgment routine: Code to determine whether or not to switch to User Boot mode

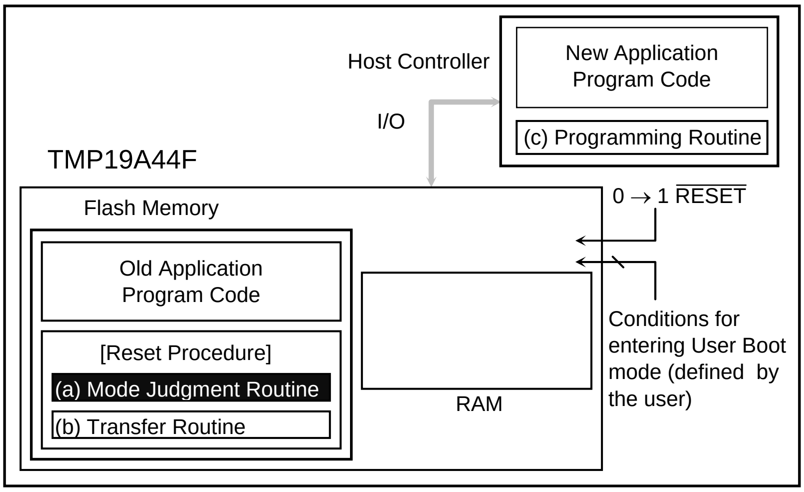
Transfer routine: Code to download new program code from a host controller

Also, prepare a programming routine on the host controller:

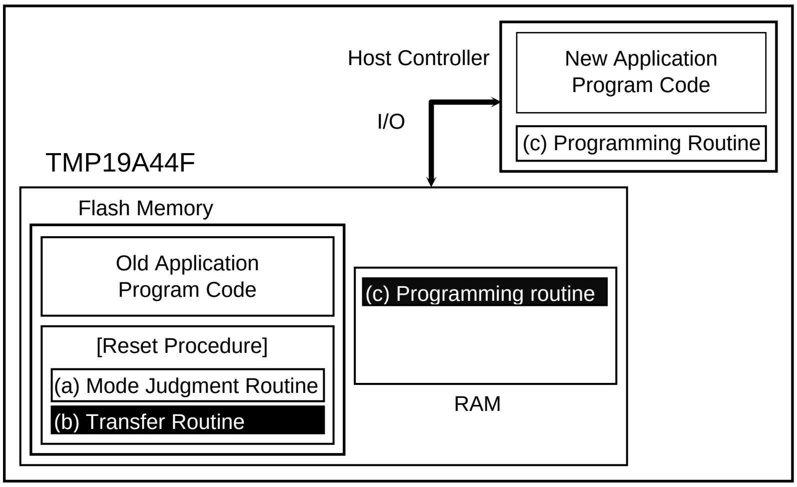
Programming routine: Code to download new program code from an external host controller and re-program the flash memory



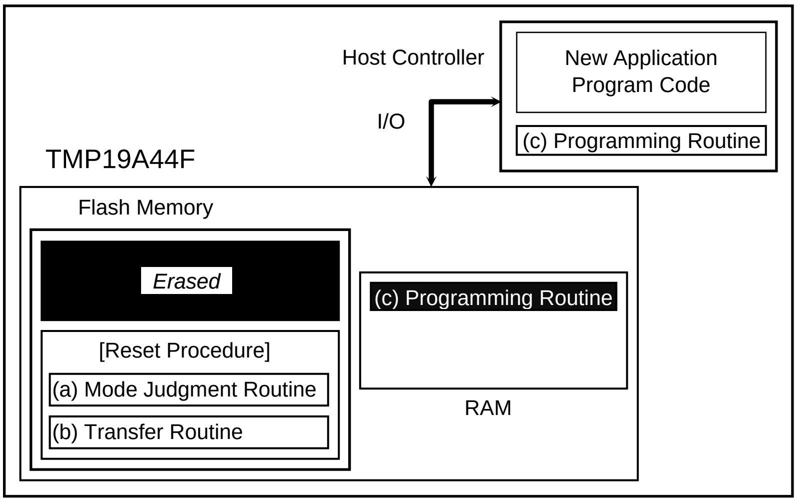
(2) After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP19A44F flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be globally disabled while in User Boot mode.)


Host Controller
New Application Program Code
(c) Programming Routine
I/O
TMP19A44F
0 → 1 RESET
Flash Memory
Old Application Program Code
[Reset Procedure]
(a) Mode Judgment Routine
(b) Transfer Routine
RAM
Conditions for entering User Boot mode (defined by the user)

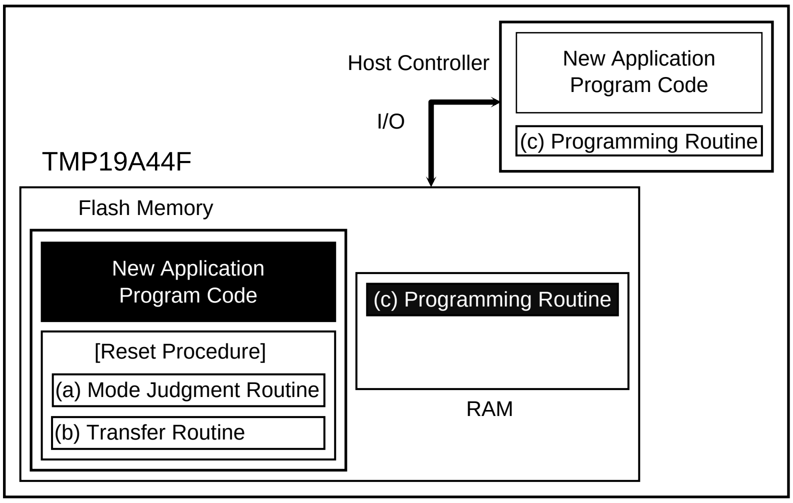
(3) Once User Boot mode is entered, execute the transfer routine to download the flash programming routine from the host controller to the TMP19A44F on-chip RAM.



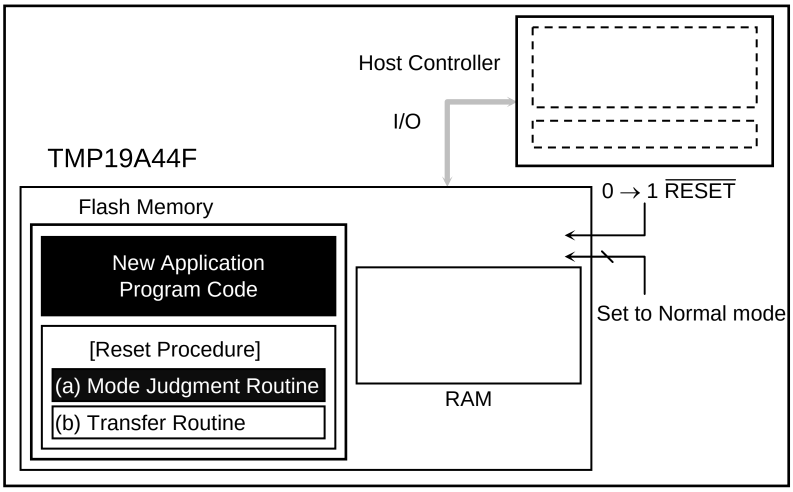
(4) Jump program execution to the flash programming routine in the on-chip RAM to erase a flash block after releasing the protection for accidental writing/ erasing of the old application program code.



(5) Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection for writing/ erasing of that flash block.



Drive $\overline{\text{RESET}}$ low to reset the TMP19A44F. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



24.2.3 Single Boot Mode

In Single Boot mode, the flash memory can be re-programmed by using a program contained in the TMP19A44F on-chip boot ROM. This boot ROM is a masked ROM. When Single Boot mode is selected upon reset, the boot ROM is mapped to the address region including the interrupt vector table while the flash memory is mapped to an address region different from it.

Single Boot mode allows for serial programming of the flash memory. Channel 0 of the SIO (SIO0) of the TMP19A44F is connected to an external host controller. Via this serial link, a programming routine is downloaded from the host controller to the TMP19A44F on-chip RAM. Then, the flash memory is re-programmed by executing the programming routine. The host sends out both commands and programming data to re-program the flash memory.

Communications between the SIO0 and the host must follow the protocol described later. To secure the contents of the flash memory, the validity of the application's password is checked before a programming routine is downloaded into the on-chip RAM. If password matching fails, the transfer of a programming routine itself is aborted.

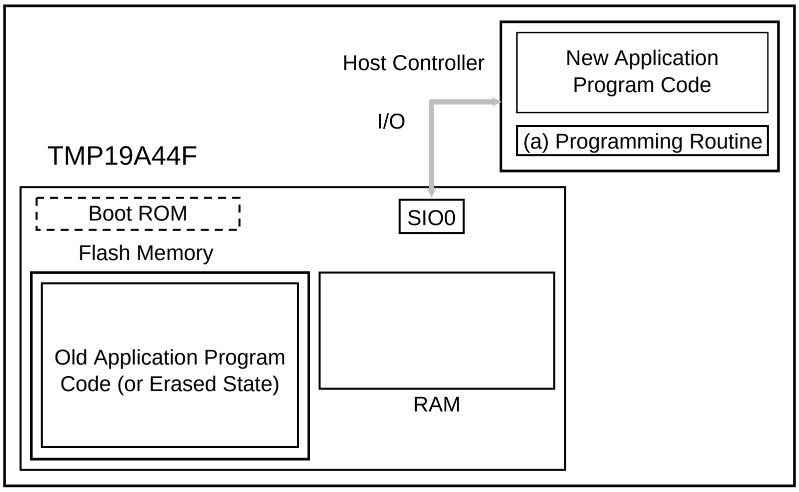
As in the case of User Boot mode, all interrupts including the non-maskable (NMI) interrupt must be globally disabled in Single Boot mode while the flash memory is being erased or programmed. In Single Boot mode, the boot-ROM programs are executed in Normal mode.

Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental writing/ erasing during subsequent Single-Chip (Normal mode) operations. For a detailed description of the erase and program sequence, refer to On-Board Programming and Erasure.

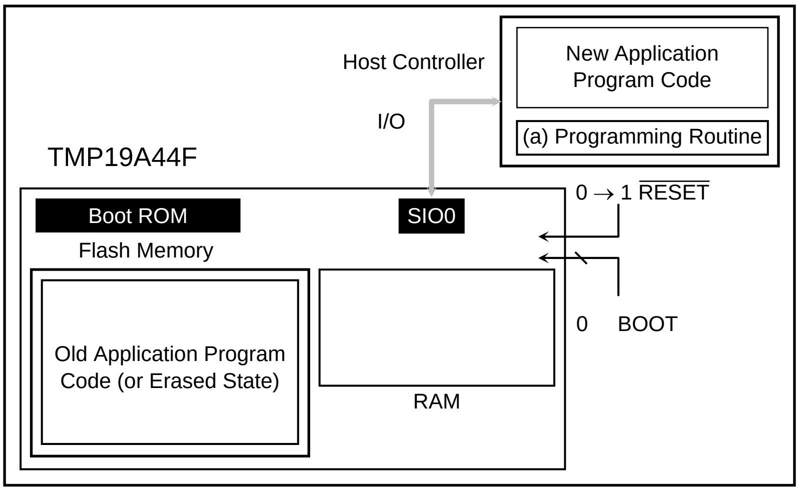
Boot Mode

(2-A) General Procedure: Using the Program in the On-Chip Boot ROM

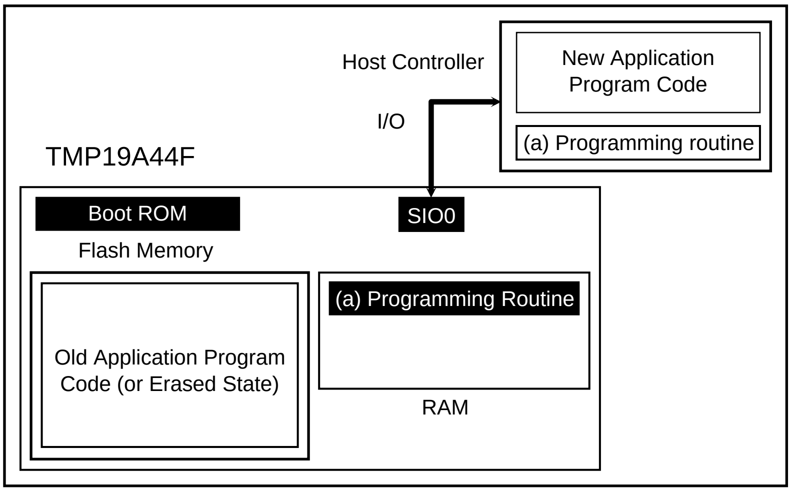
(1) The flash block containing the older version of the program code need not be erased before executing the programming routine. Since a programming routine and programming data are transferred via the SIO0, the SIO0 must be connected to a host controller. Prepare a programming routine on the host controller.



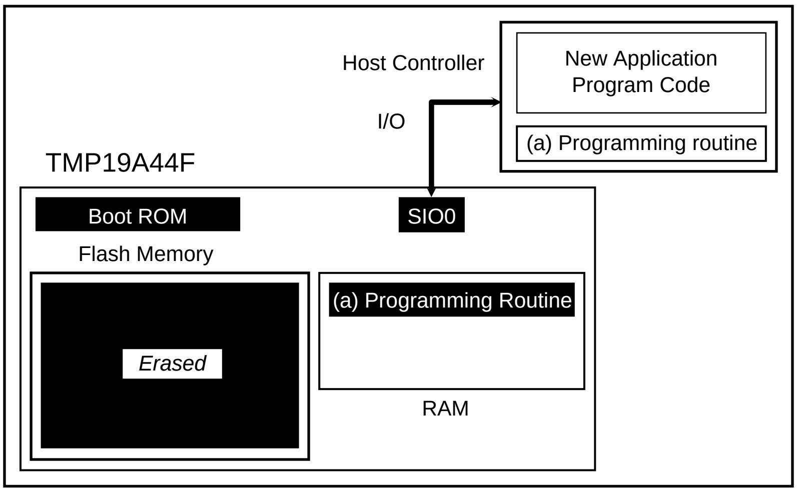
(2) Reset the TMP19A44F with the mode setting pins held at appropriate logic values, so that the CPU re-boots from the on-chip boot ROM. The 12-byte password transferred from the host controller is first compared to the contents of special flash memory locations. (If the flash block has already been erased, the password is 0xFFFF.)


Host Controller
New Application Program Code
(a) Programming Routine
I/O
TMP19A44F
Boot ROM
SIO0
0 → 1 $\overline{\text{RESET}}$
Flash Memory
0 BOOT
Old Application Program Code (or Erased State)
RAM

(3) If the password was correct, the boot program downloads, via the SIO0, the programming routine from the host controller into the on-chip RAM of the TMP19A44F. The programming routine must be stored in the address range 0xFFFD_A400 – 0xFFFD_FFFF.



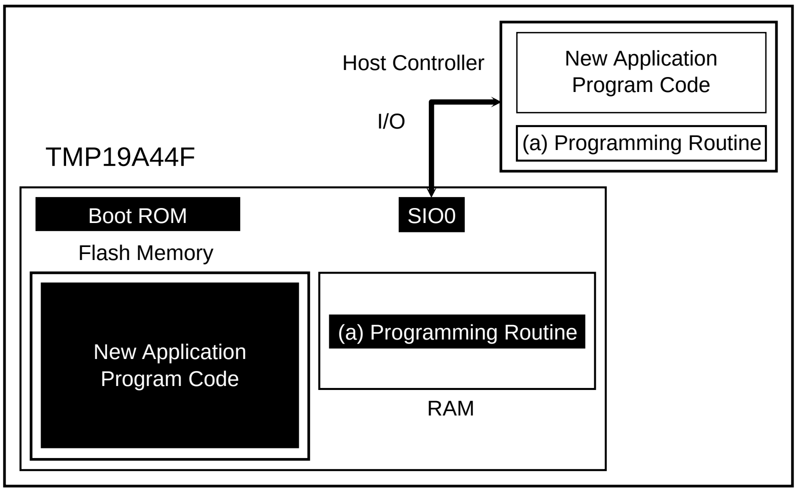
(4) The CPU jumps to the programming routine in the on-chip RAM to erase the flash block containing the old application program code. The Block Erase or Chip Erase command may be used.



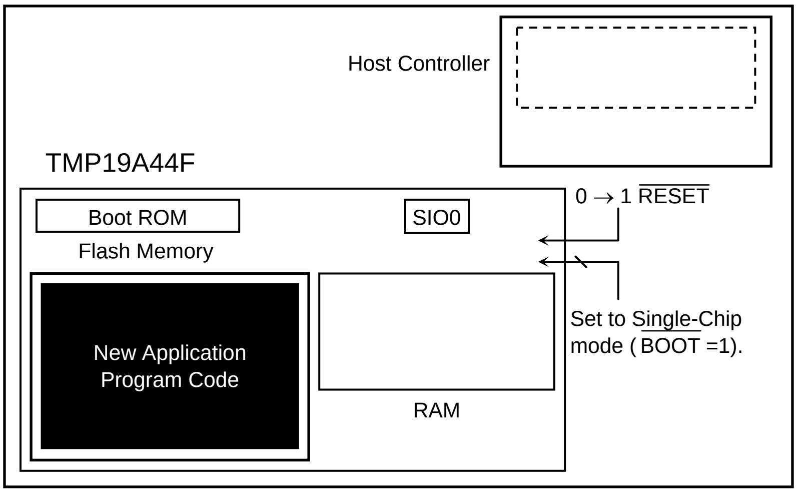
(5) Next, the programming routine downloads new application program code from the host controller and programs it into the erased flash block. Once programming is complete, protection for writing/ erasing of that flash block is turned on.

It is not allowed to move program control from the programming routine back to the boot ROM.

In the example below, new program code comes from the same host controller via the same SIO channel as for the programming routine. However, once the programming routine has begun to execute, it is free to change the transfer path and the source of the transfer. Create board hardware and a programming routine to suit your particular needs.



(6) When programming of the flash memory is complete, power off the board and disconnect the cable leading from the host to the target board. Turn on the power again so that the TMP19A44F re-boots in Single-Chip (Normal) mode to execute the new program.


Host Controller
TMP19A44F
0 → 1 $\overline{\text{RESET}}$
Boot ROM
SIO0
Flash Memory
Set to Single-Chip mode ($\overline{\text{BOOT}}$ =1).
New Application Program Code
RAM

(1) Mode Setting

For on-board programming, boot the TMP19A44F in Single Boot mode, as follows:

$\overline{\text{BOOT}}$ = 0

$\overline{\text{RESET}}$ = 0 → 1

Set the $\overline{\text{RESET}}$ input at logic 0, and the $\overline{\text{BOOT}}$ input at the logic values shown above, and then release $\overline{\text{RESET}}$ (high).

(2) Memory Map

Fig. 24.3 shows a comparison of the memory maps in Normal and Single Boot modes. In single Boot mode, the on-chip flash memory is mapped to physical addresses (0x000_0000 through 0x4007_FFFF), virtual addresses (0x0000_0000 through 0x0007_FFFF), and the on-chip boot ROM is mapped to physical addresses 0x1FC0_0000 through 0x1FC0_0FFF.

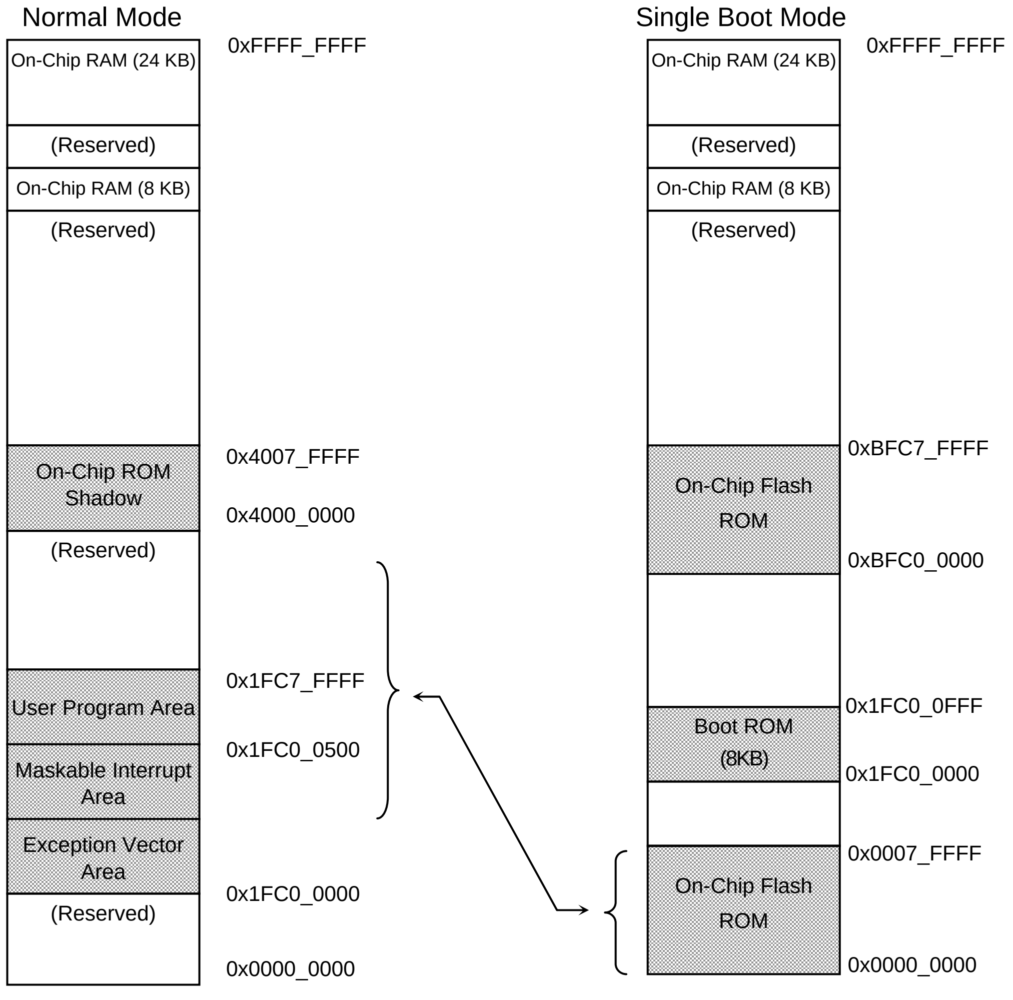


Fig. 24.3 Memory Maps for Normal and Single Boot Modes (Physical Addresses)

(3) Interface Specification

In Single Boot mode, an SIO channel is used for communications with a programming controller. Both UART (asynchronous) and I/O Interface (synchronous) modes are supported. The communication formats are shown below. In the subsections that follow, virtual addresses are indicated, unless otherwise noted.

- UART mode

Communication channel:	SIO Channel 0 (SIO0)
Transfer mode:	UART (asynchronous) mode, half-duplex, LSB first
Data length:	8 bits
Parity bits:	None
STOP bits:	1
Baud rate:	Arbitrary baud rate

- I/O Interface mode

Communication channel:	SIO Channel 0 (SIO0)
Transfer mode:	I/O Interface mode, full-duplex, LSB first
Synchronization clock (SCLK0):	Input mode
Handshaking signal:	P63 output mode
Baud rate:	Arbitrary baud rate

Table 24.3 Required Pin Connections

Pin		Interface	
		UART Mode	I/O Interface Mode
Power Supply Pins	DVCC3	Required	Required
	DVSS	Required	Required
Mode-Setting Pin	$\overline{\text{BOOT}}$	Required	Required
Reset Pin	$\overline{\text{RESET}}$	Required	Required
Communication Pins	TXD0 (P60)	Required	Required
	RXD0 (P61)	Required	Required
	SCLK0 (P62)	Not Required	Required (Input Mode)
	P63	Not Required	Required (OUTPUT Mode)

(4) Data Transfer Format

The host controller is to issue one of the commands listed in Table 24.4 to the target board. Table 24.6 illustrate the sequence of two-way communications that should occur in response to each command.

