TOSHIBA

32bit TX System RISC
TX19A Family

TMP19A43CD/CZXBG

Rev2.0 2007.Aug.31

32-bit RISC Microprocessor - TX19 Family
TMP19A43CZXBG, CDXBG

TMP19A43FZXBG, FDXBG

1. Overview and Features

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

TMP19A43 is a 32-bit RISC microprocessor with a TX19A processor core and various peripheral functions integrated into one package. It can operate at low voltage with low power consumption.

Features of TMP19A43 are as follows:

(1) TX19A processor core

1) Improved code efficiency and operating performance have been realized through the use of two ISA (Instruction Set Architecture) modes - 16- and 32-bit ISA modes.
   - The 16-bit ISA mode instructions are compatible with the MIPS16™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.
  - 5-stage pipeline
  - Built-in high-speed memory
  - DSP function: A 32-bit multiplication and accumulation operation can be executed with one clock.
- Low power dissipation
  - Optimized design using a low power dissipation 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 ROM	Built-in RAM
TMP19A43CZXBG	384Kbyte	20Kbyte
TMP19A43CDXBG	512Kbyte	24Kbyte
TMP19A43FZXBG	384Kbyte (Flash)	20Kbyte
TMP19A43FDXBG	512Kbyte (Flash)	24Kbyte

- ROM correction function: 1 word × 8 blocks, 8 words × 4 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 (2 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 : 16 channels
- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG output (every 4 channels, synchronous outputs are possible)
- Input capture function
- 2-phase pulse input counter function (4 channels assigned to perform this function): Multiplication-by-4 mode

(6) 32-bit timer
- 32-bit input capture register : 4 channels
- 32-bit compare register : 8 channels
- 32-bit time base timer : 1 channel

(7) Clock timer : 1 channel

(8) General-purpose serial interface : 3 channels
- Selectable between the UART mode and the synchronization mode

(9) High-speed serial interface : 3 channels
- Selectable between the UART mode and the high-speed synchronization mode (maximum speed: 10 Mbps in the high-speed synchronization mode @40MHz)

(10) Serial bus interface : 1 channel
- Selectable between the I^2C bus mode and the clock synchronization mode

(11) 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
- Timer monitor function
- Conversion time 1.15 μsec(@ 40MHz)

(12) 8-bit D/A converter : 2 channels

(13) Watchdog timer : 1 channel

(14) Interrupt function
- CPU: 2 factors .....software interrupt instruction
- Internal: 46 factors.....The order of precedence can be set over 7 levels (except the watchdog timer interrupt).
- External: 48 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.

(15) Input and output ports .....143 terminals

(16) Standby function
- Three standby modes (IDLE, SLEEP, STOP)

(17) Clock generator
- Built-in PLL (multiplication by 4)
- Clock gear function: The high-speed clock can be divided into 3/4, 1/2, 1/4 or 1/8.
- Sub-clock: SLOW and SLEEP modes (32.768 kHz)

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

(19) Maximum operating frequency
- 40 MHz (PLL multiplication)

(20) Operating voltage range
- Core: 1.35 V to 1.65 V
- I/O and ADC: 2.7 V to 3.6 V
- DAC: 2.3 V to 2.7 V

(21) Package

P-FBGA193 (12 mm × 12 mm, 0.65 mm pitch)

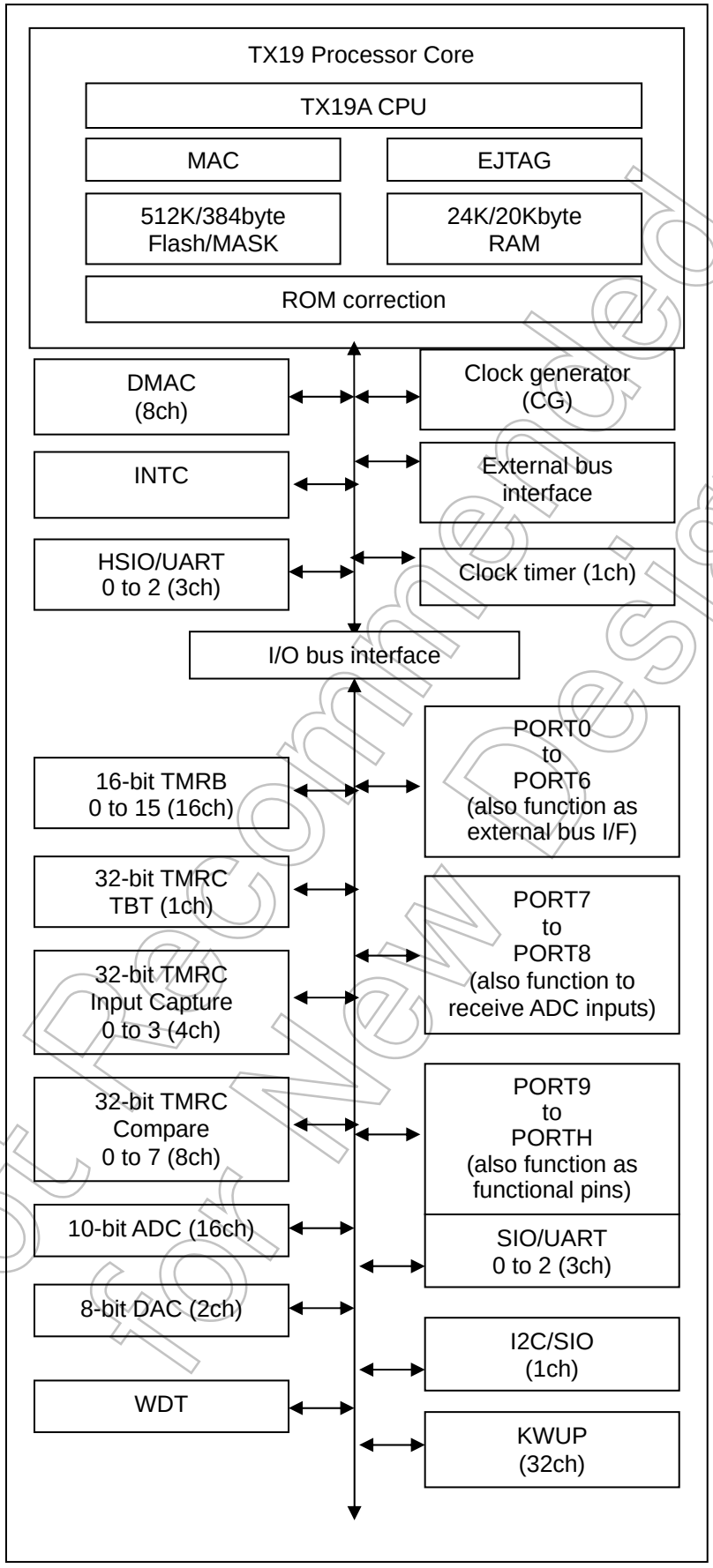


Fig. 1-1 TMP19A43 Block Diagram

2. Pin Layout and Pin Functions

This section shows the pin layout of TMP19A43 and describes the names and functions of input and output pins.

2.1 Pin Layout (Top view)

Fig. 2-1 Pin Layout Diagram (P-FBGA193) shows the pin layout of TMP19A43.

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														C16	C17
D1	D2		D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14		D16	D17
E1	E2		E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14		E16	E17
F1	F2		F4	F5	F6							F13	F14		F16	F17
G1	G2		G4	G5								G13	G14		G16	G17
H1	H2		H4	H5								H13	H14		H16	H17
J1	J2		J4	J5								J13	J14		J16	J17
K1	K2		K4	K5								K13	K14		K16	K17
L1	L2		L4	L5								L13	L14		L16	L17
M1	M2		M4	M5								M13	M14		M16	M17
N1	N2		N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14		N16	N17
P1	P2		P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14		P16	P17
R1	R2														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 Diagram (P-FBGA193)

2.2 Pin Numbers and Names

Table 2-1 shows the pin numbers and names of TMP19A43.

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	DVSS	D2	PF3/KEY19/DACK4	G2	P95/SCLK2/CTS2	M1	PB5/HTXD1	R2	P33/WAIT/RDY
A2	P81/AN9/KEY05	D4	P71/AN1	G4	P94/RXD2	M2	PB4/HSCLK0/HCTS0	R16	P45/BUSMD
A3	P83/AN11/KEY07	D5	P73/AN3	G5	P93/TXD2	M4	PB3/HRXD0	R17	P46/ENDIAN
A4	P85/AN13/INT7	D6	P74/AN4/KEY00	G13	PH1/TPC1/TPD1	M5	TEST4	T1	P37/ALE/TC3IN
A5	P87/AN15/INT9	D7	P76/AN6/KEY02	G14	PH7/TPC7/TPD7	M13	FVCC3	T2	P34/BUSRQ/TBEOUT
A6	DA0	D8	PD5/TBDOUT	G16	PCST4	M14	PG3/TPD3	T3	P30/RD
A7	CVREF0	D9	PD3/TBBOUT	G17	DCLK	M16	PG4/TPD4	T4	P02/D2/AD2
A8	DA1	D10	PD0/HTXD2	H1	PC1/TCOUT0	M17	PG5/TPD2	T5	P06/D6/AD6
A9	CVREF1	D11	PE0/KEY8	H2	PC0/TBTIN/KEY30	N1	PB7/HSCLK1/HCTS1	T6	P12/D10/AD10/A10
A10	PD2/HSCLK2/HCTS2	D12	PE3/KEY11	H4	P97/TBAOUT	N2	PB6/HRXD1	T7	P16/D14/AD14/A14
A11	PE2/KEY10	D13	PA2/INT2/TB7IN0	H5	DVCC3	N4	P00/D0/AD0	T8	P21/A17/A1/TB0IN1
A12	PE5/KEY13	D14	PH4/TPC4/TPD4	H13	PH2/TPC2/TPD2	N5	P04/D4/AD4	T9	P24/A20/A4/TB4IN0
A13	PE7/KEY15	D16	PA3/INT3/TB7IN1	H14	TRST	N6	P10/D8/AD8/A8	T10	P26/A22/A6/TB5IN0
A14	X1	D17	XT1	H16	TMS	N7	P14/D12/AD12/A12	T11	P52/A2/INTE
A15	X2	E1	PF6/KEY22/TCOUT6	H17	EJE	N8	FVCC3	T12	P56/A6/TB2OUT/KEY28
A16	CVCCH	E2	PF5/KEY21/TCOUT5	J1	PC4/TCOUT3	N9	DVSS	T13	P62/A10/SCLK0/CTS0
A17	CVSS	E4	P70/AN0	J2	PC3/TCOUT2	N10	DVCC15	T14	P66/A14/TB4OUT
B1	PF0/KEY16/DREQ0	E5	P72/AN2	J4	PC2/TCOUT1	N11	P50/A0/INTC	T15	P40/CS0/KEY24
B2	P80/AN8/KEY04	E6	VREFH	J5	DVCC15	N12	P54/A4/TB0OUT	T16	P42/CS2/KEY26
B3	P82/AN10/KEY06	E7	AVSS	J13	PH3/TPC3/TPD3	N13	P60/A8/TXD0	T17	P44/SCOUT
B4	P84/AN12/INT6	E8	DAVCC	J14	DINT	N14	P64/A12/RXD1/INTB	U1	TEST2
B5	P86/AN14/INT8	E9	DAVREF	J16	TDO	N16	PG6/TPD6	U2	P35/BUSAK/TC1IN
B6	P75/AN5/KEY01	E10	DAGND	J17	DVSS	N17	PG7/TPD7	U3	P31/WR
B7	P77/AN7/KEY03	E11	DVCC3	K1	PC7/SCK	P1	BOOT	U4	P03/D3/AD3
B8	PD6/KEY31/AFTRG	E12	PA0/INT0/TB6IN0	K2	PC6/SI/SCL	P2	P32/HWR/TC0IN	U5	P07/D7/AD7
B9	PD4/TBCOUT	E13	PA1/INT1/TB6IN1	K4	PC5/SO/SDA	P4	P01/D1/AD1	U6	P13/D11/AD11/A11
B10	PD1/HRXD2	E14	PH5/TPC5/TPD5	K5	DVSS	P5	P05/D5/AD5	U7	P17/D15/AD15/A15
B11	PE1/KEY09	E16	PCST0	K13	DVCC15	P6	P11/D9/AD9/A9	U8	P22/A18/A2/TB1IN0
B12	PE4/KEY12	E17	PCST1	K14	TOVR/TSTA	P7	P15/D13/AD13/A13	U9	P25/A21/A5/TB4IN1
B13	PE6/KEY14	F1	PF7/KEY23/TCOUT7	K16	TDI	P8	P20/A16/A0/TB0IN0	U10	P27/A23/A7/TB5IN1
B14	PA5/INT5/TB8IN1	F2	P92/TB8OUT	K17	TCK	P9	P23/A19/A3/TB1IN1	U11	P53/A3/INTF
B15	PA6/ TB2IN0	F4	P91/TB7OUT	L1	PB2/HTXD0	P10	TEST0	U12	P57/A7/TB3OUT/KEY29
B16	PA7/TB2IN1	F5	P90/TB6OUT	L2	PB1/TB3IN1	P11	P51/A1/INTD	U13	P63/A11/TXD1
B17	CVCCL	F6	AVCC3	L4	PB0/TB3IN0	P12	P55/A5/TB1OUT	U14	P67/A15/TB5OUT
C1	PF2/KEY18/DREQ4	F13	PH0/TPC0/TPD0	L5	TEST1	P13	P61/A9/RXD0/INTA	U15	P41/CS1/KEY25
C2	PF1/KEY17/DACK0	F14	PH6/TPC6/TPD6	L13	DVSS	P14	P65/A13/SCLK1/CTS1	U16	P43/CS3/KEY27
C16	PA4/INT4/TB8IN0	F16	PCST2	L14	PG0/TPD0	P16	P47/TBFOUT	U17	TEST3
C17	XT2	F17	PCST3	L16	PG1/TPD1	P17	RESET		
D1	PF4/KEY20/TCOUT4	G1	P96/TB9OUT	L17	PG2/TPD2	R1	P36/RW/TC2IN		

2.3 Pin Names and Functions

Table 2-2 through Table 2-7 show the names and functions of input and output pins.

Table 2-2 Pin Names and Functions (1 of 6)

Pin name	Number of pins	Input or output	Function
P00-P07	8	Input/output	Port 0: Input/output port (with pull-up) that allows input/output to be set in units of bits
D0-D7		Input/output	Data (lower): Data bus 0 to 7 (separate bus mode)
AD0-D7		Input/output	Address data (lower): Address data bus 0 to 7 (multiplexed bus mode)
P10-P17	8	Input/output	Port 1: Input/output port (with pull-up) that allows input/output to be set in units of bits
D8-D15		Input/output	Data (upper): Data bus 8 to 15 (separate bus mode)
AD8-AD15		Input/output	Address data (upper): Address data bus 8 to 15 (multiplexed bus mode)
A8-A15		Output	Address: Address bus 8 to 15 (multiplexed bus mode)
P20-P27	8	Input/output	Port 2: Input/output port (with pull-up) that allows input/output to be set in units of bits
A16-A23		Output	Address: Address bus 15 to 23 (separate bus mode)
A0-A7		Output	Address: Address bus 0 to 7 (multiplexed bus mode)
TB0IN0,TB0IN1		Input	16-bit timer 0 input 0,1: For inputting the count/capture trigger of a 16-bit timer 0
TB1IN0,TB1IN1		Input	16-bit timer 1 input 0,1: For inputting the count/capture trigger of a 16-bit timer 1
TB4IN0,TB4IN1		Input	16-bit timer 4 input 0,1: For inputting the count/capture trigger of a 16-bit timer 4
TB5IN0,TB5IN1		Input	16-bit timer 5 input 0,1: For inputting the count/capture trigger of a 16-bit timer 5
P30	1	Output	Port 30: Port used exclusively for output
$\overline{RD}$		Output	Read: Strobe signal for reading external memory
P31	1	Output	Port 31: Port used exclusively for output
$\overline{WR}$		Output	Write: Strobe signal for writing data of D0 to D7 pins
P32	1	Input/output	Port 32: Input/output port (with pull-up)
$\overline{HWR}$		Output	Write upper-pin data: Strobe signal for writing data of D8 to D15 pins
TC0IN		Input	For inputting the capture trigger for 32-bit timer
P33	1	Input/output	Port 33: Input/output port (with pull-up)
$\overline{WAIT}$		Input	Wait: Pin for requesting CPU to put a bus in a wait state
$\overline{RDY}$		Input	Ready: Pin for notifying CPU that a bus is ready
P34	1	Input/output	Port 34: Input/output port (with pull-up)
$\overline{BUSRQ}$		Input	Bus request: Signal requesting CPU to allow an external master to take the bus control authority
TBEOUT		Output	16-bit timer E output: Pin for outputting 16-bit timer E
P35	1	Input/output	Port 35: Input/output port (with pull-up)
$\overline{BUSAK}$		Output	Bus acknowledge: Signal notifying that CPU has released the bus control authority in response to $\overline{BUSRQ}$
TC1IN		Input	For inputting the capture trigger for 32-bit timer
P36	1	Input/output	Port 36: Input/output port (with pull-up)
R/$\overline{W}$		Output	Read/write: "1" shows a read cycle or a dummy cycle. "0" shows a write cycle.
TC2IN		Input	For inputting the capture trigger for 32-bit timer
P37	1	Input/output	Port 37: Input/output port (with pull-up)
ALE		Output	Address latch enable (address latch is enabled only if access to external memory is taking place)
TC3IN		Input	For inputting the capture trigger for 32-bit timer
P40	1	Input/output	Port 40: Input/output port (with pull-up)
$\overline{CS0}$		Output	Chip select 0: "0" is output if the address is in a designated address area.
KEY24		Input	KEY on wake up input 24: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
P41	1	Input/output	Port 41: Input/output port (with pull-up)
$\overline{CS1}$		Output	Chip select 1: "0" is output if the address is in a designated address area.
KEY25		Input	KEY on wake up input 25: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
P42	1	Input/output	Port 42: Input/output port (with pull-up)
$\overline{CS2}$		Output	Chip select 2: "0" is output if the address is in a designated address area.
KEY26		Input	KEY on wake up input 26: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter

Table 2-3 Pin Names and Functions (2 of 6)

Pin name	Number of pins	Input or output	Function
P43	1	Input/output	Port 43: Input/output port (with pull-up)
$\overline{CS3}$		Output	Chip select 3: "0" is output if the address is in a designated address area.
KEY27		Input	KEY on wake up input 27: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
P44	1	Input/output	Port 44: Input/output port (with pull-up)
SCOUT		Output	System clock output: Selectable between high- and low-speed clock outputs, as in the case of CPU
P45	1	Input/output	Port 45: Input/output port (with pull-up)
BUSMD		Input	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.)
P46	1	Input/output	Port 46: Input/output port (with pull-up)
ENDIAN		Input	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.) Input with Schmitt trigger
P47	1	Input/output	Port 47: Input/output port (with pull-up)
TBFOUT		Output	16-bit timer F output: Pin for outputting a 16-bit timer F
P50-P53	4	Input/output	Port 5: Input/output port (with pull-up) that allows input/output to be set in units of bits
A0-A3		Output	Address: Address buses 0 to 3 (separate bus mode)
INTC-INTF		Input	Interrupt request pins C to F: Selectable between "H" level, "L" level, rising edge, and falling edge Input pin with Schmitt trigger with Noise filter
P54,P55	2	Input/output	Port 5: Input/output port (with pull-up) that allows input/output to be set in units of bits
A4,A5		Output	Address: Address buses 4 and 5 (separate bus mode)
TB0OUT		Output	16-bit timer 0 output: Pin for outputting a 16-bit timer 0
TB1OUT		Output	16-bit timer 1 output: Pin for outputting a 16-bit timer 1
P56,P57	2	Input/output	Port 5: Input/output port (with pull-up) that allows input/output to be set in units of bits
A6,A7		Output	Address: Address buses 6 and 7 (separate bus mode)
TB2OUT		Output	16-bit timer 2 output: Pin for outputting a 16-bit timer 2
TB3OUT		Output	16-bit timer 3 output: Pin for outputting a 16-bit timer 3
KEY28,KEY29		Input	KEY on wake up input 28 and 29: (Dynamic pull up is selectable) Input pin with Schmitt trigger with Noise filter
P60	1	Input/output	Port 60: Input/output port (with pull-up)
A8		Output	Address: Address bus 8 (separate bus mode)
TXD0		Output	Sending serial data 0: Open drain output pin depending on the program used
P61	1	Input/output	Port 61: Input/output port (with pull-up)
A9		Output	Address: Address bus 9 (separate bus mode)
RXD0		Input	Receiving serial data 0
INTA		Input	Interrupt request pin A: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges. Input pin with Schmitt trigger with Noise filter
P62	1	Input/output	Port 62: Input/output port (with pull-up)
A10		Output	Address: Address bus 10 (separate bus mode)
$\overline{SCLK0}$		Input/output	Serial clock input/output 0
CTS0		Input	Handshake input pin Open drain output pin depending on the program used
P63	1	Input/output	Port 63: Input/output port (with pull-up)
A11		Output	Address: Address bus 11 (separate bus mode)
TXD1		Output	Sending serial data 1: Open drain output pin depending on the program used
P64	1	Input/output	Port 64: Input/output port (with pull-up)
A12		Output	Address: Address bus 12 (separate bus mode)
RXD1		Input	Receiving serial data 1
INTB		Input	Interrupt request pin B: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges. Input pin with Schmitt trigger with Noise filter

Table 2-4 Pin Names and Functions (3 of 6)

Pin name	Number of pins	Input or output	Function
P65	1	Input/output	Port 65: Input/output port (with pull-up)
A13		Output	Address: Address bus 13 (separate bus mode)
$\overline{\text{SCLK1}}$		Input/output	Serial clock input/output 1
CTS1		Input	Handshake input pin. Open drain output pin depending on the program used
P66,P67	2	Input/output	Port 6: Input/output port (with pull-up) that allows input/output to be set in units of bits
A14,A15		Output	Address: Address buses 14 and 15 (separate bus mode)
TB4OUT		Output	16-bit timer 4 output: Pin for outputting a 16-bit timer 4
TB5OUT		Output	16-bit timer 5 output: Pin for outputting a 16-bit timer 5
P70-P73	4	Input	Port 7: Port used exclusively for input (with pull-up)
AIN0-AIN3		Input	Analog input: Input from A/D converter
P74-P77	4	Input	Port 7: Port used exclusively for input (with pull-up)
AIN4-AIN7		Input	Analog input: Input from A/D converter
KEY00-KEY03		Input	KEY on wake up input 00 to 03: (Dynamic pull up is selectable) Input pin with Schmitt trigger with Noise filter
P80-P83	4	Input	Port 8: Port used exclusively for input (with pull-up)
AIN8-AIN11		Input	Analog input: Input from A/D converter
KEY04-KEY07		Input	KEY on wake up input 04 to 07: (Dynamic pull up is selectable) Input pin with Schmitt trigger with Noise filter
P84-P87	4	Input	Port 8: Port used exclusively for input (with pull-up)
AIN12-AIN15		Input	Analog input: Input from A/D converter
INT6-9			Interrupt request pins 6 to 9: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges. Input pin with Schmitt trigger with Noise filter
P90-P92	3	Input/output	Port 9: Input/output port (with pull-up) that allows input/output to be set in units of bits
TB6OUT		Output	16-bit timer 6 output: Pin for outputting a 16-bit timer 6
TB7OUT		Output	16-bit timer 7 output: Pin for outputting a 16-bit timer 7
TB8OUT		Output	16-bit timer 8 output: Pin for outputting a 16-bit timer 8
P93	1	Input/output	Port 93: Input/output port (with pull-up)
TXD2		Output	Sending serial data 2: Open drain output pin depending on the program used
P94	1	Input/output	Port 94: Input/output port (with pull-up)
RXD2		Input	Receiving serial data 2
P95	1	Input/output	Port 95: Input/output port (with pull-up)
SCLK2		Input/output	Serial clock input/output 2
$\overline{\text{CTS2}}$		Input	Handshake input pin Open drain output pin depending on the program used
P96,P97	2	Input/output	Ports 96 and 97: Input/output port (with pull-up) that allows input/output to be set in units of bits
TB9OUT		Output	16-bit timer 9 output: Pin for outputting a 16-bit timer 9
TBAOUT		Output	16-bit timer A output: Pin for outputting a 16-bit timer A
PA0	1	Input/output	Port A0: Input/output port (with pull-up)
TB6IN0		Input	16-bit timer 6 input 0: For inputting the capture trigger of a 16-bit timer 6
INT0		Input	Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges. Input pin with Schmitt trigger with Noise filter
PA1	1	Input/output	Port A1: Input/output port (with pull-up)
TB6IN1		Input	16-bit timer 6 input 1: For inputting the capture trigger of a 16-bit timer 6
INT1		Input	Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges Input pin with Schmitt trigger with Noise filter
PA2	1	Input/output	Port A2: Input/output port (with pull-up)
TB7IN0		Input	16-bit timer 7 input 0: For inputting the capture trigger of a 16-bit timer 7
INT2		Input	Interrupt request pin 0: Selectable "H" level, "L" level, rising edge, falling edge, and both rising and falling edges. Input pin with Schmitt trigger with Noise filter
PA3	1	Input/output	Port A3: Input/output port (with pull-up)
TB7IN1		Input	16-bit timer 7 input 1: For inputting the capture trigger of a 16-bit timer 7
INT3		Input	Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges. Input pin with Schmitt trigger with Noise filter

Table 2-5 Pin Names and Functions (4 of 6)

Pin name	Number of pins	Input or output	Function
PA4	1	Input/output	Port A4: Input/output port (with pull-up)
TB8IN0		Input	16-bit timer 8 input 0: For inputting the capture trigger of a 16-bit timer 8
INT4		Input	Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges Input pin with Schmitt trigger with Noise filter
PA5	1	Input/output	Port A5: Input/output port (with pull-up)
TB8IN1		Input	16-bit timer 8 input 1: For inputting the capture trigger of a 16-bit timer 8
INT5		Input	Interrupt request pin 1: Selectable between "H" level, "L" level, rising edge, falling edge, and both rising and falling edges Input pin with Schmitt trigger with Noise filter
PA6	1	Input/output	Port A6: Input/output port (with pull-up)
TB2IN0		Input	16-bit timer 2 input 0: For inputting the capture trigger of a 16-bit timer 2
PA7		Input/output	Port A7: Input/output port (with pull-up)
TB2IN1		Input	16-bit timer 2 input 1: For inputting the capture trigger of a 16-bit timer 2
PB0	1	Input/output	Port B0: Input/output port (with pull-up)
TB3IN0		Input	16-bit timer 3 input 0: For inputting the capture trigger of a 16-bit timer 3
PB1	1	Input/output	Port B1: Input/output port (with pull-up)
TB3IN1		Input	16-bit timer 3 input 1: For inputting the capture trigger of a 16-bit timer 3
PB2	1	Input/output	Port B2: Input/output port (with pull-up)
HTXD0		Output	Sending serial data 0 at high speeds: Open drain output pin depending on the program used
PB3	1	Input/output	Port B3: Input/output port (with pull-up)
HRXD0		Input	Receiving serial data 0 at high speeds
PB4	1	Input/output	Port B4: Input/output port (with pull-up)
HSCLK0		Input/output	High-speed serial clock input/output 0
$\overline{\text{HCTS0}}$		Input	Handshake input pin: Open drain output pin depending on the program used
PB5	1	Input/output	Port B5: Input/output port (with pull-up)
HTXD1		Output	Sending serial data 1 at high speeds: Open drain output pin depending on the program used
PB6	1	Input/output	Port B6: Input/output port (with pull-up)
HRXD1		Input	Receiving serial data 1 at high speeds
PB7	1	Input/output	Port B7: Input/output port (with pull-up)
HSCLK1		Input/output	High-speed serial clock input/output 1
$\overline{\text{HCTS1}}$		Input	Handshake input pin: Open drain output pin depending on the program used
PC0	1	Input/output	Port C0: Input/output port (with pull-up)
TBTIN		Input	32-bit time base timer input: For inputting a 32-bit time base timer
KEY30			KEY on wake up input 30: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
PC1-PC4	4	Input/output	Ports C1 to C4: Input/output ports (with pull-up) that allow input/output to be set in units of bits
TCOUT0-TCOUT3		Output	Outputting 32-bit timer if the result of a comparison is a match
PC5	1	Input/output	Port C5: Input/output port (with pull-up)
SO		Output	Pin for sending data if the serial bus interface operates in the SIO mode
SDA		Input/output	Pin for sending and receiving data if the serial bus interface operates in the I2C mode Open drain output pin depending on the program used Input with Schmitt trigger
PC6	1	Input/output	Port C6: Input/output port (with pull-up)
SI		Input	Pin for receiving data if the serial bus interface operates in the SIO mode
SCL		Input/output	Pin for inputting and outputting a clock if the serial bus interface operates in the I2C mode Open drain output pin depending on the program used Input with Schmitt trigger
PC7	1	Input/output	Port C7: Input/output port (with pull-up)
SCK		Input/output	Pin for inputting and outputting a clock if the serial bus interface operates in the SIO mode Open drain output pin depending on the program used