Table 24.4 Single Boot Mode Commands

Code	Command
10H	RAM Transfer

(5) Restrictions on internal memories

Single Boot Mode places restrictions on the internal RAM and ROM as shown in Table 24.5 Restrictions in Single Boot Mode.

Table 24.5 Restrictions in Single Boot Mode

Memory	Details
Internal RAM	BOOT ROM is mapped to 0xFFFF_A000 to 0xFFFF_A3FF. Store the RAM transfer program in 0xFFFF_A400～0xFFFF_FFFF.
Internal ROM	0x0000_0470 to 0x0000_047F are assigned for storing software ID information and passwords. Storing program in these addresses is not recommendable.

Table 24.6 Transfer Format for the RAM Transfer Command

	Byte	Data Transferred from the Controller to the TMP19A44	Baud Rate	Data Transferred from the TMP19A44 to the Controller
Boot ROM	1st byte	Serial operation mode and baud rate For UART mode 86H For I/O Interface mode 30H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK for the serial operation mode byte For UART mode Normal acknowledge 86H (The boot program aborts if the baud rate is can not be set correctly.) For I/O Interface mode Normal acknowledge 30H
	3rd byte	Command code (10H)		—
	4th byte	—		ACK for the command code byte (Note 2) Normal acknowledge 10H Negative acknowledge x1H Communication error x8H
	5th byte thru 16th byte	Password sequence (12 bytes) (0x3F8F_FFF4 thru 0x3F8F_FFFF)		—
	17th byte	Checksum value for bytes 5–16		—
	18th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge x1H Communication error x8H
	19th byte	RAM storage start address (bits 31–24)		—
	20th byte	RAM storage start address (bits 23–16)		—
	21st byte	RAM storage start address (bits 15–8)		—
	22nd byte	RAM storage start address (bits 7–0)		—
	23rd byte	RAM storage byte count (bits 15–8)		—
	24th byte	RAM storage byte count (bits 7–0)		—
	25th byte	Checksum value for bytes 19–24		—
	26th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Negative acknowledge x1H Communication error x8H
	27th byte thru mth byte	RAM storage data		—
	(m + 1)th byte	Checksum value for bytes 27–m		—
	(m + 2)th byte	—		ACK for the checksum byte (Note 2) Normal acknowledge 10H Non-acknowledge x1H Communications error x8H
RAM	(m + 3)th byte	—		Jump to RAM storage start address

Note 1: In I/O Interface mode, the baud rate for the transfers of the first and second bytes must be 1/16 of the desired baud rate.

Note 2: In case of any negative acknowledge, the boot program returns to a state in which it waits for a command code (3rd byte). In I/O Interface mode, if a communication error occurs, a negative acknowledge does not occur.

Note 3: The 19th to 25th bytes must be within the RAM address range 0xFFFF_A400–0xFFFF_FFFF

.

(6) Overview of the Boot Program Commands

When Single Boot mode is selected, the boot program is automatically executed on startup. The boot program offers these four commands, the details of which are provided on the subsections 1) through 4).

1. RAM Transfer command

   The RAM Transfer command stores program code transferred from a host controller to the on-chip RAM and executes the program once the transfer is successfully completed. The maximum program size is 24 Kbytes. The RAM storage start address must be within the range.

   The RAM Transfer command can be used to download a flash programming routine of your own; this provides the ability to control on-board programming of the flash memory in a unique manner. The programming routine must utilize the flash memory command sequences described in Section 24.3.

   Before initiating a transfer, the RAM Transfer command checks a password sequence coming from the controller against that stored in the flash memory. If they do not match, the RAM Transfer command aborts.

   Once the RAM Transfer command is complete, the whole on-chip RAM is accessible.

1) RAM Transfer Command (see Table 24.6)

1. The 1st byte specifies which one of the two serial operation modes is used. For a detailed description of how the serial operation mode is determined, see Section 0. If it is determined as UART mode, the boot program then checks if the SIO0 is programmable to the baud rate at which the 1st byte was transferred. During the first-byte interval, the RXE bit in the SC0MOD0 register is cleared.

- To communicate in UART mode

  Send, from the controller to the target board, 86H in UART data format at the desired baud rate. If the serial operation mode is determined as UART, then the boot program checks if the SIO0 can be programmed to the baud rate at which the first byte was transferred. If that baud rate is not possible, the boot program aborts, disabling any subsequent communications.

- To communicate in I/O Interface mode

  Send, from the controller to the target board, 30H in I/O Interface data format at 1/16 of the desired baud rate. Also send the 2nd byte at the same baud rate. Then send all subsequent bytes at a rate equal to the desired baud rate.

  In I/O Interface mode, the CPU sees the serial receive pin as if it were a general input port in monitoring its logic transitions. If the baud rate of the incoming data is high or the chip’s operating frequency is high, the CPU may not be able to keep up with the speed of logic transitions. To prevent such situations, the 1st and 2nd bytes must be transferred at 1/16 of the desired baud rate; then the boot program calculates 16 times that as the desired baud rate.

  When the serial operation mode is determined as I/O Interface mode, the SIO0 is configured for SCLK Input mode. Beginning with the third byte, the controller must ensure that its AC timing restrictions are satisfied at the selected baud rate. In the case of I/O Interface mode, the boot program does not check the receive error flag; thus there is no such thing as error acknowledge (x8H).

2. The 2nd byte, transmitted from the target board to the controller, is an acknowledge response to the 1st byte. The boot program echoes back the first byte: 86H for UART mode and 30H for I/O Interface mode.

- UART mode

  If the SIO0 can be programmed to the baud rate at which the 1st byte was transferred, the boot program programs the SC0BRCR and sends back 86H to the controller as an acknowledge. If the SIO0 is not programmable at that baud rate, the boot program simply aborts with no error indication.

  Following the 1st byte, the controller should allow for a time-out period of five seconds. If it does not receive 86H within the allowed time-out period, the controller should give up the communication.

  The boot program sets the RXE bit in the SC0MOD0 register to enable reception before loading the SIO transmit buffer with 86H.

- I/O Interface mode

  The boot program programs the SC0MOD0 and SC0CR registers to configure the SIO0 in I/O Interface mode (clocked by the rising edge of SCLK0), writes 30H to the SC0BUF. Then, the SIO0 waits for the SCLK0 signal to come from the controller. Following the transmission of the 1st byte, the controller should send the SCLK clock to the target board after a certain idle time (several microseconds). This must be done at 1/16 the desire baud rate. If the 2nd byte, which is from the target board to the controller, is 30H, then the controller should take it as a go-ahead. The controller must then deliver the 3rd byte to the target board at a rate equal to the desired baud rate. The boot program sets the RXE bit in the SC0MOD0 register to enable reception before loading the SIO transmit buffer with 30H.

3. The 3rd byte, which the target board receives from the controller, is a command. The code for the RAM Transfer command is 10H.

4. The 4th byte, transmitted from the target board to the controller, is an acknowledge response to the 3rd byte. Before sending back the acknowledge response, the boot program checks for a receive error. If there was a receive error, the boot program transmits x8H and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command. When the SIO0 is configured for I/O Interface mode, the boot program does not check for a receive error.

If the 3rd byte is equal to any of the command codes listed in Table 24.4, the boot program echoes it back to the controller. When the RAM Transfer command was received, the boot program echoes back a value of 10H and then branches to the RAM Transfer routine. Once this branch is taken, a password check is done. Password checking is detailed in Section “Password”.

If the 3rd byte is not a valid command, the boot program sends back x1H to the controller and returns to the state in which it waits for a command again. In this case, the upper four bits of the acknowledge response are undefined — they hold the same values as the upper four bits of the previously issued command.

5. The 5th to 16th bytes, which the target board receives from the controller, are a 12-byte password. The 5th byte is compared to the contents of address 0x0000_0474 in the flash memory; the 6th byte is compared to the contents of address 0x0000_0475 in the flash memory; likewise, the 16th byte is compared to the contents of address 0x0000_047F in the flash memory. If the password checking fails, the RAM Transfer routine sets the password error flag.

6. The 17th byte is a checksum value for the password sequence (5th to 16th bytes). To calculate the checksum value for the 12-byte password, add the 12 bytes together, drop the carries and take the two’s complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section “Checksum Calculation”.

7. The 18th byte, transmitted from the target board to the controller, is an acknowledge response to the 5th to 17th bytes.

First, the RAM Transfer routine checks for a receive error in the 5th to 17th bytes. If there was a receive error, the boot program sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 5th to 16th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

Finally, the RAM Transfer routine examines the result of the password check. The following two cases are treated as a password error. In these cases, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

- Irrespective of the result of the password comparison, all of the 12 bytes of a password in the flash memory are the same value other than FFH.
- Not the entire password bytes transmitted from the controller matched those contained in the flash memory.

When all the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

8. The 19th to 22nd bytes, which the target board receives from the controller, indicate the start address of the RAM region where subsequent data (e.g., a flash programming routine) should be stored. The 19th byte corresponds to bits 31–24 of the address, and the 22nd byte corresponds to bits 7–0 of the address.

9. The 23rd and 24th bytes, which the target board receives from the controller, indicate the number of bytes that will be transferred from the controller to be stored in the RAM. The 23rd byte corresponds to bits 15–8 of the number of bytes to be transferred, and the 24th byte corresponds to bits 7–0 of the number of bytes.

10. The 25th byte is a checksum value for the 19th to 24th bytes. To calculate the checksum value, add all these bytes together, drop the carries and take the two’s complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section.

11. The 26th byte, transmitted from the target board to the controller, is an acknowledge response to the 19th to 25th bytes of data. First, the RAM Transfer routine checks for a receive error in the 19th to 25th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 19th to 24th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again.

The RAM storage start address must be within the range 0xFFFF_A400–0xFFFF_FFFF.

When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

12. The 27th to mth bytes from the controller are stored in the on-chip RAM of the TMP19A44. Storage begins at the address specified by the 19th–22nd bytes and continues for the number of bytes specified by the 23rd–24th bytes.

13. The (m+1)th byte is a checksum value. To calculate the checksum value, add the 27th to mth bytes together, drop the carries and take the two's complement of the total sum. Transmit this checksum value from the controller to the target board. The checksum calculation is described in details in Section "Checksum Calculation ".

14. The (m+2)th byte is a acknowledge response to the 27th to (m+1)th bytes.

First, the RAM Transfer routine checks for a receive error in the 27th to (m+1)th bytes. If there was a receive error, the RAM Transfer routine sends back 18H and returns to the state in which it waits for a command (i.e., the 3rd byte) again. In this case, the upper four bits of the acknowledge response are the same as those of the previously issued command (i.e., all 1s). When the SIO0 is configured for I/O Interface mode, the RAM Transfer routine does not check for a receive error.

Next, the RAM Transfer routine performs the checksum operation to ensure data integrity. Adding the series of the 27th to (m)th bytes must result in zero (with the carry dropped). If it is not zero, one or more bytes of data has been corrupted. In case of a checksum error, the RAM Transfer routine sends back 11H to the controller and returns to the state in which it waits for a command (i.e., the 3rd byte) again. When the above checks have been successful, the RAM Transfer routine returns a normal acknowledge response (10H) to the controller.

15. If the (m+2)th byte was a normal acknowledge response, a branch is made to the address specified by the 19th to 22nd bytes in 32-bit ISA mode.

7) Acknowledge Responses

The boot program represents processing states with specific codes. Table 24.7 to Table 24.14 show the values of possible acknowledge responses to the received data. The upper four bits of the acknowledge response are equal to those of the command being executed. Bit 3 of the code indicates a receive error. Bit 0 indicates an invalid command error, a checksum error or a password error. Bit 1 and bit 2 are always 0. Receive error checking is not done in I/O Interface mode.

Table 24.7 ACK Response to the Serial Operation Mode Byte

Return Value	Meaning
0x86	The SIO can be configured to operate in UART mode. (See Note)
0x30	The SIO can be configured to operate in I/O Interface mode.

Note: If the serial operation mode is determined as UART, the boot program checks if the SIO can be programmed to the baud rate at which the operation mode byte was transferred. If that baud rate is not possible, the boot program aborts, without sending back any response.

Table 24.8 ACK Response to the Command Byte

Return Value	Meaning
0x?8 (See Note)	A receive error occurred while getting a command code.
0x?1 (See Note)	An undefined command code was received. (Reception was completed normally.)
0x10	The RAM Transfer command was received.

Note: The upper four bits of the ACK response are the same as those of the previous command code.

8) Determination of a Serial Operation Mode

The first byte from the controller determines the serial operation mode. To use UART mode for communications between the controller and the target board, the controller must first send a value of 86H at a desired baud rate to the target board. To use I/O Interface mode, the controller must send a value of 30H at 1/16 the desired baud rate. Fig. 24.4 shows the waveforms for the first byte.

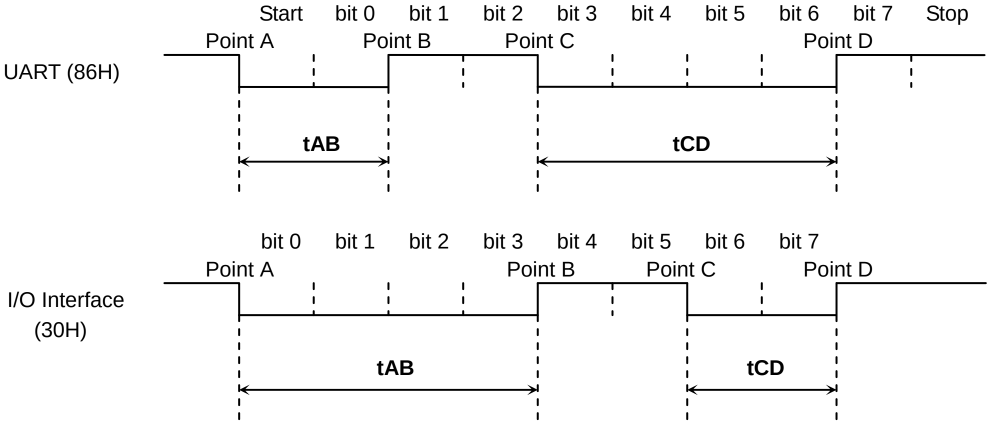


Fig. 24.4 Serial Operation Mode Byte

After $\overline{\text{RESET}}$ is released, the boot program monitors the first serial byte from the controller, with the SIO reception disabled, and calculates the intervals of tAB, tAC and tAD. Fig. 24.5 shows a flowchart describing the steps to determine the intervals of tAB, tAC and tAD. As shown in the flowchart, the boot program captures timer counts each time a logic transition occurs in the first serial byte. Consequently, the calculated tAB, tAC and tAD intervals are bound to have slight errors. If the transfer goes at a high baud rate, the CPU might not be able to keep up with the speed of logic transitions at the serial receive pin. In particular, I/O Interface mode is more prone to this problem since its baud rate is generally much higher than that for UART mode. To avoid such a situation, the controller should send the first serial byte at 1/16 the desired baud rate.

The flowchart in Fig. 24.6 shows how the boot program distinguishes between UART and I/O Interface modes. If the length of tAB is equal to or less than the length of tCD, the serial operation mode is determined as UART mode. If the length of tAB is greater than the length of tCD, the serial operation mode is determined as I/O Interface mode. Bear in mind that if the baud rate is too high or the timer operating frequency is too low, the timer resolution will be coarse, relative to the intervals between logic transitions. This becomes a problem due to inherent errors caused by the way in which timer counts are captured by software; consequently the boot program might not be able to determine the serial operation mode correctly. To prevent this problem, reset UART mode within the programming routine.

For example, the serial operation mode may be determined to be I/O Interface mode when the intended mode is UART mode. To avoid such a situation, when UART mode is utilized, the controller should allow for a time-out period within which it expects to receive an echo-back (86H) from the target board. The controller should give up the communication if it fails to get that echo-back within the allowed time. When I/O Interface mode is utilized, once the first serial byte has been transmitted, the controller should send the SCLK clock after a certain idle time to get an acknowledge response. If the received acknowledge response is not 30H, the controller should give up further communications.

When the intended mode is I/O interface mode, the first byte does not have to be 0x30 as long as tAB is greater than tCD as shown above. 0x91, 0xA1 or 0xB1 can be sent as the first byte code to determine the falling edges of Point A and Point C and the rising edges of Point B and Point D. If tAB is greater than tCD and SIO is selected by the resolution of the operation mode determination, the second byte code is 0x30 even though the transmitted code on the first byte is not 0x30 (The first byte code to determine I/O interface mode is described as 0x30).

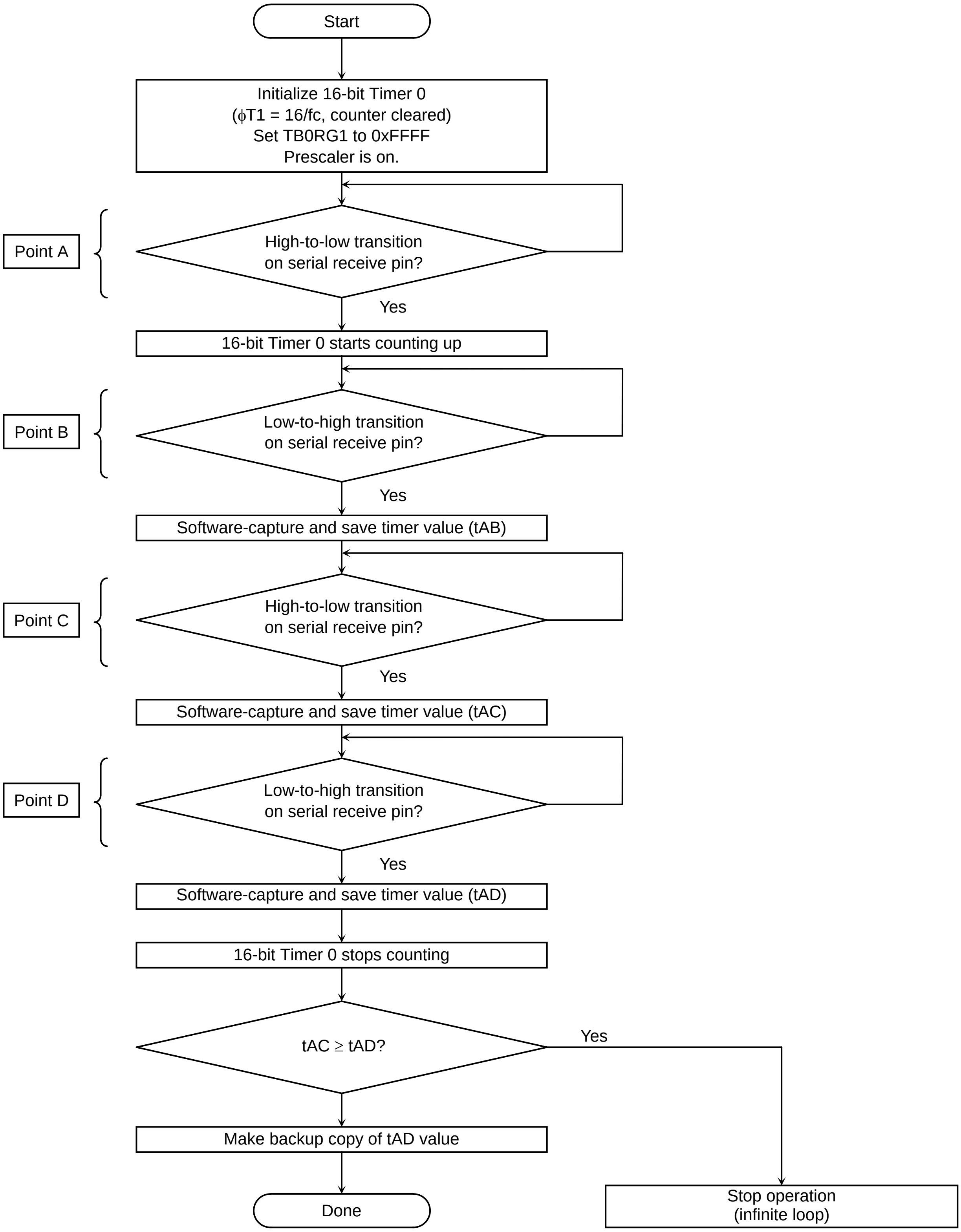

Start
Initialize 16-bit Timer 0
(ϕT1 = 16/fc, counter cleared)
Set TB0RG1 to 0xFFFF
Prescaler is on.
Point A
High-to-low transition on serial receive pin?
Yes
16-bit Timer 0 starts counting up
Point B
Low-to-high transition on serial receive pin?
Yes
Software-capture and save timer value (tAB)
Point C
High-to-low transition on serial receive pin?
Yes
Software-capture and save timer value (tAC)
Point D
Low-to-high transition on serial receive pin?
Yes
Software-capture and save timer value (tAD)
16-bit Timer 0 stops counting
tAC ≥ tAD?
Yes
Make backup copy of tAD value
Done
Stop operation
(infinite loop)

Fig. 24.5 Serial Operation Mode Byte Reception Flow

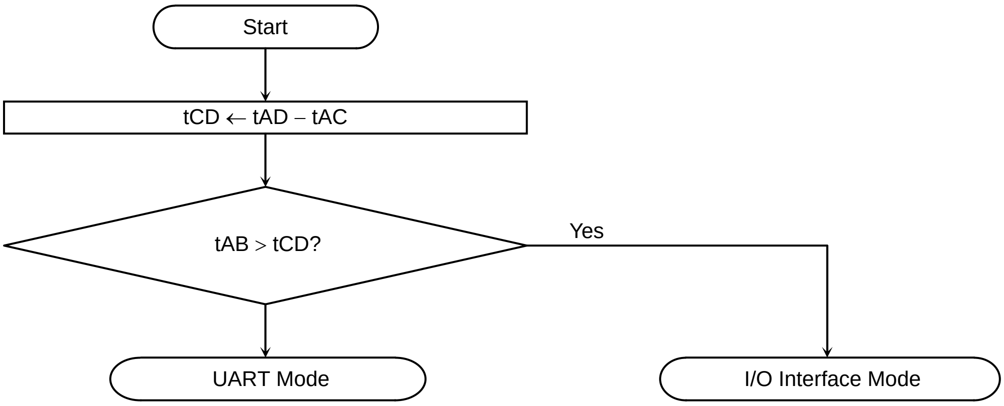


Fig. 24.6 Serial Operation Mode Determination Flow

9) Password

The RAM Transfer command (10H) causes the boot program to perform a password check. Following an echo-back of the command code, the boot program checks the contents of the 12-byte password area (0x0000_0474 to 0x0000_047F) within the flash memory.

If all these address locations contain the same bytes of data other than FFH, a password area error occurs. In this case, the boot program returns an error acknowledge (11H) in response to the checksum byte (the 17th byte), regardless of whether the password sequence sent from the controller is all FFHs.

The password sequence received from the controller (5th to 16th bytes) is compared to the password stored in the flash memory. All of the 12 bytes must match to pass the password check. Otherwise, a password error occurs, which causes the boot program to return an error acknowledge in response to the checksum byte (the 17th byte).

The password check is performed even if the security function is enabled.

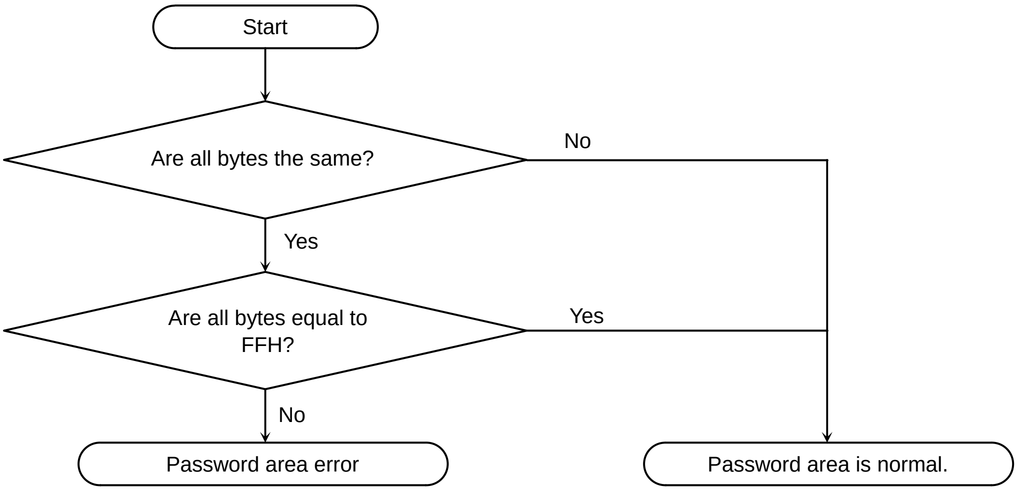
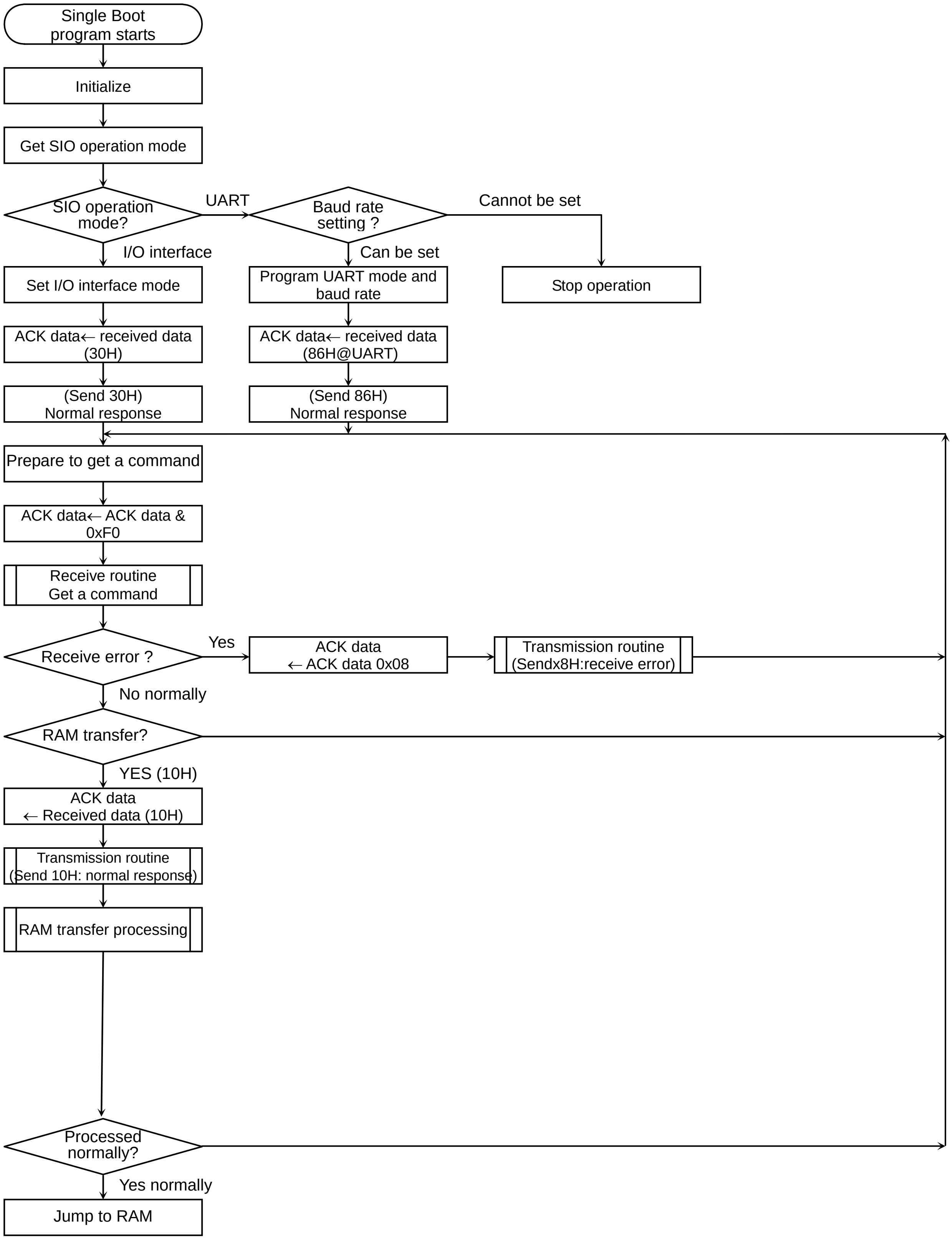


Fig. 24.7 Password Area Check Flow

10) General Boot Program Flowchart

Fig. 24.8 shows an overall flowchart of the boot program.



Single Boot program starts
Initialize
Get SIO operation mode
SIO operation mode?
UART
Baud rate setting ?
Cannot be set
I/O interface
Can be set
Set I/O interface mode
Program UART mode and baud rate
Stop operation
ACK data← received data (30H)
ACK data← received data (86H@UART)
(Send 30H) Normal response
(Send 86H) Normal response
Prepare to get a command
ACK data← ACK data & 0xF0
Receive routine Get a command
Receive error ?
Yes
ACK data ← ACK data 0x08
Transmission routine (Sendx8H:receive error)
No normally
RAM transfer?
YES (10H)
ACK data ← Received data (10H)
Transmission routine (Send 10H: normal response)
RAM transfer processing
Processed normally?
Yes normally
Jump to RAM

Fig. 24.8 Overall Boot Program Flow

24.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM or from an external memory device after shifting to the user boot mode. In this section, flash memory addresses are represented in virtual addresses unless otherwise noted.

24.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands. In writing or erasing, use the SW command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 24.9 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically.
Write protect	The write or erase function can be individually inhibited for each block. When all blocks are set for protection, the entire protection function is automatically enabled.
Protect function	By writing a 4-bit protection code, the write or erase function can be individually inhibited for each area.

Note that addressing of operation commands is different from the case of standard commands due to the specific interface arrangements with the CPU. Also note that the flash memory is written in 32-bit blocks. So, 32-bit (word) data transfer commands must be used in writing the flash memory.

(1) Block configuration

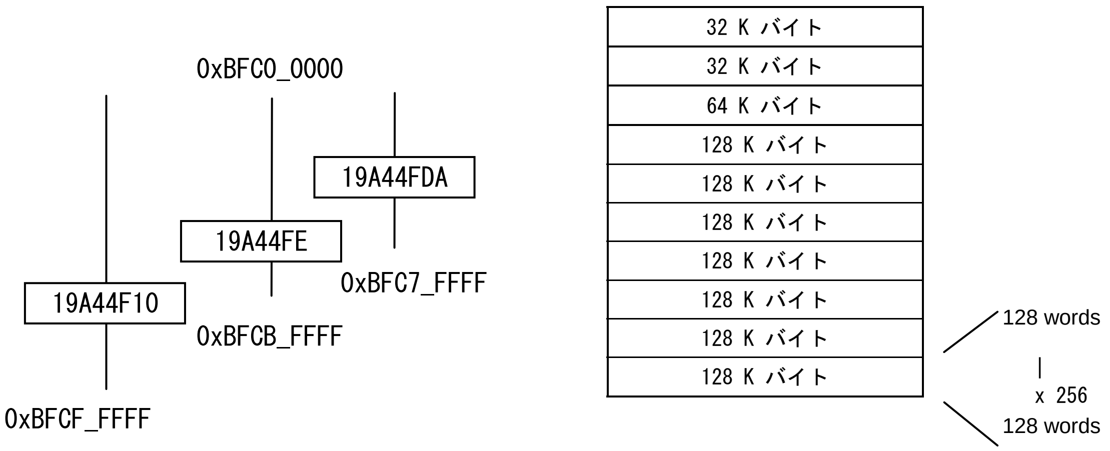


Fig. 24.9 Block Configuration of Flash Memory

(2) Basic operation

Generally speaking, this flash memory device has the following two operation modes:

The mode to read memory data (Read mode)

The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exceptions other than debug exceptions and reset while a DSU probe is connected. Any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated.

1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

Read/reset command and Read command (software reset)

When an automatic operation is abnormally terminated, the flash memory cannot return to the read mode by itself (When FLCS<FlashBusy> = 0, data read from the flash memory is undefined.) In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used to return to the read mode. The Read command is used to return to the read mode after executing the SW command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.

With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.

2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read mode.

While commands are generally comprised of several bus cycles, the operation to apply the SW command to the flash memory is called "bus write cycle." The bus write cycles are to be in a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of a command write operation is in accordance with a predefined specific sequence. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode. The address [31:20] in each bus write cycle should be the virtual address [31:20] of command execution. It will be explained later for the address bits [19:8].

(Note 1) Command sequences are executed from outside the flash memory area.

(Note 2) The interval between bus write cycles for this device must be 15 system clock cycles or longer. The command sequencer in the flash memory device requires a certain time period to recognize a bus write cycle. If more than one bus write cycles are executed within this time period, normal operation cannot be expected. For adjusting the applicable bus write cycle interval using a software timer to be operated at the operating frequency, use the section 10) "ID-Read" to check for the appropriateness.

(Note 3) Between the bus write cycles, never use any load command (such as LW, LH, or LB) to the flash memory or perform a DMA transmission by specifying the flash area as the source address. Also, don't execute a Jump command to the flash memory. While a command sequence is being executed, don't generate any interrupt such as maskable interrupts (except debug exceptions when a DSU probe is connected).

If such an operation is made, it can result in an unexpected read access to the flash memory and the command sequencer may not be able to correctly recognize the command. While it could cause an abnormal termination of the command sequence, it is also possible that the written command is incorrectly recognized.

(Note 4) The SYNC command must be executed immediately after the SW command for each bus write cycle.

(Note 5) For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle that the FLCS[0]R <FlashBusy> bit is set to "1." It is recommended to subsequently execute a Read command.

(Note 6) Upon issuing a command, if any address or data is incorrectly written, be sure to perform a system reset operation or issue a reset command to return to the read mode again.

3) Reset

Hardware reset

The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the RESET input pin of this device is set to V_{IL} or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. The CPU reset is also used in returning to the read mode when an automatic operation is abnormally terminated or when any mode set by a command is to be canceled. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section 24.2.1 "Reset Operation" for CPU reset operations. After a given reset input, the CPU will read the reset vector data from the flash memory and starts operation after the reset is removed.

4) Automatic Page Programming

Writing to a flash memory device is to make "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For making "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data in 128 word blocks. A 128 word block is defined by a same [31:9] address and it starts from the address [8:0] = 0 and ends at the address [8:0] = 0x1FF. This programming unit is hereafter referred to as a "page."

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by the FLCS [0] <FlashBusy> register.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more times irrespective of the data cell value whether it is "1" or "0." Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page two or more times without erasing the content can cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the fourth bus write cycle of the command cycle is completed. On and after the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at a time). Be sure to use the SW command in writing commands on and after the fourth bus cycle. In this, any SW command shall not be placed across word boundary. On and after the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0." For example, if the top address of a page is not to be written, set the input data of the fourth bus write cycle to 0xFFFFFFFF to command write the data.

Once the fourth bus cycle is executed, it is in the automatic programming operation. This condition can be checked by monitoring the register bit FLCS [0] <FlashBusy> (SeeTable 24.10). Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted. When a single page has been command written normally terminating the automatic page writing process, the FLCS [0] <FlashBusy> bit is set to "1" and it returns to the read mode.

When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FLCS [0] <FlashBusy> (See Table 24.10). If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to use the reset command or hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

Note: Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.

5) Automatic chip erase

The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FLCS [0] <FlashBusy> (See Table 24.10). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If all the protected blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the mode and will not return to the read mode.

For returning to the read mode, it is necessary to use the reset command or hardware reset to reset the flash memory or the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

6) Automatic block erase (one block at a time)

The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FLCS [0] <FlashBusy> (See Table 24.10). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, use the reset command or hardware reset to reset the flash memory or the device.

7) Automatic programming of protection bits

This device is implemented with four protection bits. The protection bits can be individually set in the automatic programming. The applicable protection bit is specified in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each area. The protection status of each area can be checked by the FLCS <PROTECT 3:0> register to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FLCS <FlashBusy> (See Table 24.10). Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, the flash memory cannot be read from any area outside the flash memory such as the internal RAM.

Note: Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. The FLCS <FlashBusy> bit turns to "0" after entering the seventh bus write cycle.

8) Automatic erasing of protection bits

The protection condition can be canceled by the automatic protection bit erase operation. The target bits are specified in the seventh bus write cycle and when the command is completed, the device is in a condition all the blocks are erased. This operation can be checked by monitoring FLCS <FlashBusy>. Also, you can check the protection condition by monitoring FLCS <PROTECT 3:0>.

When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the device. If this is done, it is necessary to check the status of protection bits by FLCS <PROTECT 3:0> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

The FLCS <FlashBusy> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.

9) Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the area protection status.

Table 24.10 Flash Control Register

FLCS
(0xFF00_0100)

	7	6	5	4	3	2	1	0
bit Symbol	—	—	—	—	—	—	—	FlashBusy
Read/Write	R							
After reset	0	0	0	0	0	0	0	1
Function	“0” can be read.							0:BUSY 1:READY
	15	14	13	12	11	10	9	8
bit Symbol	—	—	—	—	—	—	—	—
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
bit Symbol	—	—	—	—	PROTECT3	PROTECT2	PROTECT1	PROTECT0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function					Indicates protection condition.			
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” can be read.							

Bit 0: FlashBusy flag bit

The FlashBusy 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. By applying a hardware reset, it returns to "1."

(note)Please issue it after confirming the command issue is always a ready state.

A normal command not only is sent when the command is issued to a busy inside but also there is a possibility that the command after that cannot be input. In that case, please return by system reset or the reset command.

Bits [19:16]: Protection status bits (

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 block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

10) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (any input data other than 0xF can be used). On and after the fourth bus write cycle, when an LW command (to read an arbitrary flash memory area) is executed after an SW command, the ID value will be loaded (execute a SYNC command immediately after the LW command). Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and LW/SYNC commands can be repetitively executed. For returning to the read mode, reset the system or use the Read or Read/reset command.

(Important) The "interval between bus write cycles" between successive command sequences must be 15 system clock cycles or longer irrespective of the operating frequency used. This device doesn't have any function to automatically adjust the interval between bus write cycles regarding execution of multiple SW commands to the flash memory. Therefore, if an inadequate interval is used between two sets of bus write cycles, the flash memory cannot be written as expected. Prior to setting the device to work in the onboard programming mode, adjust the bus write cycle interval using a software timer, etc., to verify that the ID-Read command can be successfully executed at the operating frequency of the application program. In the onboard programming mode, use the bus write cycle interval at which the ID-Read command can be operated normally to execute command sequences to rewrite the flash memory.

(3) List of Command Sequences

Table 24.11 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle	Second bus cycle	Third bus cycle	Fourth bus cycle	Fifth bus cycle	Sixth bus cycle	Seventh bus cycle
	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.	Addr.
	Data	Data	Data	Data	Data	Data	Data
Read	0xXX	RA					
	0xF0	RD					
Read/reset	0x55XX	0xAAXX	0x55XX	RA			
	0xAA	0x55	0xF0	RD			
ID-Read	0x55XX	0xAAXX	0x55XX	IA	0xXX	–	
	0xAA	0x55	0x90	0x00	ID	–	
Automatic page programming (note)	0x55XX	0xAAXX	0x55XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	0x55XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Auto Block erase (note)	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Protection bit programming	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	0x55XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	0x55XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

(4) Supplementary explanation

RA: Read address

RD: Read data

IA: ID address

ID:ID data

PA: Program page address

PD: Program data (32-bit data)

After the fourth bus cycle, enter data in the order of the address for a page.

BA: Block address

PBA: Protection bit address

(Note 1) Always set "0" to the address bits [1:0] in the entire bus cycle. (Setting values to bits [7:2] are undefined.)

(Note 2) Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by SW commands. Use "Data" in the table for the rt register [7:0] of SW commands. The address [31:16] in each bus write cycle should be the target flash memory address [31:16] of the command sequence. Use "Addr." in the table for the address [15:0].

(Note 3) In executing the bus write cycles, the interval between each bus write cycle shall be 15 system clocks or more.

(Note 4) The "Sync command" must be executed immediately after completing each bus write cycle.

(Note 5) Execute the "Sync command" immediately following the "LW command" after the fourth bus write cycle of the ID-Read command.

(5) Address bit configuration for bus write cycles

Table 24.12 Address Bit Configuration for Bus Write Cycles

Address	Addr [31:20]	Addr [19]	Addr [18:17]	Addr [16]	Addr [15]	Addr [14]	Addr [13]	Addr [12:9]	Addr [8]	Addr [7:0]
Normal commands	**Normal bus write cycle address configuration**									
	Flash area	"0" is recommended			Command [15:8]					Addr [1:0]=0 (fixed), Others: 0 (recommended)
	0xbfc0									0x00

Block erase	**BA: Block address (Set the sixth bus write cycle address for block erase operation)**									
	Flash area	Block selection Addr[19:15]				Addr[1:0]=0 (fixed), Others: 0 (recommended)				
	Block 0:0xbfc00000 Block 1:0xbfc08000 Block 2:0xbfc10000 Block 3:0xbfc20000 Block 4:0xbfc40000 Block 5:0xbfc60000 Block 6:0xbfc80000 Block 7:0xbfca0000 Block 8:0xbfcc0000 Block 9:0xbfce0000									

Auto page program-ming	**PA: Program page address (Set the fourth bus write cycle address for page programming operation)**									
	Flash area	Block selection [19:15]				Page selection [14:9]			Addr[1:0]=0 (fixed), Others: 0 (recommended)	

ID-READ	**IA: ID address (Set the fourth bus write cycle address for ID-Read operation)**									
	Flash area	"0" is recommended			ID address [15:14]		Addr[1:0]=0 (fixed), Others: 0 (recommended)			
	0xbfc0									

Protection bit program-ming	**PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming)**									
	Flash area	“0”推奨	Protection bit write [18:9] [18:11]=00100000 [10:9] Area 0:00 Area 1:01 Area 2:10 Area 3:11							Addr[1:0]=0 (fixed), Others: 0 (recommended)
	Area 0:0xbfc10000 Area 1:0xbfc10200 Area 2:0xbfc10400 Area 3:0xbfc10600									

Protection bit erase	**PBA: Protection bit address (Set the seventh bus write cycle address for protection bit erasure)**									
	Flash area	"0" is recommended	Erase protection [18：17]00		Addr[1:0]=0 (fixed), Others: 0 (recommended)					
	0xbfc00000 As for the Protection bit, the batch deletion is done.									

(Note) **Table 24.11 "Flash Memory Access from the Internal CPU" can also be used.**

(Note) **Address setting can be performed according to the "Normal bus write cycle address configuration" from the first bus cycle.**

(Note) **"0" is recommended" can be changed as necessary.**

Table 24.13 Block Erase Address Table

Product			BA	Address Range		Size
				Flash Memory Address	**When applied to the projected area**	
TMP19A44F10XBG	TMP19A44FEXBG	TMP19A44FDAXB	Block 0	BFC0_0000 - BFC0_7FFF	0x0000_0000~0x0000_7FFF	32 KB
			Block 1	BFC0_8000 - BFC0_FFFF	0x0000_8000~0x0000_FFFF	32 KB
			Block 2	BFC1_0000 - BFC1_FFFF	0x0001_0000~0x0001_FFFF	64 KB
			Block 3	BFC2_0000 - BFC3_FFFF	0x0002_0000~0x0003_FFFF	128 KB
			Block 4	BFC4_0000 - BFC5_FFFF	0x0004_0000~0x0005_FFFF	128 KB
			Block 5	BFC6_0000 - BFC7_FFFF	0x0006_0000~0x0007_FFFF	128 KB
			Block 6	BFC8_0000 - BFC9_FFFF	0x0008_0000~0x0009_FFFF	128 KB
			Block 7	BFCA_0000 - BFCB_FFFF	0x000A_0000~0x000B_FFFF	128 KB
			Block 8	BFCC_0000 - BFCD_FFFF	0x000C_0000~0x000D_FFFF	128 KB
			Block 9	BFCE_0000 - BFCF_FFFF	0x000E_0000~0x000F_FFFF	128 KB

Example: When BA0 is to be selected, any single address in the range 0xBFC0_0000 to 0xBFC0_7FFF may be entered.

(Note) As for the addresses from the first to the sixth bus cycles, specify the upper 4 bit with the corresponding flash memory addresses of the blocks to be erased.

Table 24.14 Protection Bit Programming/ Erasing Address Table

		Programming (address)										Erasing (address)	
		18	17	16	15	14	13	12	11	10	9	18	17
Protect bit	bit0	0	0	1	0	0	0	0	0	0	0	0	0
	bit1	0	0	1	0	0	0	0	0	0	1		
	bit2	0	0	1	0	0	0	0	0	1	0		
	bit3	0	0	1	0	0	0	0	0	1	1		

Table 24.15 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following LW command (ID)

IA [15:14]	ID [7: 0]	Code
00b	0x98	Manufacturer code
01b	0x5A	Device code
10b	Reserved	---
11b	0x10：FE/F10 0x12：FD	Macro code

25. Various protecting functions

25.1 Overview

The ROM protect function for designating the internal ROM (flash) area as a read-protected area and the DSU protect function for prohibiting the use of DSU (DSU-Probe) are built into the TMP19A44. The read protect functions specifically include the following:

- Flash protection
- ROM data protection
- DSU protection

25.2 Features

- Flash Protection
  It protects data in FLASH memory by prohibiting accidental rewriting and deletion.
  This function covers the area where protecting bit is set.

- ROM data Protection
  It prevents operand of FLASH data from being read incorrectly.
  Area attribution is specified with protecting bit and the ROMSEC register.
  * Deleting protection bit and clearing the ROMSEC register and the DSUSEC register are under protection as well.

- DSU Protection
  It prevents incorrect program analysis using DSU.

25.3 Protecting Functions

25.3.1 Flash Protection

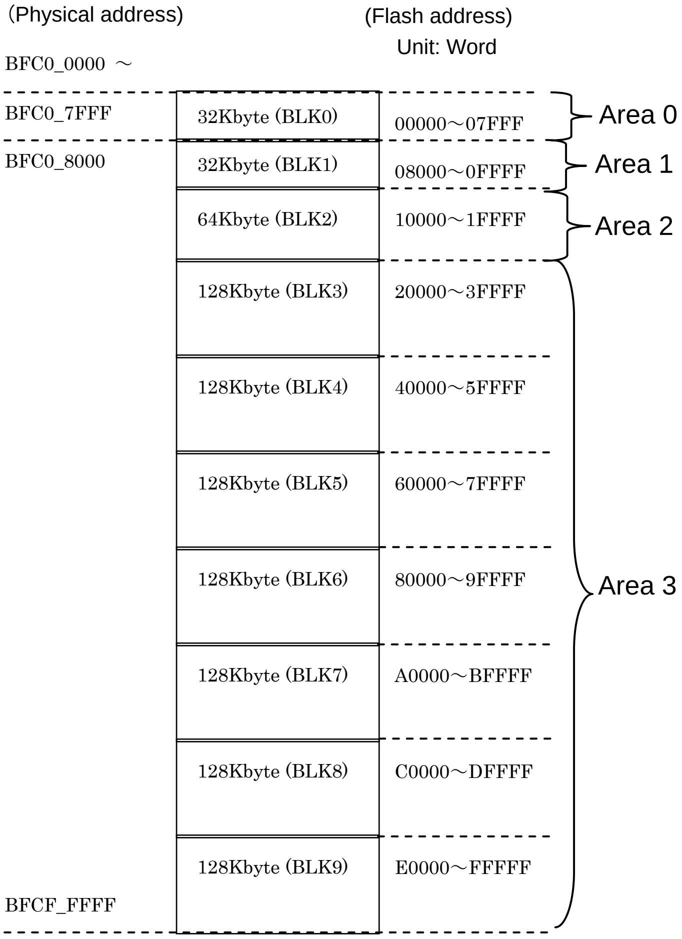
An internal FLASH can prohibit the operation of writing and the deletion per protected area. This function is called the flash protection.
To make the protection effective, set “1” to the protecting bit corresponding to the area to protect. By clearing this bit to “0”, the protection can be released (see “Chapter 24 Flash Memory Operation” for how to program.)
The protecting bit can be monitored with FLCS register.

a) Operation mode: Valid in single or single boot mode.
b) ROM division: The entire ROM area is divided into 4 areas. Protection can be set for each area.
d) Area attribution: Defines area as protected or normal (area without protection).