Table 2-6 Pin Names and Functions (5 of 6)

Pin name	Number of pins	Input or output	Function
PD0 HTXD2	1	Input/output Output	Port D0: Input/output port (with pull-up) Sending serial data 2 at high speeds: Open drain output pin depending on the program used
PD1 HRXD2	1	Input/output Input	Port D1: Input/output port (with pull-up) Receiving serial data 2 at high speeds
PD2 HSCLK2 $\overline{\text{HCTS2}}$	1	Input/output Input/output Input	Port D2: Input/output port (with pull-up) High-speed serial clock input/output 2 Handshake input pin: Open drain output pin depending on the program used
PD3-PD5 TBBOUT- TBDOUT	3	Input/output Output	Ports D3 to D5: Input/output ports (with pull-up) that allow input/output to be set in units of bits 16-bit timers B, C and D outputs: Pin for outputting 16-bit timers B, C and D
PD6 ADTRG KEY31	1	Input/output Input Input	Port D6: Input/output port (with pull-up) that allows input/output to be set in units of bits Pin (with Schmitt trigger) for starting A/D trigger or A/D converter from an external source KEY on wake up input 31: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
PE0-PE7 KEY08-KEY15	8	Input/output Input	Port E: Input/output port (with pull-up) that allows input/output to be set in units of bits KEY on wake up input 08 to 15: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
PF0,PF2 $\overline{\text{DREQ0,4}}$ KEY16,KEY18	2	Input/output Input Input	Port F: Input/output port (with pull-up) 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) Input with Schmitt trigger with Noise filter
PF1,PF3 $\overline{\text{DACK0,4}}$ KEY17,KEY19	2	Input/output Output Input	Port F: Input/output port (with pull-up) that allows input/output to be set in units of bits DMA acknowledge signals 0 and 4: Signal showing that DREQ0 and DREQ4 have acknowledged a DMA transfer request KEY on wake up input 16 to 19: (Dynamic pull up is selectable) Input with Schmitt trigger with Noise filter
PF4-PF7 KEY20-KEY23 TCOUT4- TCOUT7	4	Input/output Input Output	Port F: Input/output port (with pull-up) that allows input/output to be set in units of bits KEY on wake up input 20 to 23: (Dynamic pull up is selectable) Input with Schmitt trigger Outputting 32-bit timer if the result of a comparison is a match with Noise filter
PG0-PG7 TPD0-TPD7	8	Input/output Output	Port G: Input/output port (with pull-up) that allows input/output to be set in units of bits Outputting trace data from the data access address: Signal for DSU-ICE
PH0-PH7 TPC0-TPC7 TPD0-TPD7	8	Input/output Output Output	Port H: Input/output port (with pull-up) that allows input/output to be set in units of bits Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
DCLK	1	Output	Debug clock: Signal for DSU-ICE
EJE	1	Input	DSU-ICE enable: Signal for DSU-ICE (with Schmitt trigger) (with pull-up) with Noise filter
PCST4-0	4	Output	PC trace status: Signal for DSU-ICE
$\overline{\text{DINT}}$	1	Input	Debug interrupt: Signal for DSU-ICE (input with Schmitt trigger and pull-up) with Noise filter
TOVR/TSR	1	Output	Outputting the status of PD data overflow status: Signal for DSU-ICE
TCK	1	Input	Test clock input: Signal for testing DSU-ICE (with Schmitt trigger and pull-up) with Noise filter
TMS	1	Input	Test mode select input: Signal for testing DSU-ICE (with Schmitt trigger and pull-up)
TDI	1	Input	Test data input E: Signal for testing JTAG (with Schmitt trigger and pull-up)
TDO	1	Output	Test data output: Signal for testing DSU-ICE
$\overline{\text{TRST}}$	1	Input	Test reset input: Signal for testing DSU-ICE (with Schmitt trigger and pull-down) with Noise filter
$\overline{\text{RESET}}$	1	Input	Reset: Initializing LSI (with pull-up) Input with Schmitt trigger with Noise filter
X1/X2	2	Input/output	Pin for connecting a high-speed oscillator (X1: Input with Schmitt trigger)
XT1/XT2	2	Input/output	Pin for connecting a low-speed oscillator (XT1: Input with Schmitt trigger)

Table 2-7 Pin Names and Functions (6 of 6)

Pin name	Number of pins	Input or output	Function
$\overline{\text{BOOT}}$	1	Input	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. (With pull-up)
VREFH	1	Input	Pin (H) for supplying the A/D converter with a reference power supply Connect this pin to AVCC3 if the A/D converter is not used.
AVCC3	1	–	Pin for supplying the A/D converter with a power supply. Connect it to a power supply even if the A/D converter is not used.
AVSS	1	–	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
TEST0	1	Input	TEST pin: To be fixed to DVCC3 (with Schmitt trigger)
TEST1	1	Input	TEST pin: To be fixed to DVCC3
TEST2	1	Input	TEST pin: Set to OPEN.
TEST3	1	Input	TEST pin: Set to OPEN.
TEST4	1	Input	TEST pin: Set to OPEN.
CVCCH	1	–	Pin for supplying a high-frequency oscillator with power: 1.5 V power supply
CVCCL	1	–	Pin for supplying a low-frequency oscillator with power: 3 V power supply
CVSS	1	–	Oscillator GND pin (0 V)
DVCC15	3	–	Power supply pin: 1.5 V power supply
DVCC3	4	–	Power supply pin: 3 V power supply
DVSS	5	–	Power supply pin: GND pin (0 V)
DAVCC	1	–	Power supply pin for the D/A converter: 2.5 V power supply If the D/A converter is not used, connect (fix) this pin to GND.
CVREF	1	–	Reference power supply pin for the D/A converter If the D/A converter is not used, connect (fix) this pin to GND.
DAGND	1	–	GND pin (0 V) for the D/A converter Connect this pin to GND even if the D/A converter is not used.
CVREF0	1	–	Pin for connecting a stabilizing capacitor to the D/A converter
CVREF1	1	–	Pin for connecting a stabilizing capacitor to the D/A converter
DA0	1	Output	D/A converter 0 output pin
DA1	1	Output	D/A converter 1 output pin

2.4 Pin Names and Power Supply Pins

Table 2-8 Pin Names and Power Supplies

Pin name	Power supply	Pin name	Power supply
P0	DVCC3	PCST4-0	DVCC3
P1	DVCC3	DCLK	DVCC3
P2	DVCC3	$\overline{\text{EJE}}$	DVCC3
P3	DVCC3	$\overline{\text{TRST}}$	DVCC3
P4	DVCC3	TDI	DVCC3
P5	DVCC3	TDO	DVCC3
P6	DVCC3	TMS	DVCC3
P7	AVCC3	TCK	DVCC3
P8	AVCC3	$\overline{\text{DINT}}$	DVCC3
P9	DVCC3	TOVR/TSTA	DVCC3
PA	DVCC3	BUSMD	DVCC3
PB	DVCC3	$\overline{\text{BOOT}}$	DVCC3
PC	DVCC3	X1, X2	CVCCH
PD	DVCC3	XT1, XT2	CVCCL
PE	DVCC3	$\overline{\text{RESET}}$	DVCC3
PF	DVCC3	DA0,1	DAVCC
PG	DVCC3		
PH	DVCC3		

2.5 Pin Numbers and Power Supply Pins

Table 2-9 Pin Numbers and Power Supplies

Power supply	Pin number	Voltage range
DVCC15	J5, K13, N10	1.35 V to 1.65 V
DVCC3	E11, H5	1.65 V to 3.6 V
AVCC3	F6	2.7 V to 3.6 V
FVCC3	M13, N8	2.7 V to 3.6 V
CVCCH	A16	1.35 V to 1.65 V
CVCCL	B17	2.7 V to 3.6 V
DAVCC	E8	2.3 V to 2.7 V

3. Processor Core

The TMP19A43 has a high-performance 32-bit processor core (TX19A processor core). For information on the operations of this processor core, please refer to the "TX19A Family Architecture."

This chapter describes the functions unique to the TMP19A43 that are not explained in that document.

3.1 Reset Operation

To reset the device, ensure that the power supply voltage is in the operating voltage range, the oscillation of the internal high-frequency oscillator has stabilized at the specified frequency and that the $\overline{\text{RESET}}$ input has been "0" for at least 12 system clocks (2.4 μs during external 10 MHz operation).

Note that the PLL multiplication clock is quadrupled and the clock gear is initialized to the 1/8 mode during the reset period. When the reset request is authorized, the system control coprocessor (CP0) register of the TX19A processor core is initialized. For further details, please refer to the chapter about architecture.

After the reset exception handling is executed, the program branches off to the exception handler. The address to which the program branches off to (address where exception handling starts) is called an exception vector address. This exception vector address of a reset exception (for example, nonmaskable interrupt) is 0xBFC0_0000H (virtual address).

The register of the internal I/O is initialized.

The port pin (including the pin that can also be used by the internal I/O) is set to a general-purpose input or output port mode.

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

(Note 2) 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.

4. Memory Map

Fig. 4-1 shows the memory map of the TMP19A43FDXBG/TMP19A43CDXBG.

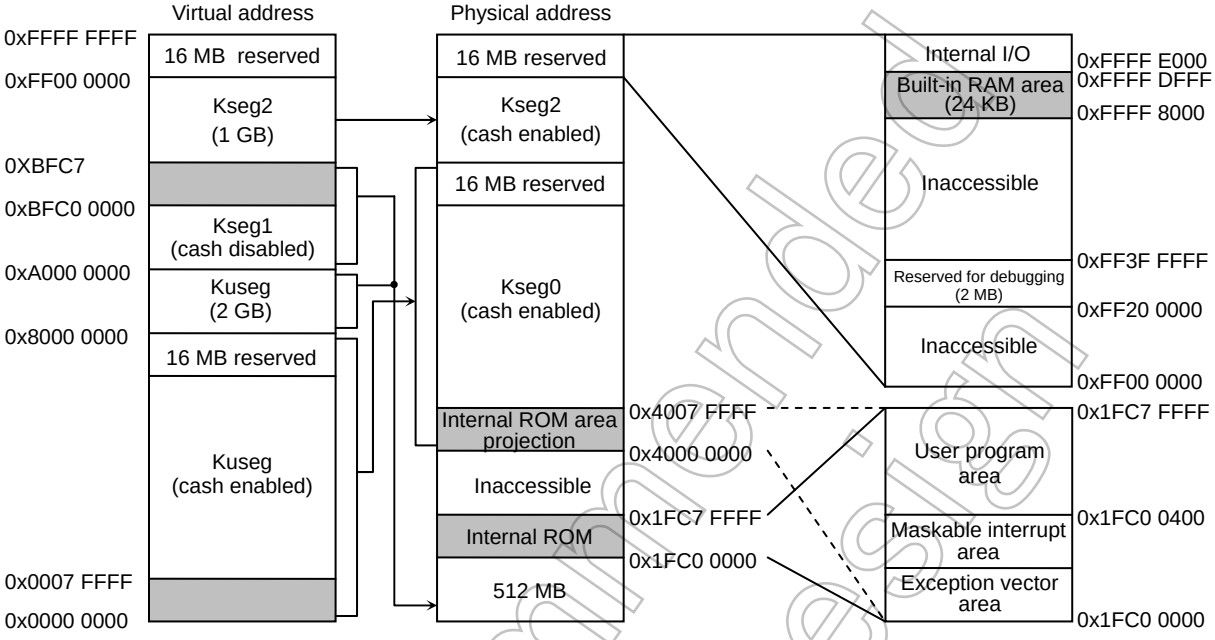


Fig. 4-1 Memory Map

Fig. 4-2 shows the memory map of the TMP19A43FZXBG/TMP19A43CZXBG.

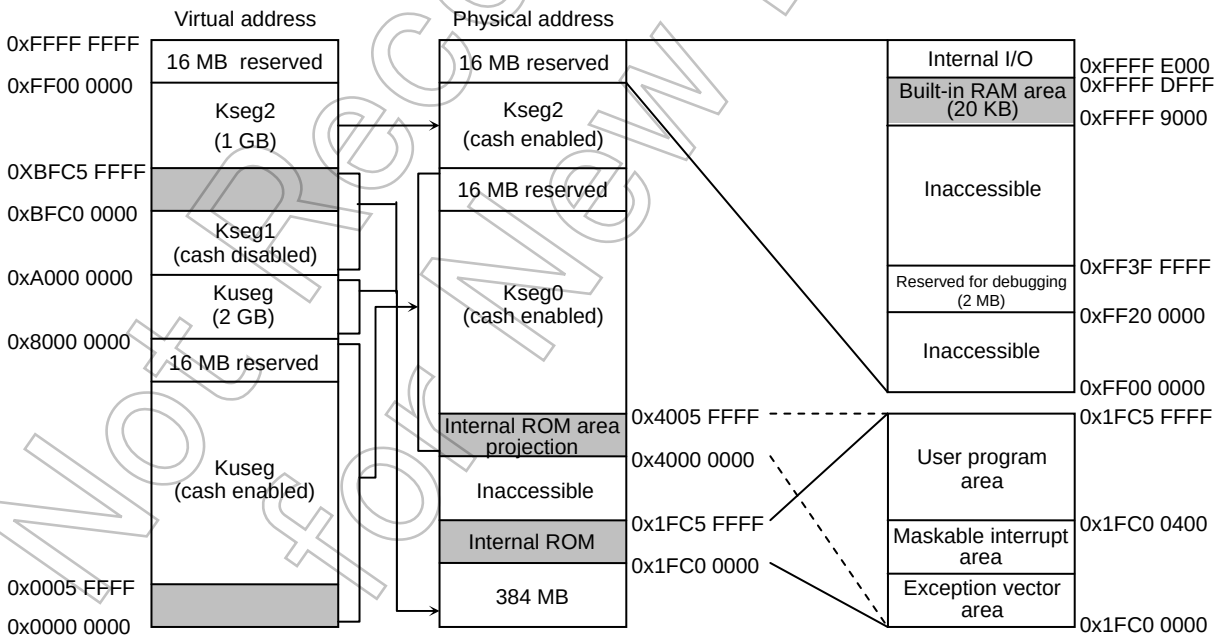


Fig. 4-2 Memory Map

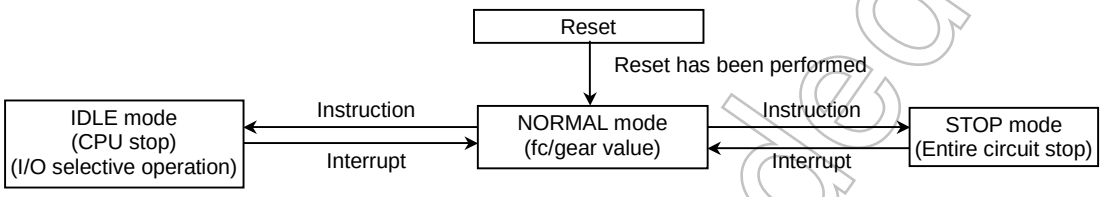
(Note 1) The internal ROM is mapped to:
0x1FC0_0000-0x1FC5_FFFF (384 KB)
0x1FC0_0000-0x1FC7_FFFF (512 KB)
The internal RAM is mapped to:
0xFFFF_9000-0xFFFF_DFFF (20 KB)
0xFFFF_8000-0xFFFF_DFFF (24 KB)

(Note 2) For the TMP19A43, a physical space of only 16 MB is available as external address space to be accessed. It is possible to place this 16-MB physical address space in a chip select area of your choice inside the 3.5-GB physical address space of the CPU.
Access to internal memory, internal I/O space and reserved areas is given priority over access to the external address space. Therefore, access to the external address space is denied if any of the internal memory, internal I/O space or reserved areas are being accessed.

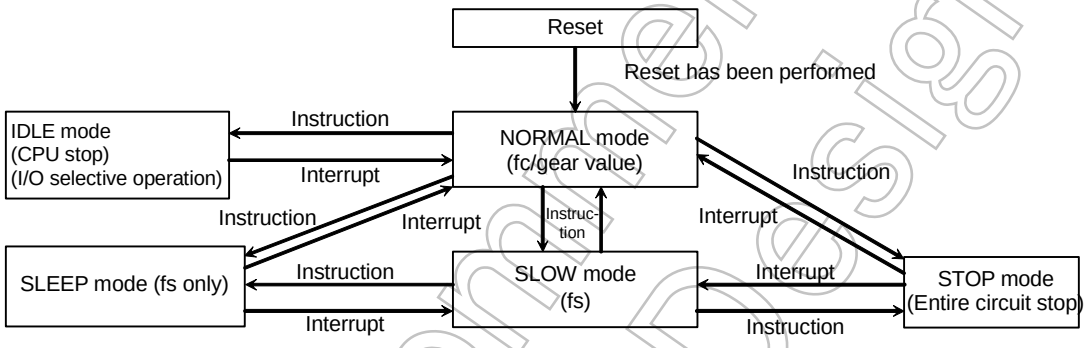
(Note 3) Do not place an instruction in the last four words of a physical area, specifically the last four words of an area where memory is mounted for external ROM extension (this varies depending on the system of the user).
Internal ROM: 0x1FC5_FFF0-0x1FC5_FFFF (384 KB)
Internal ROM: 0x1FC7_FFF0-0x1FC7_FFFF (512 KB)

5. Clock/Standby Control

The system operation modes contain the standby modes in which the processor core operations are stopped to reduce power dissipation. 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

Fig. 5-1 State Transition Diagram of Each Operation Mode

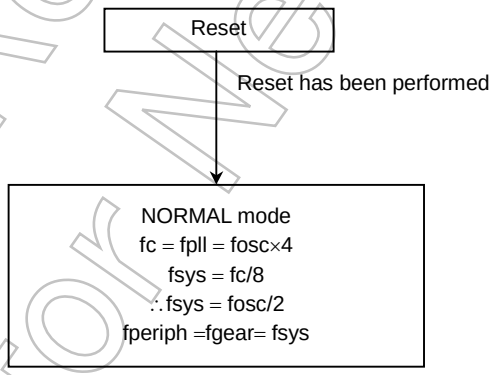


Fig. 5-2 Default State of the System Clock

fosc:	**Clock frequency to be input via the X1 and X2 pins**
fpll:	**Clock frequency multiplied (quadrupled) 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** **The CPU, ROM, RAM, DMAC, INTC and HSIO all operate according to this clock. The internal peripheral I/O operates according to the fsys/2 clock.**
fperiph:	**Clock frequency selected by SYSCR1<FPSEL> (Clock to be input to the peripheral I/O prescaler)**

5.1 Clock System Block Diagram

5.1.1 Main System Clock

- Allows for oscillator connection or external clock input.
- Clock gear (3/4, 1/2, 1/4, 1/8)
  (Default is 1/8.)
- Input frequency (high frequency)

Input frequency range	Maximum operating frequency	Lowest operating frequency
8 to 10 (MHz)*	40 MHz	4 MHz

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

- Input frequency (low frequency)

Input frequency range	Maximum operating frequency	Lowest operating frequency
30 KHz to 34 KHz	34 kHz	15 kHz

- When the low-speed clock gear 1/2 is used: 15 KHz (MIN)

(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.

5.1.2 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 and fperiph/16

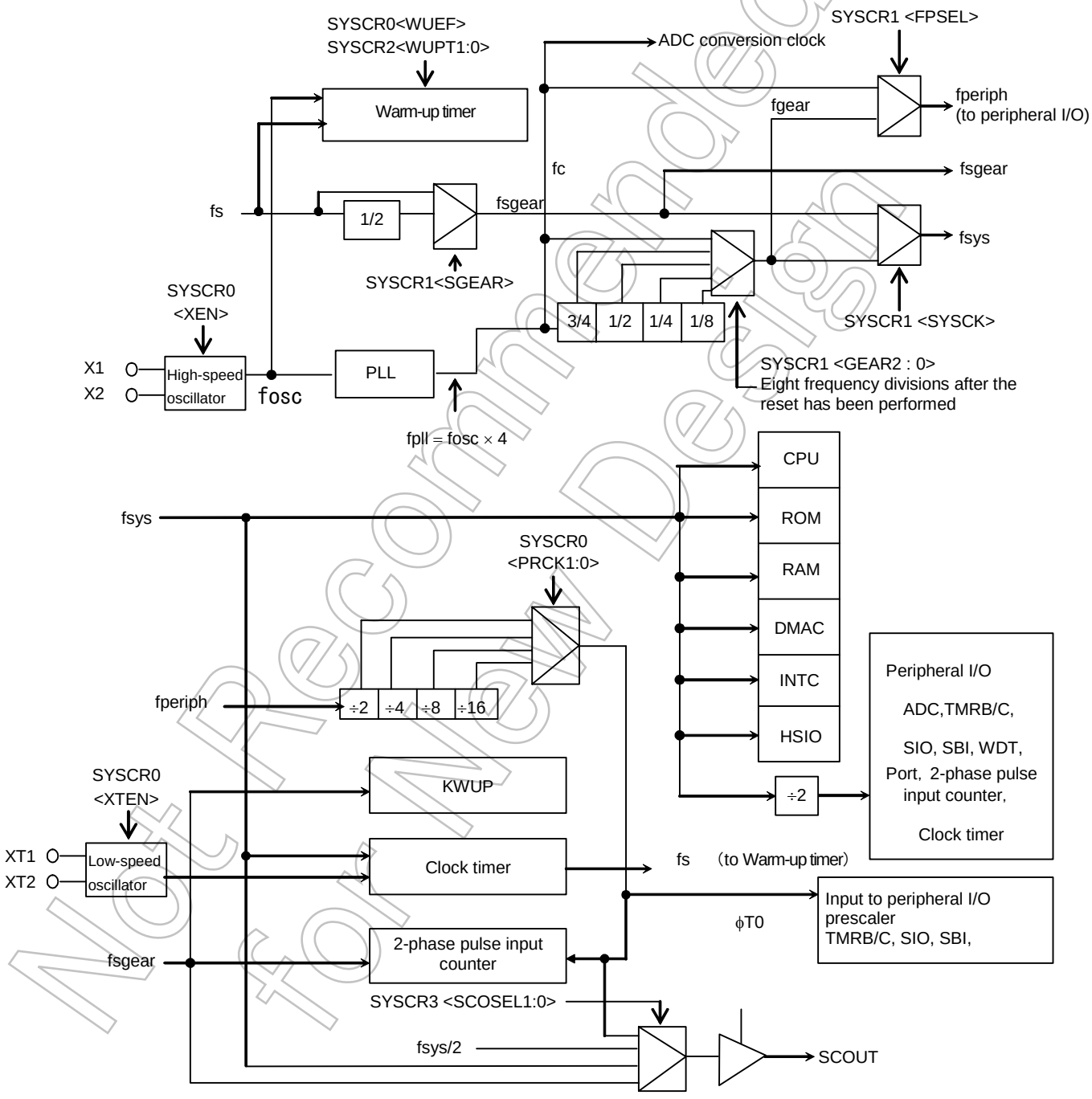


Fig. 5-3 Clock and Standby Related Block Diagram

5.2 CG Registers

5.2.1 System Control Registers

SYSCR0 (0xFFFF_EE00) (0xFFFF_EE03) LITTLE BIG

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

SYSCR1 (0xFFFF_EE01) (0xFFFF_EE02) LITTLE BIG

	7	6	5	4	3	2	1	0
Bit symbol		SYSCKFLG	SYSCK	FPSEL	SGEAR	GEAR2	GEAR1	GEAR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	1	1	1
Function	This can be read as "0."	System clock status flag 0: High speed (fc) 1: Low speed (fs)	Select system clock 0: High speed (fgear) 1: Low speed (fs)	Select fperiph 0: fgear 1: fc	Select gear of low-speed clock 0: fs/1 1: fs/2	Select gear of high-speed clock (fc) 000: fc 001: reserved 010: fc3/4 011: reserved 100: fc/2 101: reserved 110: fc/4 111: fc/8		

SYSCR2 (0xFFFF_EE02) (0xFFFF_EE01) LITTLE BIG

	7	6	5	4	3	2	1	0
Bit symbol	DRVOSCH		WUPT1	WUPT0	STBY1	STBY0		DRVE
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	1	0	1	1	0	0
Function	High-speed oscillator current control 0: High capability 1: Low capability	This can be read as "0."	Select oscillator warm-up time 00: No WUP 01: 2^8 /Input frequency 10: 2^{14} /Input frequency 11: 2^{16} /Input frequency		Select standby mode 00: Reserved 01: STOP 10: SLEEP 11: IDLE		This can be read as "0."	1: Drive the pin even in the STOP mode.

SYSCR3 (0xFFFF_EE03) (0xFFFF_EE00) LITTLE BIG

	7	6	5	4	3	2	1	0
Bit symbol		SCOSEL1	SCOSEL0	ALESEL				
Read/Write	R	R/W	R/W	R/W	R			
After reset	0	0	1	1	0			
Function	This can be read as "0."	Select SCOUT output 00: fsgear 01: fsys/2 10: fsys 11: φT0		Set ALE output width 0: fsys×1 1: fsys×2	This can be read as "0."			

- **Don't switch the SYSCK and the GEAR<2:0> simultaneously.**
- **If the system enters the STOP mode with SYSCR2<DRVOSCH> set at 1 (low capability), the setting will change to 0 (high capability) after the STOP mode is released.**
- **SYSCK can be switched when both of XEN and XTEN are set to "1."**
- **Be sure to set the RXEN and the RXTEN to 1 (oscillation) for the oscillator selected at the SYSCK. If a wrong setting is made, the oscillator selected by the SYSCK will oscillate.**
- **The clock that has been selected with SYSCK oscillates without fail after making clear the STOP mode.**

5.3 System Clock Controller

By resetting the system clock controller, the controller status is initialized to <XEN>="1,"<XTEN>="0" and <GEAR2:0>="111," and the system clock fsys changes to fc/8. (fc=fosc (original oscillation frequency)×4, because the original oscillation is quadrupled by PLL.) For example, when a 10-MHz oscillator is connected to the X1 or X2 pin, fsys becomes 5 MHz (=10×4×1/8) after the reset.

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

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

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

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

Table 5-1 shows the warm-up time when switching occurs.

(Note 1) The time for warm-up is required even when an external clock (oscillator, etc.) is used and providing stable oscillation because the internal PLL is used even in this case.