● FLASH protection/ ROM data protection/ DSU protection

Item	Use of protecting bit	Use of ROMSEC register	Use of DSUSEC register	Function
FLASH protection	Yes	-	-	- Prohibits Flash operation (Chip/Block deletion, page writing)
ROM data protection	Yes (with restriction at deletion)	Yes (with restriction at clearing)	- (with restriction at clearing)	- Prohibits ROM reading - Controls data deletion at protecting bit deletion - Prohibits clearing register used for ROM data protection - Prohibits clearing register used for DSU protection - Prohibits clearing error flag
DSU protection	Yes -	-	Yes -	- Prohibits break. - Prohibits trace (including memory store)

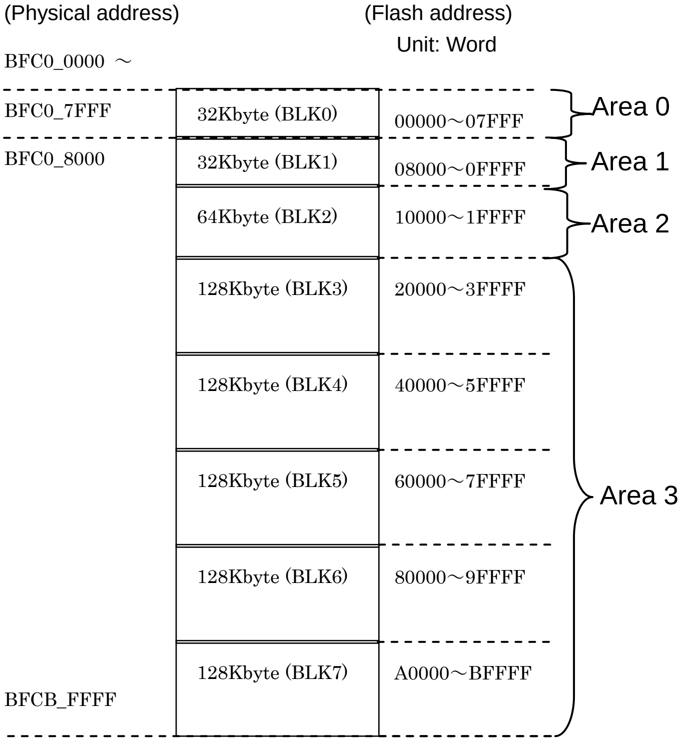
●TMP19A44F10XBG（1024KB）



The protection can be set for each area.

Overall protection
Protected area = 0~3

●TMP19A44FEXBG（768KB）

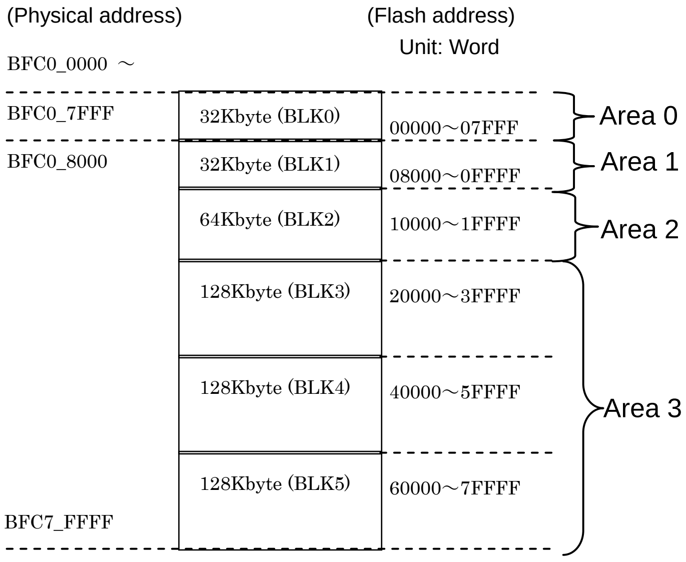


(Physical address)		(Flash address) Unit: Word	
BFC0_0000 ～ BFC0_7FFF	32Kbyte (BLK0)	00000～07FFF	Area 0
BFC0_8000	32Kbyte (BLK1)	08000～0FFFF	Area 1
	64Kbyte (BLK2)	10000～1FFFF	Area 2
	128Kbyte (BLK3)	20000～3FFFF	Area 3
	128Kbyte (BLK4)	40000～5FFFF	
	128Kbyte (BLK5)	60000～7FFFF	
	128Kbyte (BLK6)	80000～9FFFF	
BFCB_FFFF	128Kbyte (BLK7)	A0000～BFFFF	

The protection can be set for each area.

Overall protection
Protected area = 0~3

●TMP19A44FDAXBG（512KB）



(Physical address)		(Flash address) Unit: Word	
BFC0_0000 ～ BFC0_7FFF	32Kbyte (BLK0)	00000～07FFF	Area 0
BFC0_8000	32Kbyte (BLK1)	08000～0FFFF	Area 1
	64Kbyte (BLK2)	10000～1FFFF	Area 2
	128Kbyte (BLK3)	20000～3FFFF	Area 3
	128Kbyte (BLK4)	40000～5FFFF	
BFC7_FFFF	128Kbyte (BLK5)	60000～7FFFF	

ROM Data Protection

●TMP19A44F10XBG (evaluation sample)

The ROM data protection restricts data reading from internal FLASH.

This function does not cover:

- access to unprotected area
- access to protected area by ROM correction (the area is considered to be accessed by an internal RAM)
- access to an internal RAM, internal Boot-ROM, external memory and DSU (debug area)
- access to operand by an internal DMAC
- access by FLASH writer

・ How to make the protection valid

Set SECBIT, which corresponds to area to protect, of the SECBIT register to "1" (enabled). (All bits are set to "1"and enabled after power-on.)

・Procedure of detecting protection

1) Issuing a protecting bit erase command from normal area.
2) Detecting protection.
3) Deleting the entire data in FLASH.
4) Deleting a protecting bit.

・Procedure of SECBIT protection

1) Writing corresponding SECBIT from normal area.
2) Detecting protection (no value updated).

・ Procedure of ROM data protection

1) Reading protected area from normal area.
2) Detecting protection.

(Flash writer is used)	Initial data in an internal ROM can be read.
(Others)	"0x0098" can be read.

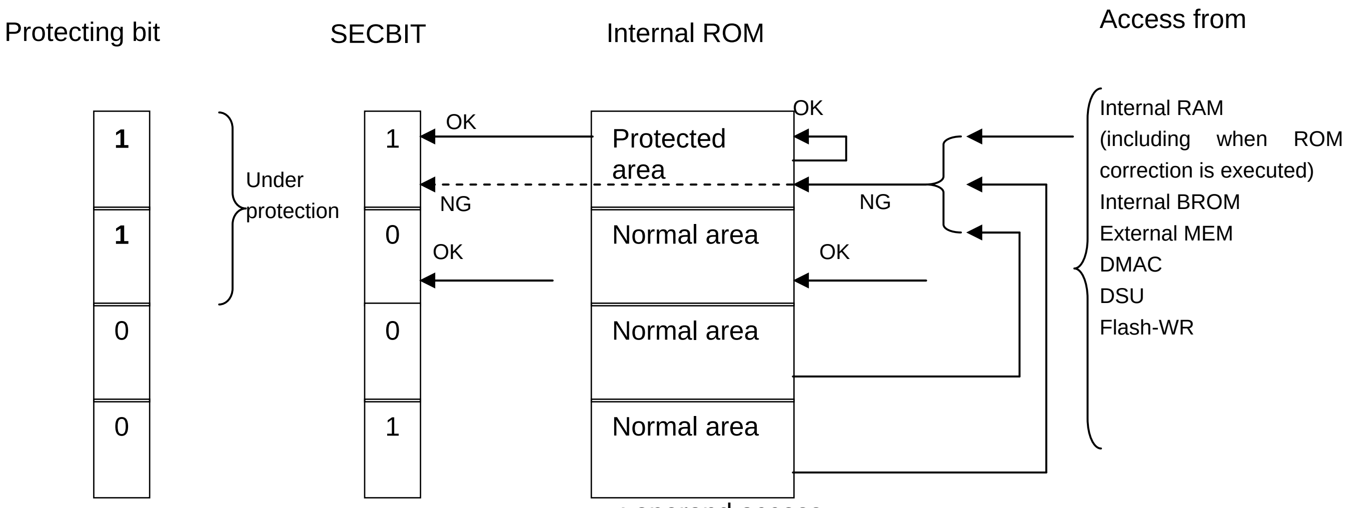
・ Procedure of DSU protection

1) Either of the following takes place:
   - Executing an instruction in protected area (for inhibiting break, event/ PC trace).
   - Executing operand bus cycle caused by an instruction in protected area (for inhibiting data trace).
2) Detecting protection.
3) Inhibiting break (executes EJTAGBOOT and step) and trace (including store to trace memory).

25.4 Definition of operations/ terms

- Protected area valid: Corresponding SECBIT bit is set to “1” (enabled).
- Detecting ROM data protection : Control by ROSMEC (operand mask etc.) is in operation.
- Detecting DSU protection : Control by DSUSEC (inhibiting break/ trace etc.) is in operation

●Example of detecting protection



←: operand access
OK: correct access
NG: incorrect access

25.5 Registers

Flash Control Register (FLCS)

FLCS
(0xFF00_0100)

	7	6	5	4	3	2	1	0
bit Symbol	—	—	—	—	—	—	—	FlashBusy
Read/Write	R							
After reset	0	0	0	0	0	0	0	1
Function	“0” can be read.							0:BUSY 1:READY
	15	14	13	12	11	10	9	8
bit Symbol	—	—	—	—	—	—	—	—
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
bit Symbol	—	—	—	—	PROTECT3	PROTECT2	PROTECT1	PROTECT0
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Indicates protection condition.							
	31	30	29	28	27	26	25	24
bit Symbol	—	—	—	—				
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” can be read.							

FlashBusy : FLASH READY/BUSY signal
0: BUSY
1: READY

PROTECT3:0 : Under protection.

ROM Data Protection Enable Bit Register (SECBIT)

SECBIT
(0xFF00_0200)

	7	6	5	4	3	2	1	0
bit Symbol	–	—	—	—	SECBIT3	SECBIT2	SECBIT1	SECBIT0
Read/Write	R				R/W			
After reset	0	0	0	0	1	1	1	1
Function	“0” can be read.				xxx1 :Enabling area 0 protection xx1x :Enabling area 1 protection x1xx :Enabling area 2 protection 1xxx :Enabling area 3 protection			
	15	14	13	12	11	10	9	8
bit Symbol	–	—	—	—	—	—	—	—
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
bit Symbol	–	—	—	—	—	—	—	—
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
bit Symbol	–	—	—	—	—	—	—	—
Read/Write	R							
After reset	0							
Function	“0” can be read.							

ROM Protection Lock Register

		7	6	5	4	3	2	1	0
SECCODE	Bit Symbol								
(0xFF00_0208)	Read/Write	W							
	After reset	Undefined							
	Function	See note.							
		15	14	13	12	11	10	9	8
	Bit Symbol								
	Read/Write	W							
	After reset	Undefined							
	Function	See note.							
		23	22	21	20	19	18	17	16
	Bit Symbol								
	Read/Write	W							
	After reset	Undefined							
	Function	See note.							
		31	30	29	28	27	26	25	24
	Bit Symbol								
	Read/Write	W							
	After reset	Undefined							
	Function	See note.							

(Note) Setting 0x0000_003d to the SECCODE register enables writing to the SECBIT register.

DSU Protection Enable Bit Register

DSUSECBIT

(0xFF00_0204)

	7	6	5	4	3	2	1	0
Bit Symbol					DSUSECBIT3	DSUSECBIT2	DSUSECBIT1	DSUSECBIT0
Read/Write	R				R/W			
After reset	0				1			
Function	“0” can be read.				xxx1 :Enabling area 0 protection xx1x :Enabling area 1 protection x1xx :Enabling area 2 protection 1xxx :Enabling area 3 protection			
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” can be read.							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0							
Function	“0” can be read.							

DSU protection enable bit register setting:
1: Protection enabled
0: Protection disabled

Protection for each area can be set.
xxx1: Enabling area 0 protection
xx1x: Enabling area 1 protection
x1xx: Enabling area 2 protection
1xxx: Enabling area 3 protection

Power-on reset sets the entire bits to “1”.

DSU Protection Control Register

		7	6	5	4	3	2	1	0
DSUSECCODE	Bit Symbol	DSECODE 07	DSECODE 06	DSECODE 05	DSECODE 04	DSECODE 03	DSECODE 02	DSECODE 01	DSECODE 00
(0xFF00_020C)	Read/Write	W							
	After reset	0							
	Function	Write “0x0000_00C5”.							
		15	14	13	12	11	10	9	8
	Bit Symbol	DSECODE 15	DSECODE 14	DSECODE 13	DSECODE 12	DSECODE 11	DSECODE 10	DSECODE 09	DSECODE 08
	Read/Write	W							
	After reset	0							
	Function	Write “0x0000_00C5”.							
		23	22	21	20	19	18	17	16
	Bit Symbol	DSECODE 23	DSECODE 22	DSECODE 21	DSECODE 20	DSECODE 19	DSECODE 18	DSECODE 17	DSECODE 16
	Read/Write	W							
	After reset	0							
	Function	Write “0x0000_00C5”.							
		31	30	29	28	27	26	25	24
	Bit Symbol	DSECODE 31	DSECODE 30	DSECODE 29	DSECODE 28	DSECODE 27	DSECODE 26	DSECODE 25	DSECODE 24
	Read/Write	W							
	After reset	0							
	Function	Write “0x0000_00C5”.							

(Note 1) To access this register, 32-bit access is required.
(Note 2) This is read-only register. Undefined values are read.

Setting 0x0000_00c5 to the SUSECCODE register enables writing to the DSUSECBIT register.

Example) If area 0 is protected:
The instruction from area 0 can write both “0” and “1” to the SECBIT and DSUSECBIT registers.
The instruction from area 1, 2 and 3 can write “1” to the SECBIT and DSUSECBIT registers.

26. Backup module

26.1 Features

The backup mode, one of the system operation modes, enables the 19A44 to operate in the low power consumption. By cutting electricity to the entire block, such as CPU or other peripheral I/Ps, other than the backup module, this mode significantly reduces power consumption.

26.2 Overview

The backup mode operates in lower power consumption than STOP/SLEEP modes.

Backup mode	Condition for releasing standby mode
Backup STOP	Two-phase counter (counting enabled）, INT, Key-on wake-up (static), STOP release ,reset
Backup SLEEP	Two-phase counter (asynchronous input) ,INT, Key-on wake-up (dynamic/static), real time clock,reset

26.3 Operation in Backup Mode

・DSU

Transition to the backup mode is available while DSU is connected.
When the transition takes place, the power supply to DSU is not cut and DSU retains data but stops operating.

・Two-phase counter

High-speed counting is available in both Backup STOP and Backup SLEEP modes.

・PORT

Output/Pull-up: retains the condition set in the PORTKEEP register.
Input: disabled (two-phase counter, key-on wake-up and INT input are enabled.)
Key-on wake-up operates by dynamic pull-up (input enabled).

26.4 Block Diagram

Fig. 26.1 shows the block diagram of the backup module.

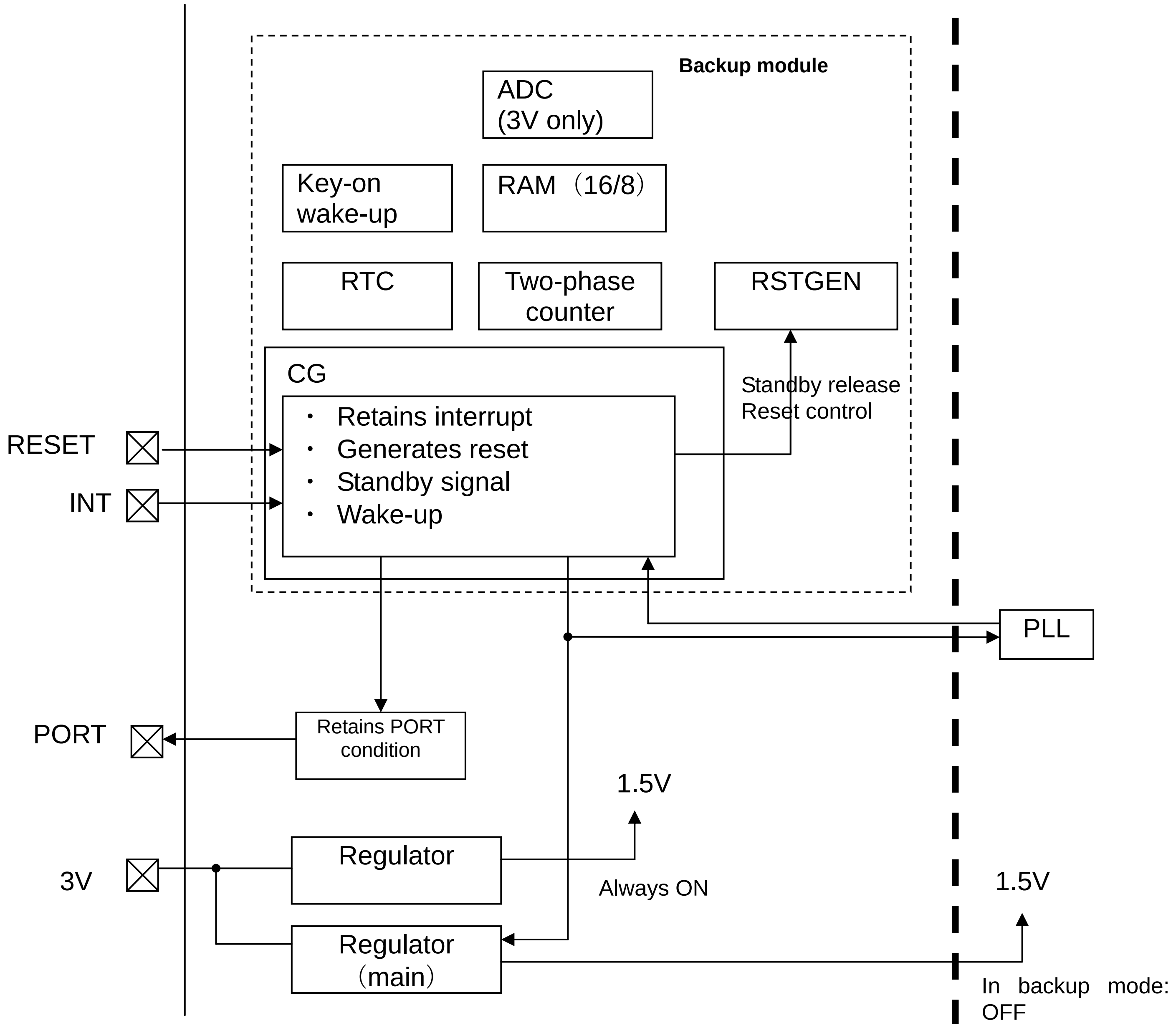


Fig. 26.1 Block Diagram of Backup Module

26.5 Registers

26.5.1 Standby Control Register

STBYCR0
LITTLE (0xFF00_1908)
BIG (0xFF00_190B)

	7	6	5	4	3	2	1	0
Bit symbol						STBY2	STBY1	STBY0
Read/Write	R					R/W	R/W	R/W
After reset	0					0	1	1
Function	“0” is read.					Standby mode selection 000: Reserved 001: STOP 010: SLEEP 011: IDLE 100: Reserved 101: Backup STOP 110: Backup SLEEP 111: Reserved		

STBYCR1
LITTLE (0xFF00_1909)
BIG (0xFF00_190A)

	15	14	13	12	11	10	9	8
Bit symbol							RXTEN	RXEN
Read/Write	R						R/W	R/W
After reset	0						0	1
Function	“0” is read.						Low-speed oscillator operation after releasing STOP mode 0: Stop 1: Oscillating	High-speed oscillator operation after releasing STOP mode 0: Stop 1: Oscillating

STBYCR2
LITTLE (0xFF00_190A)
BIG (0xFF00_1909)

	23	22	21	20	19	18	17	16
Bit symbol							PTKEEP	DRVE
Read/Write	R						R/W	R/W
After reset	0						0	0
Function	“0” is read.						0:PORT control 1: retains condition shifted from 0 to 1.	0: Not to drive the pin even in the STOP mode. 1: Drive the pin even in the STOP mode.

	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	“0” is read.							

<Bit 2:0><STBY2:0> : Selects standby mode.

<Bit 8><RXEN> : Selects high-speed oscillator operation after releasing STOP mode

<Bit 9><RXTEN> : Selects low-speed oscillator operation after releasing STOP mode

<Bit 16><DRVE> : Selects pin drive condition in STOP mode.
This setting is invalid in the backup mode.

<Bit 17><PTKEEP> : Retains port condition in the backup mode.
Reset initializes the port, therefore reconfiguration is required.

26.5.2 Reset Flag Register

RSTFLG
(0xFF00_191C)

	7	6	5	4	3	2	1	0
Bit symbol					BUPRSTF	WDTRSTF	PINRSTF	PONRSTF
Read/Write	R				R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	1
Function	“0” is read.				Backup reset flag 0: “0” is written. 1:Resetting backup mode	Watchdog timer reset flag 0: “0” is written. 1:Reset by watchdog timer	RESET pin flag 0: “0” is written. 1:Reset from RESET pin	Power-on Reset flag 0: “0” is written. 1:Power On Reset
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	“0” is read.							

<Bit 0><PONRSTF> : Power-on reset sets this bit to “1”.

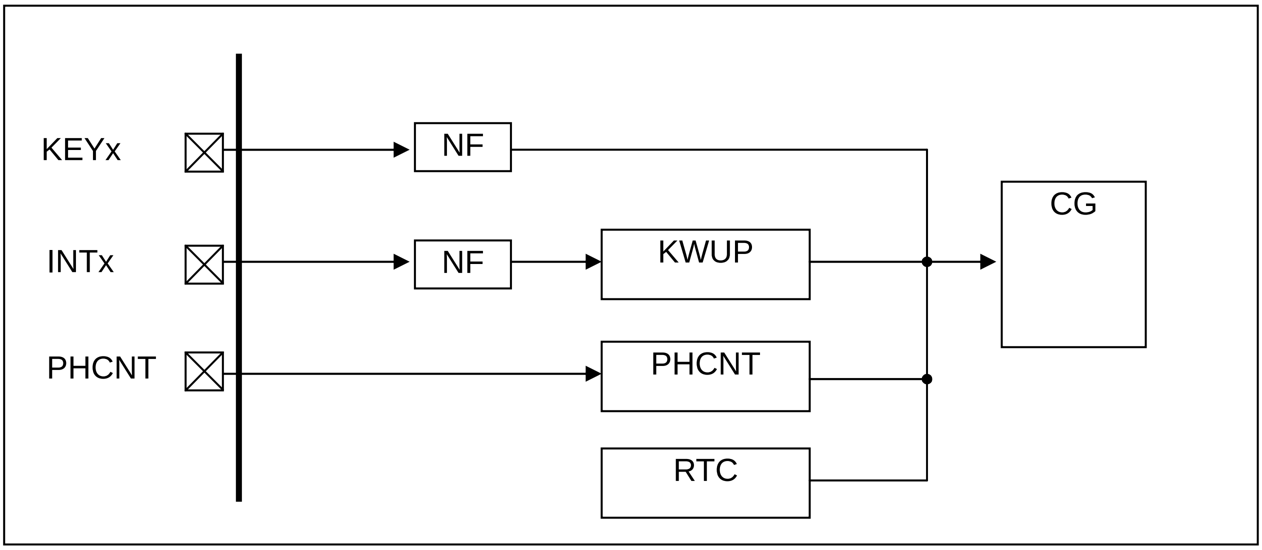
<Bit 1><PINRSTF> : Reset from RESET pin sets this bit to “1”.

<Bit 2><WDTRSTF> : Reset by a watch dog timer sets this bit to “1”.

<Bit 3><BUPRSTF> : Returning from backup mode sets this bit to “1”.

Writing "0" resets a flag.

26.6 Return Circuit



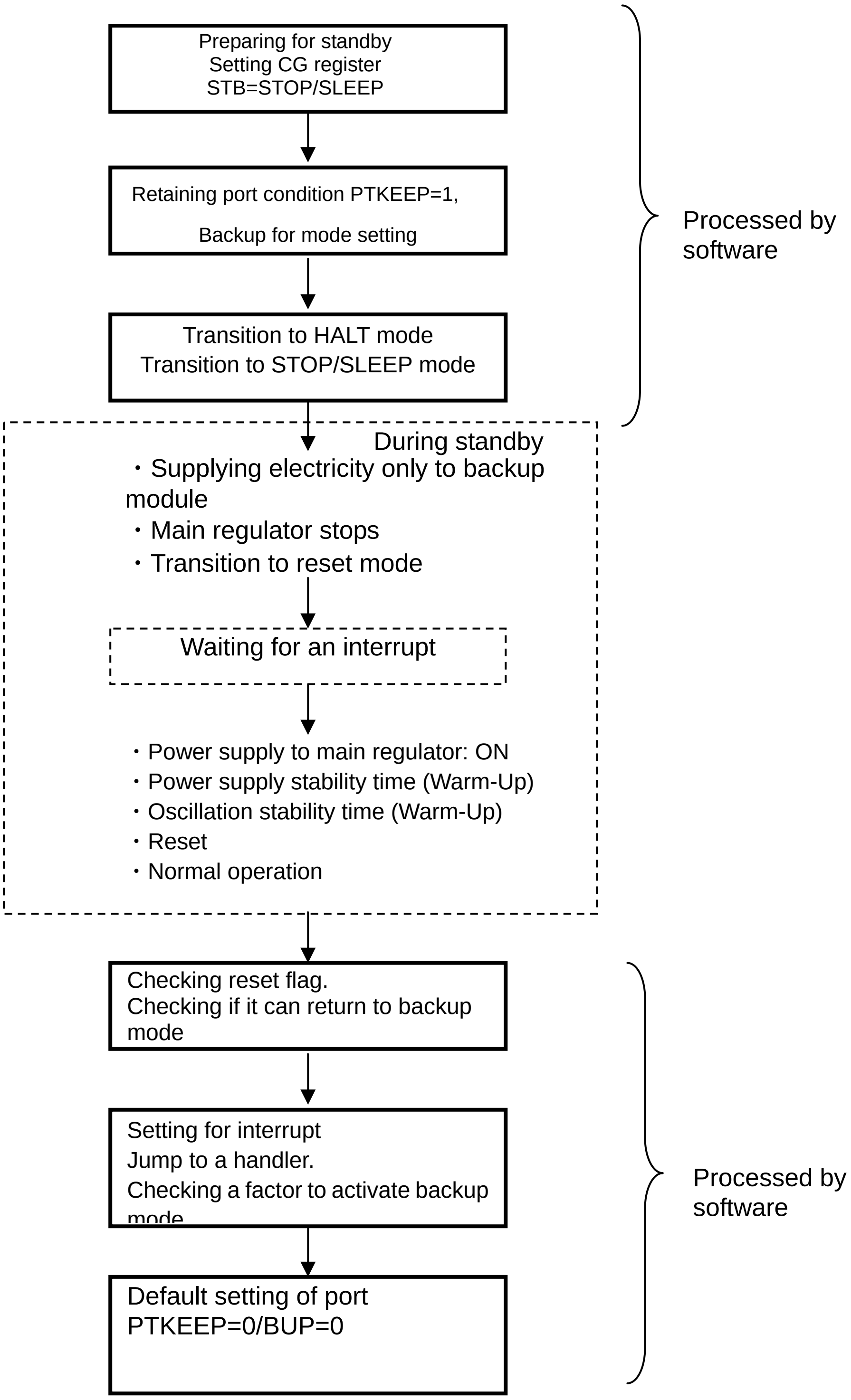
In the backup mode, a return signal of an external factor or the real time clock is transferred to the internal clock generator.

The interrupt factor is retained in the clock generator, and the main regulator is powered on.

The instruction is executed from the initial address in the same manner as reset operation.

The flags in the Reset Flag Register indicate which interrupt factor is used.

26.7 Transition flowchart


Preparing for standby
Setting CG register
STB=STOP/SLEEP
Retaining port condition PTKEEP=1,
Backup for mode setting
Transition to HALT mode
Transition to STOP/SLEEP mode
Processed by software
During standby
・Supplying electricity only to backup module
・Main regulator stops
・Transition to reset mode
Waiting for an interrupt
・Power supply to main regulator: ON
・Power supply stability time (Warm-Up)
・Oscillation stability time (Warm-Up)
・Reset
・Normal operation
Checking reset flag.
Checking if it can return to backup mode
Setting for interrupt
Jump to a handler.
Checking a factor to activate backup mode
Default setting of port
PTKEEP=0/BUP=0
Processed by software

27. 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 SCKSEL.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 (SCKSEL.SYSCK = 0) and a clock gear factor of 1/fc (SYSCR1.GEAR[2:0] = 000).

27.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC3} (I/O)	− 0.3 to 3.9	V
		AVCC3（A/D）	− 0.3 to 3.9	
		DVCC3	− 0.3 to 3.9	
Supply voltage		V_{IN}	− 0.3toV_{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 consumption (Ta = 85°C)		PD	600	mW
Soldering temperature (10 s)		T_{SOLDER}	260	°C
Storage temperature		T_{STG}	−40to125	°C
Operating Temperature	Exceptduring Flash W/E	T_{OPR}	-20 to 85	°C
	During Flash W/E		0 to 70	
Write/erase cycles		N_{EW}	100	cycle

Note: Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

27.2 DC ELECTRICAL CHARACTERISTICS (1/3)

Ta=−20 to 85°C

Parameter		Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
Supply voltage	AVCC3 n= DVCC3=3.3V CVSS=DVSS= AVSS= 0V	DVCC3	fosc = 8 to 10MHz fs = 30kHz to 34kHz fsys = 30KHz to 34KHz 4MHz to 80MHz	2.7		3.6	V
Low-level input voltage	P7 to P8	V_{IL1}	2.7V≤AVCC3n≤3.6V	− 0.3		0.3 AVCC3n	V
	Normal port	V_{IL2}	2.7V≤DVCC3≤3.6V			0.3 DVCC3	
	Schmitt-Triggered port	V_{IL3}	2.7V≤DVCC3≤3.6V			0.2 DVCC3	
	XT1	V_{IL5}	2.7V≤DVCC3≤3.6V			0.1DVCC3	

Note 1: Ta = 25°C, DVCC3= AVCC3n=3.3V, unless otherwise noted.

Ta=−20 to 85°C

Parameter		Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
High-level input voltage	P7 to P8	V_{IH1}	2.7V≤AVCC3≤3.6V	0.7 AVCC3		DVCC3 + 0.3	V
	Normal port	V_{IH2}	2.7V≤DVCC3≤3.6V	0.7 DVCC3			
	Schmitt-Triggered port	V_{IH3}	2.7V≤DVCC3≤3.6V	0.8 DVCC3			
	XT1	V_{IH5}	2.7V≤DVCC3≤3.6V	0.9 CVCCL			
Low-level output voltage		V_{OL}	I_{OL} = 2mA DVCC3≥2.7V			0.4	V
High-level output voltage		V_{OH}	I_{OH} = − 2mA DVCC3≥2.7V	2.4 (DVCC3=0.3)			

Note 1: Ta = 25°C, DVCC3= AVCC3n=3.3V, unless otherwise noted

27.3 DC ELECTRICAL CHARACTERISTICS (2/3)

Ta=－20 to 85°C

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
Input leakage current	I_{LI}	$0.0 \le V_{IN} \le$DVCC3 $0.0 \le V_{IN} \le$AVCC3n		0.02	± 5	μA
Output leakage current	I_{LO}	$0.2 \le V_{IN} \le$DVCC3 − 0.2 $0.2 \le V_{IN} \le$AVCC3n− 0.2		0.05	± 10	
Pull-up resister at Reset	RRST	DVCC3= 2.7Vto3.6V	20	50	150	kΩ
Schmitt-Triggered port	VTH	2.7V≤DVCC3≤3.6V	0.3	0.6		V
Programmable pull-up/ pull-down resistor	PKH	DVCC3 = 2.7V to3.6V	20	50	150	kΩ
Pin capacitance (Except power supply pins)	C_{IO}	fc = 1MHz			10	pF

Note 1: Ta = 25°C, DVCC3= AVCC3n=3.3V, unless otherwise noted.

27.4 DC ELECTRICAL CHARACTERISTICS (3/3)

DVCC3=AVCC3n=2.7V to 3.6V Ta=－20 to 85°C

TMP19A44FDAXBG

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL(Note 2): Gear = 1/1	I_{CC}	f_{sys} =-80 MHz (f_{osc} = -10 MHz)		45	55	mA
				30	40	
IDLE(Doze) (Note 3)				25	40	
IDLE(Halt) (Note 3)				23	35	
SLOW (Note 4)		fs = 32.768kHz		3	10	μA
SLEEP (Note 4)				250	3000	μA
Backup SLEEP				25	145	μA
STOP				200	2800	μA
Backup STOP				22	140	

(Note 1) Ta = 25°C, DVCC3= AVCC3=3.3V, unless otherwise noted

(Note 2) I_{CC} NORMAL:

Measured with the CPU dhrystone operating (DSU is excluded.), RAM, FLASH.

All functions operating. A/D excluded.

(Note 3) I_{CC} IDLE :

Measured with all functions stoping.

(Note 4) I_{CC} SLOW, SLEEP and Backup SLEEP:

Measured with RTC on low-speed

(Note 5) EJE=1

DVCC3=AVCC3n=2.7V to 3.6V

Ta=－20 to 85°C

TMP19A44FEXBG/F10XBG

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL(Note 2): Gear = 1/1	I_{CC}	f_{sys} =-80 MHz (f_{OSC} = -10 MHz)		50	60	mA
				30	40	
IDLE(Doze) (Note 3)				27	45	
IDLE(Halt) (Note 3)				24	40	
SLOW (Note 4)		fs = 32.768kHz		5	16	μA
SLEEP (Note 4)				340	6200	μA
Backup SLEEP				33	710	μA
STOP				290	6000	μA
Backup STOP				30	700	

(Note 6) Ta = 25°C, DVCC3= AVCC3n=3.3V, unless otherwise noted

(Note 7) I_{CC} NORMAL:

Measured with the CPU dhrystone operating (DSU is excluded.), RAM, FLASH.

All functions operating. A/D excluded.

(Note 8) I_{CC} IDLE :

Measured with all functions stoping.

(Note 9) I_{CC} SLOW, SLEEP and Backup SLEEP:

Measured with RTC on low-speed

(Note 10) EJE=1

27.5 10-bit ADC Electrical Characteristics

DVCC3=AVCC3n=VREFH=2.7V to 3.6V,
AVSS＝DVSS ,Ta=－20 to 85°C
AVCC3 load capacitance= 3.3μF, VREFH load capacitance= 3.3μF

Parameter		Symbol	Rating	Min	Typ	Max	Unit
Analog reference voltage (＋)		VREFH		2.7	3.3	3.6	V
Analog reference voltage (－)		VREFL		AVSSn	AVSSn	AVSSn	V
Analog input voltage		VAIN		VREFLn		VREFHn	V
Analog supply current	A/D conversion	IREF			2	5	mA
	Non-A/D conversion				0.02	5	μA
consumption current	A/D conversion	—			7	10	mA
INL error		—	AIN resistance ≦1.3kΩ AIN load capacitance≦0.1μF Conversion time≧1.15μs			±3	LSB
DNL error						±2	
Offset error						±4	
Fullscale error						±4	

(Note 1) 1LSB＝ (VREFH – VREFL) / 1024[V]

(Note 2) No guarantee about Relative accuracy in the multiple-channel operation

27.6 AC Electrical Characteristic

27.6.1 Separate Bus mode

(1) DVCC3＝AVCC3n=2.7Vto3.6V, Ta = －20 to 85°C

BUSCR<ALESEL>	= ”01”
BxxCS<BxW>	= “0_1010”
BxxCS<BxCSCV>	=“ 001”
BxxCS<BxRCV>	= “01”
BxxCS<BxWCV>	=“01”

No.	Parameter	Symbol	Equation		40 MHz (fsys)(Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	x		12.5		ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	x (1 + ALE) – 19		6		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x (1 + CSR) – 19		6		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (2 + ALE + W) – 46		116.5	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) - 46		91.5	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 + W) – 13		124.5		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 (1+ CSR + RWR) - 19		18.5		ns
9	$\overline{WR}$ /$\overline{HWR}$ width low	t_{WW}	x (1 + W) - 13		124.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		19		19	ns
11	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) - 19		118.5		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x (1 + CSR) - 19		6		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x (1 + ALE + W - 2) - 46		79	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (W - 2) - 10	x (W - 2 + N) – 46	90	104	ns

Note :

ALE: Number of ALE insertion

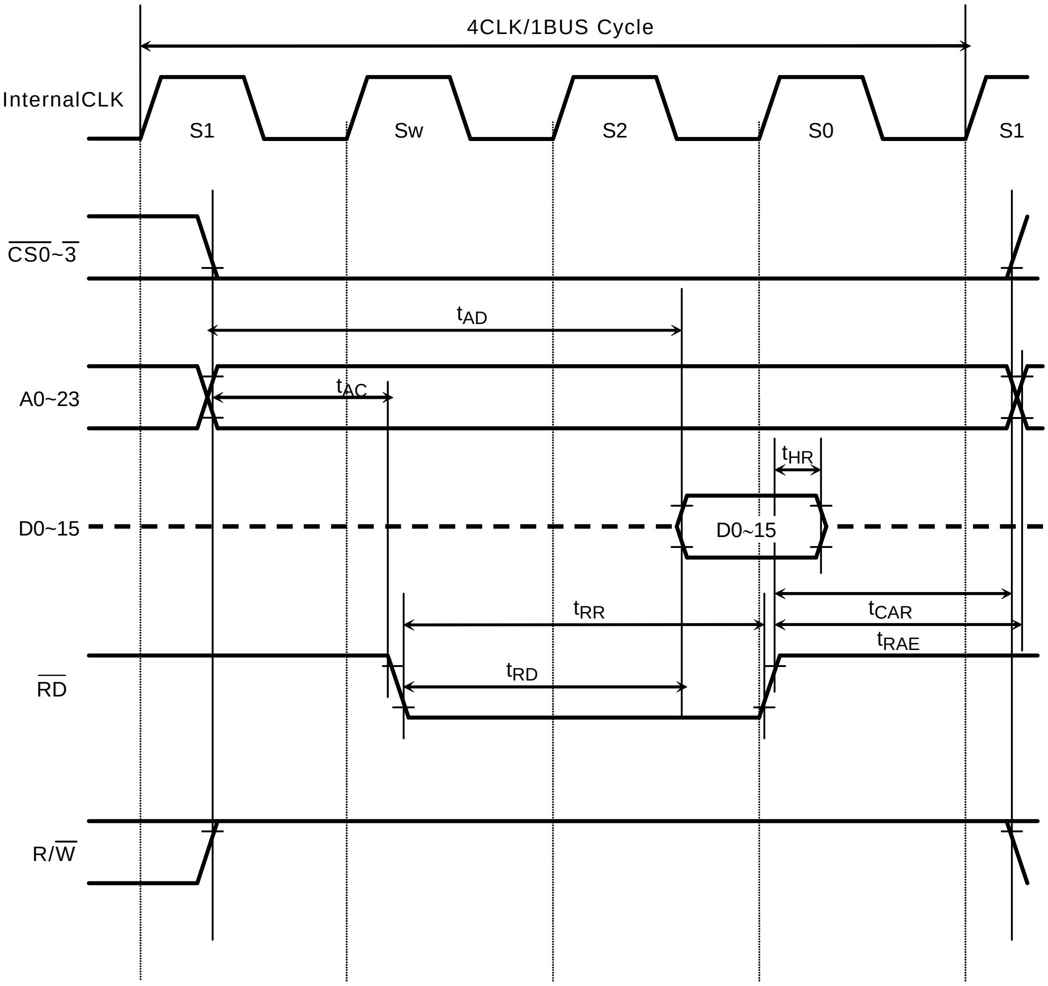
W: Number of Auto wait insertion ,

N : Number of external wait insertion

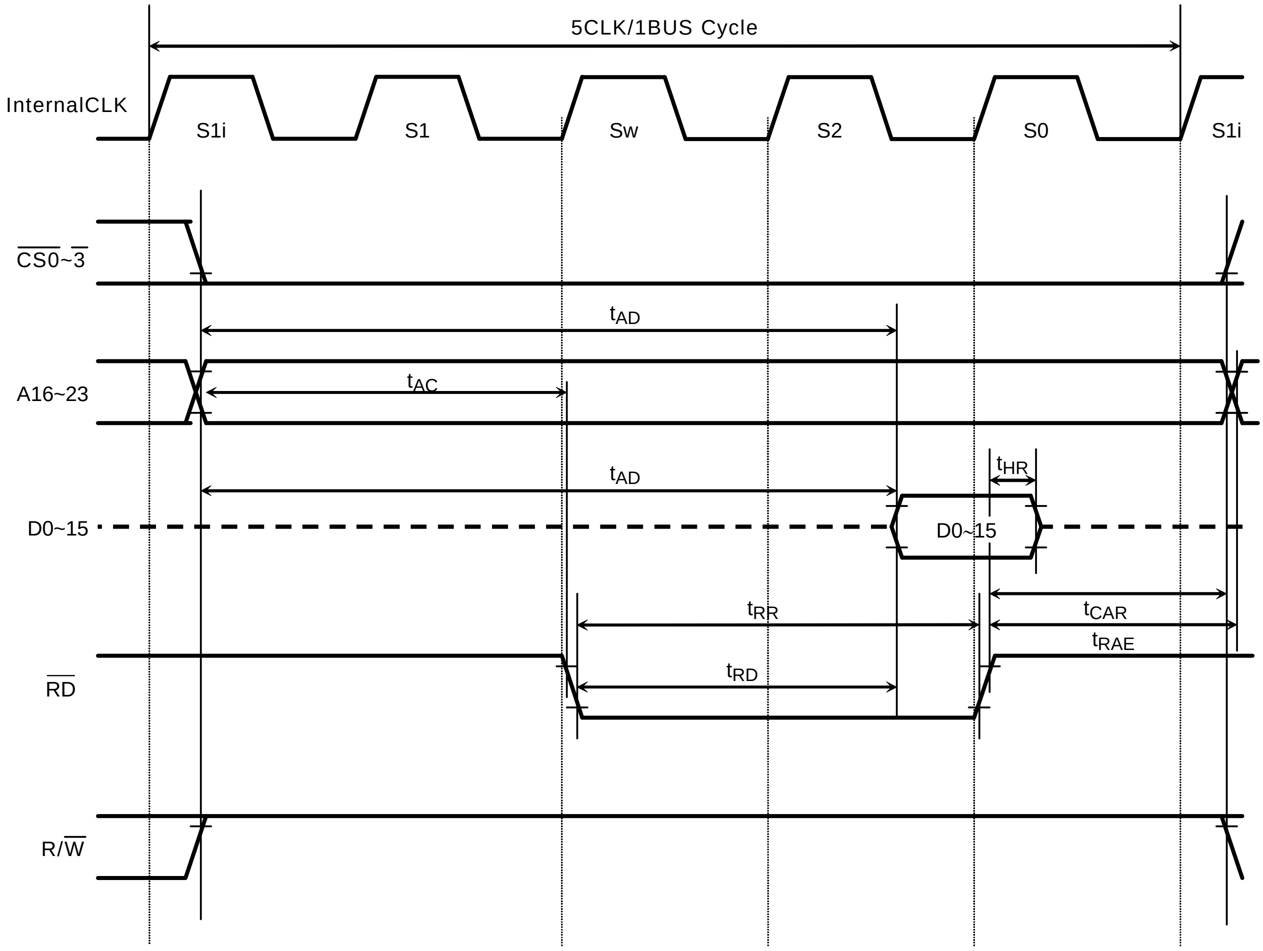
AC measurement conditions:

Output levels:	High = 0.8DVCC3 V/Low 0.2DVCC3 V, CL = 30 pF
Input levels:	High = 0.7DVCC3 V/Low 0.3DVCC3 V

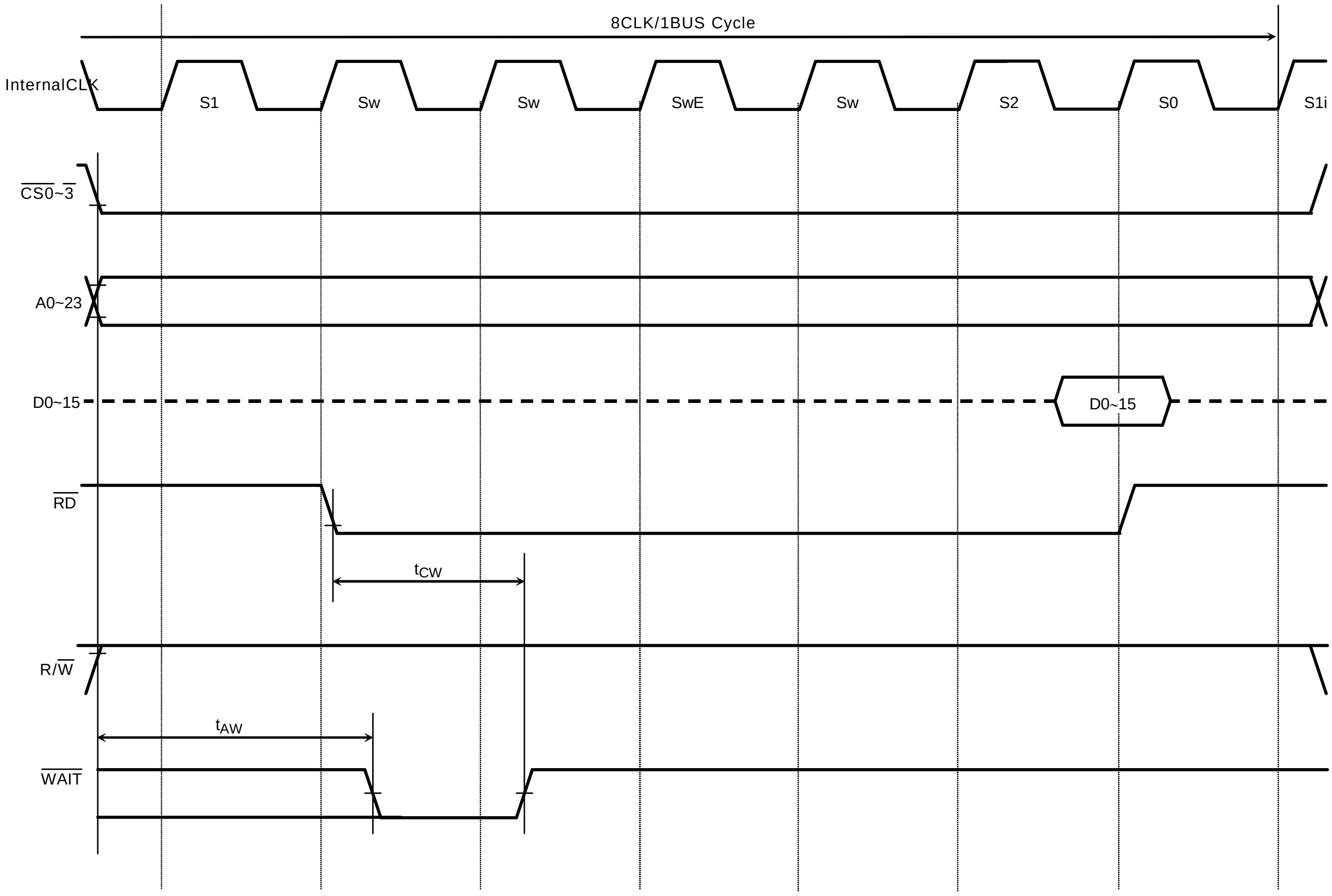
(1) Read cycle timing (BUSCR <ALESEL>= 0, 1 programmed wait state)