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

Table 5-1 Warm-up Time

Warm-up time options SYSCR2<WUPT1 : 0>	High-speed clock (fosc)	Low-speed clock (fs)
01 (2^8/oscillation frequency)	25.6 (μs)	7.8 (ms)
10 (2^{14}/oscillation frequency)	1.638 (ms)	500 (ms)
11 (2^{16}/ oscillation frequency)	6.554 (ms)	2000 (ms)

These values are calculated under the following conditions:
fosc = 10 MHz,
fs = 32.768 kHz

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

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

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

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

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

5.3.2 System Clock Pin Output Function

The system clock, fsys, fsys/2 or fs, can be output from the P44/SCOUT pin. By setting the port 4 related registers, P4CR<P44C> to "1" and P4FC<P44F> to "1," the P44/SCOUT pin becomes the SCOUT output pin. The output clock is selected by setting the SYSCR3<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 fsgear clock.				
<SCOSEL1:0> = "01"	Output the fsys/2 clock.		Fixed to "0" or "1."		
<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.

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

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

● Reducing the driving capability of the high-speed oscillator

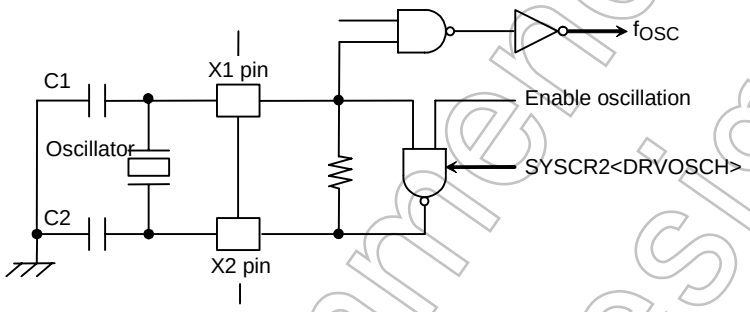


Fig. 5-4 Oscillator Driving Capability

5.3.4 Clock Frequency Division for Low-Speed System Clock

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

Set the clock frequency division during high-speed oscillation.

5.4 Prescaler Clock Controller

Each internal I/O (TMRB0-F, TMRC, SIO0-2 and SBI) has a prescaler for dividing a clock. The clock ϕT0 to be input to each prescaler is obtained by selecting the "fperiph" clock at the SYSCR1<FPSEL> and the SYSCR0<PRCK1:0> and then dividing the clock according to the setting of SYSCR0<PRCK1:0>. After the controller is reset, fperiph/16 is selected as ϕT0. For details, please refer to Fig. 5-5 System Clock Transition Diagram.

5.5 Clock Multiplication Circuit (PLL)

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

5.6 Standby Controller

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

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

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

IDLE: Only the CPU is stopped in this mode.

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

Table 5-3 shows a list of IDLE setting registers.

Table 5-3 Internal I/O setting registers for the IDLE mode

Internal I/O	IDLE mode setting register
TMRB0-F	TBxRUN<I2TBx>
TMRC	TCCR<I2TBT>
SIO0-3	SCxMOD1<I2Sx>
HSIO0-3	HSCxMOD1<I2Sx>
I2C/SIO(SBI)	SBIBR1<I2SBIx>
A/D converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>

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

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

SLEEP: Only the internal low-speed oscillator, the clock timer, the 2-phase pulse input counter and the dynamic pull-up circuit (KWUP) 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 1:0	HALT RP=0	DOZE RP=1
RESERVED	00	X	X
STOP	01	STOP	X
SLEEP	10	SLEEP	X
IDLE	11	HALT	DOZE

5.6.1 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	fs only	×	Clock timer, 2-phase pulse input counter and KWUP	×
	Stop	×	×	×	×

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

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

5.6.2 Block Operations in Each Mode

Table 5-5 Block Operating Status in Each Operation Mode

Block	NORMAL	SLOW	IDLE (Doze)	IDLE (Halt)	SLEEP	STOP
TX19A processor core	○	○	×	×	×	×
DMAC	○	○	○	×	×	×
INTC	○	○	○	○	×	×
External bus I/F	○	○	○	×	×	×
IO port		○		×	×	×
ADC	○	×	ON/OFF selectable for each module		×	×
DAC	○	×			×	×
SIO	○	×			×	×
HSIO	○	×			×	×
I2C	○	×			×	×
TMRB	○	×			×	×
TMRC	○	×			×	×
WDT	○	×			×	×
2-phase pulse input counter	○	○			○ (fs only)	×
Dynamic pull-up (KWUP)	○	○	○	○	○	○ (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 SYSCR1<XEN>.

5.6.3 Releasing the Standby State

The standby state can be released by an interrupt request when the interrupt level is higher than the interrupt mask level, or by the reset. The standby release source that can be used is determined by a combination of the standby mode and the state of the interrupt mask register <IM15:8> assigned to the status register in the system control coprocessor (CPO) of the TX19A processor core. Details are shown in Table 5-6.

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

- Release by the reset

  Any standby state can be released by the reset.
  Note that releasing of the STOP mode requires sufficient reset time to allow the oscillator operation to become stable. (Table 5-1 Warm-up Time).

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

Standby Release Sources and Standby Release Operations
(Interrupt level)>(Interrupt mask)

Table 5-6

Interrupt accepting state			Interrupt enabled EI= "1"			Interrupt disabled EI= "0"		
Standby mode			IDLE (programmable)	SLEEP	STOP	IDLE (programmable)	SLEEP	STOP
Standby release source	Interrupt	INTWDT	◎	×	×	◎	–	–
		INT0-B	◎	◎	◎ (Note 1)	○	○	○ (Note 1)
		KWUP00-31	◎	◎	◎ (Note 1)	○	○	○ (Note 1)
		INTRTC	◎	◎	×	○	○	×
		INTTB2-3 (Note 2)	◎	◎	×	○	○	×
		INTTB0-F	◎	×	×	○	×	×
		INTRX0-2,INTTX0-2 HINTRX00-2,HINTTX0-2	◎	×	×	○	×	×
		INTS0	◎	×	×	○	×	×
		INTAD/INTADHP/INTADM	◎	×	×	○	×	×

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

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

× : Cannot be used for releasing the standby mode

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

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

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

(Note 3) 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 4) 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.

5.6.4 STOP Mode

In the STOP mode, all the internal circuits, including the internal oscillators, are brought to a stop. The pin states in the STOP mode vary depending on the setting of the SYSCR2<DRVE>. Table 5.8 shows the pin states in the STOP mode. When the STOP mode is released, the system clock output is started after the elapse of warm-up time at the warm-up counter to allow the internal oscillators to stabilize. After the STOP mode is released, the system returns to the operation mode that was active immediately before the STOP mode (NORMAL or SLOW), and starts the operation.

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

(Note) To shift from the NORMAL mode to the STOP mode on the TMP19A43, do not set the SYSCR2<WUPT1:0> to "00" or "01" for the warm-up time setting. The internal system recovery time cannot be satisfied when the system recovers from the STOP mode.

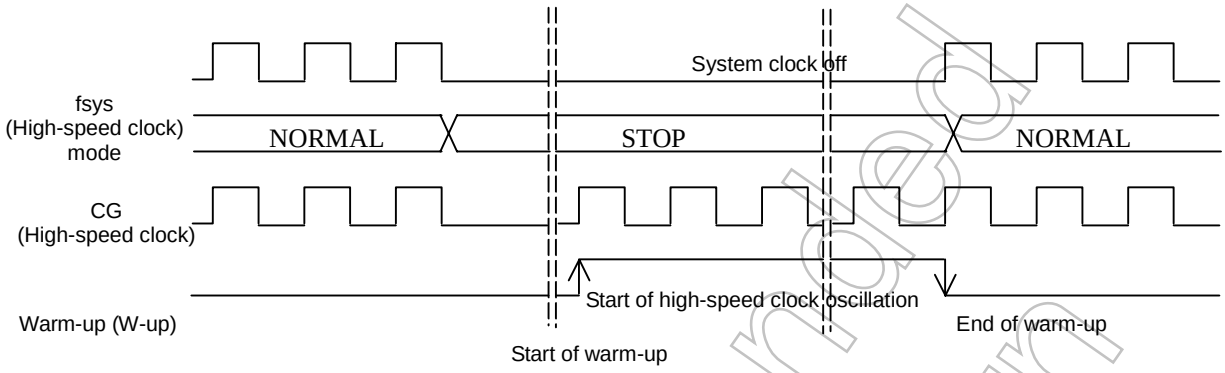
Table 5-7 Warm-up Settings for Transitions of Operation Modes

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

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

5.6.5 Recovery from the STOP or SLEEP Mode

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

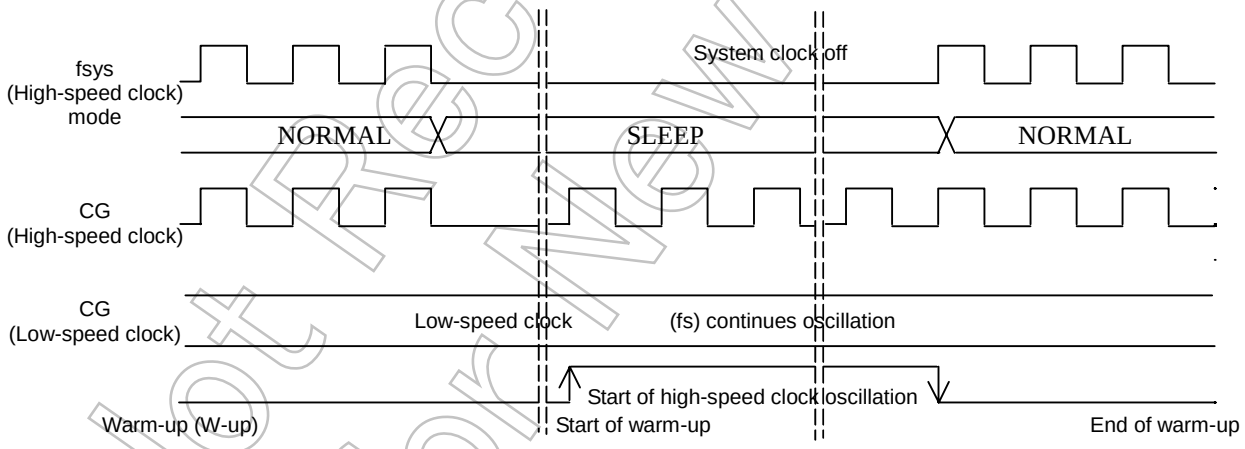


when @fosc=10 MHz

Selection of warm-up time SYSCR2<WUPT1:0>	Warm-up time (fosc)
01 (2^8/fosc)	Setting disabled
10 (2^{14}/fosc)	1.638 ms
11 (2^{16}/fosc)	6.554 ms

(Note) The internal system recovery time cannot be satisfied. Do not set <WUPT1:0> to "01."

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

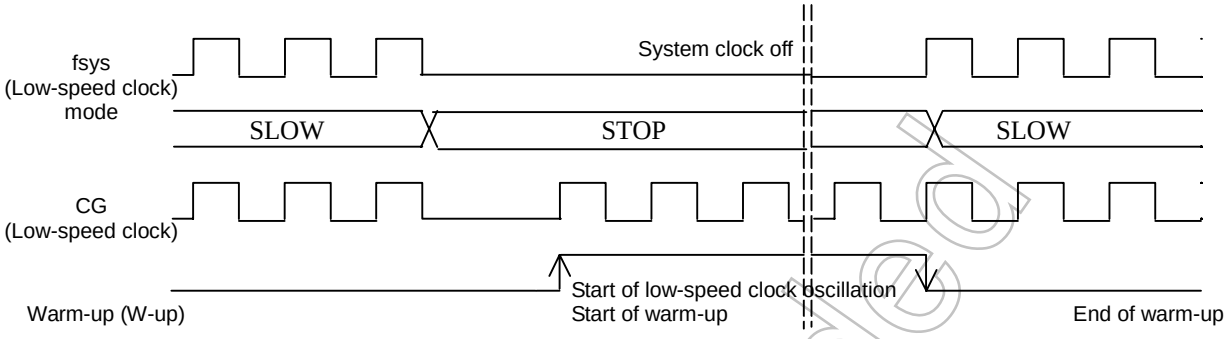


when @fosc=10 MHz

Selection of warm-up time SYSCR2<WUPT1:0>	Warm-up time (fosc)
01 (2^8/fosc)	Setting disabled
10 (2^{14}/fosc)	1.638 ms
11 (2^{16}/fosc)	6.554 ms

(Note) The internal system recovery time cannot be satisfied. Do not set <WUPT1:0> to "01."

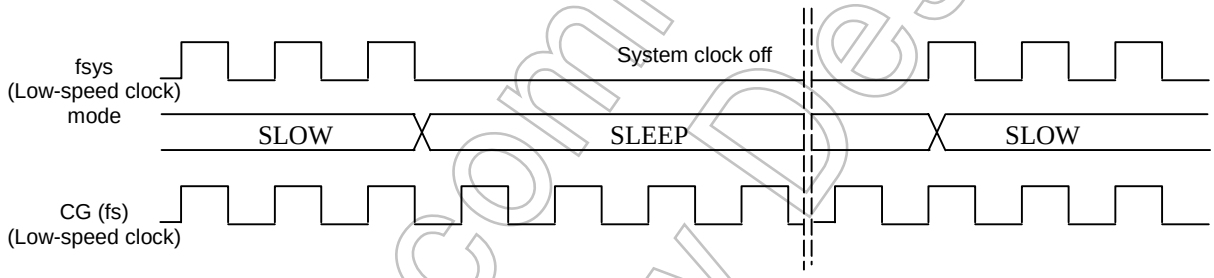
3. Transition of operation modes: SLOW → STOP → SLOW



when @fs=32.768 kHz

Selection of warm-up time SYSCR2<WUPT1:0>	Warm-up time (fs)
11 (2^{16}/fs)	2000 ms

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



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

Table 5-8 Pin States in the STOP Mode in Each State of SYSCR2<DRVE> (1/2)

Pin name	Input/output	<DRVE>=0	<DRVE>=1
P00-P07	Input mode Output mode AD0-AD7, D0-D7	— —	— Output
P10-P17	Input mode Output mode, A8-A15 AD8-AD15, D8-D15 A8-A7 (Output mode)	— — Output	— Output Output
P20-P27	Input mode Output mode, A0-A7/A16-A23 (Output mode)	— — Output	Input Output Output
P30 (/RD), P31 (/WR)	Output pin /RD,/WR(Output mode)	— Output	Output Output
P32(/HWR), P35(/BUSAK, P36(R/W)	Input mode Output mode /HWR,/BUSAK, R/W(Output mode)	— — Output	Input Output Output
P37 (ALE)	Input mode Output mode ALE (Output mode)	— — "L" level output	Input Output "L" level output
P40-P43	Input mode Output mode /CS0-/CS2 (Output mode) KEY24-KEY27 (Input mode)	— — Output Input	Input Output Output Input
P44 –P47	Input mode Output mode	— —	Input Output
P50-P55	Input mode Output mode, A0-A5 (Output mode)	— — Output	Input Output
P56, P57	Input mode Output mode A6, A7 (Output mode) KEY28, KEY29 (Input mode)	— — Output Input	Input Output Input
P61, P64	Input mode Output mode A9, A12 (Output mode) INTA, INTB (Input mode)	— — Output Input	— Output Input
P60, P62, P63, P65-P67	Input mode Output mode, A8, A10, A11, A13-A15 (Output mode)	— — Output	— Output
P70-73	Input mode	—	—
P74-77	Input mode KEY00-KEY03 (Input mode)	Input Input	— Input
P80-P83	Input mode KEY04-KEY07 (Input mode)	Input Input	— Input
P84-P87	Input mode INT6-INT9 (Input mode)	Input Input	— Input
P9	Input mode Output mode	— —	Input Output
PA0-PA5	Input mode Output mode INT0-INT5 (Input mode)	— — Input	Input Output Input
PA6, PA7	Input mode Output mode	— —	Input Output
PB0-PB7	Input mode Output mode	— —	Input Output
PC0	Input mode Output mode KEY30 (Input mode)	— — Input	Input Output Input
PC1-PC7	Input mode Output mode	— —	Input Output

Pin name	Input/Output	<DRVE>=0	<DRVE>=1
PD0-PD5	Input mode Output mode	— —	Input Output
PD6	Input mode Output mode KEY30 (Input mode)	— — Input	Input Output Input
PE0-PE7	Input mode Output mode KEY08-KEY15 (Input mode)	— — Input	Input Output Input
PF0-PF7	Input mode Output mode KEY16-KEY23 (Input mode)	— — Input	Input Output Input
PG, PH	Input mode Output mode	— —	Input Output
$\overline{\text{RESET}}$	Input pin	Input	Input
TEST	Input pin	Input	Input
X1	Input pin	—	—
X2	Output pin	"H" level output	"H" level output
XT1	Input pin	—	—
XT2	Output pin	"H" level output	"H" level output

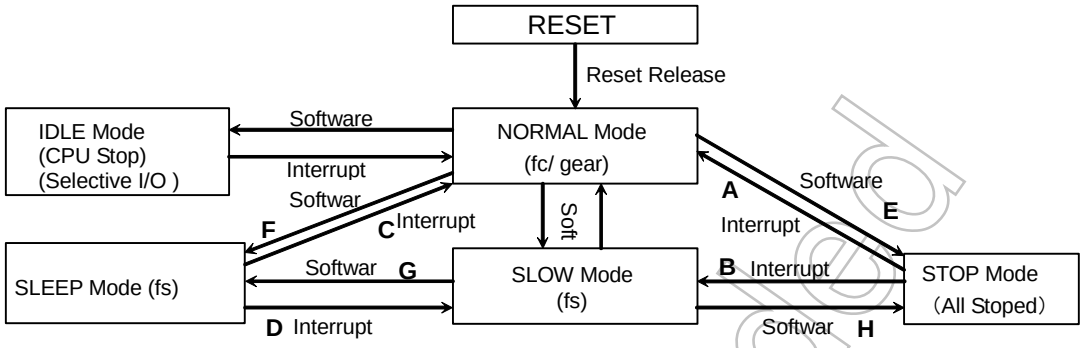
— : Indicates that the input is disabled for the input mode and the input pin and the impedance becomes high for the output mode and the output pin. Note that the input is enabled when the port function register (PxFC) is "1" and the port control register (PxCR) is "0."

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

Output : The pin is in the output state.

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

(note) 19A43 requires a recovery time from Warming up state as following



State Transition Diagram

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

WUP Trigger	State Transition		Minimum required Operation time before WAIT instruction done (sec)
Software release	E	STOP	16 / fs in NOMAL mode
	F	SLEEP	16 x fs in NOMAL mode

6. Exceptions/Interrupts

6.1 Overview

The TMP19A43 device is configured with the following 50 maskable interrupt factors and 15 exceptions including NMI. In this section, general exceptions and debug exceptions are described simply as "exceptions" and interrupts are described as "interrupts."

- 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 (data access)
  - Co-processor unusable exception
  - Reserved instruction exception
  - Integer overflow exception
  - Trap exception
  - System call exception
  - Breakpoint exception
- Debug exception
  - Single step exception
  - Debug breakpoint exception
- Interrupts
  - Maskable software interrupts (2 factors)
  - Maskable hardware interrupts: 46 internal factors and 48 external factors (INT0 - F, KEY00 - 31)

The TMP19A43 device not only processes interrupt requests from internal hardware peripherals and external inputs but also forces transition to exception handling processes as a means of notifying any error status generated in normal instruction sequences.

By using the register bank called "shadow register set" newly implemented in the TX19A processor core, it is now unnecessary to save the general purpose register (GPR) contents elsewhere upon interrupt response thus leading to very fast interrupt response.

The device is capable of handling multiple interrupts according to seven programmable interrupt levels (priority orders). Also, it can mask interrupt requests with a priority level the same or lower than a specified mask level.

6.2 Exception Vector

The starting address of an exception handler is defined to be "exception vector address." The exception vector address for a reset exception and non-maskable interrupts is 0xBFC0_0000. The exception vector address for a debug exception can be either 0xBFC0_0480 (EJTAG ProbEn = 0) or 0xFF20_0200 (EJTAG ProbEn = 1) depending on the internal signal <ProbeEn>. For other exceptions, the corresponding exception vector addresses are determined depending on the values of Status <BEV> and Cause <IV> of the system control coprocessor register (CP0).

Table 6.21 Exception Vector Table (Virtual Address)

Exception	BEV=0	BEV=1
Reset, NMI	0xBFC0_0000	0xBFC0_0000
Debug exceptions (En=0)	0xBFC0_0480	0xBFC0_0480
Debug exceptions (En=1)	0xFF20_0200	0xFF20_0200
Interrupts (IV=0)	0x8000_0180	0xBFC0_0380
Interrupts (IV=1)	0x8000_0200	0xBFC0_0400
Others general exceptions	0x8000_0180	0xBFC0_0380

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

6.3 Reset Exception

A reset exception is generated by either setting the external reset pin to "L" or counting the WDT beyond a "reset" count. When a reset exception is generated, peripheral hardware registers and the CP0 register are initialized and it jumps to the exception vector address 0xBFC0_0000. The PC value of reset exception generation will be stored in ErrorEPC of the CP0 register.

Since a reset exception causes to set the status bit <ERL> of the CP0 register to "1" disabling interrupt requests, the Status <ERL> bit must be cleared to "0" in a startup routine (reset exception handler) or by any other means if interrupts are to be used.

Refer to the section "Exception Handling, Reset Exception" of the separate volume "TX19A Core Architecture" for detailed operation upon generation of reset exception.

6.4 Non-maskable Interrupt (NMI)

An NMI interrupt is generated when WDT is counted to an NMI set count or when a bus error area is accessed by store access including DMA transfer. When an NMI interrupt is generated, the status bits <ERL> and <NMI> of the CP0 register are set to "1" and it jumps to the exception vector address 0xBFC0_0000.

The PC value of NMI generation will be stored in ErrorEPC of the CP0 register. Note that any NMI due to a bus error upon a store instruction causes an exception that is not synchronized with instruction sequence. Therefore, the PC value of an instruction being executed at the time of error generation will be stored instead of the PC value for the instruction that actually caused the error. Upon NMI generation, when the shadow register set is enabled, SSCR <CSS> will be overwritten by the value of SSCR <PSS> but the register bank will not be switched because the value of SSCR <CSS> is not updated. The reason why only the SSCR <PSS> value is updated is because it is necessary to prevent the register bank from being changed when SSCR <PSS> is overwritten by the value of SSCR <CSS> due to an ERET instruction executed upon returning from NMI.

The cause of NMI generation can be determined by NMIFLG <WDT> and <WBER> of CG. (Refer to the Section 6.10, NMI Flag Register Reference.) Refer to the section "Exception Handling, Non-Maskable Interruptions" of the separate volume "TX19A Core Architecture" for detailed operation upon generation of NMI.

6.5 General Exceptions (Other than Reset Exception and NMI)

A general exception will be generated when a specific instruction such as SYSCALL is executed or when any abnormalities such as an illegal instruction fetch is detected. When a general exception is generated and if Status <BEV> of the CP0 register is "1," it jumps to the exception vector address 0xBFC0_380. The cause of a general exception can be determined by Cause <ExCode> of the CP0 register.

The PC value at a general exception will be stored in EPC of the CP0 register. Note that any bus error exception (data access) is not synchronized with instruction sequence so the PC value of an instruction being executed at the time of error generation will be stored instead of the PC value for the instruction that actually caused the error. Upon a general exception, when the shadow register set is enabled, SSCR <CSS> will be overwritten by the value of SSCR <PSS> but the register bank will not be switched because the value of SSCR <CSS> is not updated. The reason why only the SSCR <PSS> value is updated is because it is necessary to prevent the register bank from being changed when SSCR <PSS> is overwritten by the value of SSCR <CSS> due to an ERET instruction executed upon returning from the exception.

Any illegal address that caused an address error exception (instruction fetch or load/store) or bus error (instruction fetch/data access) will be stored in BadVAddr of the CP0 register.

Refer to the corresponding sections of "Exception Handling" of the separate volume "TX19A Core Architecture" for detailed operation upon generation of general exceptions.

(Note 1) Address error exceptions (load/store) will not be generated in DMS transfer operations. In DMA transfer, address errors can be detected as configuration errors (CSRx <Conf> of DMAC).

(Note 2) Bus errors (data access) may be generated either by load instructions or by load accesses of DMA transfer operations.

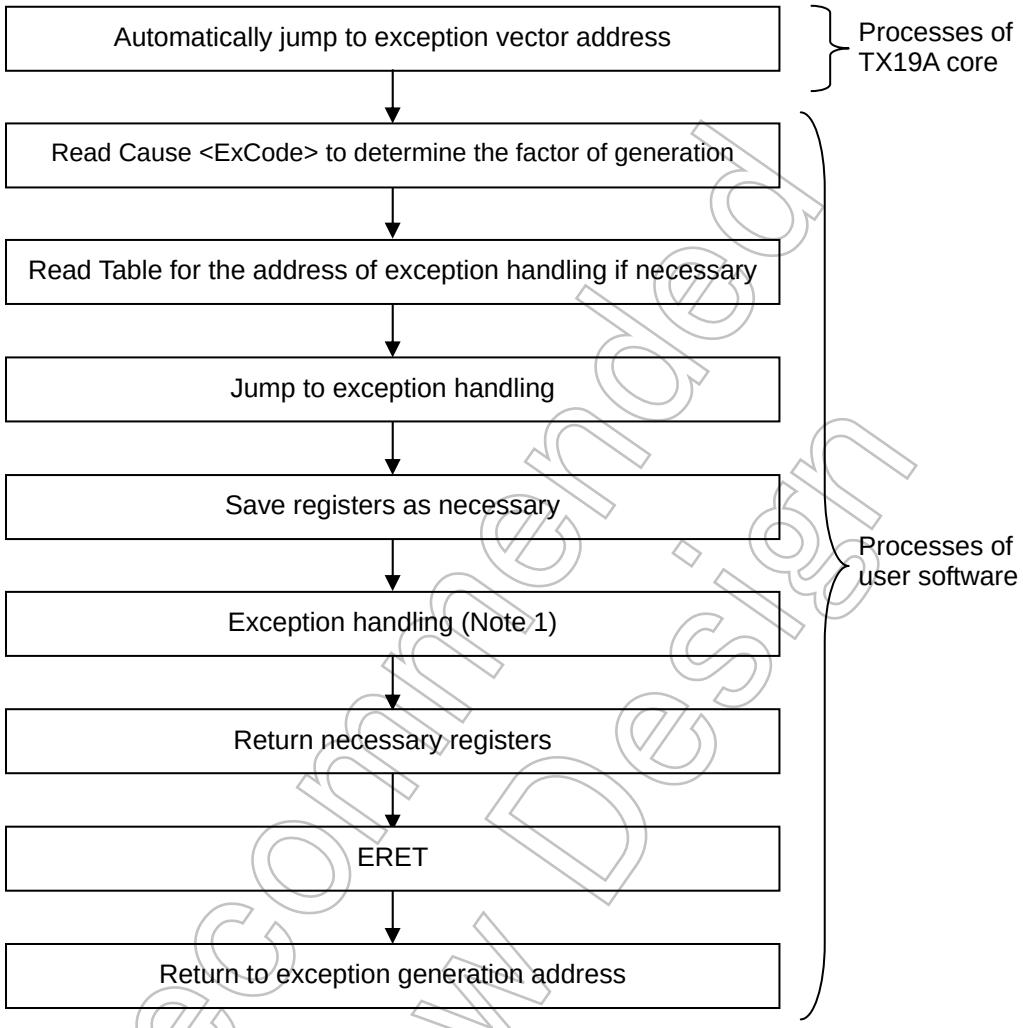


Fig. 6-1 Example Sequence of General Exceptions (Other than Reset Exception and NMI)

(Note 1) Since general exceptions (other than reset exception/NMI and excluding trap exceptions, system call exceptions, and breakpoint exceptions) indicate some sort of abnormal conditions, the system tends to be reset.

(Note 2) Upon generation of a general exception other than reset exception/NMI, excluding bus error exceptions (instruction fetch/data access), the PC that caused the exception will be stored in EPC. Therefore, returning the system by simply using ERET may cause the same exception again.

6.6 Debug Exceptions

Single step exceptions and debug breakpoint exceptions are the types of debug exceptions. These types of exceptions are seldom used in user programs.