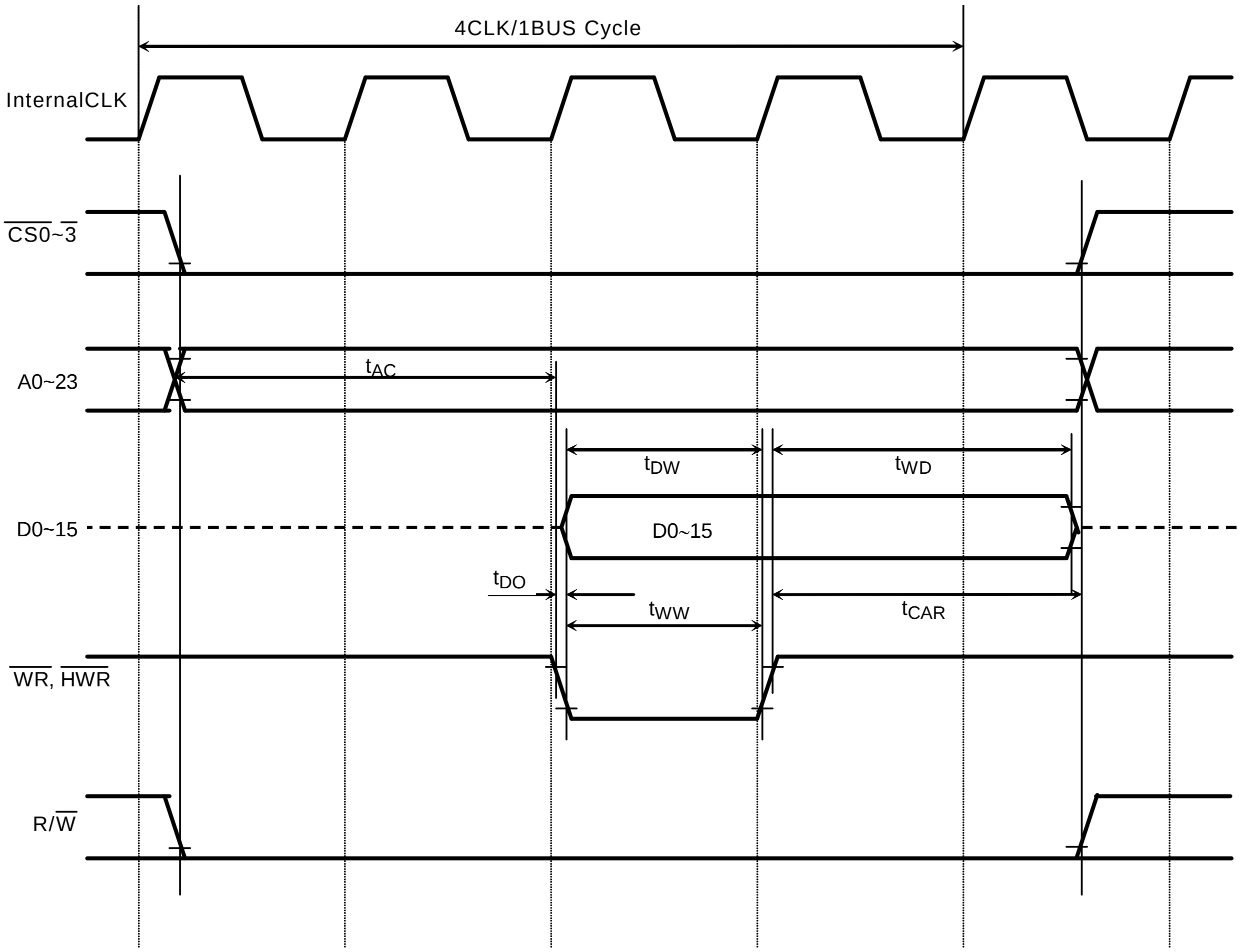
(2) Read cycle timing (BUSCR <ALESEL> = 1, 1 programmed wait state)



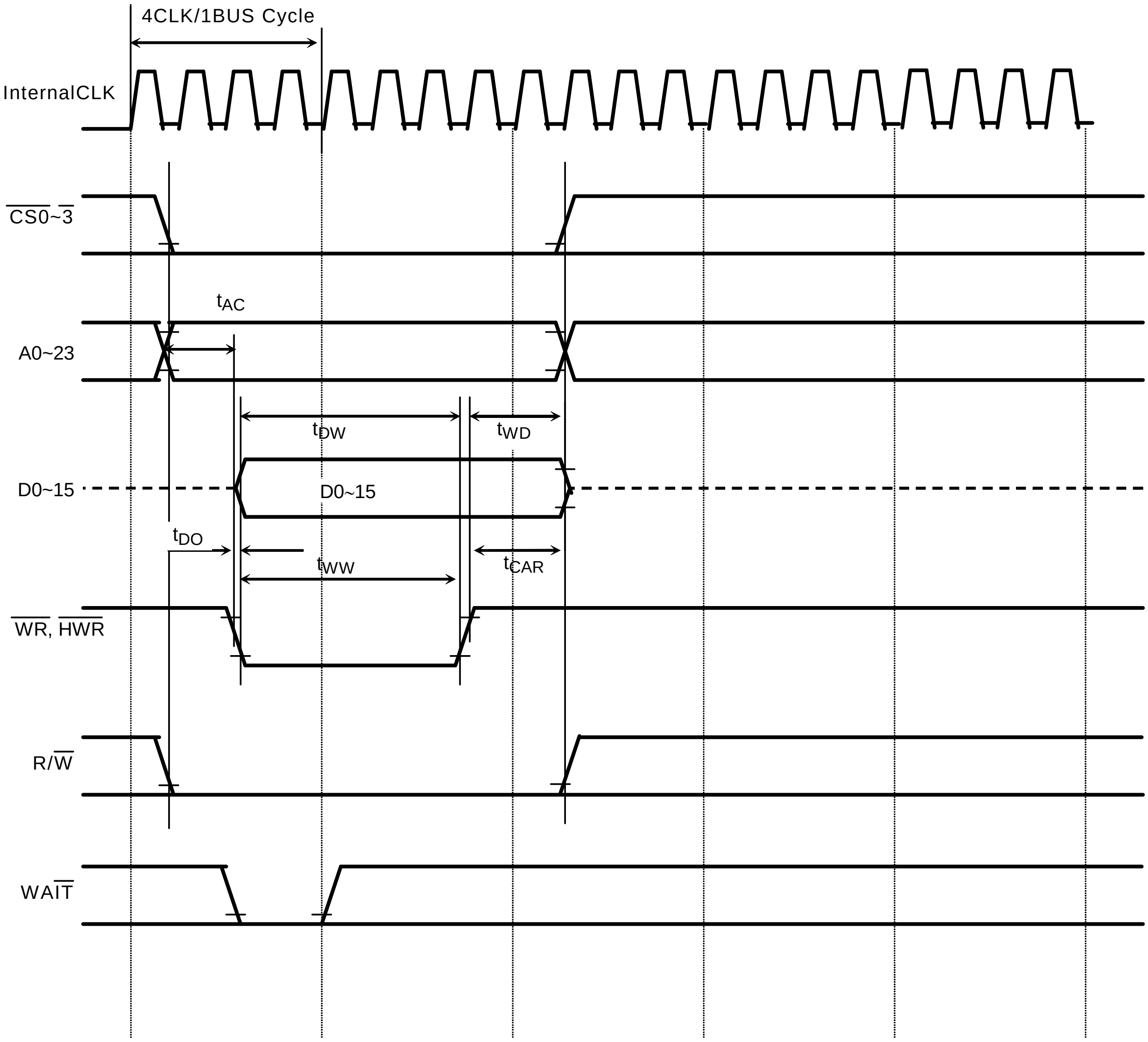
(3)Read cycle timing BUSCR <ALESEL> = 1, 4 externally generated wait states with N = 1)



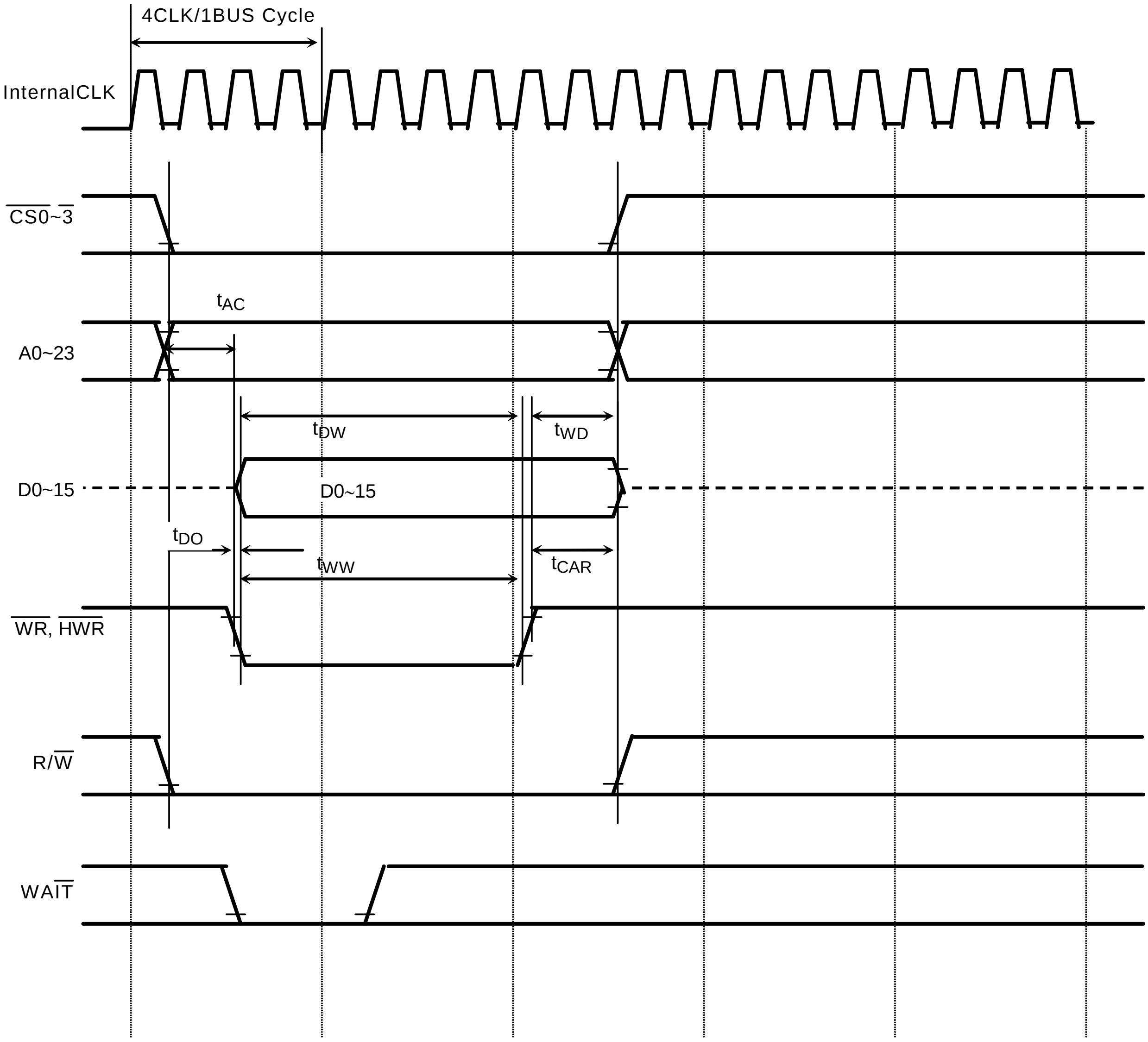
(4) Write cycle timing (BUSCR <ALESEL> = 1, zero wait sate)



(5) Write cycle timing (BUSCR <ALESEL> =1, auto 2 wait state+2N(N=1))



(6) Write cycle timing (BUSCR <ALESEL>= 1, auto 3 wait state+2N(N=1))



27.6.2 Multiplex Bus mode

(1) DVCC3＝AVCC3n= 2.7Vto3.6V,Ta =－20to85°C

BUSCR<ALESEL> ="01"

BxxCS<BxW> ="0_1010"

BxxCS<BxCSCV> ="001"

BxxCS<BxRCV> ="01"

BxxCS<BxWCV> ="01"

No.	Parameter	Symbol	Equation		40 MHz (fsys)(Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	x		12.5		ns
2	A0-A15 VALID TO ALE LOW	t_{AL}	x (1 + ALE) – 19		18.5		ns
3	A0-A15 HOLD AFTER ALE LOW	t_{LA}	x – 12		0.5		ns
4	ALE pulse width high	t_{LL}	x (1 + ALE) – 6		31.5		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x – 12		0.5		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x(1+CSR+RWR) – 19		18.5		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	x (1 + ALE) − 19		18.5		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	x (1 + ALE) − 19		18.5		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x (1＋CSR) − 19		6		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (2 + ALE + W) − 46		129	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (2 + ALE + W) − 46		129	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + W) − 46		91.5	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + W) − 13		124.5		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(1+CSR+ RWR) -19		18.5		ns
16	$\overline{WR}$/$\overline{HWR}$ width low	t_{WW}	x (1 + W) − 13		124.5		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + W) − 19		118.5		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x (1 + CSR) - 19		6		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x (1 + ALE + W - 2) - 46		91.5	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x (1 + ALE + W - 2) - 46		91.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (W – 2) – 10	x (W – 2 + N) - 46	90	104	ns

Note :

ALE: Number of ALE insertion

W: Number of Auto wait insertion ,

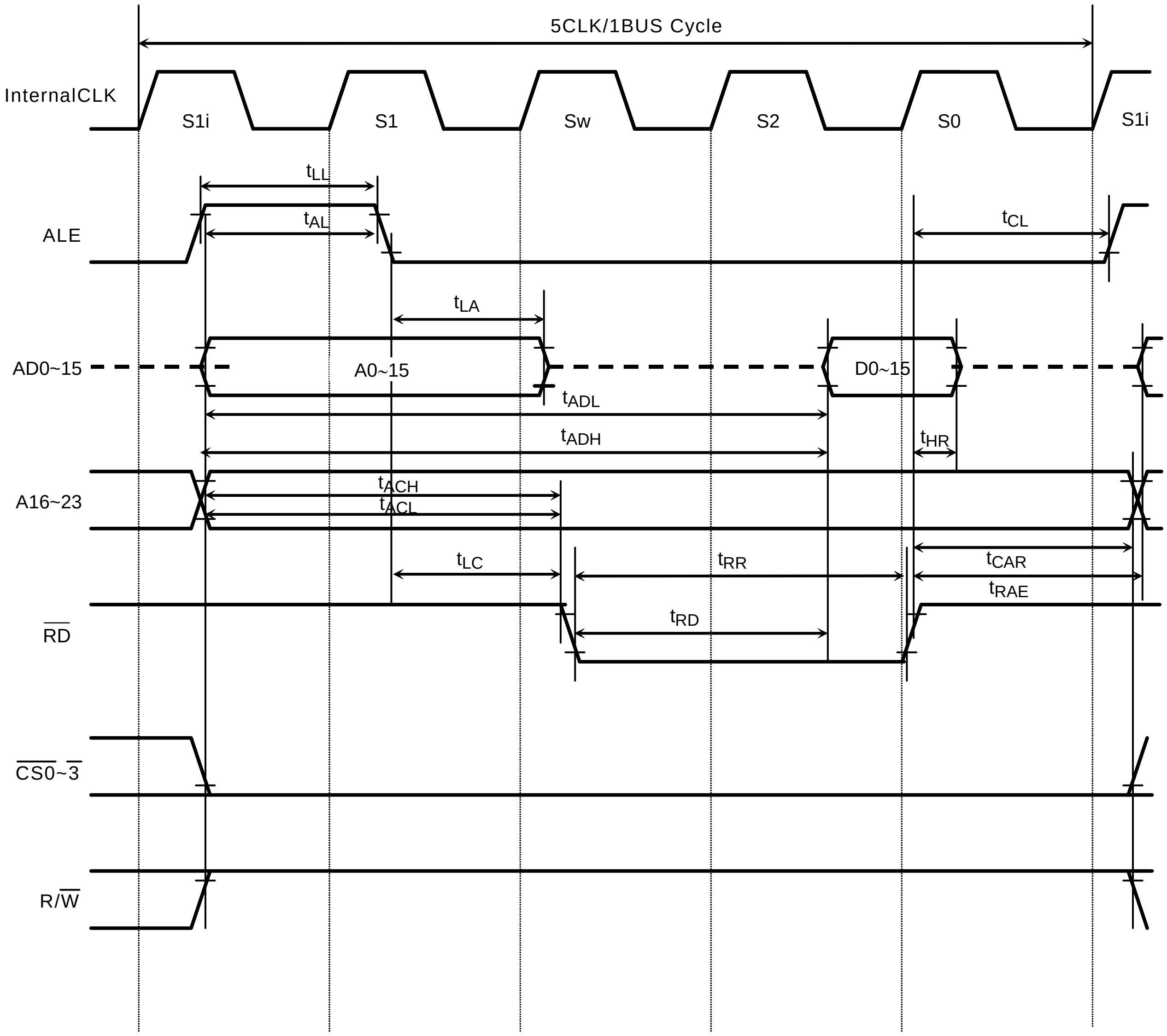
N : Number of external wait insertion

AC measurement conditions:

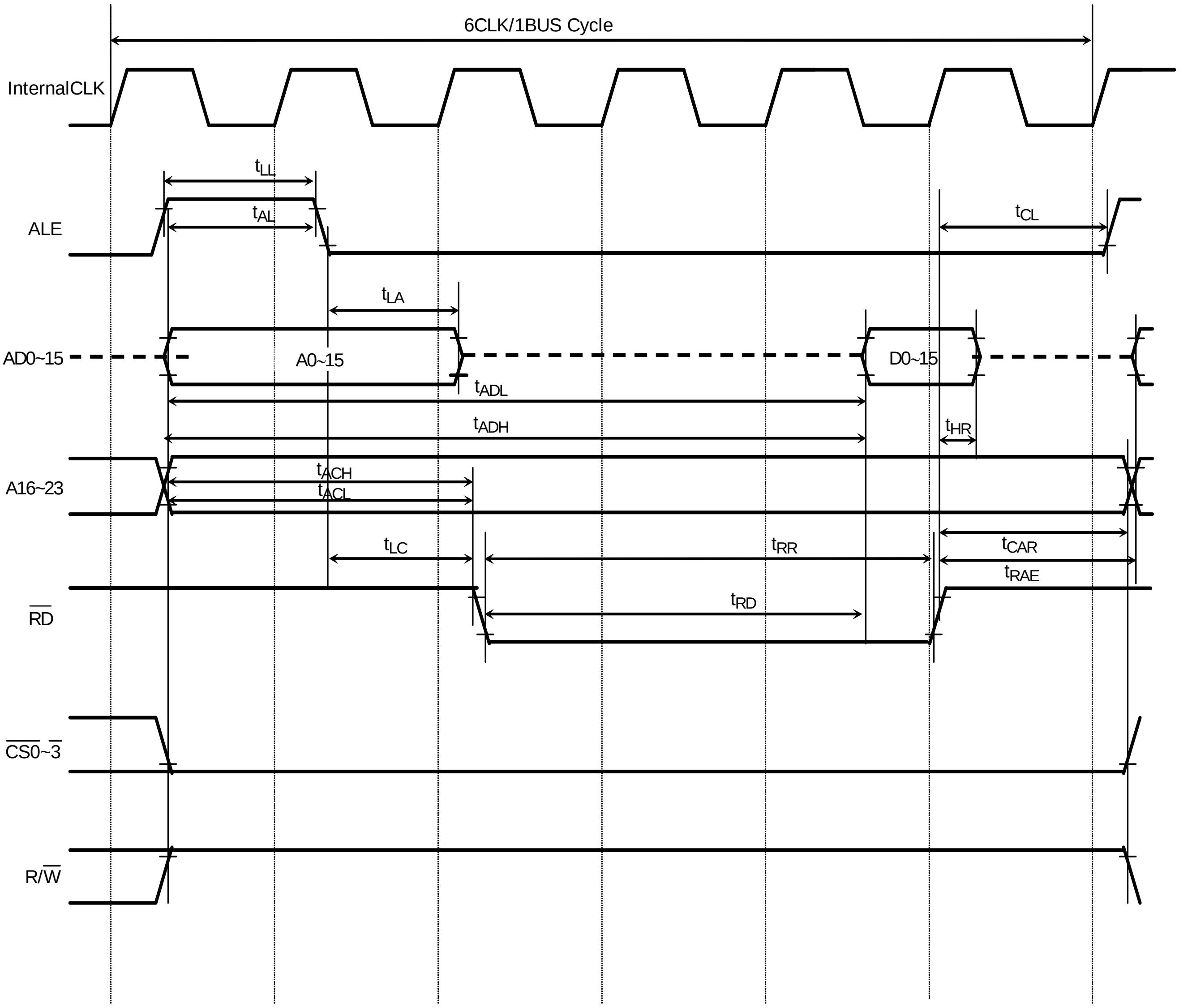
Output levels: High = 0.8DVCC3 V/Low 0.2DVCC3 V, CL = 30 pF

Input levels: High = 0.7DVCC3 V/Low 0.3DVCC3 V

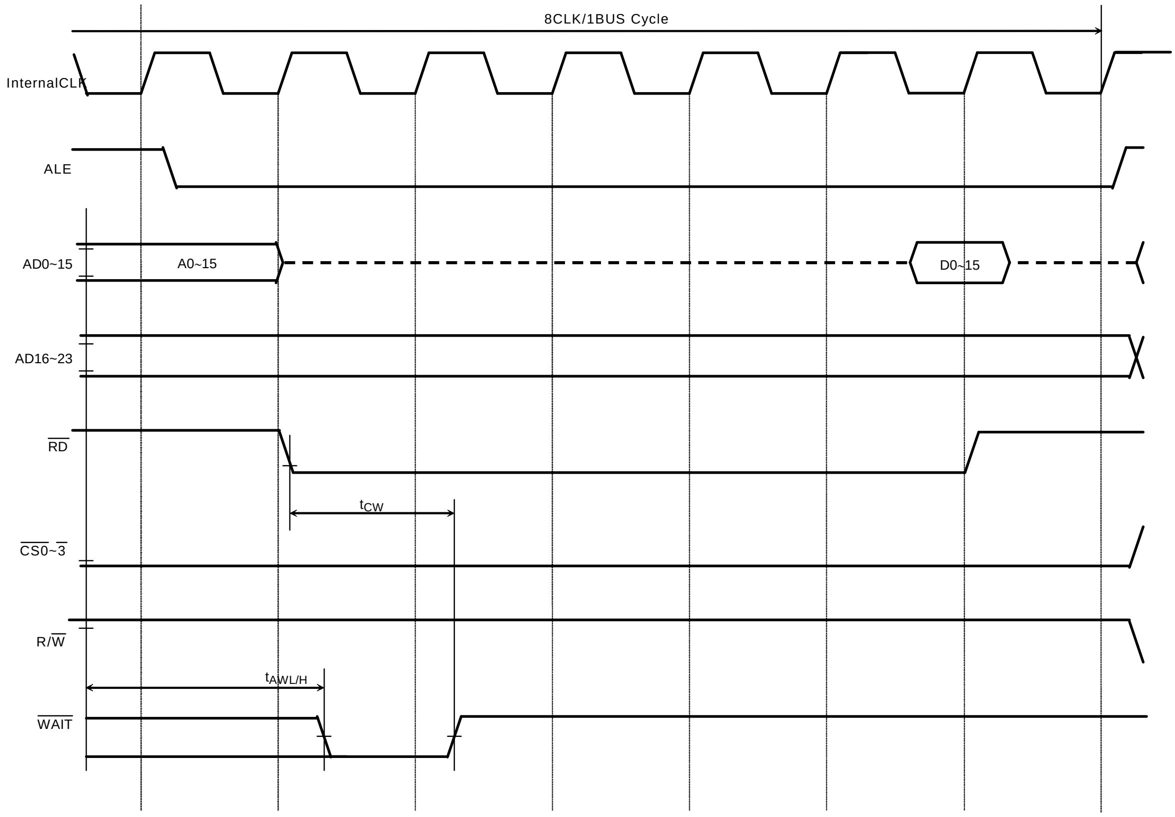
(1) Read cycle timing, ALE width = 1 clock cycle, 1 programmed wait state



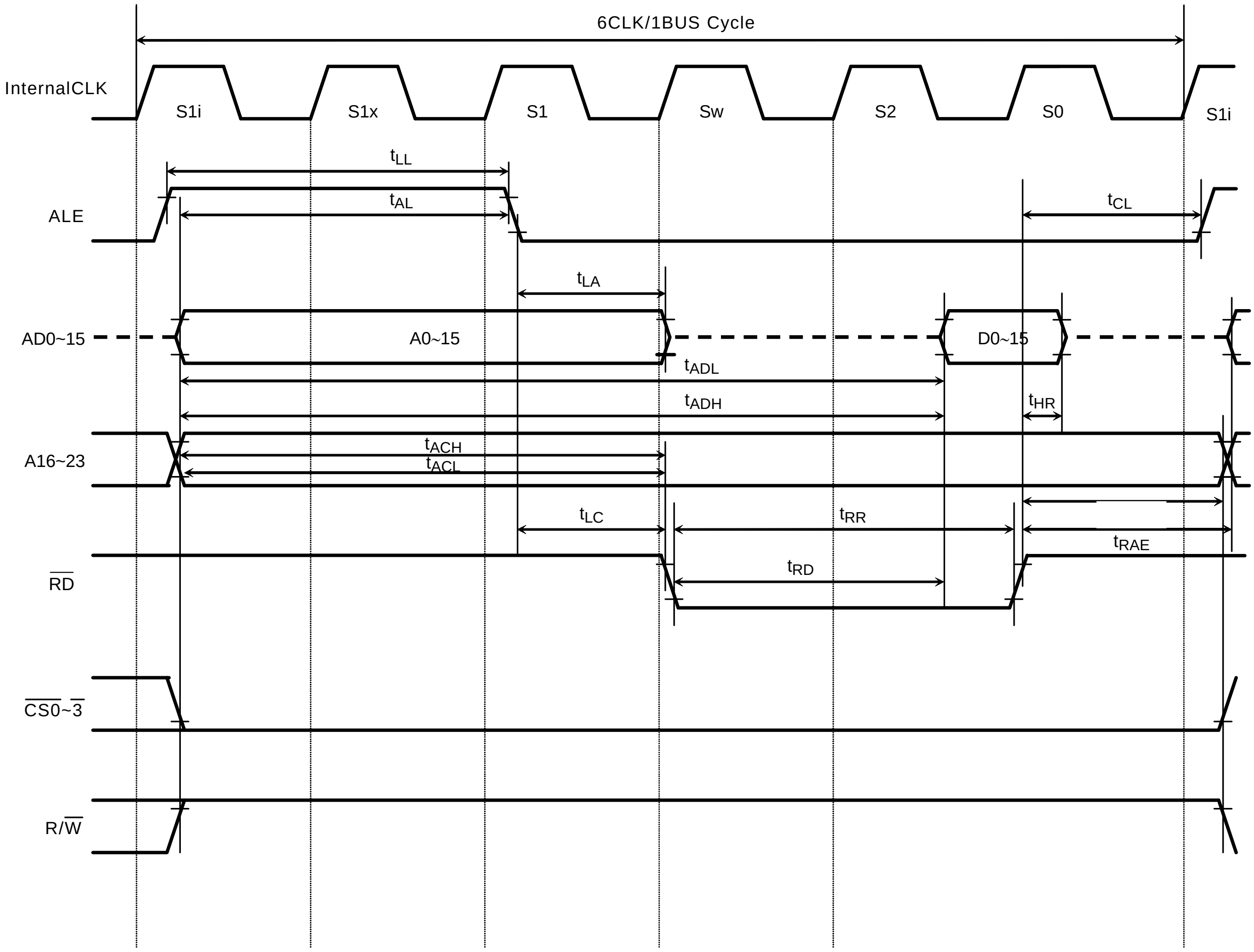
(2) Read cycle timing, ALE width = 1 clock cycle, 2 programmed wait state



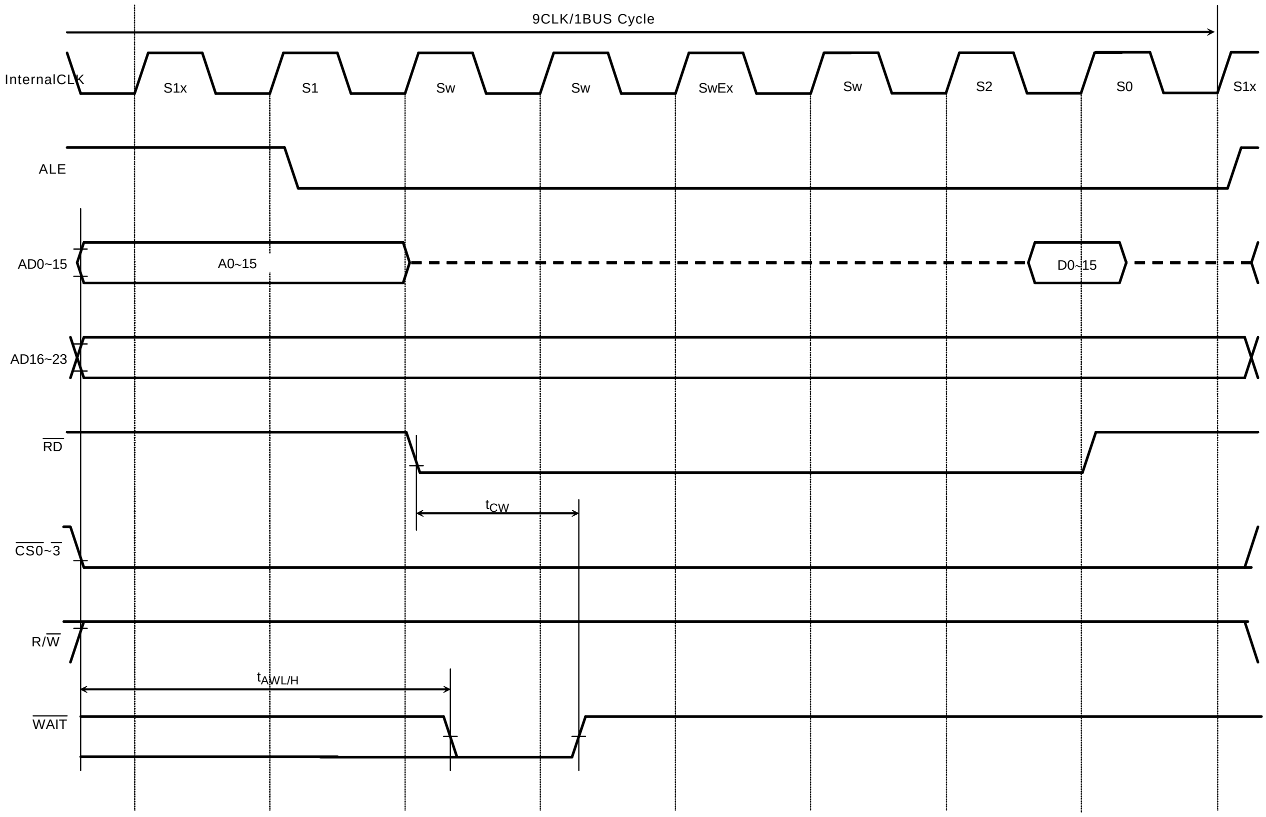
(3) Read cycle timing, ALE width = 1 clock cycle, 4 programmed wait state



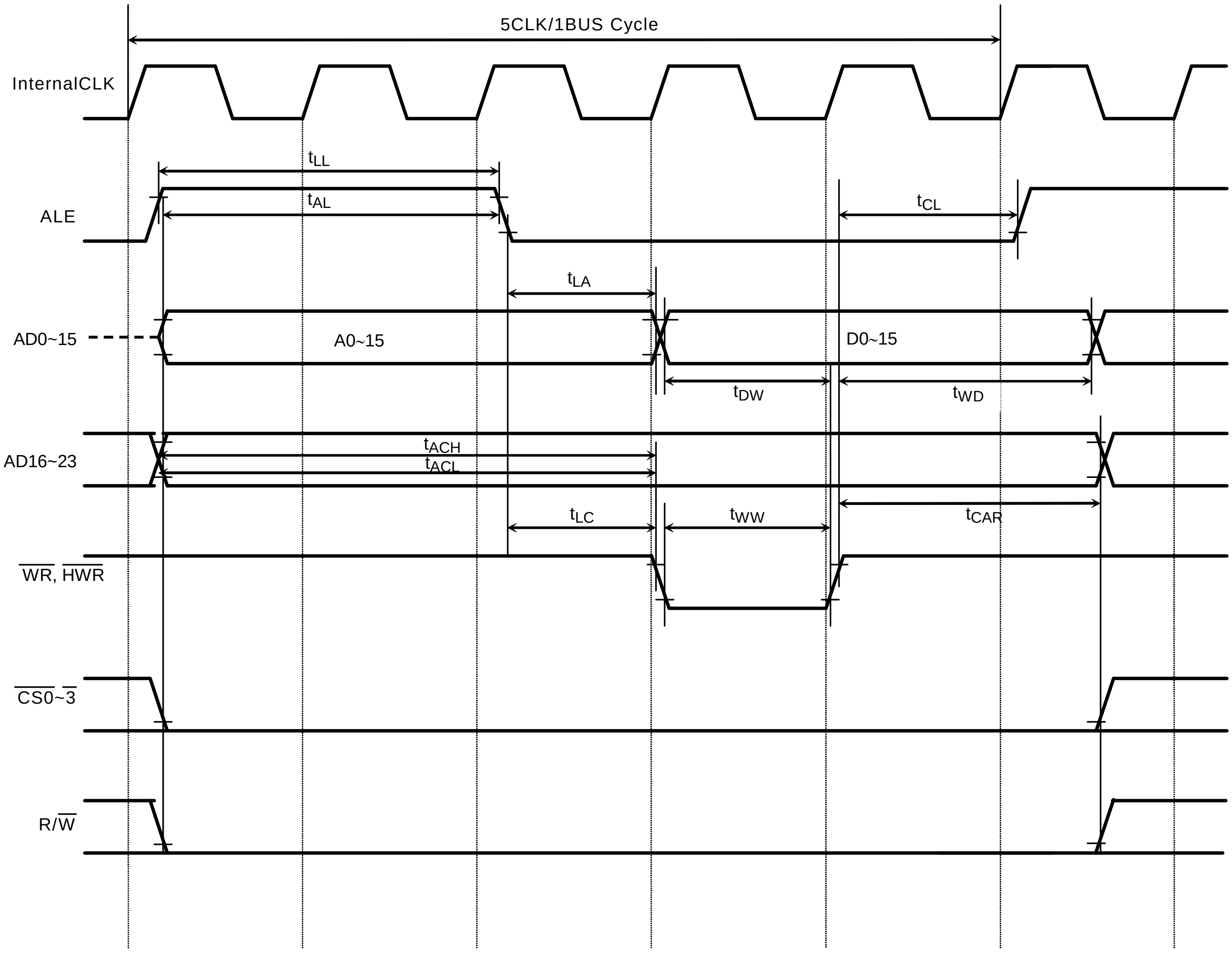
(4) Read cycle timing, ALE width = 2 clock cycle, 1 programmed wait state



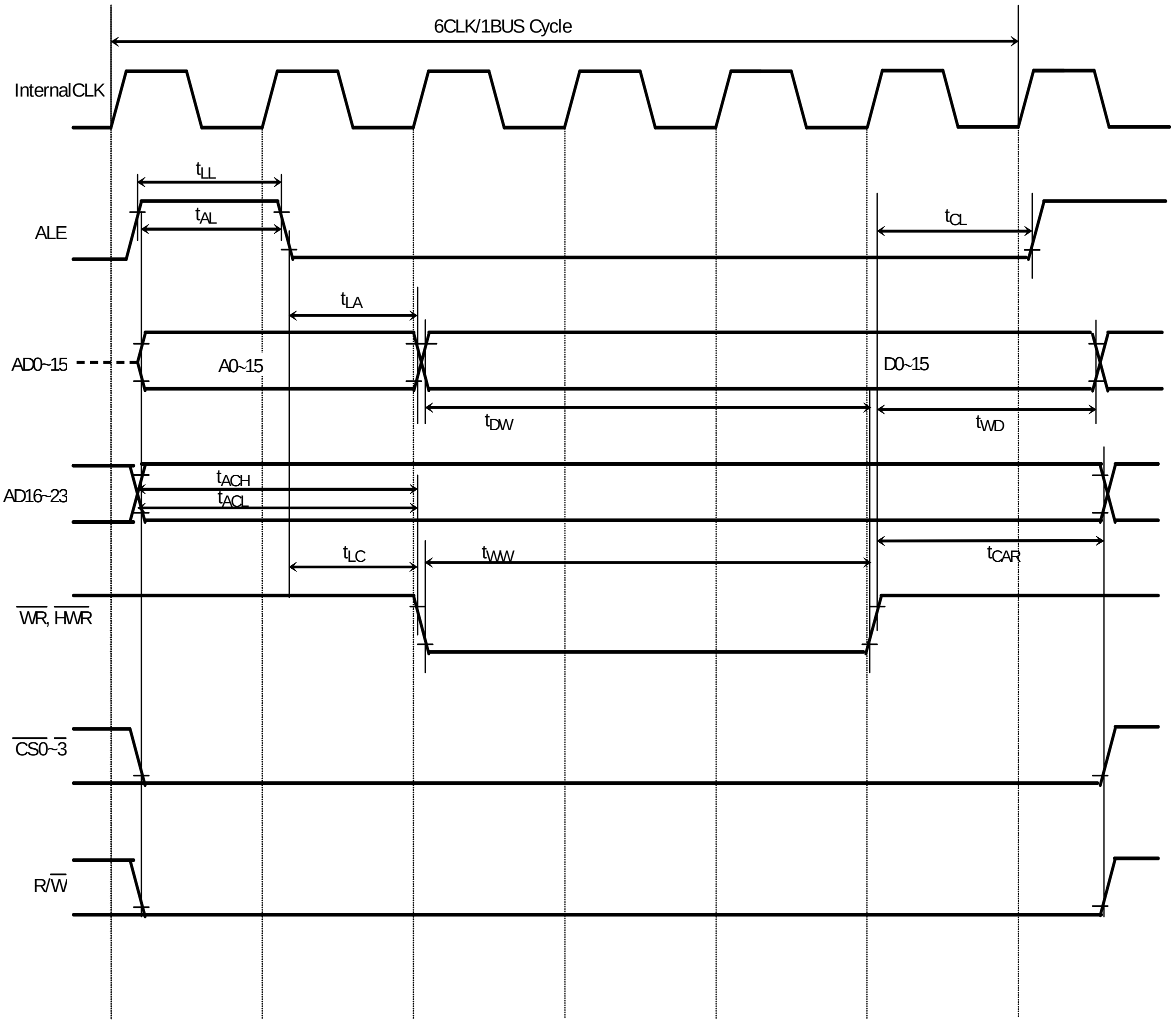
(5) Read cycle timing, ALE width = 2 clock cycle, 4 programmed wait state



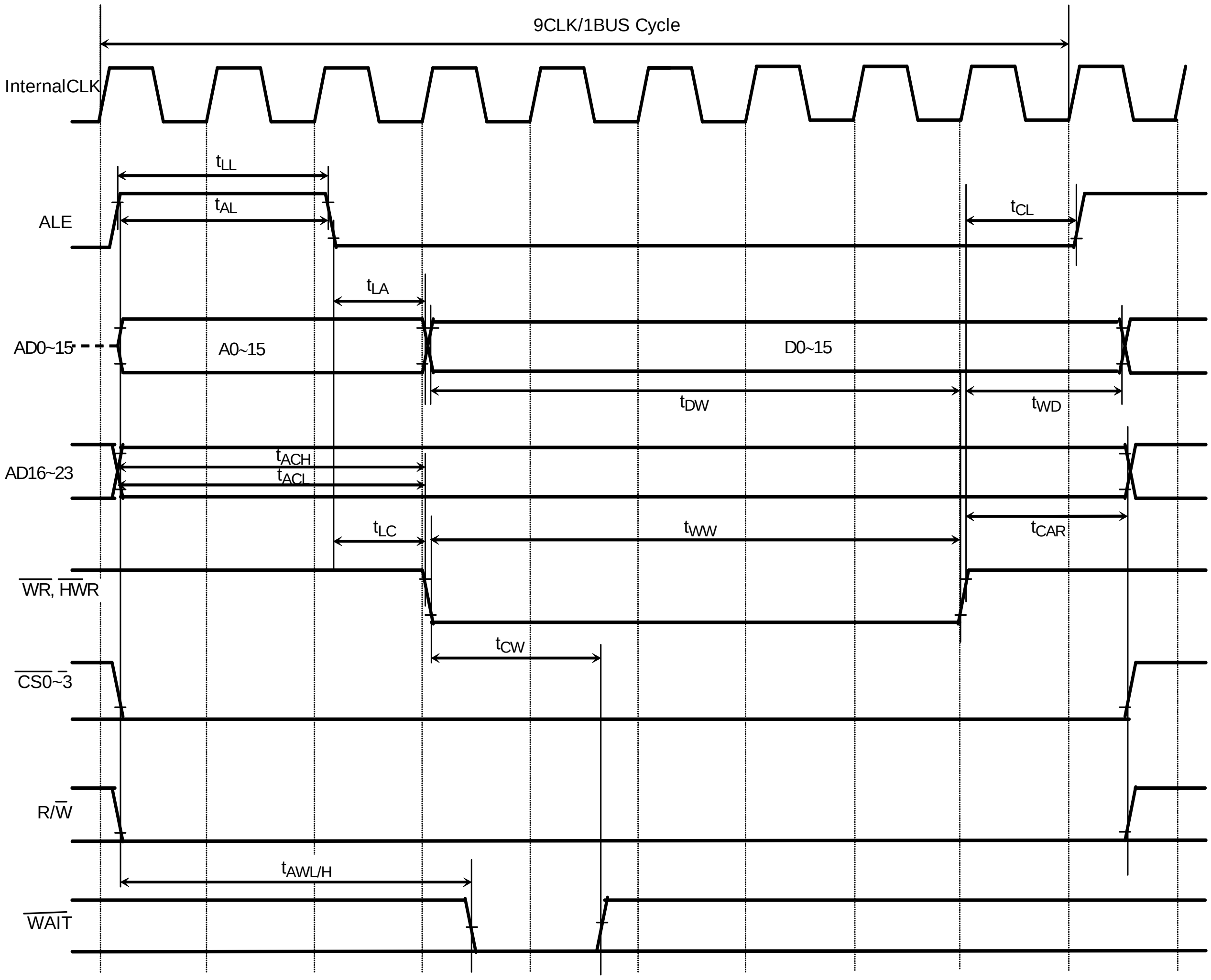
(6) Write cycle timing, ALE width = 2 clock cycles, zero wait state



(7) Write cycle timing, ALE width = 1 clock cycles, 2 wait state



(8) Write cycle timing, ALE width = 2 clock cycles, 4 wait state

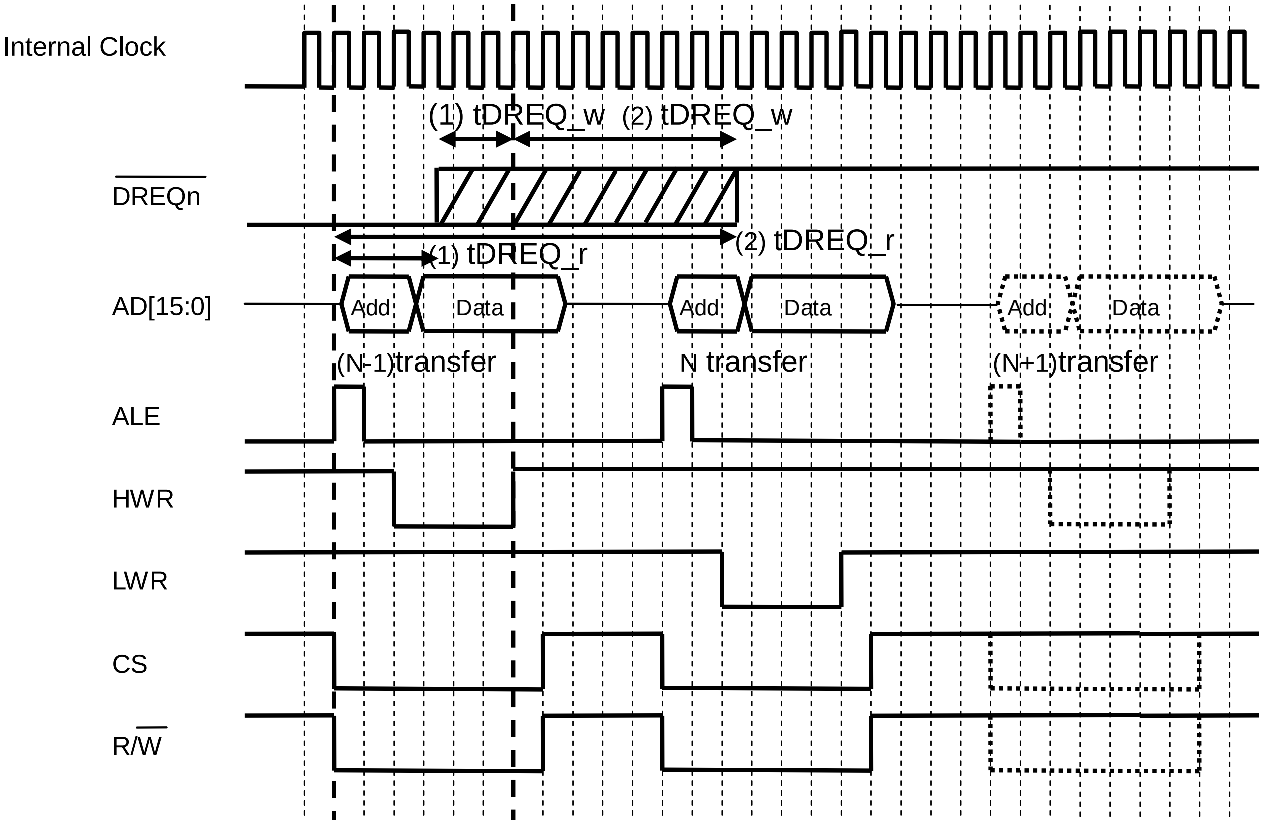


27.7 Transfer with DMA Request

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.