Also note that enabling the shadow register set will not be effective in debug exceptions.

Refer to the section "Exception Handling, Debug Exception" of the separate volume "TX19A Core Architecture" for detailed operation upon generation of debug exceptions.

6.7 Maskable Software Interrupts

Two-factor maskable software interrupts (hereinafter referred to simply as "software interrupts") can be generated by individually setting "1" to the Cause <IP [1:0]> bits of the CP0 register.

Software interrupts can be accepted in no less than three clocks after setting values to the Cause <IP [1:0]> bits of the CP0 register.

In order for a software interrupt request to be accepted, it is necessary regarding the CP0 register that its Status <IE> is set to "1" and Status <ERL/EXL> is cleared to "0" while Status <IM [1:0]> is "1." Also, software interrupts can be individually masked by setting Status <IM [1:0]> of the CP0 register to "0." If software and hardware interrupts coincide, the hardware interrupt overrides the software interrupt.

Upon software interrupts, when the shadow register set is enabled, SSCR <CSS> will be overwritten by the value of SSCR <PSS> but the register bank will not be switched because the value of SSCR <CSS> is not updated. The reason why only the SSCR <PSS> value is updated is because it is necessary to prevent the register bank from being changed when SSCR <PSS> is overwritten by the value of SSCR <CSS> due to an ERET instruction executed upon returning from the software interrupt. Software interrupts are processed in a process flow such as shown in Fig. 6-2.

(Note 1) Please read out the data in IVR after a software interrupt is generated. To read out the data is a trigger to notify the core of a hardware interrupt.

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

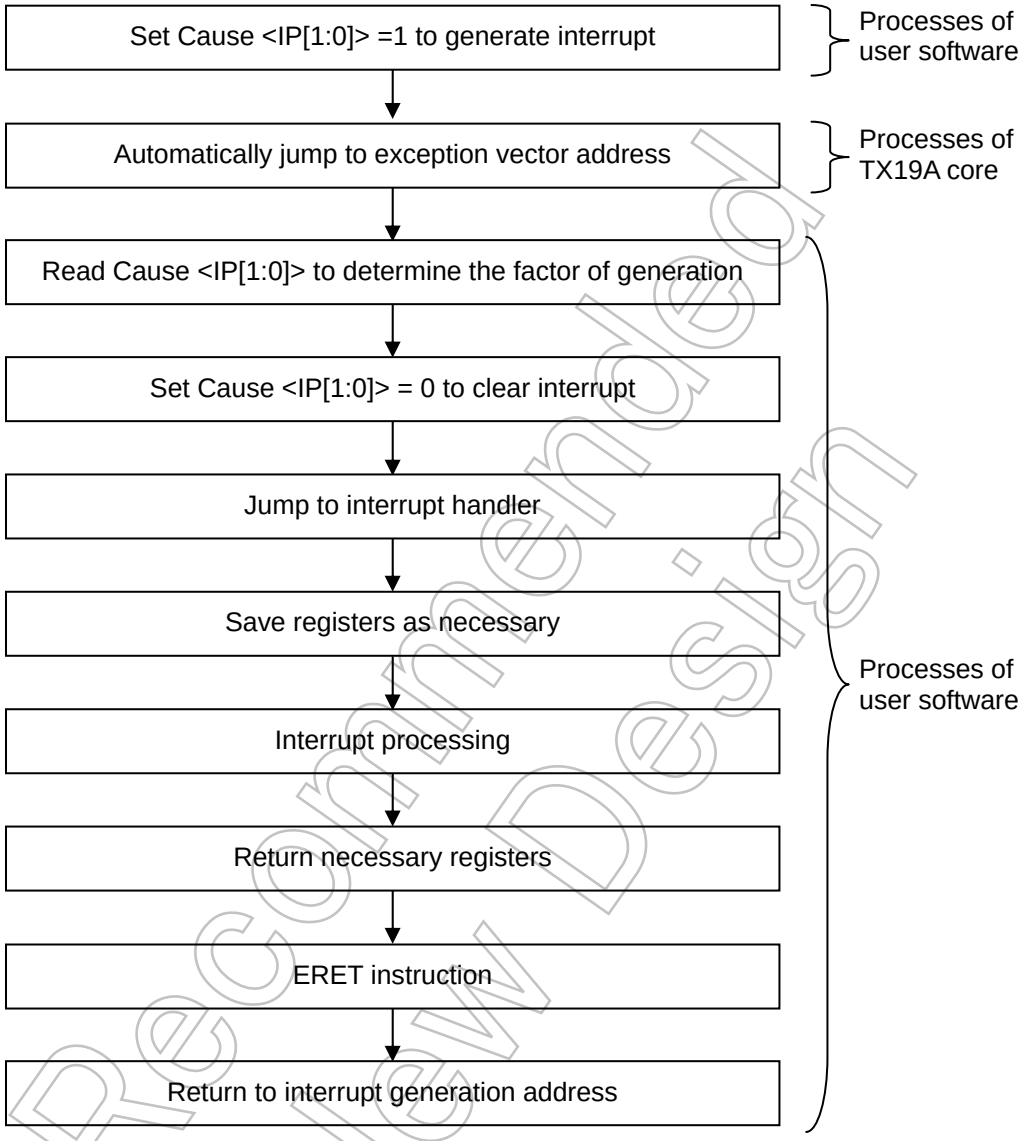


Fig. 6-2 Example of Software Interrupt Operation

(Note 1) A software interrupt is accepted in no less than three clocks after the instruction that enabled the interrupt and the PC at the time of acceptance is stored in EPC.

6.8 Maskable Hardware Interrupts

6.8.1 Features

The maskable hardware interrupts (hereinafter referred to as "hardware interrupts") are 64 factor interrupt requests for which the interrupt controller (INTC) can individually assign an interrupt level out of seven interrupt (priority) levels.

In order for a hardware interrupt request to be accepted, it is necessary regarding the CP0 register that its Status <IE> is set to "1" and Status <ERL/EXL> is cleared to "0" while Status <IM [4:2]> is set to "1."

If more than one interrupts are generated at the same time, the hardware interrupts are accepted in accordance with the priority order of the interrupt levels. If more than one interrupts of a same interrupt level are generated at the same time, these interrupts are accepted in the order of the interrupt number as listed in Table 6.8.1.

When an interrupt request is accepted, the Status <EXL> bit of the CP0 register is set to "1," further interrupts are disabled, and ILEV<CMASK> of INTC is automatically updated to the interrupt level set for the interrupt request. Note that Status <IE> of the CP0 register remains set to "1" in interrupt response operations.

In processing hardware interrupts, each interrupt level is associated with a register bank called a "shadow register set." When an interrupt request is accepted, the register bank is switched to the register bank of which number is the same as with the corresponding interrupt level. Through this mechanism, it is unnecessary for the user program to save the general purpose register (GPR) contents elsewhere upon interrupt response thus ensuring fast interrupt response. CP0 register SSCR<SSD>="0")

For accepting multiple interrupts, Status <EXL> of the CP0 register is cleared to "0" to permit further interrupts. In this, because ILEV <CMASK> of INTC has been updated to the interrupt level set for the interrupt request already accepted, only further interrupts of which level is higher than the present interrupt level can be accepted. Refer to Section 6.8.7 "Example of Multiple Interrupt Setting" for more details of multiple interrupts.

Also, by appropriately setting the ILEV <CMASK> register of INTC, you can mask interrupt requests of which interrupt level is lower than a programmed mask level.

Any interrupt request can be used as a trigger to start a DMA transfer sequence.

While detailed operation of hardware interrupts is provided below, please also refer to the section "Exception Handling, Maskable Interrupts (Interrupts)" of the separate volume "TX19A Core Architecture" for more details.

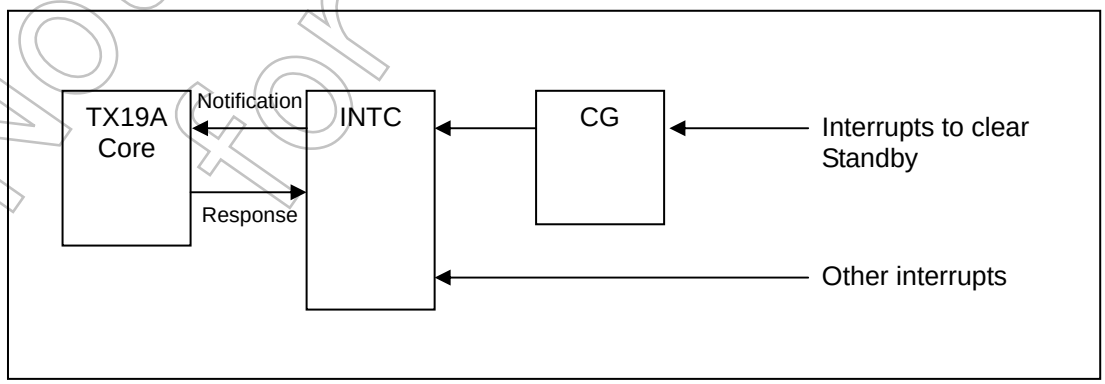


Fig. 6-3 Interrupt Notification Diagram

Table 6.8.1 List of Hardware Interrupt Factors

Interrupt Number	IVR[7:0]	Interrupt Factor	Interrupt Control Register	Address
0	0x000	Software set	IMC0	0xFFFF_E000
1	0x004	INT0 pin		
2	0x008	INT1 pin		
3	0x00C	INT2 pin		
4	0x010	INT3 pin	IMC1	0xFFFF_E004
5	0x014	INT4 pin		
6	0x018	INT5 pin		
7	0x01C	INT6 pin		
8	0x020	INT7 pin	IMC2	0xFFFF_E008
9	0x024	INT8 pin		
10	0x028	INT9 pin		
11	0x02C	INTA pin		
12	0x030	INTB pin	IMC3	0xFFFF_E00C
13	0x034	INTC pin		
14	0x038	INTD pin		
15	0x03C	INTE pin		
16	0x040	INTF pin	IMC4	0xFFFF_E010
17	0x044	KWUP		
18	0x048	INTRX0 : Serial receive (channel.0)		
19	0x04C	INTTX0 : Serial transmit (channel.0)		
20	0x050	INTRX1 : Serial receive (channel.1)	IMC5	0xFFFF_E014
21	0x054	INTTX1 : Serial transmit (channel.1)		
22	0x058	INTRX2 : Serial receive (channel.2)		
23	0x05C	INTTX2 : Serial transmit (channel.2)		
24	0x060	HINTRX0 : High speed serial receive (Hchannel.0)	IMC6	0xFFFF_E018
25	0x064	HINTTX0 : High speed serial transmit (Hchannel.0)		
26	0x068	HINTRX1 : High speed serial receive (Hchannel.1)		
27	0x06C	HINTTX1 : High speed serial transmit (Hchannel.1)		
28	0x070	HINTRX2 : High speed serial receive (Hchannel.2)	IMC7	0xFFFF_E01C
29	0x074	HINTTX2 : High speed serial transmit (Hchannel.2)		
30	0x078	INTS0 : Serial bus interface 0		
31	0x07C	INTADHP : Highest priority ADC complete interrupt		
32	0x080	INTADM : ADC monitor function interrupt	IMC8	0xFFFF_E020
33	0x084	INTTB0 : 16-bit timer 0		
34	0x088	INTTB1 : 16-bit timer 1		
35	0x08C	INTTB2 : 16-bit timer 2		
36	0x090	INTTB3 : 16-bit timer 3	IMC9	0xFFFF_E024
37	0x094	INTTB4 : 16-bit timer 4		
38	0x098	INTTB5 : 16-bit timer 5		
39	0x09C	INTTB6 : 16-bit timer 6		
40	0x0A0	INTTB7 : 16-bit timer 7	IMCA	0xFFFF_E028
41	0x0A4	INTTB8 : 16-bit timer 8		
42	0x0A8	INTTB9 : 16-bit timer 9		
43	0x0AC	INTTBA : 16-bit timer A		
44	0x0B0	INTTBB : 16-bit timer B	IMCB	0xFFFF_E02C
45	0x0B4	INTTBC : 16-bit timer C		
46	0x0B8	INTTBD : 16-bit timer D		
47	0x0BC	INTTBE : 16-bit timer E		
48	0x0C0	INTTBF : 16-bit timer F	IMCC	0xFFFF_E030
49	0x0C4	INTCAPG0 : Input capture group 0		
50	0x0C8	Reserved		
51	0x0CC	INTCMP0 : Compare interrupt 0		
52	0x0D0	INTCMP1 : Compare interrupt 1	IMCD	0xFFFF_E034
53	0x0D4	INTCMP2 : Compare interrupt 2		
54	0x0D8	INTCMP3 : Compare interrupt 3		
55	0x0DC	INTCMP4 : Compare interrupt 4		
56	0x0E0	INTCMP5 : Compare interrupt 5	IMCE	0xFFFF_E038
57	0x0E4	INTCMP6 : Compare interrupt 6		
58	0x0E8	INTCMP7 : Compare interrupt 7		
59	0x0EC	INTTBT : Overflow interrupt		
60	0x0F0	INTRTC : Clock timer interrupt	IMCF	0xFFFF_E03C
61	0x0F4	INTAD : ADC completed		
62	0x0F8	INTDMA0 : Completion of DMA transfer (channel.0)		
63	0x0FC	INTDMA1 : Completion of DMA transfer (channel.1)		

(Note 1) While IMCxx is a 32 bit register, 8 bit/16 bit access is also accepted.

(Note 2) Each factor can clear the IDLE mode.

Table 6.8.2 Interrupt Factors to Cancel Stop/Sleep Modes

Number	Interrupt Factor	Note
0	INT0	External interrupt 0
1	INT1	External interrupt 1
2	INT2	External interrupt 2
3	INT3	External interrupt 3
4	INT4	External interrupt 4
5	INT5	External interrupt 5
6	INT6	External interrupt 6
7	INT7	External interrupt 7
8	INT8	External interrupt 8
9	INT9	External interrupt 9
10	INTA	External interrupt A
11	INTB	External interrupt B
12	KWUP	Key on wake up interrupt
13	INTRTC	Clock timer interrupt
14	INTTB2	Two-phase pulse input counter interrupt 2
15	INTTB3	Two-phase pulse input counter interrupt 3

* Number 0 to 12 interrupt factors can cancel Stop, Sleep, and Idle modes.
* Number 13 to 15 interrupt factors can cancel the Sleep mode.

6.8.2 Detecting Interrupt Requests

Each of interrupt factors has its own interrupt detection sequence as described in Table 6.8. Upon detection, an interrupt request is notified to INTC for priority arbitration and then notified to the TX19A processor core. Refer to Table 6.8 for the detection level available for each interrupt factor.

Table 6.8.3 Location of Interrupt Request Detection

Interrupt	Detected by	Interrupt Notification Route
(1) Interrupts from external pins INT0 - INTB	CG	PORT → CG (detection) → INTC (arbitration) → TX19A core
	INTC	PORT → INTC (detection/arbitration) → TX19A core
(2) Interrupts from external pins INTC - INTF	INTC	PORT → INTC (detection/arbitration) → TX19A core
(3) Key on wakeup interrupt KWUP00-31	CG	PORT → CG (detection) → INTC (arbitration) → TX19A core
	INTC	PORT → INTC (detection/arbitration) → TX19A core
(4) RTC interrupt RTC	CG	PORT → CG (detection) → INTC (arbitration) → TX19A core
(5) Other interrupts	INTC	Peripheral circuit → INTC (detection/arbitration) → TX19A core

6.8.3 Interrupt Priority Arbitration

1. Seven levels of interrupt 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 IMCxx <ILxxx> of INTC. The higher the interrupt level set, the higher the priority. If the value is set to "000" meaning interrupt level of 0, no interrupts will be generated by the factor. Also note that any factors of interrupt level 0 are not suspended.

2. Interrupt Level Notification

   When an interrupt request is generated, INTC compares the interrupt level with the mask level. If the interrupt level is higher than the mask level set in ILEV <CMASK>, it notifies the TX19A processor of the interrupt request.

   If more than one interrupts are generated at the same time, the interrupts are notified in accordance with the priority order of these interrupt levels. If more than one interrupts of a same interrupt level are generated at the same time, these interrupts are notified in the order of the interrupt number as listed in Table 6.8.1.

   When an interrupt request of the same interrupt factor is received again before the previous interrupt has been cleared, only the first interrupt can be accepted.

3. INTC Register Update

   When an interrupt request is accepted by the TX19A core, the highest interrupt level at that point in time will be set to ILEV <CMASK> and the corresponding vector value is set to IVR. Once CMASK and IVR are set, any interrupt with a higher interrupt level cannot update them or cause notification to the core until the IVR value is read.

(Note 1) So, be sure to read the IVR value before attempting to change the ILEV value. If the ILEV value is changed before reading IVR, an unexpected interrupt request may be generated.

6.8.4 Hardware Interrupt Operation

When a hardware interrupt is generated, the TX19A core will go through the following steps to jump to the corresponding exception vector address as given in Table 6.21 according to the Status <BEV> and Cause <IV> bits of the CP0 register.

(1) Sets Status <EXL> of CP0 register to "1."

(2) Sets the PC value at the interrupt generation to EPC of the CP0 register.

(3) If the shadow register set is enabled (CP0 register SSCR <SSD> = 0), SSCR <CSS/PSS> of the CP0 register will be updated and it switches to the register bank of the same interrupt level number.

(4) The values of ILEV <CMASK/PMASKx> of INTC will be updated and the mask level is set to the interrupt level of the interrupt request accepted.

(5) Sets IVR [7:0] to the corresponding value listed in Table 6.8.1.

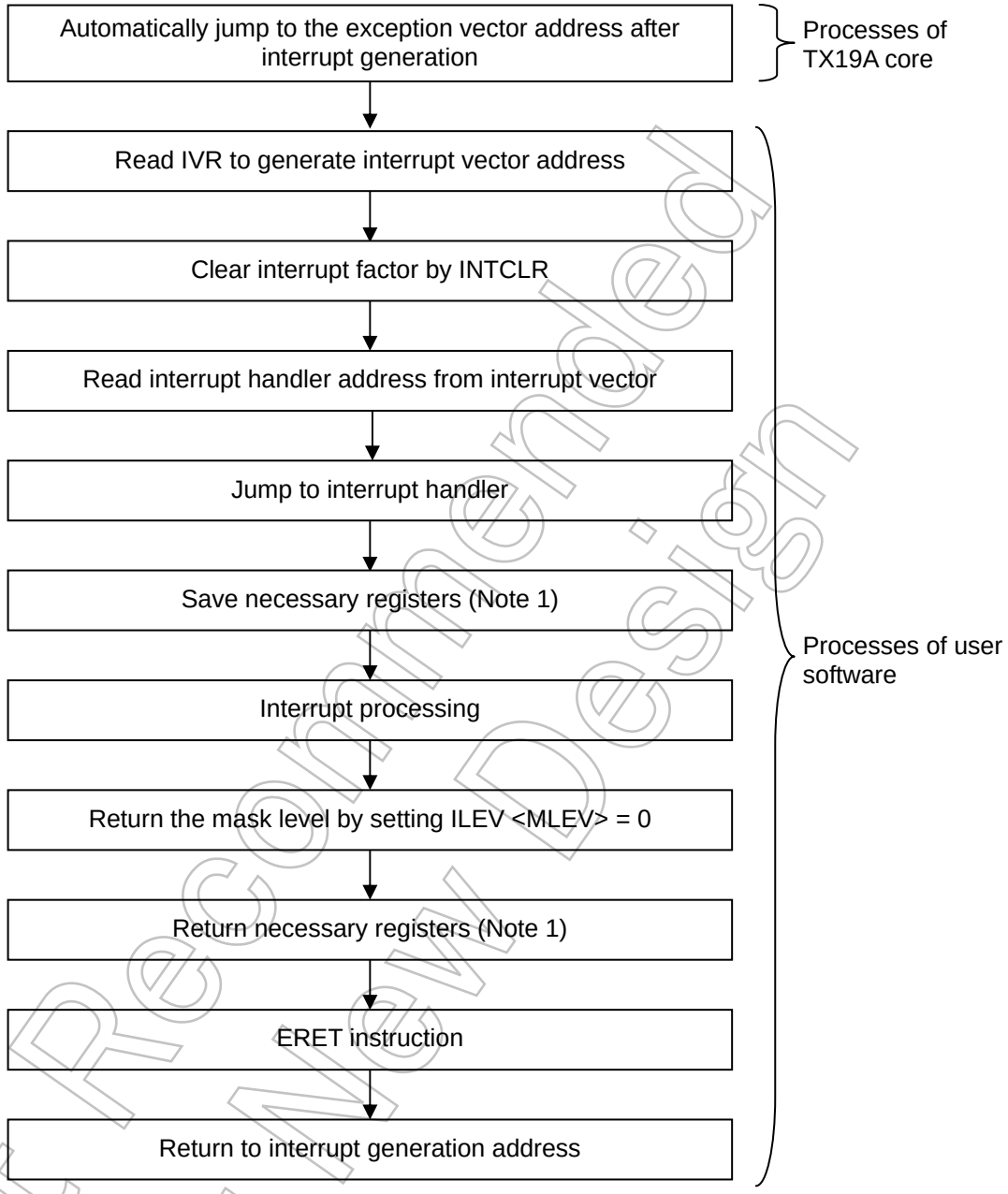


Fig. 6.8.4 Basic Operation of Hardware Interrupts (Example)

(Note 1) By using the shadow register set (setting CP0 register SSCR <SSD> = 0), most of general purpose register contents can be automatically saved in TX19A core.

6.8.5 Initialization for Interrupts

Before using interrupts, it is necessary to appropriately configure them. Necessary settings that have to be made regardless of the interrupt factors are described in Section 6.8.5.1 "Common Initialization" and settings specifically required for certain factors and applications are described in Section 6.8.5.2 "Initialization for Individual Interrupt Factors."

6.8.5.1 Common Initialization

In order to use interrupts, the following settings are necessary:

(1) Set Status <IM [4:2]> of CP0 register to "111."

(2) Set the base address of the interrupt vector table to IVR [31:8] of INTC.

(3) Set the interrupt handler addresses for the respective interrupt factors to the addresses obtained as the sum of the base address of "the interrupt vector table and the IVR [7:0] values corresponding to the respective interrupt factors."

Example of the above step (1): When the interrupt exception vector address 0xBFC00400 is used

```
lui     r2,0x1040           ; CU0=1, BEV =1 (r2 =0x1040_xxxx)
addiu   r2,r2,0x1C00        ; IM4,IM3,IM2 =1 (r2 =0x1040_1C00)
mtc0    r2,r12
```

Example of the above step (2): If Vector Table is used as the label of the interrupt vector table

```
lui     r3,hi(VectorTable)
addiu   r3,r3,lo(VectorTable)   ; r3 = VectorTable address
lui     r2,hi(IVR)              ; r2 =0xFFFF_xxxx (Upper 16 bits of IVR address)
sw      r3,lo(IVR)(r2)          ; Set address of VectorTable to IVR[31:8]
```

Example of the above step (3): If the base address of interrupt vector is set to 0xBFC20000

```
_VectorTable section code isa32 abs=0xBFC20000
VectorTable:
dw      _SWINT          ; 0 --- software interrupt
dw      _INT0           ; 1 --- INT0
dw      _INT1           ; 2 --- INT1
dw      _INT2           ; 3 --- INT2
dw      _INT3           ; 4 --- INT3
dw      _INT4           ; 5 --- INT4
dw      _INT5           ; 6 --- INT5
dw      _INT6           ; 7 --- INT6
dw      _INT7           ; 8 --- INT7
```

(Note 1) The above examples assume the use of Assembler made by Toshiba. If any third party Assembler is used, it may generate syntax errors; you are advised to modify the above statements according to the Assembler to be used.

6.8.5.2 Initialization for Individual Interrupt Factors

The registers to be set in using different interrupt factors are as listed below:

Table 6.8.4 Registers to be Set for Detecting Interrupts

Interrupt	Registers to be Set	Interrupt detection levels available (setting in active condition)
(1) Interrupts from external pins INT0 - INTB	PxFC(PORT) PxCR(PORT) IMCxx(INTC)	With INTC, "L" and "H" levels and falling and rising edges can be set.
	PxFC(PORT) PxCR(PORT) IMCGx(CG) IMCxx(INTC)	If it is to be used for recovery from Standby mode, set "L" and "H" levels and falling and rising edges for CG while INTC must be set to "H."
(2) Interrupts from external pins INTC - INTF	PxFC(PORT) PxCR(PORT) IMCxx(INTC)	With INTC, "L" and "H" levels and falling and rising edges can be set.
(3) Key on wakeup interrupt KWUP00 - 31	PxFC(PORT) PxCR(PORT) IMCxx(INTC)	With INTC and KWUP circuit, "L" and "H" levels and falling and rising edges can be set.
	PxFC(PORT) PxCR(PORT) IMCGx(CG) IMCxx(INTC)	If it is to be used for recovery from Standby mode, it must be set to "H" with INTC. With the KWUP circuit, "L" and "H" levels and falling/rising edges can be set.
(4) INTRTC interrupt	PxFC(PORT) PxCR(PORT) IMCGx(CG) IMCxx(INTC)	Set for rising edge with CG; it must be set to "H" with INTC.
(5) Other interrupts	IMCxx(INTC)	With INTC, "L" and "H" levels and falling/rising edges can be set.

(Note 1) In level detection, the value is checked at internal clock timing each time. Edge detection is made by comparing the previous value with the current value at internal clock timing.

In interrupt initialization, follow the order of the interrupt detection route as indicated in Table 6.8 before enabling the interrupts with the CP0 register. If any different setting order is used, an unexpected interrupt may be generated. So, be sure to clear interrupt factors before setting interrupt permission. Similarly, if interrupts are to be disabled, first disable the interrupt by the CP0 register and then set the registers accordingly in the reverse order of interrupt detection.

(1) Interrupts from external pins (INT0 to INTB)

- Use PORT PxCR to enable an input port. (Refer to 7. Port Function)
- Use PORT PxFC to set pin functions to INT0 - INTB. (Refer to 7. Port Function)
- Use PORT PxPE to set pull-up connections as appropriate. (Refer to 7. Port Function)
- Use INTC IMCx <EIMxx> to set active state. (Refer to 5.3.3 Interrupt-related Registers)
- Use IMCGx <EMCGxx> of CG for setting to enable/disable clearing of standby modes. (Refer to INTCG Registers, Interrupts to Clear STOP, SLEEP, and IDLE)
- Use INTC IMCx <EIMxx> to set active state of internal interrupt signals to be notified from CG. If rising or falling edge is set with INTC IMCx <EIMxx>, set it to falling edge (set IMCx <EIMxx> to "10"). For H/L level setting, set it to "L" level (set IMCx <EIMxx> to "00"). (Refer to 6.8.8 Registers.)

- An example setting when an external interrupt "INT3" is used to clear Stop by the falling edge:

```
Status<IE> = "0"                          ; Interrupt is disabled
PACR<PA3C> = "0"                          ; The port is set to an input port
PAFC<PA3F> ="0"                           ; The port is assigned to INT3
IMCGA<EMCG32:30> ="010"                   ; INT3 is set to falling edge
IMCGA<INT3EN> ="1"                        ; INT3 is set to clear Standby mode
EICRCG<ICRCG3:0> ="0011"                  ; Clears the INT3 standby clear request
IMC1<EIM41:40> ="01"                      ; INT3 is set to level detection
INTCLR<EICLR7:0> ="010"                   ; Clears the INT3 interrupt request
IMC1<IL42:40> ="101"                      ; Interrupt level of INT3 is set to "5."
ILEV<MLEV>/<CMASK> ="1"/"xxx"             ; Mask level is set to "xxx."
                                            (To be set simultaneously with ILEV <MLEV>)
SYNC instruction                          ; Stall until interrupt settings are enabled.
Status<IE> = "1"                          ; Interrupt is enabled
```

- An example setting when an external interrupt "INT3" is to be disabled:

```
Status<IE> = "0"                          ; Interrupt is disabled
IMC1<IL42:40> = "000"                     ; INT3 interrupt is disabled.
INTCLR<EICLR7:0> ="010"                   ; Clears the INT3 interrupt request
```

(2) Interrupts from external pins (INTC to INTF)

- Use PORT PxIER to enable an input port. (Refer to 7. Port Function)
- Use PORT PxFR to set pin functions to INTC - INTF. (Refer to 7. Port Function)
- Use INTC IMCx <EIMxx> to set active state. (Refer to 6.8.8 Registers.)