DVCC3=AVCC3n=2.7V to 3.6V,

Ta = −20 to 85°C

Parameter	Symbol	Equation		80 MHz (fsys)		Unit
		(1) Min	(2) Max	Min	Max	
$\overline{RD}$ asserted to $\overline{DREQn}$ negated (external device to on-chip RAM transfer)	tDREQ_r	(W+1)x	(2W＋ALE＋8)x−51	25	86.5	ns
$\overline{WR}$ / $\overline{HWR}$ rising to $\overline{DREQn}$ negated (on-chip RAM to external device transfer)	tDREQ_w	-(W+2)x	(5+WAIT)x−51.8	-37.5	23.2	ns

27.8 Serial Channel Timing

(1) I/O Interface mode（DVCC3=2.7V to 3.6V）

In the table below, the letter x represents the fsys cycle period, which varies depending on the programming of the clock gear function.

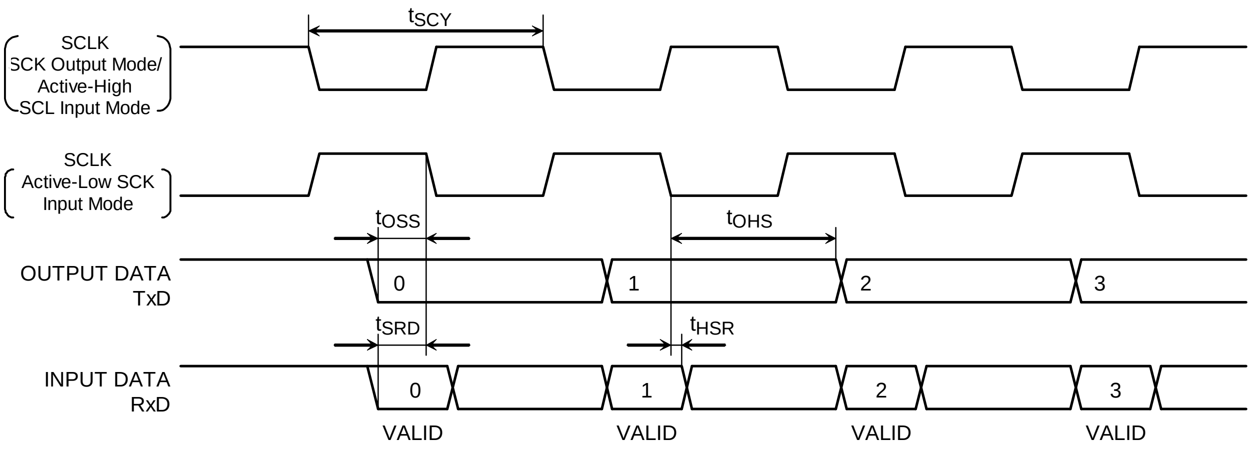
1) SCLK input mode（SIO0 to SIO2）

Parameter	Sym bol	Equation		40/ 80 MHz		Unit
		Min	Max	Min	Max	
SCLK period	tSCY	6x		150		ns
SCLK Clock High width(input)	TscH	t_{SCY} /2		75		ns
SCLK Clock Low width (input)	TscL	t_{SCY} /2		75		ns
TxD data to SCLK rise or fall*	tOSS	t_{SCY} /2 – 2x - 45		-20		ns
TxD data hold after SCLK rise or fall*	tOHS	t_{SCY} /2		75		ns
RxD data valid to SCLK rise or fall*	tSRD	30		30		ns
RxD data hold after SCLK rise or fall*	tHSR	x + 30		55		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

2) SCLK output mode （SIO0 to SIO2）

Parameter	Sym bol	Equation		40/ 80 MHz		Unit
		Min	Max	Min	Max	
SCLK period	tSCY	4x		100		ns
TxD data to SCLK rise or fall*	tOSS	t_{SCY} /2 - 20		45		ns
TxD data hold after SCLK rise or fall*	tOHS	t_{SCY} /2 - 20		45		ns
RxD data valid to SCLK rise or fall*	tSRD	45		45		ns
RxD data hold after SCLK rise or fall*	tHSR	0		0		ns



27.9 High Speed Serial Channel Timing

(1) I/O Interface mode（DVCC3=2.7V to 3.6V）

In the table below, the letter x represents the fsys cycle period, which varies depending on the programming of the clock gear function.

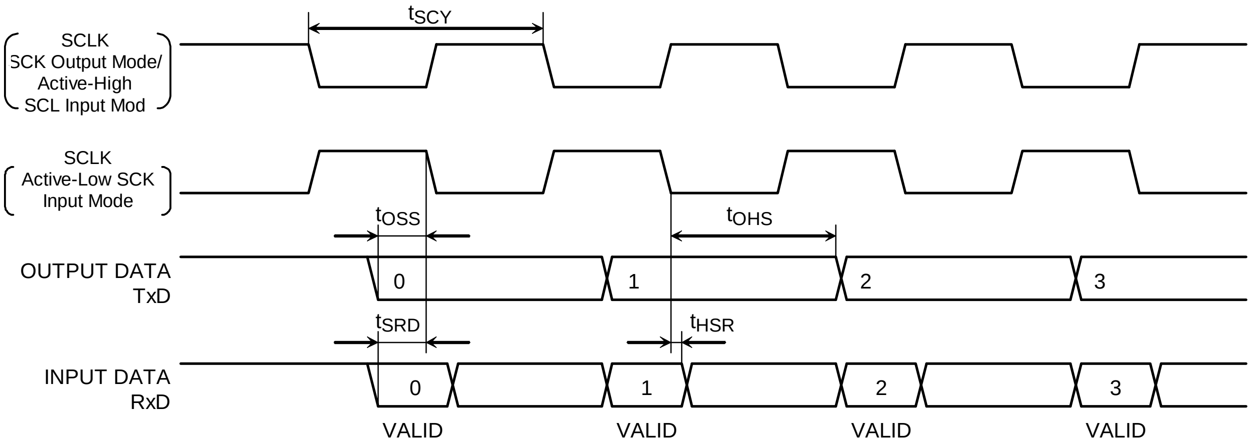
1) HSCLK input mode（HSIO0 to HSIO2）

Parameter	Symbol	Equation		40/ 80 MHz		Unit
		Min	Max	Min	Max	
HSCLK period	tSCY	8x		100		ns
HSCLK Clock High width(input)	TscH	t_{SCY} /2		50		ns
HSCLK Clock Low width (input)	TscL	t_{SCY} /2		50		ns
TxD data to HSCLK rise or fall*	tOSS	t_{SCY} /2 - 3x – 45		-32.5		ns
		t_{SCY} /2 - 3x – 36		-23.5		
TxD data hold after HSCLK rise or fall*	tOHS	t_{SCY} /2		50		ns
RxD data valid to HSCLK rise or fall*	tSRD	30		30		ns
RxD data hold after HSCLK rise or fall*	tHSR	x/2 + 30		36.25		ns

* HSCLK rise or fall: Measured relative to the programmed active edge of HSCLK.

2) HSCLK output mode （HSIO0 to HSIO2）

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
HSCLK period	tSCY	4		100		ns
TxD data to HSCLK rise or fall*	tOSS	(tSCY /2) -10		40		ns
TxD data hold after HSCLK rise or fall*	tOHS	(tSCY /2) -10		40		ns
RxD data valid to HSCLK rise or fall*	tSRD	45		45		ns
RxD data hold after HSCLK rise or fall*	tHSR	0		0		ns



27.10 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 SBI0CR.

Parameter	Symbol	Equation		Standard mode		Fast mode		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	tSCＬ	0		0	100	0	400	kHz
Hold time for START condition	tHD:STA			4.0		0.6		μs
SCL clock low width (Input) (Note 1)	tLOW			4.7		1.3		μs
SCL clock high width (Output) (Note 2)	tHIGH			4.0		0.6		μs
Setup time for a repeated START condition	tSU;STA	(Note 5)		4.7		0.6		μs
Data hold time (Input) (Note 3, 4)	tHD;DAT			0.0		0.0		μs
Data setup time	tSU;DAT			250		100		ns
Setup time for STOP condition	tSU;STO			4.0		0.6		μs
Bus free time between STOP and START conditions	tBUF	(Note 5)		4.7		1.3		μs

Note 1: SCL clock low width (output) is calculated with: $(2^{n-1}+58)/(fsys/2)$

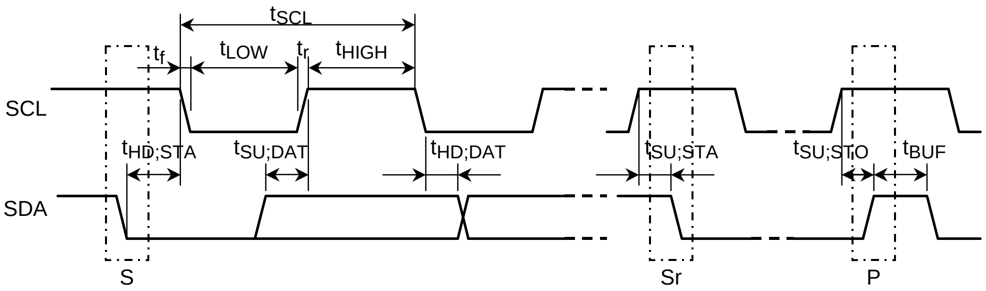
Note 2: SCL clock high width (output) is calculated with $(2^{n-1}+12)/(fsys/2)$

Notice: On I^{2}C-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^{2}C-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, this 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.

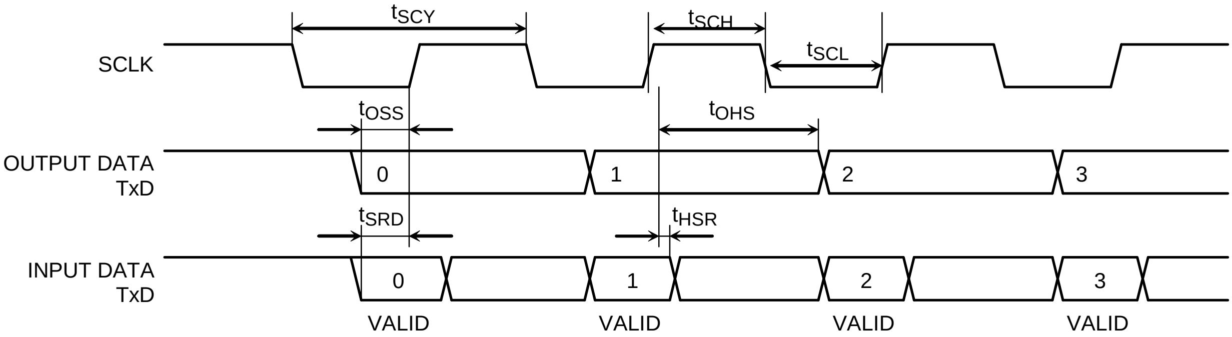
The electrical specifications below are for an SCK signal with a 50% duty cycle.

3) SCK Input mode

Parameter	Symbol	Equation		40/ 80 MHz		Unit
		Min	Max	Min	Max	
SCK period	tSCY	16x		400		ns
SCK Clock High width(input)	TscH	tSCY /2		200		ns
SCKClock Low width(input)	TscH	tSCY /2		200		ns
SO data to SCK rise	tOSS	(tSCY/2) − (6x + 20)		30		ns
SO data hold after SCK rise	tOHS	(tSCY/2) + 4x		300		ns
SI data valid to SCK rise	tSRD	0		0		ns
SI data hold after SCK rise	tHSR	4x + 10		110		ns

4) SCK Output mode

Parameter	Symbol	Equation		40/ 80 MHz		Unit
		Min	Max	Min	Max	
SCK period (programmable)	t_{SCY}	2^n ・T		800		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - 20$		380		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) - 20$		380		ns
SI data valid to SCK rise	t_{SRD}	2x + 30		55		ns
SI data hold after SCK rise	t_{HSR}	0		0		ns



27.11 Event Counter

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2X + 100		125		ns
Clock high pulse width	t_{VCKH}	2X + 100		125		ns

27.12 Timer Capture

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2X + 100		125		ns
High pulse width	t_{CPH}	2X + 100		125		ns

27.13 General Interrupts

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTA	t_{INTAL}	X + 100		112.5		ns
High pulse width for INT0-INTA	t_{INTAH}	X + 100		112.5		ns

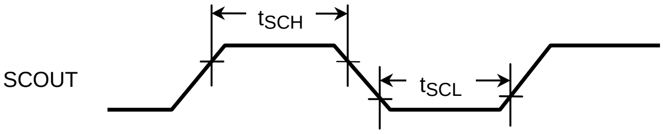
27.14 STOP /SLEEP/SLOW Wake-up Interrupts

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTB	t_{INTBL}	100		100		ns
High pulse width for INT0-INTB	t_{INTBH}	100		100		ns

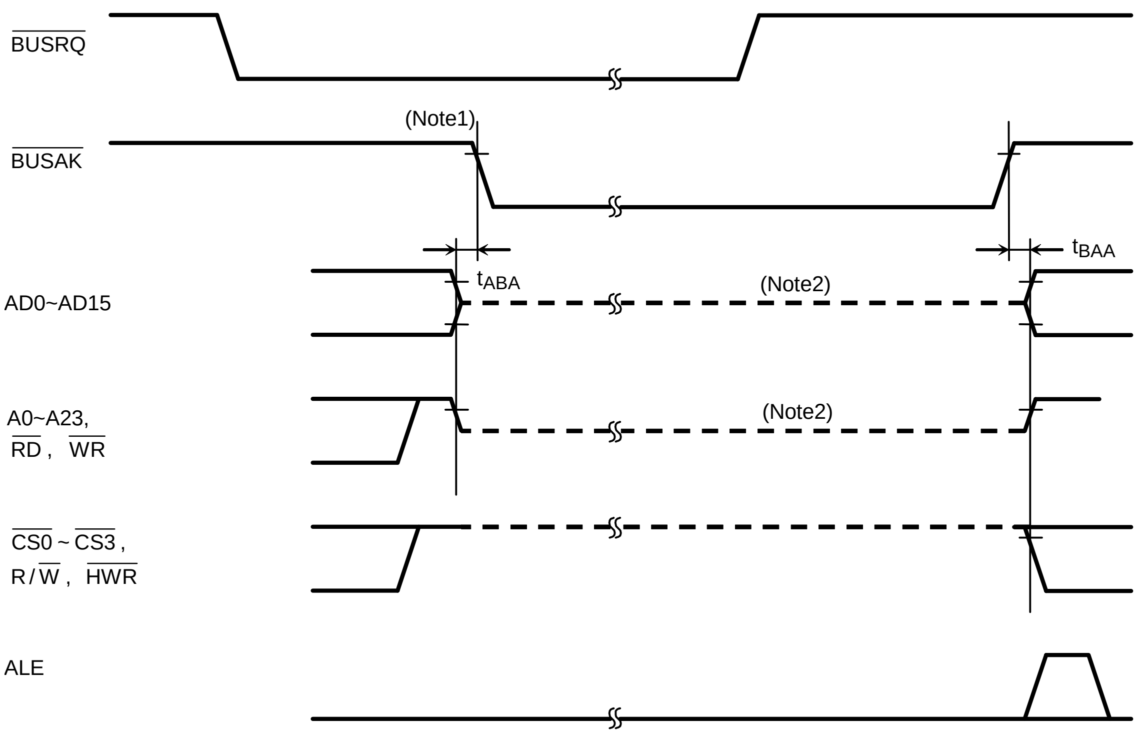
27.15 SCOUT Pin

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Clock high pulse width	t_{SCH}	0.5T − 5		1.25		ns
Clock low pulse width	t_{SCL}	0.5T − 5		1.25		ns

Note: In the above table, the letter T represents the cycle period of the SCOUT output clock.



27.16 Bus Request and Bus Acknowledge Signals


$\overline{\text{BUSRQ}}$
$\overline{\text{BUSAK}}$
(Note1)
t_{BAA}
t_{ABA}
AD0~AD15
(Note2)
A0~A23,
$\overline{\text{RD}}$, $\overline{\text{WR}}$
(Note2)
$\overline{\text{CS0}}$ ~ $\overline{\text{CS3}}$,
R/$\overline{\text{W}}$, $\overline{\text{HWR}}$
ALE

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Bus float to $\overline{\text{BUSAK}}$ asserted	t_{ABA}	0	80	0	80	ns
Bus float after $\overline{\text{BUSAK}}$ negated	t_{BAA}	0	80	0	80	ns

Note 1: If the current bus cycle has not terminated due to wait-state insertion, the TMP19A44 does not respond to $\overline{\text{BUSRQ}}$ 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.

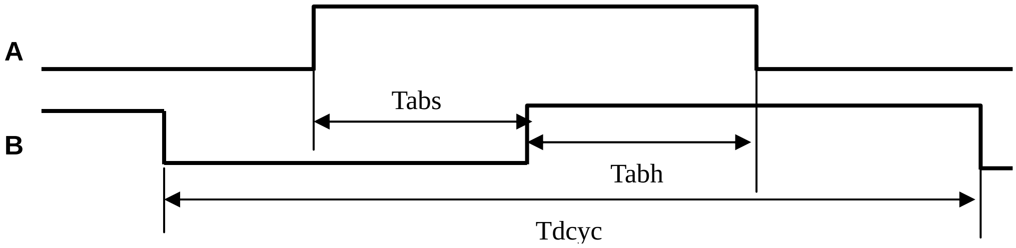
27.17 KWUP Input

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY	tky_{TBL}	100		100		ns
High pulse width for KEY	tky_{TBH}	100		100		ns

27.18 Dual Pulse Input

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Dual input pulse period	Tdcyc			TBD		ns
Dual input pulse setup	Tabs			TBD		ns
Dual input pulse hold	Tabh			TBD		ns

Y :Sampling clock (fsys/2)



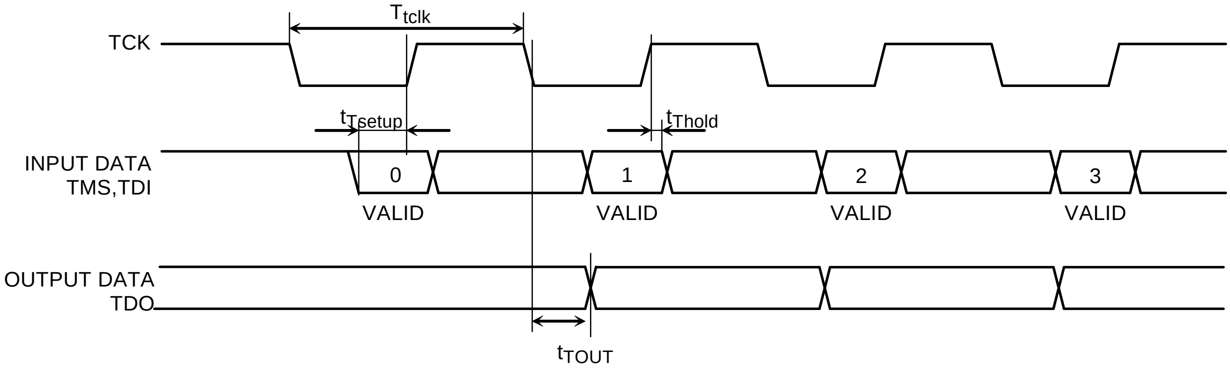
27.19 ADTRG input

Parameter	Symbol	Equation		80 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for ADTRG	tad_L	fsysy/2＋20		26.25		ns
High pulse width for ADTRG	Tadh	fsysy/2＋20		26.25		ns

27.20 EJTAG

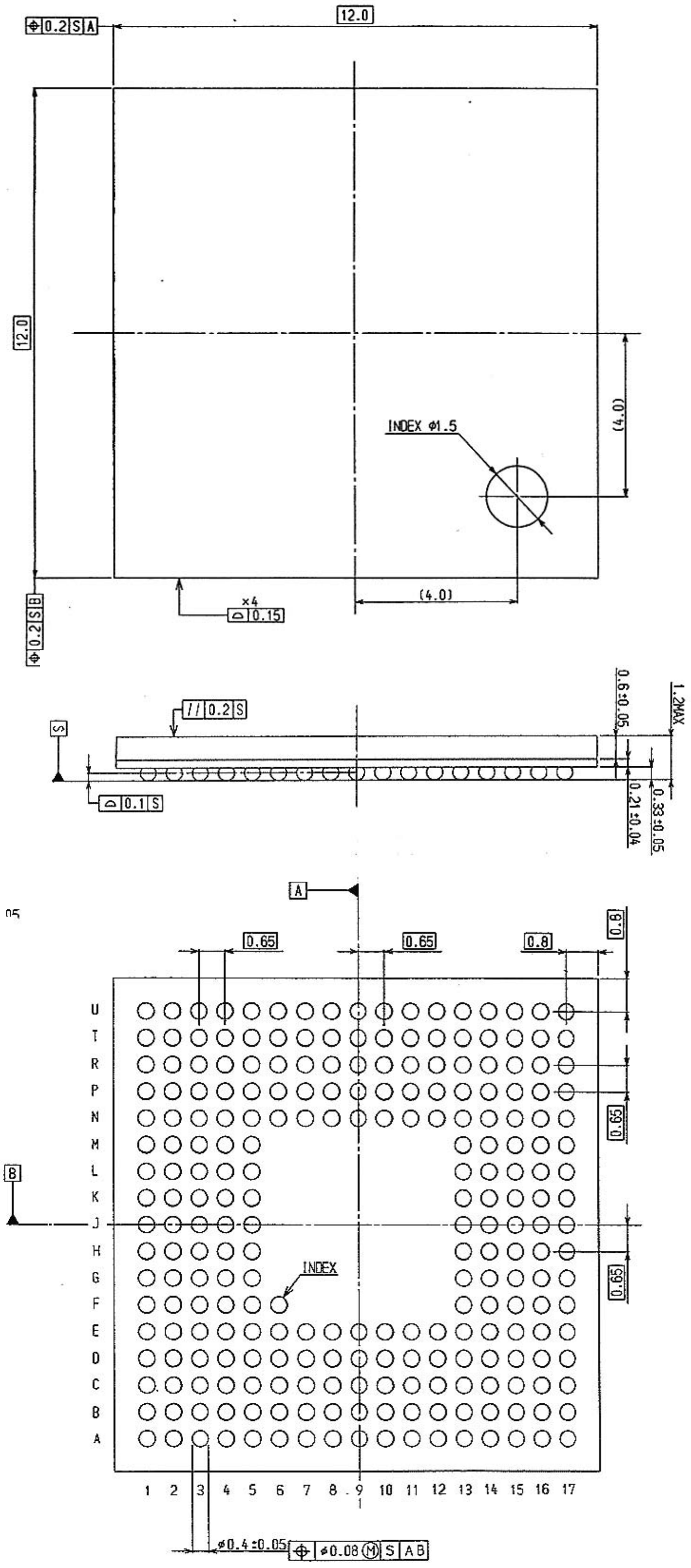
Parameter	Symbol	Equation		10 MHz(*)		Unit
		Min	Max	Min	Max	
TCK valid to TMS/TDI Data in	Ttsetup	40		40		ns
TMS/TDI hold after TCK negated	Tthold	50		50		ns
TDO hold after TCK asserted	Tt_{out}		10		10	ns

* Operating Frequency of TCK is 10MHz only



28. PKG

P-TFBGA241-1212-A5



29. RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively “TOSHIBA”), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively “Product”) without notice.

- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA’s written permission, reproduction is permissible only if reproduction is without alteration/omission.

- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**

- Product is intended for use in general electronics applications (e.g., computers, personal equipment, office equipment, measuring equipment, industrial robots and home electronics appliances) or for specific applications as expressly stated in this document. Product is neither intended nor warranted for use in equipment or systems that require extraordinarily high levels of quality and/or reliability and/or a malfunction or failure of which may cause loss of human life, bodily injury, serious property damage or serious public impact (“Unintended Use”). Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, devices related to electric power, and equipment used in finance-related fields. Do not use Product for Unintended Use unless specifically permitted in this document.

- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.

- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.

- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.

- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**

- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.

- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. TOSHIBA assumes no liability for damages or losses occurring as a result of noncompliance with applicable laws and regulations.