- An example setting when an external interrupt "INTF" is detected by the "H" level:

```
Status<IE> = "0"                          ; Interrupt is disabled
P5CR<P53C> = "0"                          ; The port is set to an input port
P5FC<P53F> = "0"                          ; The port is set to an input port
IMC4<EIM101:100> = "01"                   ; INTF is set to "H" level
INTCLR<EICLR7:0> = "0x040"                ; Clears the INTF interrupt request
IMC4<IL102:IL100> = "010"                 ; Interrupt level of INTF is set to "2."
ILEV<MLEV>/<CMASK> = "1"/ "xxx"           ; Mask level is set to "xxx." (To be set simultaneously
                                            with ILEV <MLEV>)
Status<IE> = "1"                          ; Interrupt is enabled
```

(3) Key on Wakeup interrupt, KWUP00 to 31

- Use PORT PxCR to enable the input port. (Refer to 7. Port Function)
- Use PORT PxFC to set the pin function to KEY. (Refer to 7. Port Function)
- Use PORT PxPE to set pull-up connections as appropriate. (Refer to 7. Port Function)
- Use KWUPSTxx to enable KEY ON Wakeup. (Refer to 20. KEY ON Wakeup Circuit)
- Set active state of KEY. (Refer to 20. KEY ON Wakeup Circuit.)
- Use INTC IMCx <EIMxx> to set active state. (Refer to 5.3.3 Interrupt-related Registers)
- Use IMCGx <EMCGxx> of CG for setting to enable/disable clearing of Standby. (Refer to INTCG Registers, Interrupts to Clear STOP, SLEEP, and IDLE)

• An example setting when KEY08 is used as an input to clear Stop (dynamic pull-up, falling edge):

```
Status<IE> = "0"                        ; Interrupt is disabled
PECR<PE0C> = "0"                        ; The port is set to work as an input port.
PEFC<PE0F> = "0"                        ; The port is set to KEY input.
PEPE<PEE0> = "1"                        ; Pull-up is set to the port.
KWUPCNT = "0x24"                        ; The period of dynamic pull-up is set.
                                          (Example: Period; 10:1024/fs, Duration: 01:4/fs)
KWUPST08<DPE08> = "1"                   ; Dynamic pull-up is set.
KWUPST08<KEY82:80> = "010"              ; It is set to falling edge.
KWUPST08<KEY08EN> = "1"                 ; Key input is enabled.
KWUPCLR<KEYCLR3:0> = "1010"             ; Key input factor is cleared.
IMCGD<EMCGC1:C0> = "10"                 ; Standby clear setting is set to "H" level
IMCGD<KWUPEN> = "1"                     ; KWUP is set to clear Standby mode.
EICRCG<ICRCG3:0> = "1100"               ; Clears KWUP standby clear request
IMC4<EIM111:110> = "01"                 ; KWUP is set to H level.
IMC4<IL112:110> = "110"                 ; Interrupt level of KWUP is set to "6."
ILEV<MLEV>/<CMASK> ="1"/"xxx"           ; Mask level is set to "xxx."
                                          (To be set simultaneously with ILEV <MLEV>)
SYNC instruction                        ; Stall until interrupt settings are enabled.
Status<IE> = "1"                        ; Interrupt is enabled
```

(4) Other hardware interrupts

- Settings are made to use peripheral hardware devices.
- Set INTC IMCxx <EIMxx> to "10." (Refer to 6.8.8 Registers.)

(Note 1) In interrupt initialization, set INTC registers before enabling interrupts with the CP0 register. Similarly, if interrupt is to be disabled, first disable interrupt by the CP0 register and then set INTC.

6.8.5.3 Interrupt Enable

In order for an interrupt request to be accepted, all the following parameters must be set to enable the interrupt in addition to the initial settings described in Section 6.8.5 "Initialization for Interrupts."

- Set Status <ERL> of the CP0 register to "0."
- Set Status <EXL> of the CP0 register to "0."
- Set Status <IE> of the CP0 register to "1."

By these settings, interrupt is enabled two clocks after execution of the instruction and the registers are set. Note that one of the following methods may be used in setting Status <IE> of the CP0 register to "1."

- Set IER of the CP0 register to any value other than "0" using the MTC0 instruction (32 bit ISA instruction). (Note 1)
- Execute the EI instruction of 16 bit ISA. (Note 2)

(Note 1) If Toshiba C compiler is used, this is executed by the 32 bit ISA instruction "_ _EI ( ) embedded function."

(Note 2) If Toshiba C compiler is used, this instruction is executed by the 16 bit ISA instruction "_ _EI ( ) embedded function."

(Note 3) The following different methods may also be used to set Status <IE> of the CP0 register to "1."

- Set Status <IE> of the CP0 register to "1" using the MTC0 instruction of 32 bit ISA.
- Set Status <IE> of the CP0 register to "1" using the MTC0 instruction of 16 bit ISA.

6.8.5.4 Interrupt Disable

To disable interrupts, either one of the following setting procedures must be performed in addition to the settings described in Section 6.8.5 "Initialization for Interrupts." When interrupts are disabled, any interrupt request will be suspended. Also note that TMP19A43 doesn't suspend any interrupt factor that is set to interrupt level 0.

- Set Status <ERL> of the CP0 register to "1."
- Set Status <EXL> of the CP0 register to "1."
- Set Status <IE> of the CP0 register to "0."

By these settings, interrupts are disabled immediately after execution of the instruction and the registers are set two clocks later. Note that either of the following methods may be used in setting Status <IE> of the CP0 register to "0."

- Set IER of the CP0 register to "0" using the MTC0 instruction of 32 bit ISA. (Note 1)
- Execute the DI instruction of 16-bit mode ISA. (Note 2)

(Note 1) If Toshiba C compiler is used, this instruction is executed by the 32 bit ISA instruction "_ _DI ( ) embedded function."

(Note 2) If Toshiba C compiler is used, this instruction is executed by the 16 bit ISA instruction "_ _DI ( ) embedded function."

(Note 3) The following different methods may also be used to set Status <IE> of the CP0 register to "0."

- Set Status <IE> of the CP0 register to "0" using the MTC0 instruction of 32 bit ISA.
- Set Status <IE> of the CP0 register to "0" using the MTC0 instruction of 16 bit ISA.

If the factors once enabled are to be individually disabled again after setting interrupt levels by IMCx <ILxxx> of INTC, first set the Status <ERL/EXL/EI> bits of the CP0 register to disable interrupts and then disable relevant factors individually.

Example statements to individually disable interrupt factors:

```
mtc0    r0, IER          ; Interrupt is disabled (Status<IE> = "0")
sb      r0, IMCxx        ; Disable interrupt factors
sync                     ; Stall until it is write-enabled.
mtc0    r29, IER         ; Interrupt is enabled (Status<IE> = "1")
```

(Note 4) The above examples assume use of Assembler made by Toshiba. If any third party Assembler is used, it may generate syntax errors; you are advised to modify the above statements according to the Assembler to be used.

6.8.6 Interrupt Processing

This section describes detailed operation of interrupt processing using the basic flow chart of Fig. 6.8.

6.8.6.1 Interrupt Response and Return

① Hardware processes to accept interrupts

After interrupt request arbitration, INTC sets the interrupt vector and interrupt level of the interrupt request accepted to IVR and ILEV<CMASK>, respectively, to notify the TX19A processor core of the interrupt level. When the interrupt level is notified, the TX19A processor core sets Status <EXL> of the CP0 register to "1" to disable interrupts and saves the PC value at the interrupt generation to EPC. If the shadow register set is enabled (CP0 register SSCR <SSD> = 0), the processor core sets the interrupt level to SSCR <CSS> of the CP0 register and switches the register bank.

When an interrupt is accepted, any ongoing execution is suspended and it automatically jumps to the exception vector address (for interrupts). Fig. 6-4 shows the sequence of accepting interrupts.

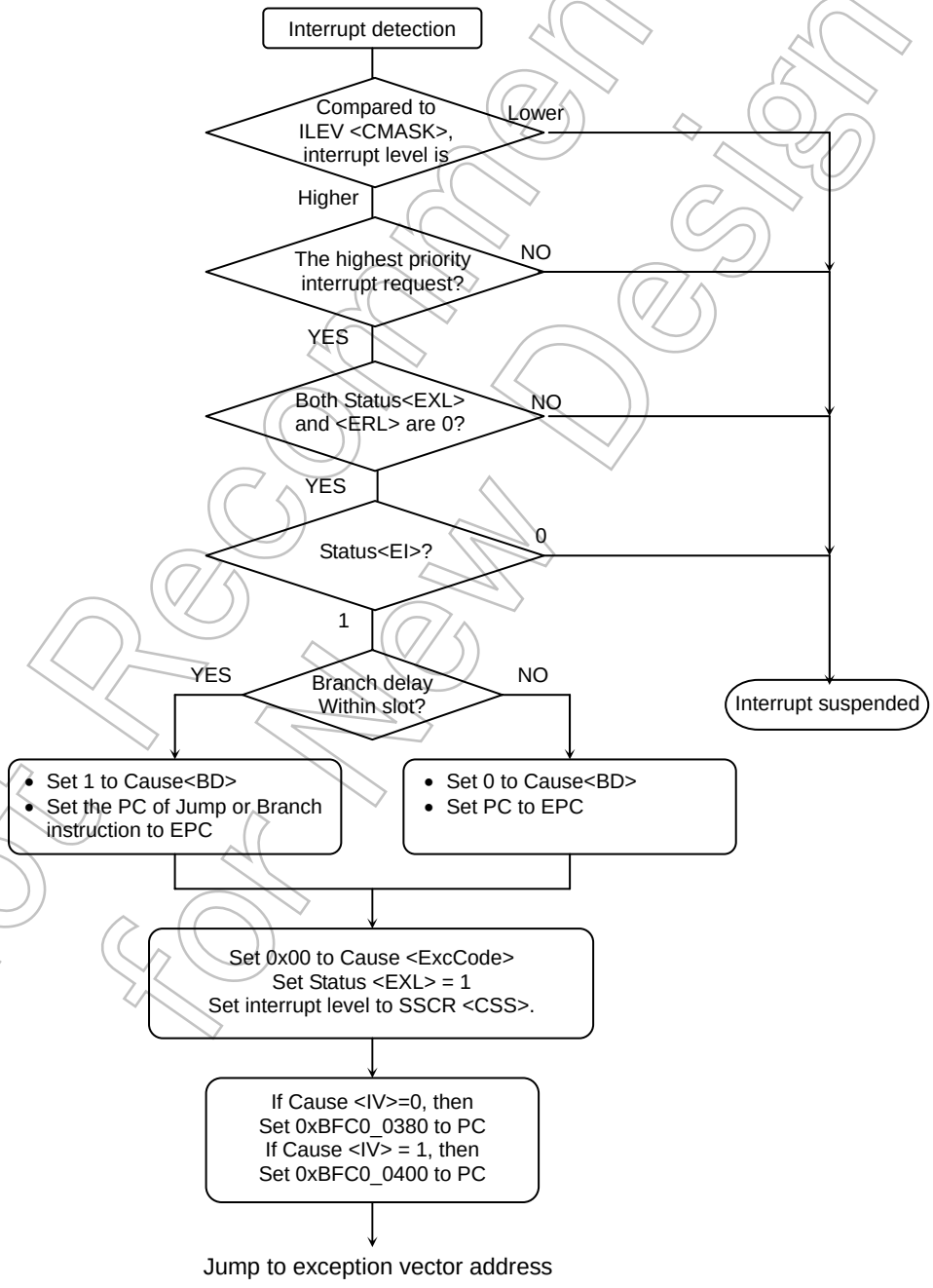


Fig. 6-4 Hardware Process Flow to Accept Interrupts

② Processes to be performed by the exception handler

After an interrupt request is accepted, it automatically jumps to the exception handler where the interrupt vector address is read from INTC IVR and the user program generates the address of the interrupt handler. As in the example statements presented in Section 6.8.5, "Initialization for Interrupts," the interrupt vector base address is set to IVR[31:8] so that the IVR value becomes the interrupt vector address.

After reading the INTC IVR value, the interrupt factor is cleared. If the interrupt factor is cleared before IVR is read, correct value cannot be read because the IVR value is also cleared.

Example exception handler statement: Exception vector address (interrupt) is 0xBFC0_0400.

VECTOR_INT section code isa32 abs=0xBFC00400

```
__InterruptVector:
lui     r26,hi(IVR)
lw      r26,lo(IVR)(r26)        ; Read IVR for interrupt vector address
lui     r27,hi(INTCLR)
sh      r26,lo(INTCLR)(r27)     ; Interrupt request is cleared
lw      r26,0(r26)              ; Read interrupt handler address from interrupt vector
jr      r26                     ; Jump to interrupt handler
nop
```

(Note 1) The above example assumes use of Assembler made by Toshiba. If any third party Assembler is used, it may generate syntax errors; you are advised to modify the above statement according to the Assembler to be used.

③ Processes to be performed by the interrupt handler

Typical tasks of the interrupt handler are to save appropriate registers and to process interrupts. If the shadow register set is enabled (CP0 register SSCR <SSD> = 0), the general purpose register values other than r26, r27, r28, and r29 (Shadow Register Set number 1 to 7) are automatically saved so the user program doesn't have to save these. Refer to the separate volume "TX19A Core Architecture" for details of general purpose registers that are to be automatically saved.

In general, registers other than GPR are dependent on user programs. The Status, EPC, SSCR, HI, LO, Cause, and Config values of the CP0 register shall be saved as appropriate.

For using multiple interrupts, interrupts are enabled by clearing Status <EXL> of the CP0 register to "0" after appropriate saving processes.

(Note 1) Note that general exceptions can be accepted even when interrupts are disabled. So, even when you don't use multiple interrupts, it is desirable to save any general purpose register and the CP0 register that could be overwritten by general exceptions.

Example interrupt handler settings to be necessary:

```
Save from SSCR to stack          ; Save SSCR values (as appropriate)
NOP instruction                  ; Stall until SSCR is switched
NOP instruction                  ; Stall until SSCR is switched
Save from EPC to stack           ; Save EPC values (as appropriate)
Save from Status to stack        ; Save Status values (as appropriate)
NOP instruction                  ; Stall before executing ERET instruction
NOP instruction                  ; Stall before executing ERET instruction
Status<EXL> = "0"                ; Interrupt enable (only for multiple interrupts)
```

(Note 1) After overwriting SSCR of the CP0 register, wait for two cycles to allow for register bank switching before attempting a register access.

④ Returning from the interrupt handler

For returning from the interrupt handler to the main process, return the register values saved at the top of the interrupt handler process and set "0" to INTC ILEV <MLEV> to clear the interrupt mask level. By executing the ERET instruction after all the return tasks are completed, Status <EXL> of the CP0 register is cleared to "0" and the EPC address returns to PC for the main process to be resumed. If the shadow register set has been enabled (CP0 register SSCR <SSD> = 0), SSCR <CSS> is updated by the ERET instruction and the Shadow Register Set number is automatically decremented for automatically returning the general purpose registers saved in the register bank.

If multiple interrupts are used, it is necessary to set Status <EXL> of the CP0 register to "1" to disable interrupts prior to executing the return process.

Example settings to return from the interrupt hander:

```
Status<EXL> = "1"                ; Interrupt disable (only for multiple interrupts)
ILEV<MLEV> = "0"                 ; Decrement the mask level
SYNC instruction                 ; Stall until mask level is decremented
Return to SSCR                   ; Return SSCR values saved (as appropriate)
NOP instruction                  ; Stall until SSCR is switched
NOP instruction                  ; Stall until SSCR is switched
Return to EPC                    ; Return SSCR values saved (as appropriate)
Return to Status                 ; Return Status values saved (as appropriate)
NOP instruction                  ; Stall before executing ERET instruction
NOP instruction                  ; Stall before executing ERET instruction
ERET instruction                 ; Status<EXL> = "0," EPC to PC, SSCR<PSS> to
                                   SSCR<CSS>
```

(Note 1) After overwriting SSCR of the CP0 register, wait for two cycles to allow for register bank switching before attempting a register access.

(Note 2) Don't access the CP0 register two instructions prior to executing the ERET instruction.

6.8.7 Example of Multiple Interrupt Setting

In "multiple interrupt" processing, a higher interrupt level interrupt is processed while an interrupt is being processed. With TMP19A43, multiple interrupts are processed through the interrupt priority arbitration function of INTC. When an interrupt request is accepted, ILEV <CMASK> of INTC is automatically updated to the interrupt level of the interrupt accepted to enable arbitration to use the priority preset by the user program.

① Additional processes required for multiple interrupts

When an interrupt is accepted, Status <EXL> of the CP0 register is set to "1" disabling further interrupts. In order to allow multiple interrupts, it is necessary to save the registers that could be overwritten by the second and the following interrupts before enabling the multiple interrupt process. For this purpose, in addition to the typical exception handler and interrupt handler processes, save the following registers before setting Status <EXL> of the CP0 register to "0" to enable interrupts.

CP0 registers that must be saved:

- EPC
- SSCR
- Status

Save the HI, LO, Cause, and Config registers as appropriate.

(Note 1) Some of the registers may be automatically saved and returned by using some interrupt function of Toshiba C compiler. Refer to "TX19A C Compiler Reference" provided with the Toshiba C compiler for more details.

② Additional return processes required for multiple interrupts

Before returning registers in the interrupt return process, it is necessary to disable interrupts using the method described in Section 6.8.5.4 "Interrupt Disable." This is to prevent the returned register values from being corrupted by multiple interrupts. Note that the ERET instruction automatically clears Status <EXL> of the CP0 register to "0." So, by setting Status <EXL> of the CP0 register to "1" to disable interrupts in the returning process, you can return from the interrupt with interrupts enabled automatically.

③ Proper use of Status <EXL> and Status <IE>

While there is no significant distinction between the Status <EXL> and Status <IE> parameters, Status <EXL> is automatically set to "1" upon interrupt generation and cleared to "0" by the ERET instruction automatically. In saving and returning register values at the initial and final phases of an interrupt process, where interrupts have to be disabled, hardware controlled Status <EXL> is normally used. Status <IE> is used for other general interrupt enable/disable control functions.

Applicable interrupt enable/disable control sequences are described in Section 6.8.7.1, "Interrupt Control for Multiple Interrupts."

6.8.7.1 Interrupt Control for Multiple Interrupts

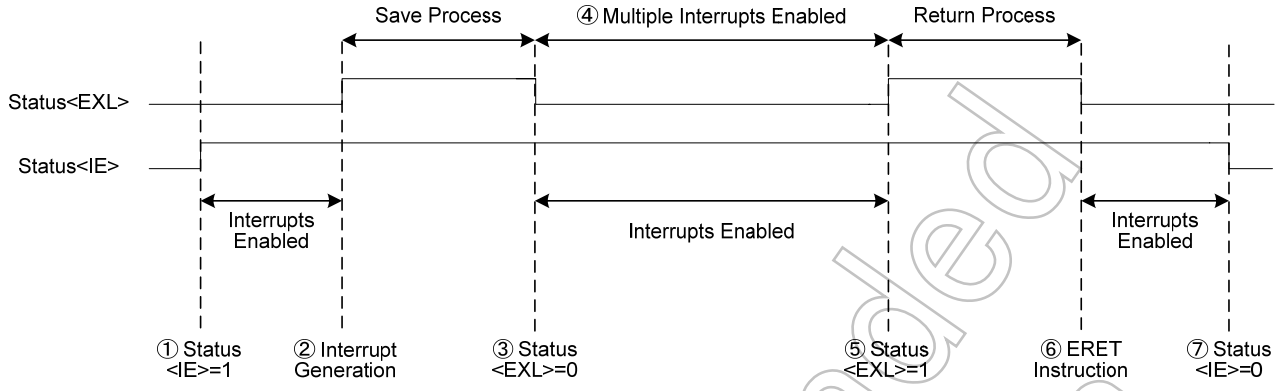


Fig. 6-5 Interrupt Enable/Disable Control Sequence for Multiple Interrupts

① Status <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.

6.8.8 Registers

6.8.8.1 Register Map

Table 6.8.5 INTC Register Map

Address	Register symbol	Register	Corresponding interrupt number
0xFFFF_E000	IMC0	Interrupt mode control register 00	0 to 3
0xFFFF_E004	IMC1	Interrupt mode control register 04	4 to 7
0xFFFF_E008	IMC2	Interrupt mode control register 08	8 to 11
0xFFFF_E00C	IMC3	Interrupt mode control register 12	12 to 15
0xFFFF_E010	IMC4	Interrupt mode control register 16	16 to 19
0xFFFF_E014	IMC5	Interrupt mode control register 20	20 to 23
0xFFFF_E018	IMC6	Interrupt mode control register 24	24 to 27
0xFFFF_E01C	IMC7	Interrupt mode control register 28	28 to 31
0xFFFF_E020	IMC8	Interrupt mode control register 32	32 to 35
0xFFFF_E024	IMC9	Interrupt mode control register 36	36 to 39
0xFFFF_E028	IMCA	Interrupt mode control register 40	40 to 43
0xFFFF_E02C	IMCB	Interrupt mode control register 44	44 to 47
0xFFFF_E030	IMCC	Interrupt mode control register 48	48 to 51
0xFFFF_E034	IMCD	Interrupt mode control register 52	52 to 55
0xFFFF_E038	IMCE	Interrupt mode control register 56	56 to 9
0xFFFF_E03C	IMCF	Interrupt mode control register 60	60 to 63
0xFFFF_E040	IVR	Interrupt vector register	
0xFFFF_E060	INTCLR	Interrupt request clear register	
0xFFFF_E10C	ILEV	Interrupt mask level register	

(Note 1) While the interrupt mode control register (IMCxx) is a 32 bit register, 8 bit/16 bit access is also accepted.

6.8.8.2 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 6.8.1 is set to IVR [7:0]. By setting the base address of interrupt vectors to IVR [31:8], a read/write register, simply reading the IVR value can provide the corresponding interrupt vector address.

Table 6.8.6 Interrupt Vector Register

IVR (0xFFFF_E040)

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

6.8.8.3 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 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."

Table 6.8.7 Interrupt Level Register

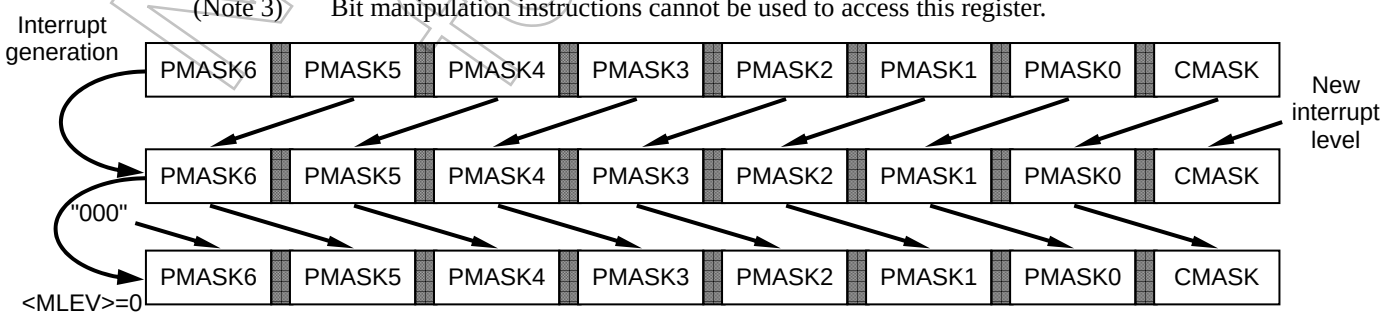
ILEV (0xFFFF_E10C)

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



6.8.8.4 Interrupt Mode Control Registers (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.

IMC0 (0xFFFF_E000)

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

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

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

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

IMC1 (0xFFFF_E004)

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

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

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

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

IMC2 (0xFFFF_E008)

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

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

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

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

IMC3 (0xFFFF_E00C)

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

	15	14	13	12	11	10	9	8
bit Symbol		EIMD1	EIMD0	DMD		ILD2	ILD1	ILD0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 13 to be the activation factor.	Always reads "0."	If DMD = 0, select the interrupt level for interrupt number 13 (INTC) 000: Disable Interrupt 001 to 111: 1 to 7 If DMD = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIME1	EIME0	DME		ILE2	ILE1	ILE0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 14 to be the activation factor.	Always reads "0."	If DME = 0, select the interrupt level for interrupt number 14 (INTD) 000: Disable Interrupt 001 to 111: 1 to 7 If DME = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIMF1	EIMF0	DMF		ILF2	ILF1	ILF0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 15 to be the activation factor.	Always reads "0."	If DMF = 0, select the interrupt level for interrupt number 15 (INTE) 000: Disable Interrupt 001 to 111: 1 to 7 If DMF = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

IMC4 (0xFFFF_E010)

	7	6	5	4	3	2	1	0
bit Symbol		EIM101	EIM100	DM10		IL102	IL101	IL100
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 16 is set as the activation factor	Always reads "0."	If DM10 = 0, select the interrupt level for interrupt number 16 (INTF) 000: Disable Interrupt 001 to 111: 1 to 7 If DM10 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM111	EIM110	DM11		IL112	IL111	IL110
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 01: "H" level **Be sure to set "01."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 17 to be the activation factor.	Always reads "0."	If DM11 = 0, select the interrupt level for interrupt number 17 (KWUP) 000: Disable Interrupt 001 to 111: 1 to 7 If DM11 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM121	EIM120	DM12		IL122	IL121	IL120
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 18 to be the activation factor.	Always reads "0."	If DM12 = 0, select the interrupt level for interrupt number 18 (INTRX0). 000: Disable Interrupt 001 to 111: 1 to 7 If DM12 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM131	EIM130	DM13		IL132	IL131	IL130
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 19 to be the activation factor.	Always reads "0."	If DM13 = 0, select the interrupt level for interrupt number 19 (INTTX0) 000: Disable Interrupt 001 to 111: 1 to 7 If DM13 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMC5 (0xFFFF_E014)

	7	6	5	4	3	2	1	0
bit Symbol		EIM141	EIM140	DM14		IL142	IL141	IL140
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 20 is set as the activation factor	Always reads "0."	If DM14 = 0, select the interrupt level for interrupt number 20 (INTRX1). 000: Disable Interrupt 001 to 111: 1 to 7 If DM14 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM151	EIM150	DM15		IL152	IL151	IL150
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 21 to be the activation factor.	Always reads "0."	If DM15 = 0, select the interrupt level for interrupt number 21 (INTTX1) 000: Disable Interrupt 001 to 111: 1 to 7 If DM15 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM161	EIM160	DM16		IL162	IL161	IL160
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 22 to be the activation factor.	Always reads "0."	If DM16 = 0, select the interrupt level for interrupt number 22 (INTRX2). 000: Disable Interrupt 001 to 111: 1 to 7 If DM16 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM171	EIM170	DM17		IL172	IL171	IL170
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 23 to be the activation factor.	Always reads "0."	If DM17 = 0, select the interrupt level for interrupt number 23 (INTTX2). 000: Disable Interrupt 001 to 111: 1 to 7 If DM17 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMC6 (0xFFFF_E018)

	7	6	5	4	3	2	1	0
bit Symbol		EIM181	EIM180	DM18		IL182	IL181	IL180
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 24 is set as the activation factor	Always reads "0."	If DM18 = 0, select the interrupt level for interrupt number 24 (INTRX3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM18 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM191	EIM190	DM19		IL192	IL191	IL190
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 25 to be the activation factor.	Always reads "0."	If DM19 = 0, select the interrupt level for interrupt number 25 (INTTX3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM19 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM1A1	EIM1A0	DM1A		IL1A2	IL1A1	IL1A0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 26 to be the activation factor.	Always reads "0."	If DM1A = 0, select the interrupt level for interrupt number 26 (INTRX4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1A = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM1B1	EIM1B0	DM1B		IL1B2	IL1B1	IL1B0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 27 to be the activation factor.	Always reads "0."	If DM1B = 0, select the interrupt level for interrupt number 27 (INTTX4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1B = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMC7 (0xFFFF_E01C)

	7	6	5	4	3	2	1	0
bit Symbol		EIM1C1	EIM1C0	DM1C		IL1C2	IL1C1	IL1C0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 28 to be the activation factor.	Always reads "0."	If DM1C = 0, select the interrupt level for interrupt number 28 (INTRX5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1C = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM1D1	EIM1D0	DM1D		IL1D2	IL1D1	IL1D0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 29 to be the activation factor.	Always reads "0."	If DM1D = 0, select the interrupt level for interrupt number 29 (INTTX5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1D = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM1E1	EIM1E0	DM1E		IL1E2	IL1E1	IL1E0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 30 to be the activation factor.	Always reads "0."	If DM1E = 0, select the interrupt level for interrupt number 30 (INTS0). 000: Disable Interrupt 001 to 111: 1 to 7 If DM1E = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

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

IMC8 (0xFFFF_E020)

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

	15	14	13	12	11	10	9	8
bit Symbol		EIM211	EIM210	DM21		IL212	IL211	IL210
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 33 to be the activation factor.	Always reads "0."	If DM21 = 0, select the interrupt level for interrupt number 33 (INTTB0). 000: Disable Interrupt 001 to 111: 1 to 7 If DM21 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM221	EIM220	DM26		IL222	IL221	IL220
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	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 34 to be the activation factor.	Always reads "0."	If DM22 = 0, select the interrupt level for interrupt number 34 (INTTB1). 000: Disable Interrupt 001 to 111: 1 to 7 If DM22 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM231	EIM230	DM23		IL232	IL231	IL230
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."** **Standby setting "01"**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 35 to be the activation factor.	Always reads "0."	If DM23 = 0, select the interrupt level for interrupt number 35 (INTTB2) 000: Disable Interrupt 001 to 111: 1 to 7 If DM23 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMC9 (0xFFFF_E024)

	7	6	5	4	3	2	1	0
bit Symbol		EIM241	EIM240	DM24		IL242	IL241	IL240
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."** **Standby setting "01"**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 36 to be the activation factor.	Always reads "0."	If DM24 = 0, select the interrupt level for interrupt number 36 (INTTB3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM24 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM251	EIM250	DM25		IL252	IL251	IL250
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 37 to be the activation factor.	Always reads "0."	If DM25 = 0, select the interrupt level for interrupt number 37 (INTTB4). 000: Disable Interrupt 001 to 111: 1 to 7 If DM25 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM261	EIM260	DM26		IL262	IL261	IL260
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 38 to be the activation factor.	Always reads "0."	If DM26 = 0, select the interrupt level for interrupt number 38 (INTTB5). 000: Disable Interrupt 001 to 111: 1 to 7 If DM26 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM271	EIM270	DM27		IL272	IL271	IL270
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 39 to be the activation factor.	Always reads "0."	If DM27 = 0, select the interrupt level for interrupt number 39 (INTTB6). 000: Disable Interrupt 001 to 111: 1 to 7 If DM27 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMCA (0xFFFF_E028)

	7	6	5	4	3	2	1	0
bit Symbol		EIM281	EIM280	DM28		IL282	IL281	IL280
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 40 to be the activation factor.	Always reads "0."	If DM28 = 0, select the interrupt level for interrupt number 40 (INTTB7). 000: Disable Interrupt 001 to 111: 1 to 7 If DM28 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM291	EIM290	DM29		IL292	IL291	IL290
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 41 to be the activation factor.	Always reads "0."	If DM29 = 0, select the interrupt level for interrupt number 41 (INTTB8). 000: Disable Interrupt 001 to 111: 1 to 7 If DM29 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM2A1	EIM2A0	DM2A		IL2A2	IL2A1	IL2A0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 42 to be the activation factor.	Always reads "0."	If DM2A = 0, select the interrupt level for interrupt number 42 (INTTB9). 000: Disable Interrupt 001 to 111: 1 to 7 If DM2A = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

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

IMCB (0xFFFF_E02C)

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

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

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

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

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

IMCC (0xFFFF_E030)

	7	6	5	4	3	2	1	0
bit Symbol		EIM301	EIM300	DM30		IL302	IL301	IL300
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 48 to be the activation factor.	Always reads "0."	If DM30 = 0, select the interrupt level for interrupt number 48 (INTTBF). 000: Disable Interrupt 001 to 111: 1 to 7 If DM30 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM311	EIM310	DM31		IL312	IL311	IL310
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 1: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 49 to be the activation factor.	Always reads "0."	If DM31 = 0, select the interrupt level for interrupt number 49 (INTCAPG0) 000: Disable Interrupt 001 to 111: 1 to 7 If DM31 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

	31	30	29	28	27	26	25	24
bit Symbol		EIM331	EIM330	DM33		IL332	IL331	IL330
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 51 to be the activation factor.	Always reads "0."	If DM33 = 0, select the interrupt level for interrupt number 51 (INTCMP0) 000: Disable Interrupt 001 to 111: 1 to 7 If DM33 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMCD (0xFFFF_E034)

	7	6	5	4	3	2	1	0
bit Symbol		EIM341	EIM340	DM34		IL342	IL341	IL340
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0		
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 52 to be the activation factor.	Always reads "0."	If DM34 = 0, select the interrupt level for interrupt number 52 (INTCMP1) 000: Disable Interrupt 001 to 111: 1 to 7 If DM34 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM351	EIM350	DM35		IL352	IL351	IL350
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 53 to be the activation factor.	Always reads "0."	If DM35 = 0, select the interrupt level for interrupt number 53 (INTCMP2) 000: Disable Interrupt 001 to 111: 1 to 7 If DM35 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM361	EIM360	DM36		IL362	IL361	IL360
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 54 to be the activation factor.	Always reads "0."	If DM36 = 0, select the interrupt level for interrupt number 54 (INTCMP3) 000: Disable Interrupt 001 to 111: 1 to 7 If DM36 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM371	EIM370	DM37		IL372	IL371	IL370
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 55 to be the activation factor.	Always reads "0."	If DM37 = 0, select the interrupt level for interrupt number 55 (INTCMP4) 000: Disable Interrupt 001 to 111: 1 to 7 If DM37 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

IMCE (0xFFFF_E038)

	7	6	5	4	3	2	1	0
bit Symbol		EIM381	EIM380	DM38		IL382	IL381	IL380
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt requet. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 56 to be the activation factor.	Always reads "0."	If DM38 = 0, select the interrupt level for interrupt number 56 (INTCMP5) 000: Disable Interrupt 001 to 111: 1 to 7 If DM38 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	15	14	13	12	11	10	9	8
bit Symbol		EIM391	EIM390	DM39		IL392	IL391	IL390
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 57 to be the activation factor.	Always reads "0."	If DM39 = 0, select the interrupt level for interrupt number 57 (INTCMP6) 000: Disable Interrupt 001 to 111: 1 to 7 If DM39 = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

	23	22	21	20	19	18	17	16
bit Symbol		EIM3A1	EIM3A0	DM3A		IL3A2	IL3A1	IL3A0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 58 to be the activation factor.	Always reads "0."	If DM3A = 0, select the interrupt level for interrupt number 58 (INTCMP7) 000: Disable Interrupt 001 to 111: 1 to 7 If DM3A = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

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

IMCF (0xFFFF_E03C)

	7	6	5	4	3	2	1	0
bit Symbol		EIM3C1	EIM3C0	DM3C		IL3C2	IL3C1	IL3C0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 10: Falling edge **Be sure to set "10."** **Standby setting "01"**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 60 to be the activation factor.	Always reads "0."	If DM3C = 0, select the interrupt level for interrupt number 60 (INTRTC) 000: Disable Interrupt 001 to 111: 1 to 7 If DM3C = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: 4 to 7		

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

	23	22	21	20	19	18	17	16
bit Symbol		EIM3E1	EIM3E0	DM3E		IL3E2	IL3E1	IL3E0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level **Be sure to set "00.".**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 62 to be the activation factor.	Always reads "0."	If DM3E = 0, select the interrupt level for interrupt number 62 (INTDMA0-3). 000: Disable Interrupt 001 to 111: 1 to 7 If DM3E = 1, select the DMAC channel. 000 to 011:--- 100 to 111: 4 to 7		

	31	30	29	28	27	26	25	24
bit Symbol		EIM3F1	EIM3F0	DM3F		IL3F2	IL3F1	IL3F0
Read/Write	R	R/W			R	R/W		
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request. 00: "L" level **Be sure to set "00."**		Set as DMAC activation factor. 0: Non-activation factor 1: Interrupt number 63 to be the activation factor.	Always reads "0."	If DM3F = 0, select the interrupt level for interrupt number 63 (INTDMA4-7). 000: Disable Interrupt 001 to 111: 1 to 7 If DM3F = 1, select the DMAC channel. 000 to 011: 0 to 3 100 to 111: ---		

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

The access to the DMAC register by DMAC is a prohibition.

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

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

Note 3: When you change an active condition (when changing to the level detection)、Please change after putting the interrupt output of the corresponding device into the state of Deasart.

(1) Don't IL="0 setting IL="0"

(2) Change Detection condition (EIM)

(3) INTCLR 、Pertinent interrupt is clear.

(4) IL It sets it to "Excluding 0".

6.8.8.5 Interrupt Request Clear Registers (INTCLR)

When it is desired to clear any interrupt request being suspended, you can do so by setting the IVR [7:0] for the corresponding interrupt factor into the INTCLR register. When an interrupt request is cleared, the IVR value is also cleared and the interrupt factor cannot be determined anymore. Do not clear an interrupt request before reading the IVR value.

Table 6.8.8 Set the IVR <IVR7:0>

value that corresponds to the interrupt request that you would like to clear

INTCLR (0xFFFF_E060)

	7	6	5	4	3	2	1	0
bit Symbol	EICLR7	EICLR6	EICLR5	EICLR4	EICLR3	EICLR2	EICLR1	EICLR0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set the IVR <IVR7:0> value that corresponds to the interrupt request that you would like to clear.							
	15	14	13	12	11	10	9	8
bit Symbol								
Read/Write	R							
After reset	0							
Function	Always reads "0."							
	23	22	21	20	19	18	17	16
bit Symbol								
Read/Write	R							
After reset	0							
Function	Always reads "0."							
	31	30	29	28	27	26	25	24
bit Symbol								
Read/Write	R							
After reset	0							
Function	Always reads "0."							

(Note 1) 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>, i.e., either "H" level, "L" level, rising edge, or falling edge, clear the interrupt request.

(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. Otherwise, such an unnecessary external transfer request should not be accepted by disabling interrupts using IMCx <ILxxx> or by canceling the corresponding DMAC activation factors using IMCx<DMxx> before accepting such external transfer requests.

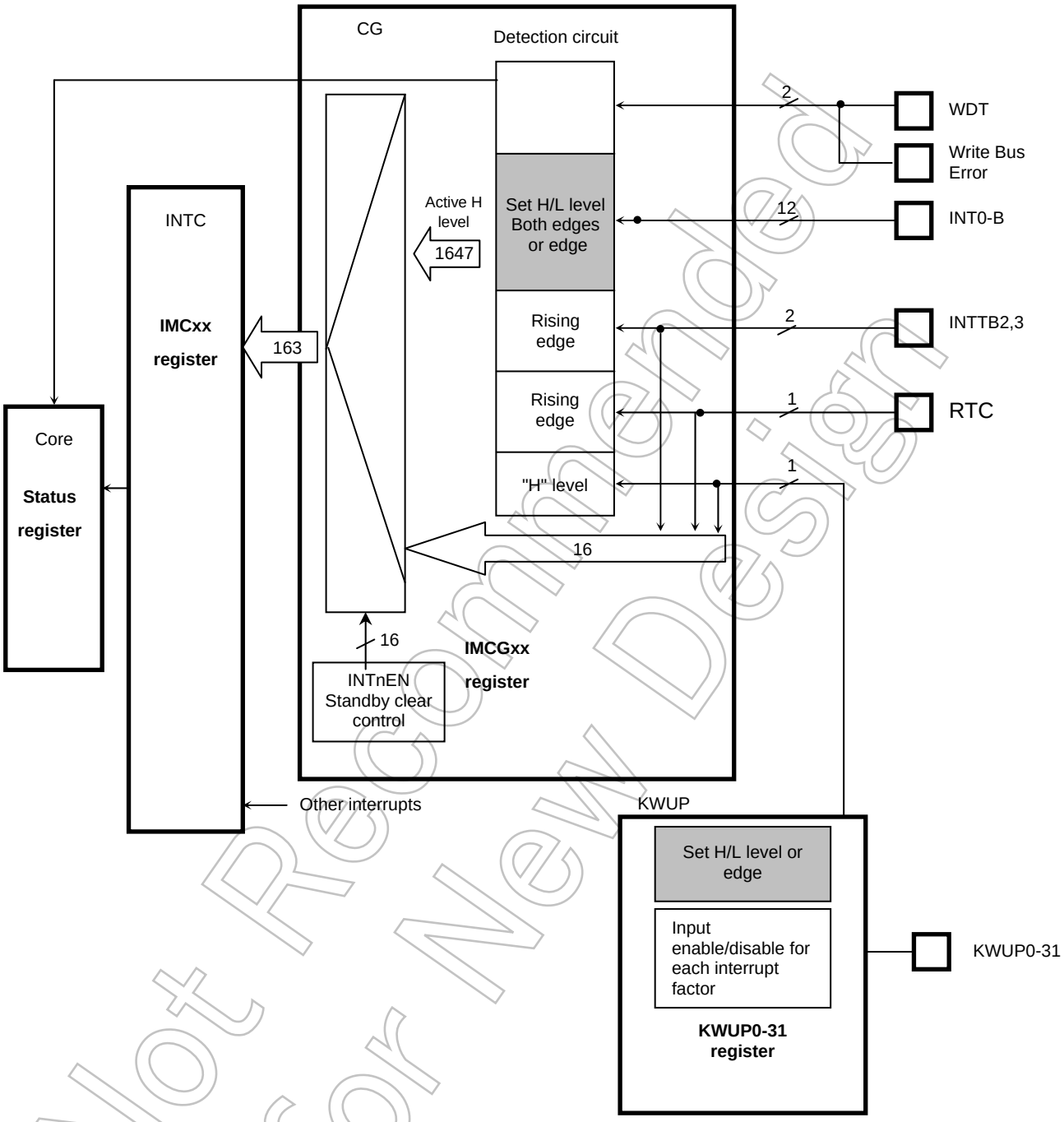


Fig. 6-6 Interrupt Connection Diagram

6.9 INTCG Registers (Interrupts to Clear STOP, SLEEP, and IDLE)

INT0 to INTB, KWUP0 to 31 (Interrupts to Clear Stop, Sleep, and Idle modes)
INTRTC, INTTB2, 3 (Two-phase pulse input counter): Sleep

IMCGA (0xFFFF_EE10)

	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	Always reads "0."	Set active state of INT0 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT0 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT1 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT1 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT2 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT2 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT3 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT3 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	INT3 Clear input 0: Disable 1: Enable

IMCGB (0xFFFF_EE14)

	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	Always reads "0."	Set active state of INT4 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT4 standby clear request 00 – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT5 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT5 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT6 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT6 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT7 standby clear request. 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			Active status of INT7 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	INT7 Clear input 0: Disable 1: Enable

IMCGC (0xFFFF_EE18)

	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	Always reads "0."	Set active state of INT8 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge 100: Both edges			Active status of INT8 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INT9 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge 100: Both edges			Active status of INT9 standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	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	Always reads "0."	Set active state of INTA standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge 100: Both edges			Active status of INTA standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	INT9 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	Always reads "0."	Set active state of INTB standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge 100: Both edges			Active status of INTB standby clear request 00: – 01: Rising edge 10: Falling edge 11: Both edges		Always reads "0."	INTB Clear input 0: Disable 1: Enable

IMCGD (0xFFFF_EE1C)

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

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

	23	22	21	20	19	18	17	16
bit Symbol			EMCGE1	EMCGE0				INTTB2EN
Read/Write	R	R/W	R/W		R			R/W
After reset	0	0	1	0	—			0
Function	Always reads "0."	Be sure to write "0."	Set active state of INTTB2 standby clear request. 11: Rising edge **Be sure to set "11."**		The lead value is irregular.		Always reads "0."	INTTB2 Clear input 0: Disable 1: Enable

	31	30	29	28	27	26	25	24
bit Symbol			EMCGF1	EMCGF0				INTTB3EN
Read/Write	R	R/W	R/W		R			R/W
After reset	0	0	1	0	—			0
Function	Always reads "0."	Be sure to write "0."	Set active state of INTTB3 standby clear request. 11: Rising edge **Be sure to set "11."**		The lead value is irregular.		Always reads "0."	INTTB3 Clear input 0: Disable 1: Enable

Note: Default values of EMCGx0 and EMCGx1 are different from the values to be used. Properly set them to the specified values before use.

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

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

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

(Note 3) For clearing the Stop and Sleep modes with TMP19A43, 16 factors, i.e., INT0 to INTB, INTRTC, INTTB2/INTTB3, and KWUP00 to 31 are available as clearing interrupts. Whether or not INT0 to INTB are to be used as clearing interrupts as well as active state edge/level selection is set with CG. Whether or not KWUP00 to 31 are to be used as STOP/SLEEP/IDLE clearing interrupts is set with CG and active state edge/level selection is set with KWUPSTn <KEYn>. Set to High level with INTC for the above 16 factors.

(Note 4) Among the above 16 factors to be assigned as Stop/Sleep/Idle clear request interrupts, INT0 to INTB don't have to be set with CG if they are to be used as normal interrupts. Use INTC to specify either H/L level, rising/falling edge, or both edges. If KWUP00 to 31 are to be used as normal interrupts, set the active level by KWUPSTn and set High level with INTC. No CG setting is necessary. Also, if INTRTC is to be used as a normal interrupt, use CG/INTC for the setting.

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

EICRCG
(0xFFFF_EE20)

	7	6	5	4	3	2	1	0
bit Symbol					ICRCG3	ICRCG2	ICRCG1	ICRCG0
Read/Write	R				R/W			
After reset	0				0			
Function	Always reads "0."				Always reads "0." Clear interrupt requests. 0000: INT0 0101: INT5 1010: INTA 0001: INT1 0110: INT6 1011: INTB 0010: INT2 0111: INT7 1100: KWUP 0011: INT3 1000: INT8 1101: INTRC 0100: INT4 1001: INT9 1110: INTTB2 1111:INTTB3			

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

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

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

(Note) To clear interrupt request of the above 16 factors that are assigned to clear Stop/Sleep/Idle modes,

① For KWUP, use KWUPCLR

② For INT0 to INTB, ,INTRTC,INTTB2,INTTB3 use the EIRCG register in the above CG block.

6.10 NMI Flag Register

NMIFLG (0xFFFF_EE24)

	7	6	5	4	3	2	1	0
bit Symbol							WDT	WBER
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."						NMI factor 1: NMI generated by WDT interrupt	NMI factor 1: NMI generated by write bus error

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

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

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

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

Although TMP19A43 doesn't have NMI interrupts as external pin inputs, NMI interrupts are available as internal interrupt factors.

6.11 Cautions in Using Interrupts

The following paragraphs describe some points to be kept in mind in using interrupts. User programs must be written in a manner to satisfy the following details.

6.11.1 Cautions Related to TX19A Processor Core

- Exceptions cannot be disabled. Note that there are some cases where two different instructions can be distinguished only by exception generation. So, properly use them according to the specific usage.
- Software interrupts are different from the "software set" to be used as one of hardware interrupt factors.
- Immediately after overwriting SSCR of the CP0 register, add two NOP instructions to allow for register bank switching as it takes two clock cycles.
- In case multiple interrupts of the same interrupt level are accepted by changing ILEV <CMASK>, it is necessary for the user program to save because the register bank will not be switched.
- Only 32-bit ISA access can be used to access IER of the CP0 register.
- Different stack pointers (r29) are used for Shadow Register Set number 0 and Shadow Register Set numbers 1 to 7; it is necessary to set them separately (twice). If it is desired to use a common stack pointer, you can do so by setting SSCR<CSS> to "1" in the main process to use Shadow Register Set number 1. In this case, when a level 1 interrupt is accepted, it is necessary for the user program to save because the register bank will not be switched.
- If an ERET instruction is executed while interrupts are disabled by setting Status <ERL> of the CP0 register to "1," it returns to the main process by using ErrorEPC of the CP0 register as the return address. As the TX19A processor core saves the interrupt return address to EPC, you should be careful if Status <ERL> is to be used for disabling interrupts.
- Don't execute an ERET instruction within two clock cycles after accessing Status, ErrorEPC, EPC, or SSCR of the CP0 register.
- If Status <ERL/EXL/IE> of the CP0 register is set to disable interrupts, interrupts are disabled at the time of instruction execution (E stage) but any value set to the register is reflected only two clocks later.
- If Status <ERL/EXL/IE> of the CP0 register is set to enable interrupts, interrupts are enabled two clocks after the instruction execution (E stage); any value set to the register is also reflected two clocks after the instruction execution (E stage).

6.11.2 Cautions Related to INTC

- If more than one interrupts of a same interrupt level are generated at the same time, interrupts are accepted from the factor of the smallest interrupt number.
- Any factor of interrupt level 0 is not suspended.
- If it is desired to individually disable interrupt factors (by setting interrupt level 0), you can do so only while interrupts are disabled.
- Default settings of IMCx <EIMxx> of INTC may be different from the settings to be used.
- The INTC ILEV register must be 32-bit accessed.
- The INTC INTCLR register must be 32-bit accessed.
- When an interrupt request is cleared by INTCLR before reading INTC IVR, the interrupt factor cannot be determined because the IVR value is cleared.
- When enabling interrupts, be sure to do so in the order of the detection route (from external to internal). When disabling, use the reverse order of the detection route (from internal to external).
- When a new value is written to INTC ILEV <CMASK>, set <MLEV> to "1" at the same time.

7. Input/Output Ports

7.1 Port 0 (P00 through P07)

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

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

If the BUSMD pin (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).

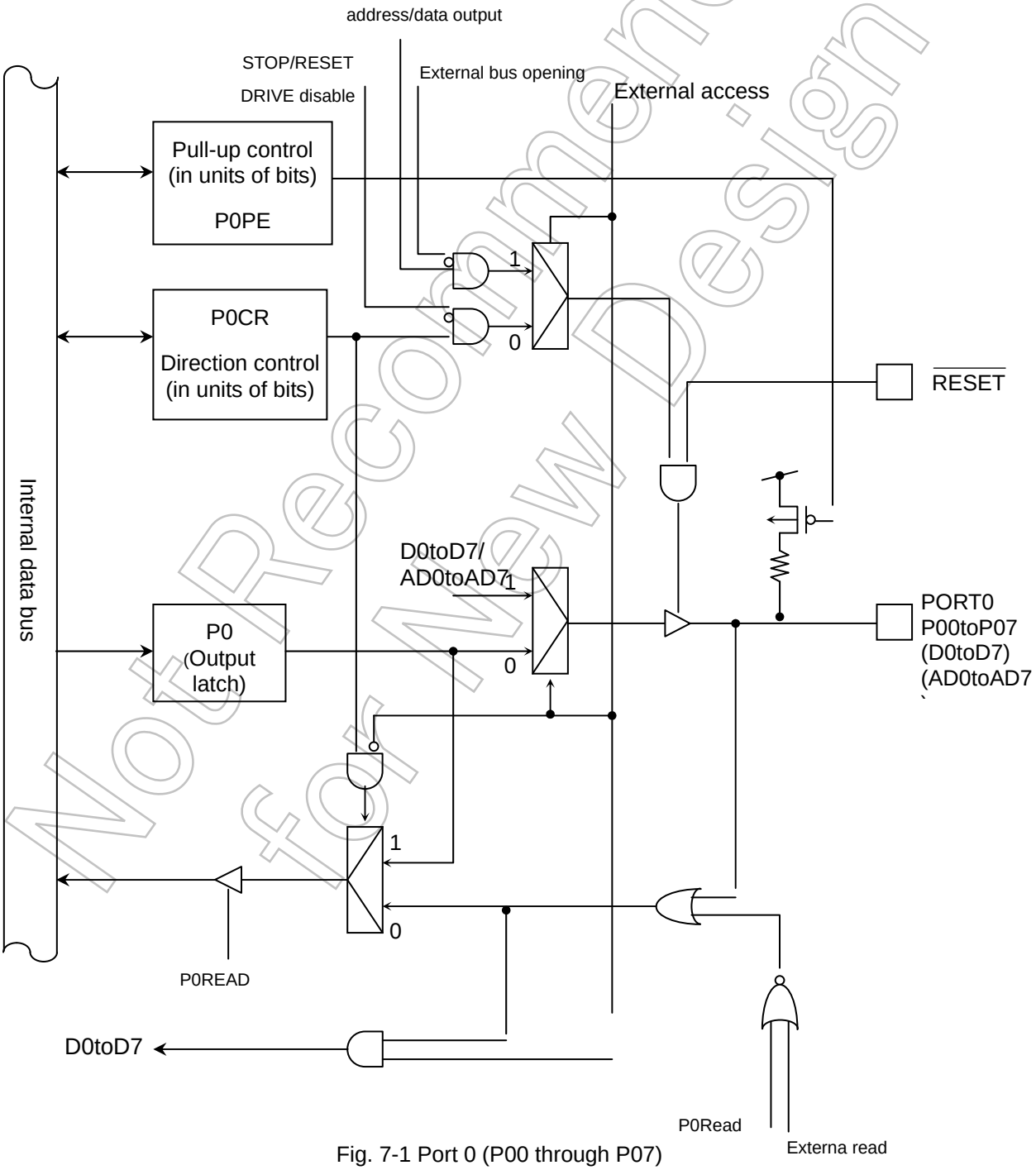


Fig. 7-1 Port 0 (P00 through P07)

Port 0 register

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

Port 0 control register

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

Port 0 Pull-up control register

		7	6	5	4	3	2	1	0
P0PE	Bit Symbol	PE07	PE06	PE05	PE04	PE03	PE02	PE01	PE00
(0xFFFF_F00C)	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 register P1FC. A reset allows all bits of the output latch P1, P1CR and P1FC to be cleared to "0" and the port 1 to be put in input mode.

Besides the general-purpose input/output function, the port 1 performs other functions: D8 through D15 function as a data bus, AD8 through AD15 function as an address data bus, and A8 through A15 function as an address bus. To access external memory, 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).

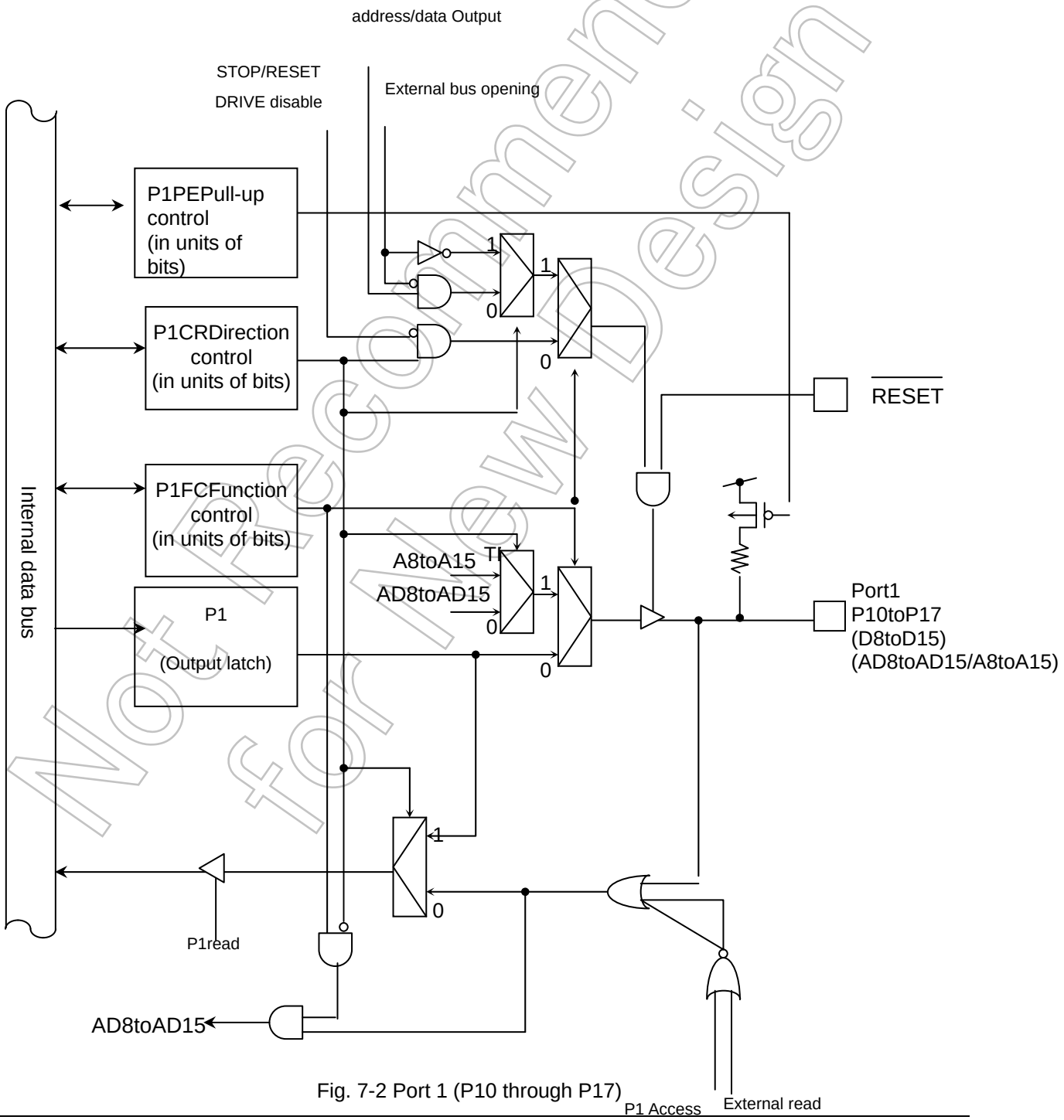


Fig. 7-2 Port 1 (P10 through P17)

Fig. 7-3 Port 1 (P10 through P17) Port 1 register

P1 (0xFFFF_F001)

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

Port 1 control register

P1CR (0xFFFF_F004)

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

Port 1 function register

P1FC (0xFFFF_F005)

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

Port 1 Pull-up control register

P1PE (0xFFFF_F00D)

	7	6	5	4	3	2	1	0
Bit Symbol	PE17	PE16	PE15	PE14	PE13	PE12	PE11	PE10
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

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

Table 7-1

7.3 Port 2 (P20 through P27)

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

The port 2 also performs a 16-bit timer input function. This function is enabled by setting the corresponding bits of P2FC and P2FC2 to "1" and the corresponding bit of P2CR to "0." A reset allows P2CR and P2FC 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).

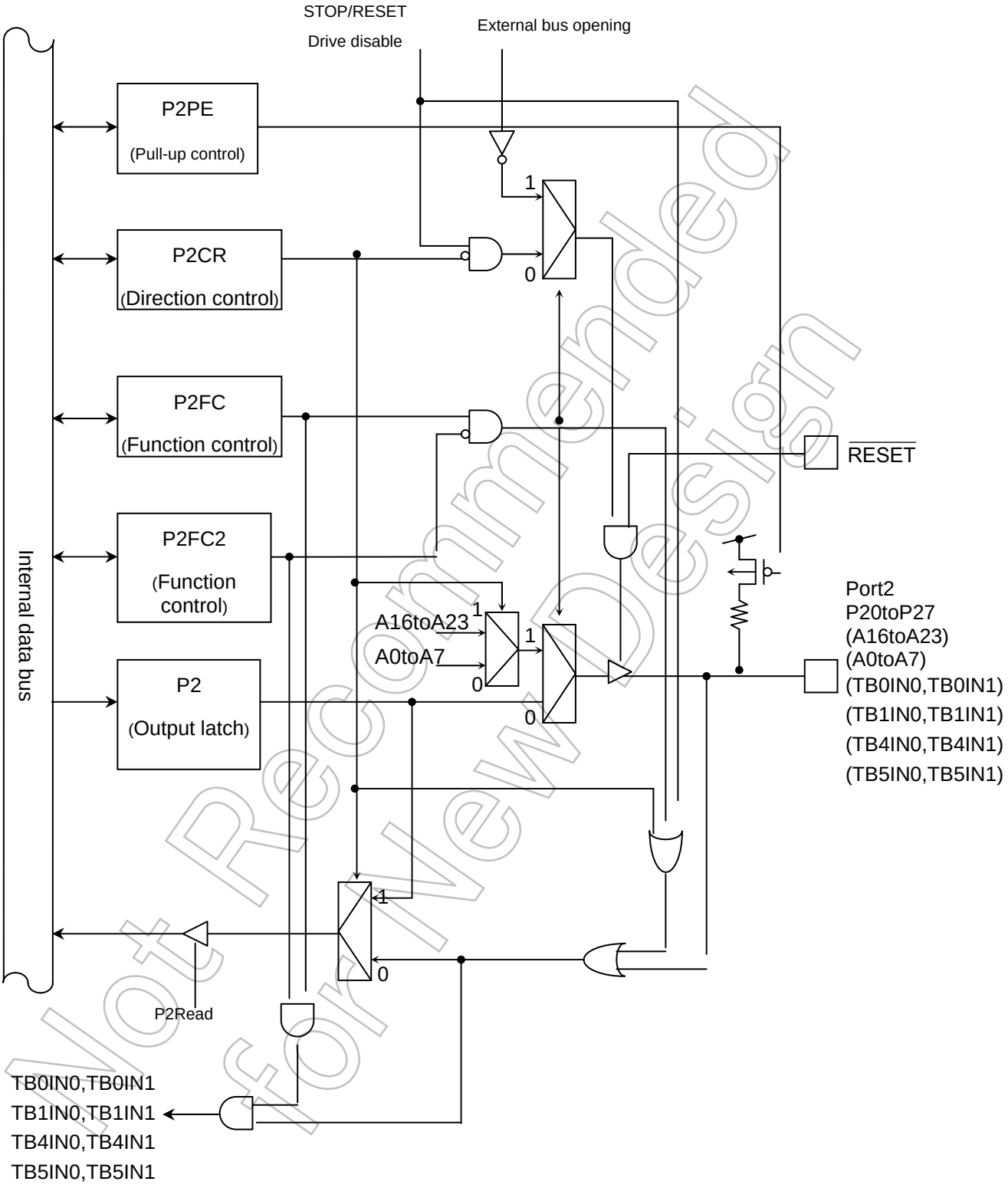


Fig. 7-3 Port 2 (P20 through P27)

Port 2 register

P2 (0xFFFF_F012)

	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 (0xFFFF_F014)

	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	<<See P2FC>>							

Port 2 function register 1

P2FC (0xFFFF_F015)

	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 : Function	0 : Port 1 : Function	0 : Port 1 : Function	0 : Port 1 : Function	0 : Port 1 : Function	0 : Port 1 : Function	0 : Port 1 : Function	0 : Port 1 : Function

Port 2 function register 2

P2FC2 (0xFFFF_F016)

	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: Address 1: TB5IN1	0: Address 1: TB5IN0	0: Address 1: TB4IN1	0: Address 1: TB4IN0	0: Address 1: TB1IN1	0: Address 1: TB1IN0	0: Address 1: TB0IN1	0: Address 1: TB0IN0

Port 2 pull-up control register

P2PE (0xFFFF_F01C)

	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

Function	Corresponding BIT of P2FC	Corresponding BIT of P2FC2	Corresponding BIT of P2CR	PORT to be used
POR2 input setting	0	*	0	PORT2
POR2 output setting	0	*	1	PORT2
Address bus (A7 through A0) output setting (*1)	1	0	0	PORT2
Address bus (A23 through A16) output setting (*1)	1	0	1	PORT2
TMRB input setting	1	1	0	PORT2

Table 7-2

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

7.4 Port 3 (P30 through P37)

The port 3 is a general-purpose, 8-bit input/output port (P30 and P31 are used exclusively for output). For this port, inputs and outputs can be specified in units of bits by using the control register P3CR and the function register P3FC. A reset allows the output latches P30 and 31 to be set to "1."

Besides the input/output port function, the port 3 performs other functions: P34 outputs a 16-bit timer, and P35, P36 and P37 perform the 32-bit capture and trigger input function. These functions are enabled by setting the corresponding bit of P3FC to "1." A reset allows P3CR and P3FC to be cleared to "0" and the port 3 to function as an input port.

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 $\overline{RD}$ signal output mode (<P30F>="1"), the $\overline{RD}$ strobe is output only when an external address area is accessed. Likewise, if the P31 pin is set to $\overline{WR}$ signal output mode (<P31F>="1"), the $\overline{WR}$ strobe is output only when an external address area is accessed.

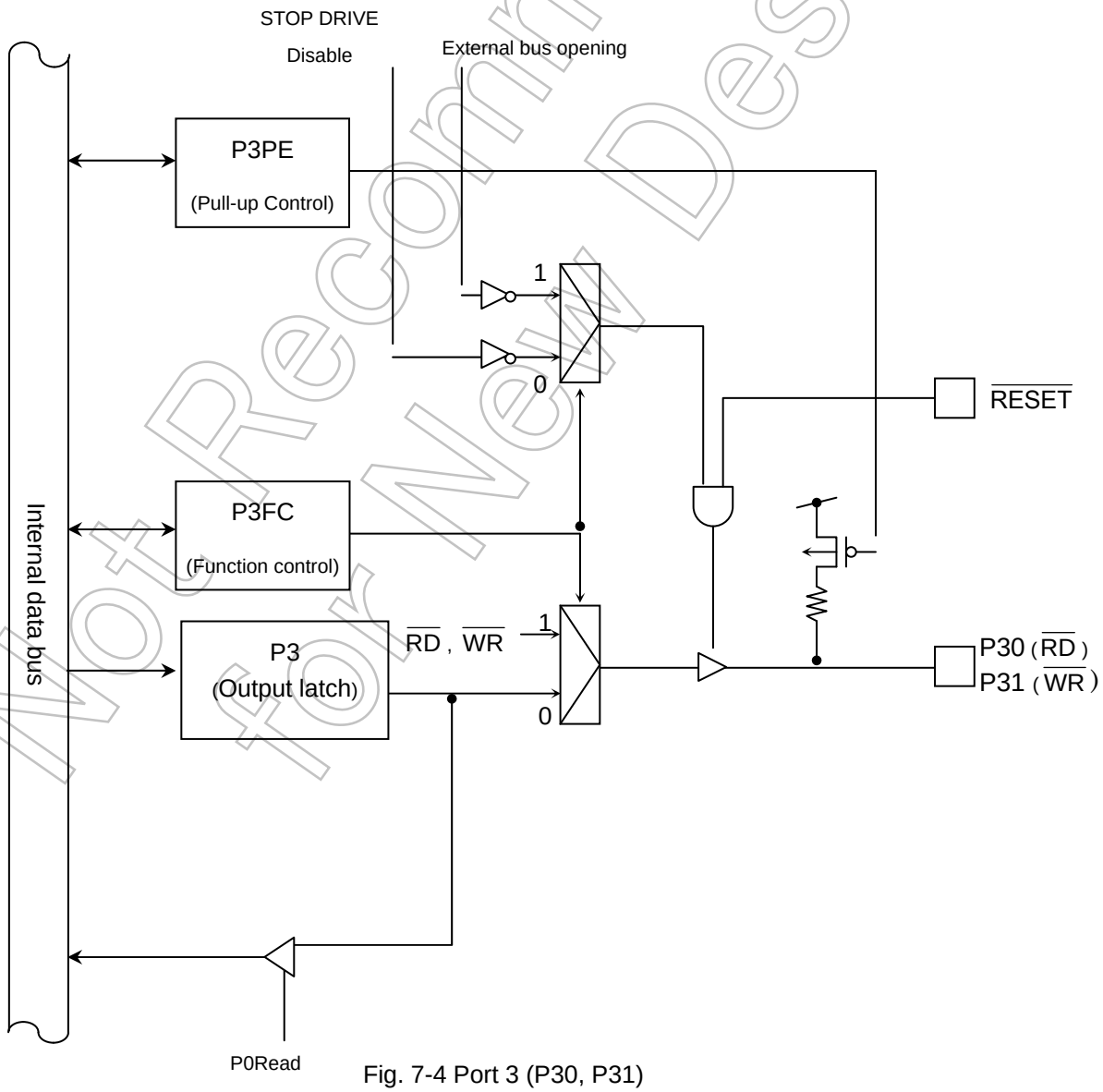


Fig. 7-4 Port 3 (P30, P31)

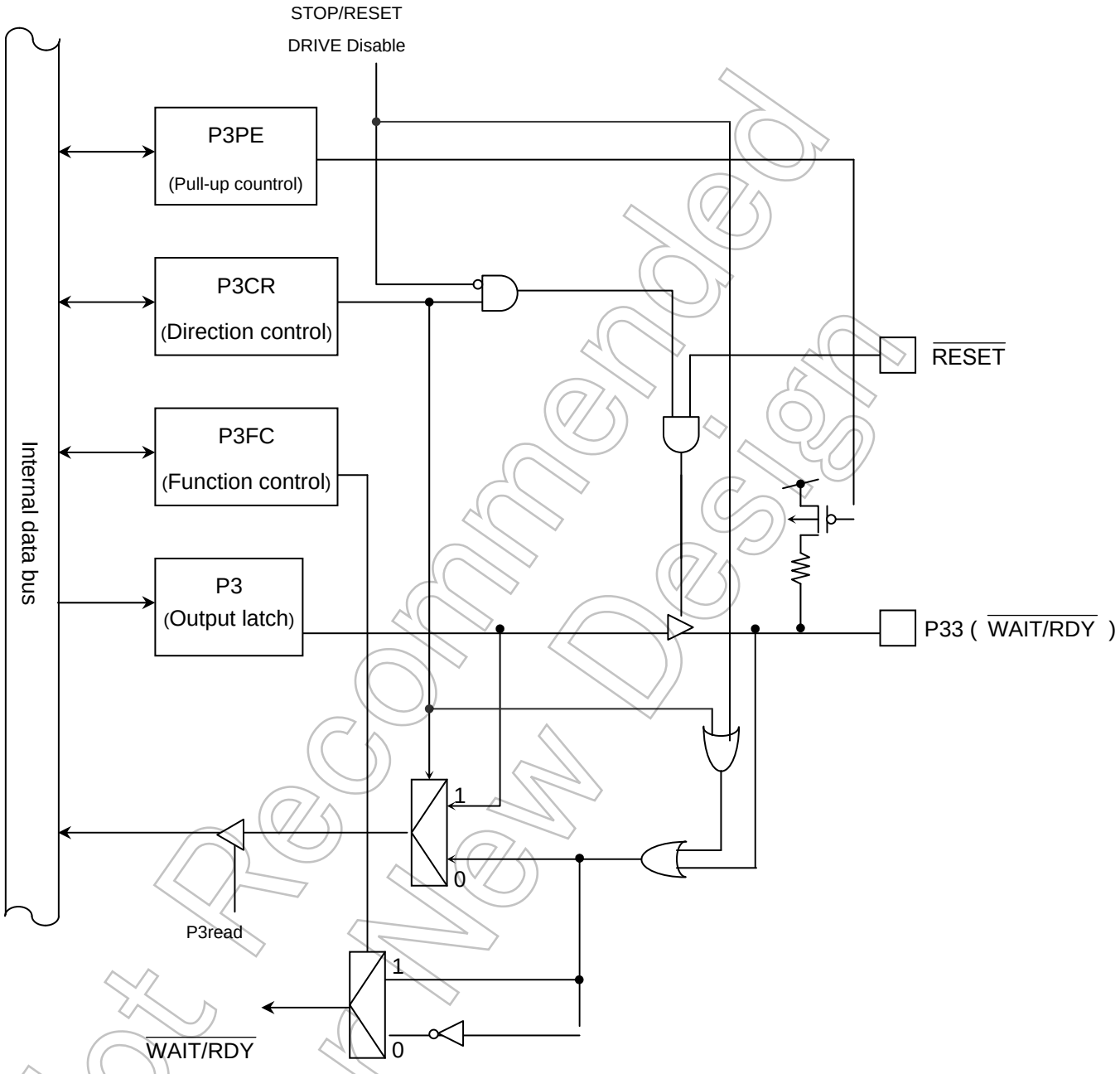


Fig. 7-5 Port 3 (P33)

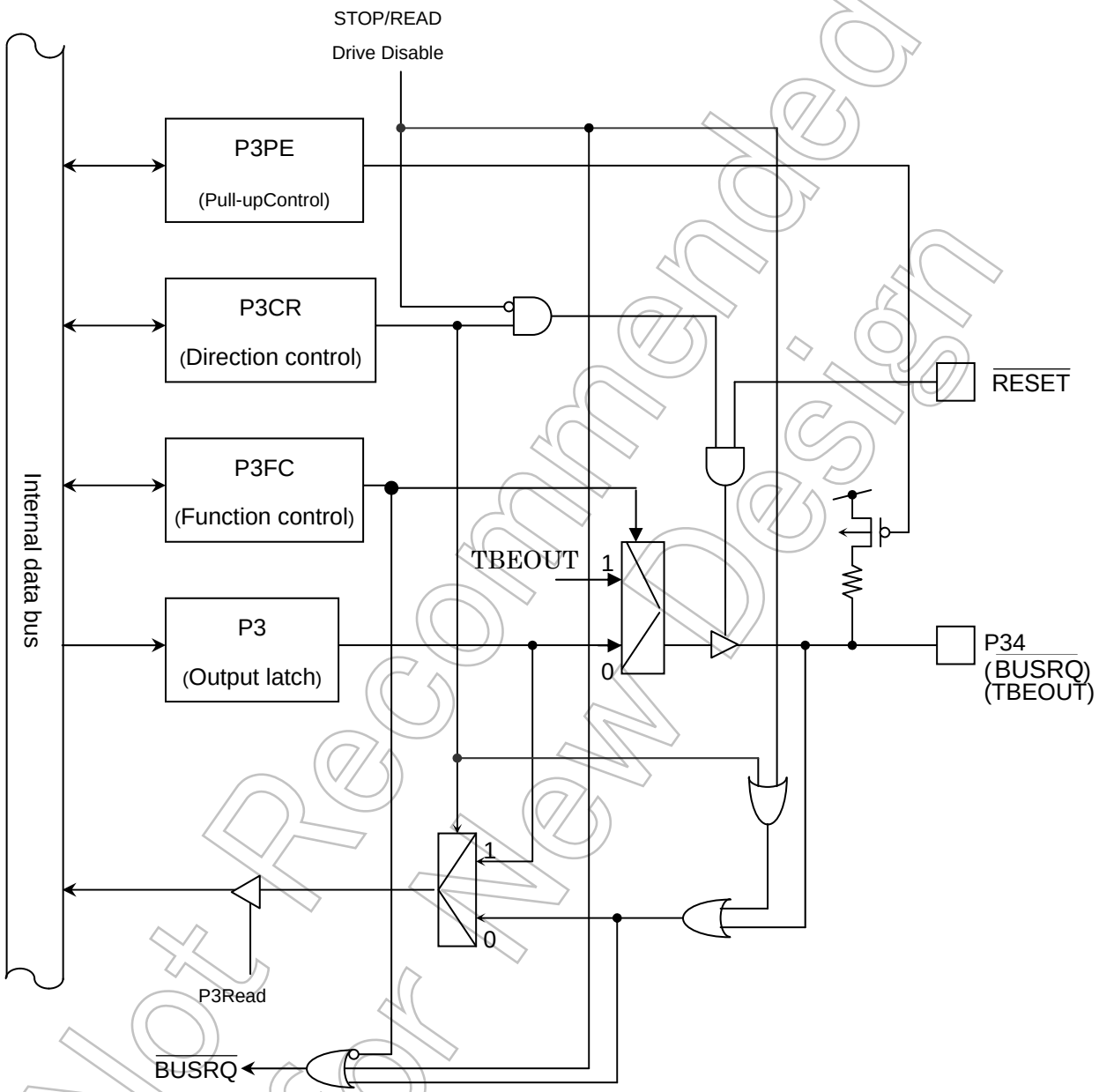


Fig. 7-6 Port 3 (P34)

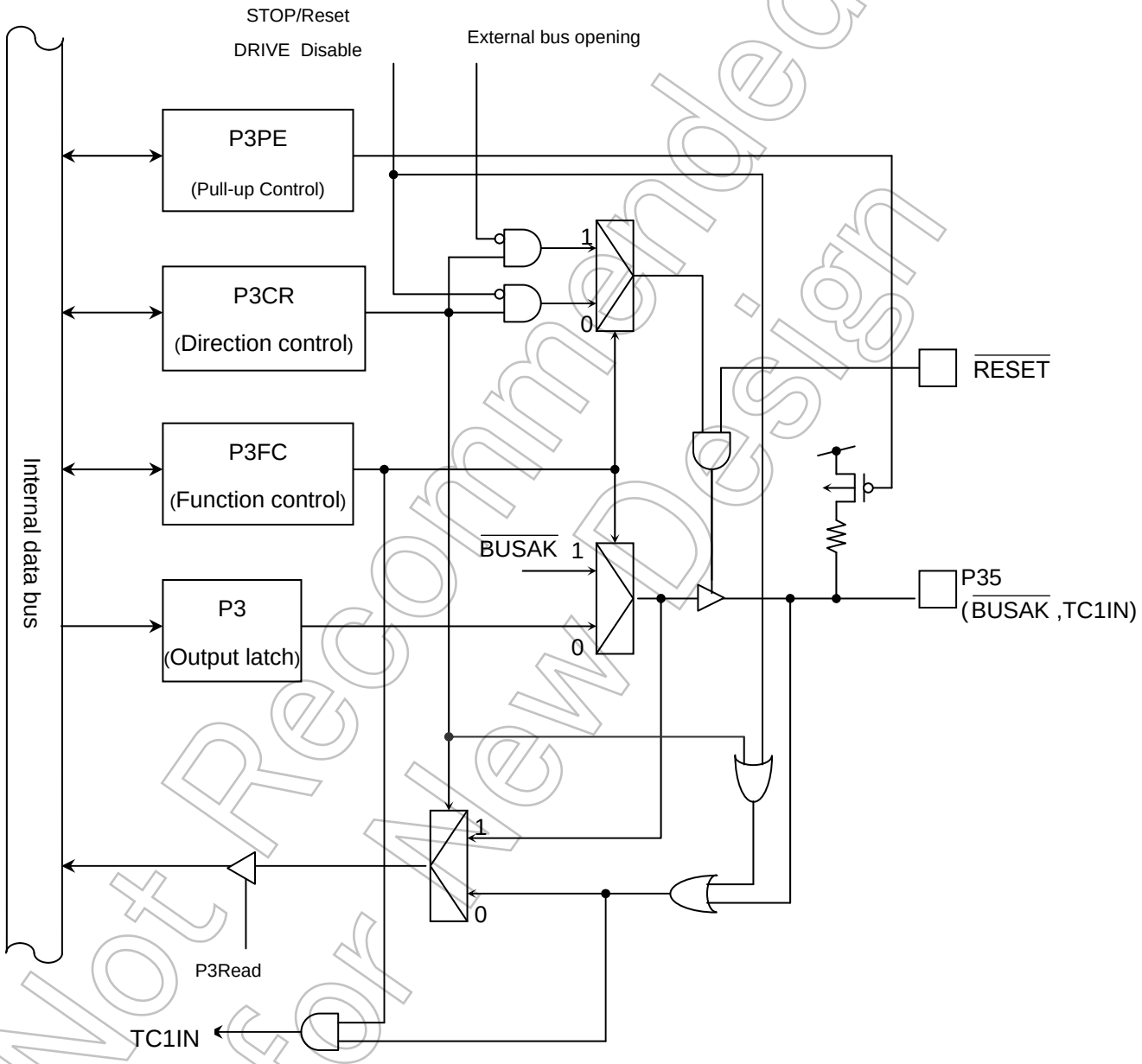


Fig. 7-7 Port 3 (P35)

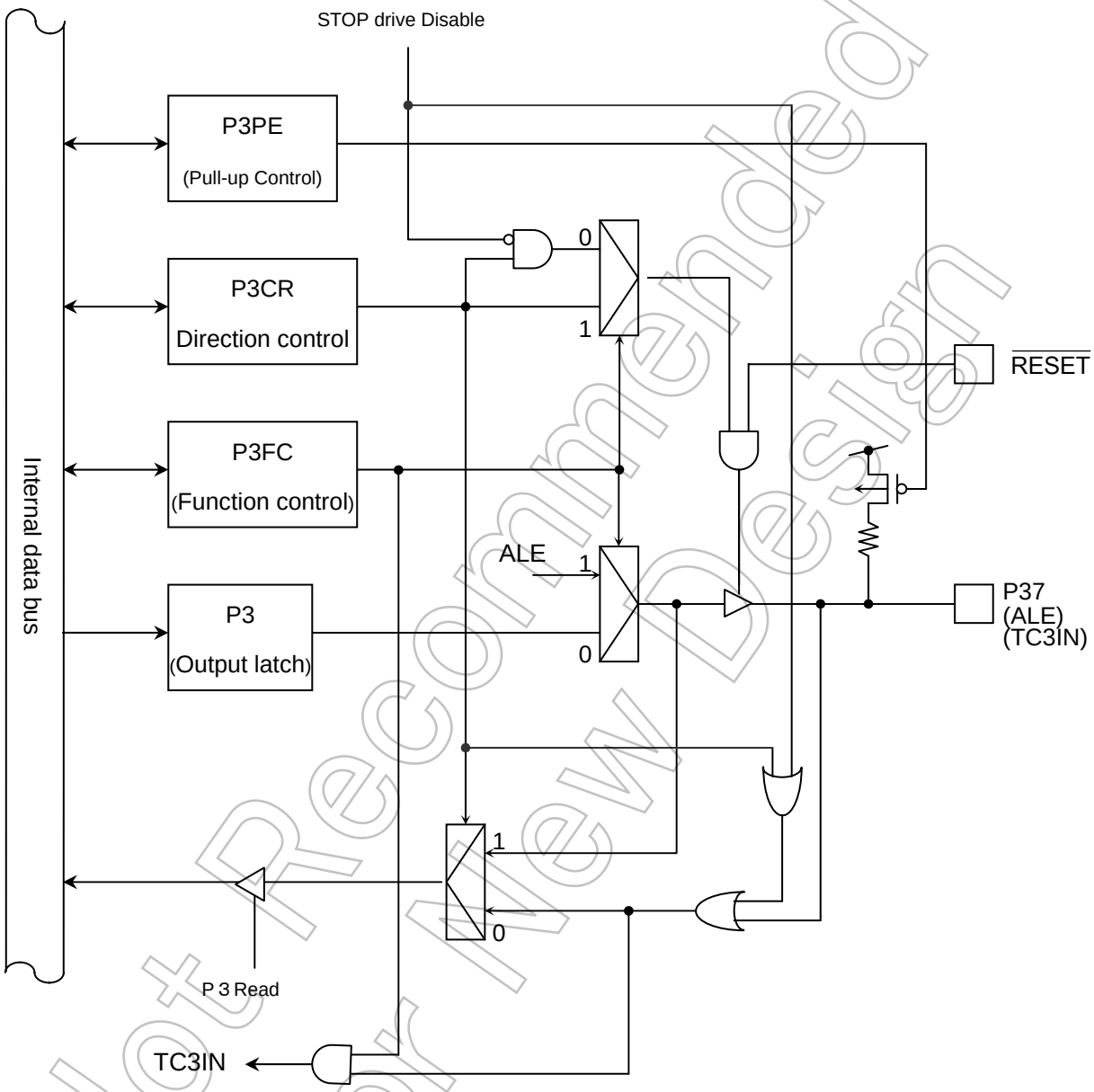


Fig. 7-8 Port 3 (P37)

Port 3 register

P3 (0xFFFF_F018)

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

Port 3 control register

P3CR (0xFFFF_F01A)

	7	6	5	4	3	2	1	0
Bit Symbol	P37C	P36C	P35C	P34C	P33C	P32C	–	–
Read/Write	R/W							
After reset	According to the bus mode	0	0	0	0	0	0	0
Function		0: Input 1: Output						

Port 3 function register 1

P3FC (0xFFFF_F01B)

	7	6	5	4	3	2	1	0
Bit Symbol	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: ALE/ TC3IN	0: PORT 1: R/$\overline{W}$	0: PORT 1: BUSAK	0: PORT 1: BUSRQ	0: PORT/ $\overline{WAIT}$ 1: PORT/ $\overline{RDY}$	0: PORT 1: $\overline{HWR}$ / TC0IN	0: PORT 1: $\overline{WR}$	0: PORT 1: $\overline{RD}$

Port 3 pull-up control register

P3PE (0xFFFF_F01D)

	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 to be used	Function	Corresponding BIT of P3FC	Corresponding BIT of P3CR	BUSMD
P30/P31	P30/31output setting	0	—	—
	RD/WRoutput setting	1	—	—
P32/P36	P32/P36input setting	*	0	—
	P32/P36output setting	0	1	—
	TC0IN/TC2ININput setting	1	0	—
	HWR/ R/WOutput setting	1	1	—
P33	P33input setting	*	0	—
	P33output setting	*	1	—
	WAITinput setting	0	0	—
	RDYinput setting	1	0	—
P34	P34input setting	*	0	—
	P34output setting	0	1	—
	BUSRQinput setting	1	0	—
	TBEOUTOutput setting	1	1	—
P35	P35input setting	*	0	—
	P35output setting	0	1	—
	TC1IN input setting	1	0	—
	BUSAK output setting	1	1	—
P37	P37input setting	*	0	—
	P37output setting	0	1	—
	TC3IN input setting	1	0	H
	ALE output setting	1	1	H

Table 7-3

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

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

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 ($\overline{CS0}$ to $\overline{CS3}$) 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.

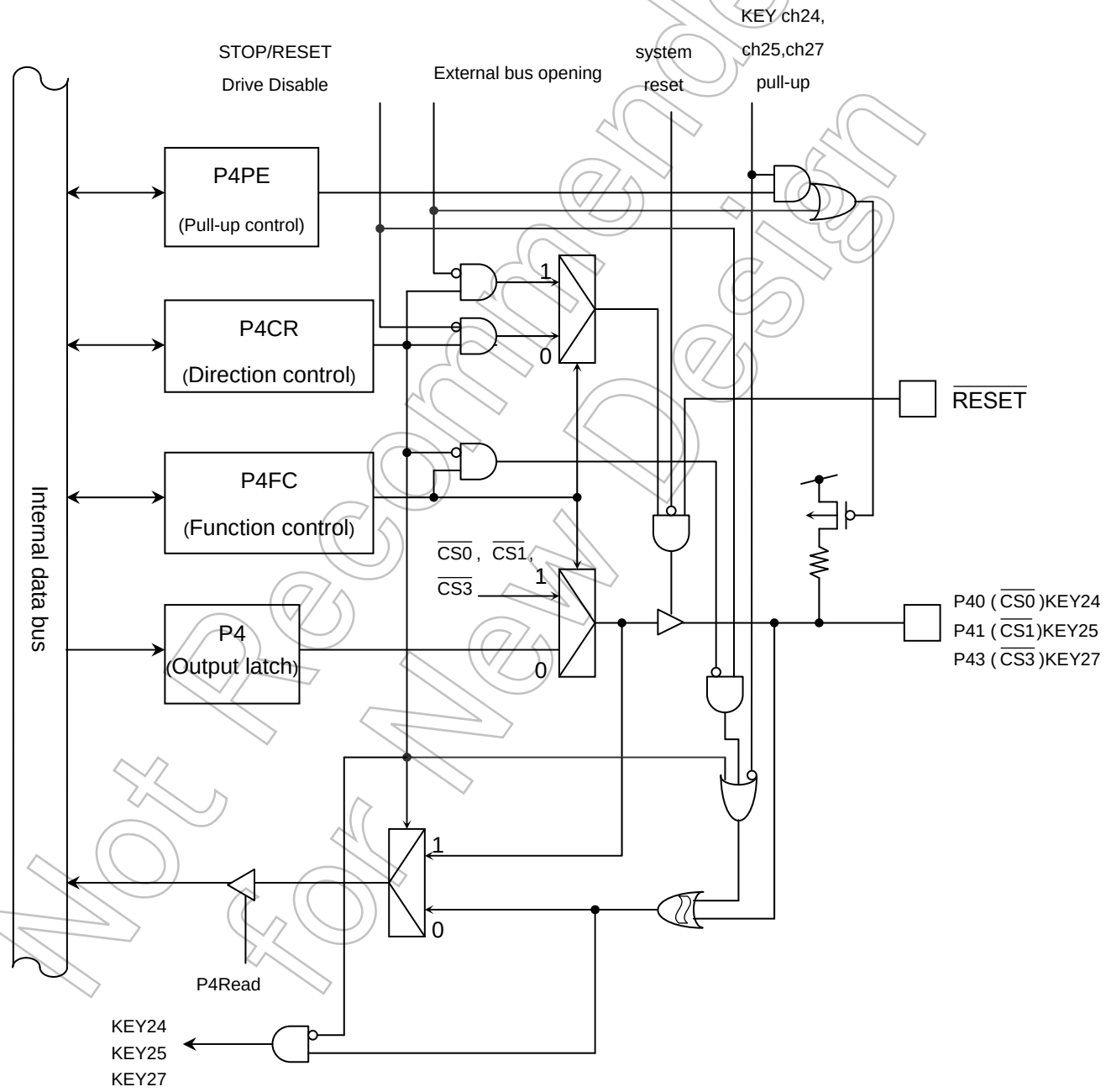


Fig. 7-9 Port 4 (P40 to P43)

If the port 4 goes into STOP mode when the KEY input is enabled, inputs are always accepted. To inhibit inputs, switch to PORT using the function register.

- Port: Inputs are accepted only during a read.
- KEY: Inputs are always accepted.

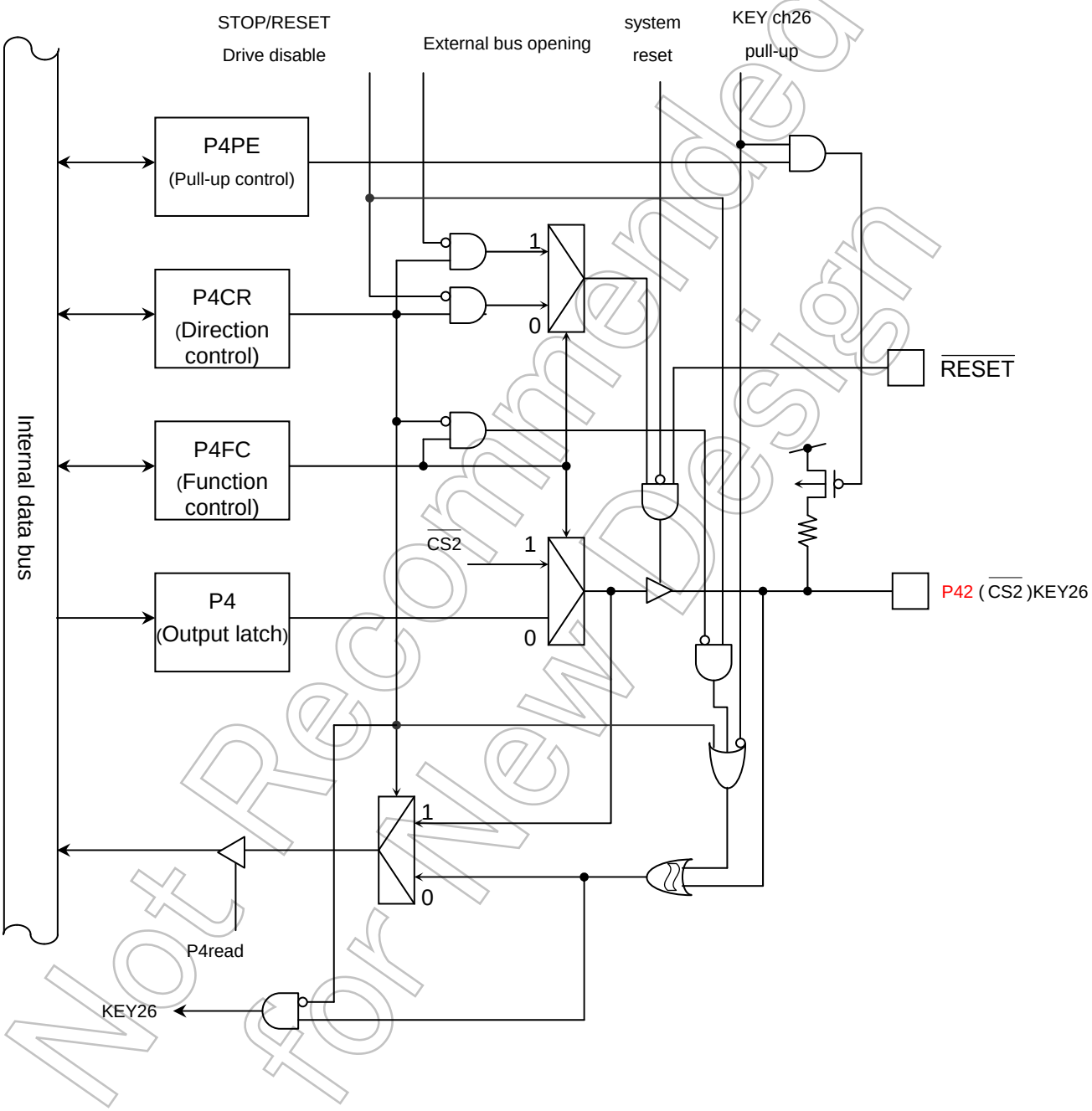


Fig. 7-10Port 4 (P42)

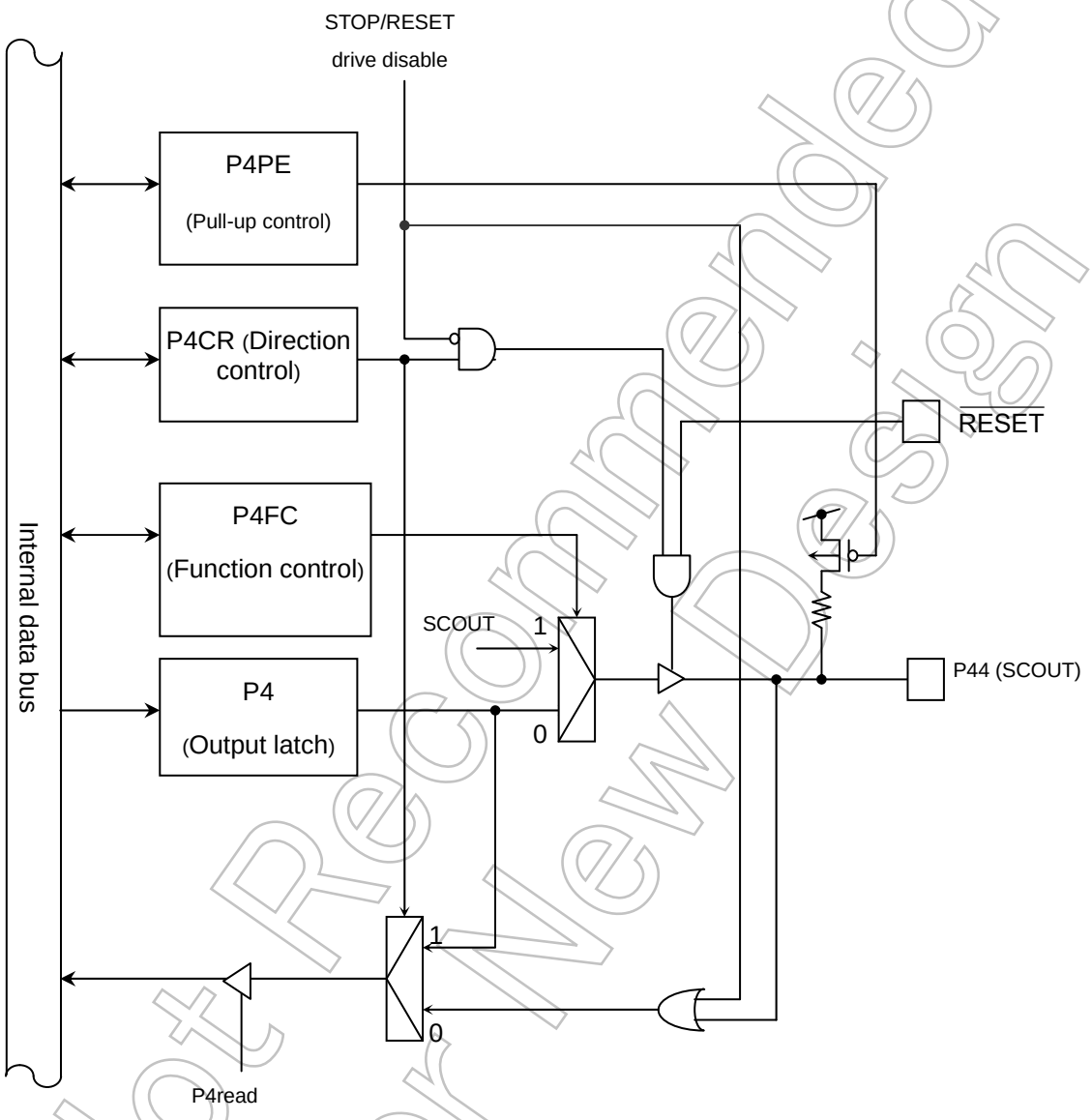


Fig. 7-11Port 4 (P44)

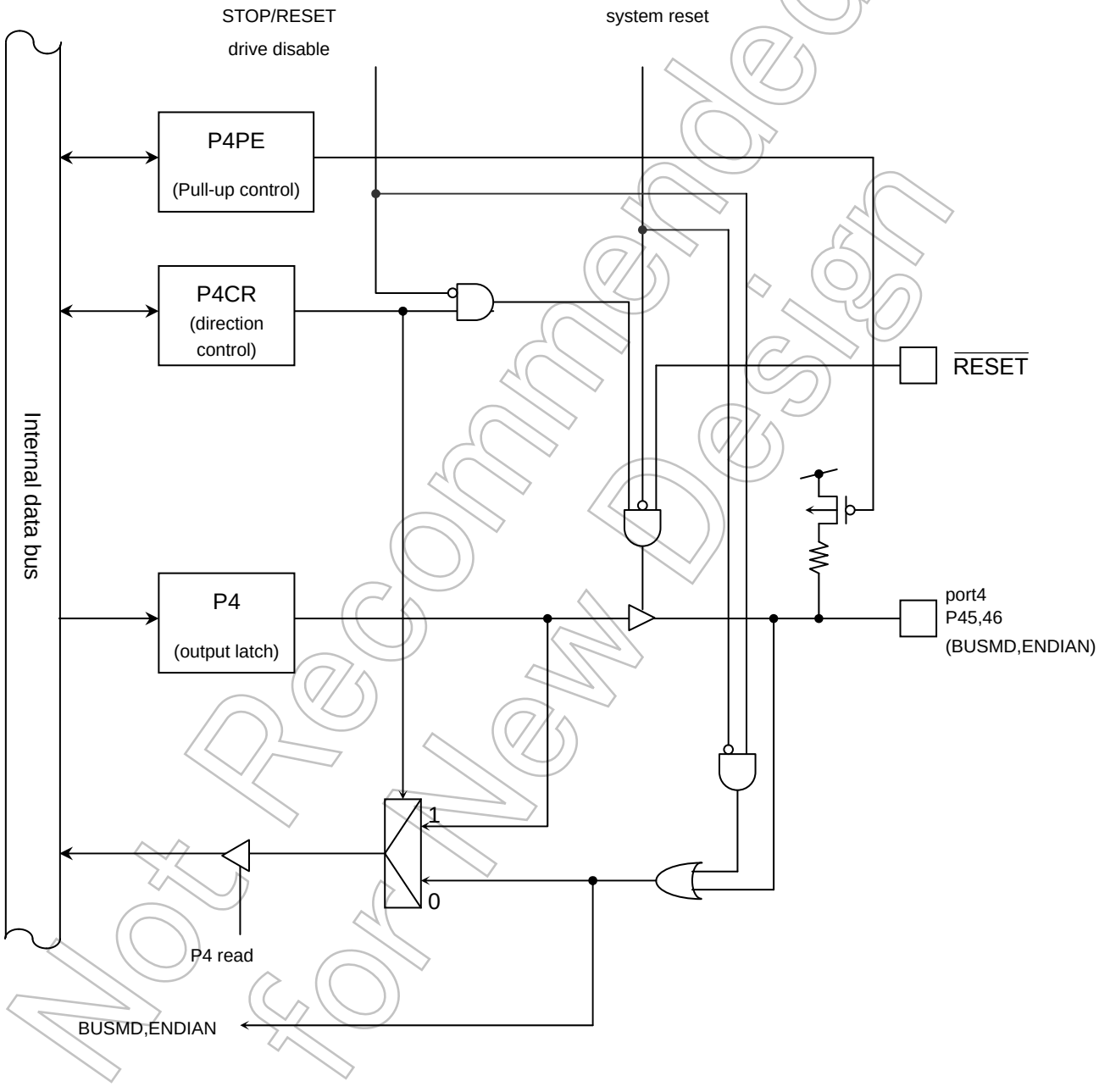


Fig. 7-12 Port 4 (P45,46)

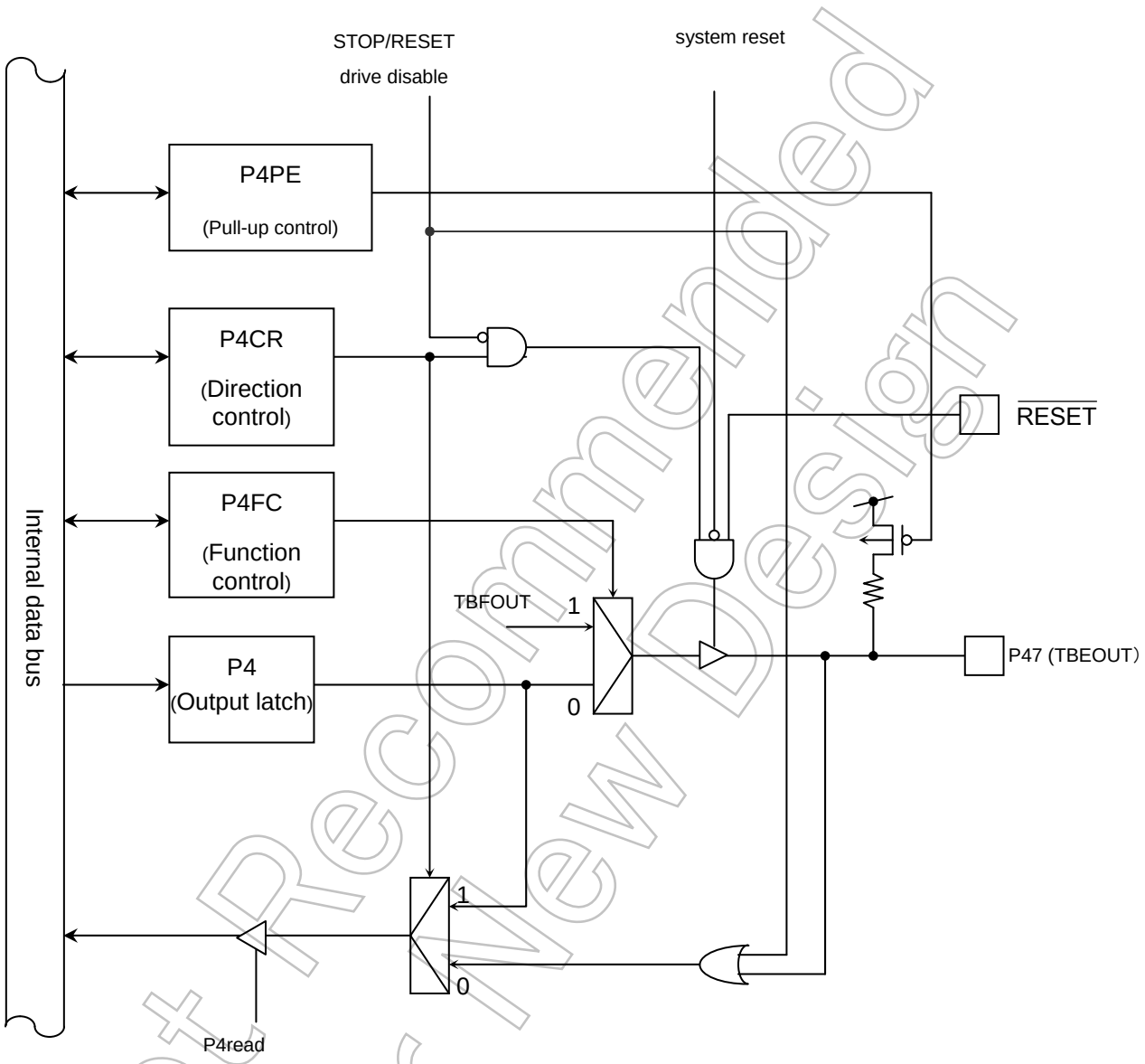


Fig. 7-13 Port 4 (P47)

Port 4 register

		7	6	5	4	3	2	1	0
P4	Bit Symbol	P47	P46	P45	P44	P43	P42	P41	P40
(0xFFFF_F01E)	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port 4 control register

		7	6	5	4	3	2	1	0
P4CR	Bit Symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
(0xFFFF_F020)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
		0: Input 1: Output							

Port 4 function register

		7	6	5	4	3	2	1	0
P4FC	Bit Symbol	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
(0xFFFF_F021)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: TBFOUT	Write 0	Write 0	0: PORT 1: SCOUT	0: PORT /KEY27 1: CS3 /KEY27	0: PORT/ KEY26 1: CS2 /KEY26	0: PORT/ KEY25 1: CS1 /KEY25	0: PORT/ KEY24 1: CS0 /KEY24

Port 4 pull-up control register

		7	6	5	4	3	2	1	0
P4PE	Bit Symbol	PE47	PE46	PE45	PE44	PE43	PE42	PE41	PE40
(0xFFFF_F025)	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 to be used	Function	Corresponding BIT of P4FC	Corresponding BIT of P4CR
P40/P41/P43	P40/P41/P43input setting	0	0
	P40/P41/P43output setting	0	1
	KEY24/25/27input setting	*	0
	CS0/CS1/CS3Output setting	1	1
P42	P42input setting	0	0
	P42output setting	0	1
	KEY26input setting	*	0
	CS2Output setting	1	1
P44	P44input setting	*	0
	P44output setting	0	1
	SCOUT setting	1	1
P45/P46	P45/P46input setting	-	0
	P45/P46output setting	–	1
	BUSMD/ENDIANinput setting	-	*
P47	P47input setting	*	0
	P47output setting	0	1
	TBEOUTOutput setting	1	1

Table 7-4

7.6 Port 5 (P50 through P57)

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

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." A reset allows P5CR and P5FC to be cleared to "0" and the port 5 to function as an input port.

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.)

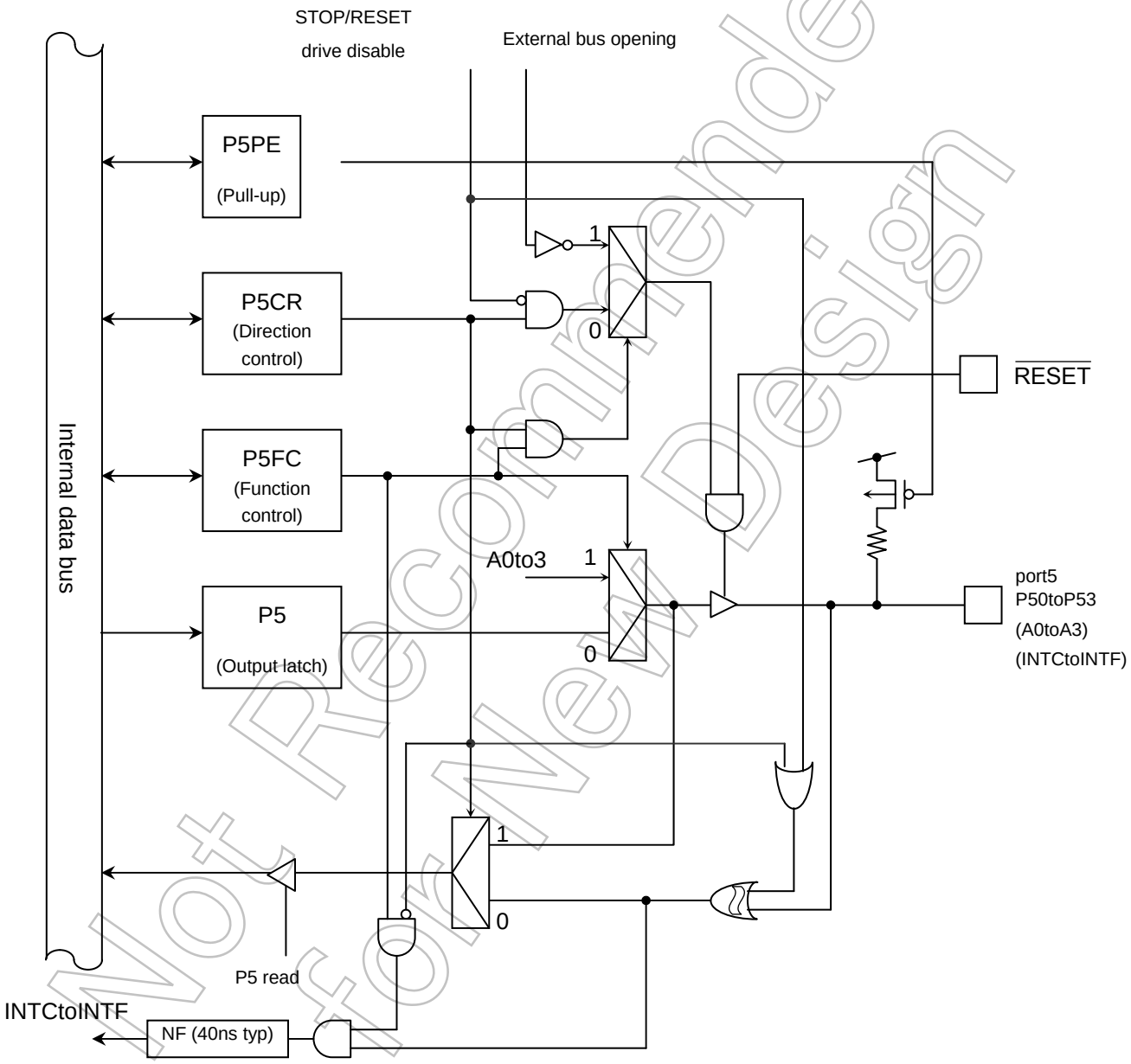


Fig. 7-14 Port 5 (P50 to P53)

If the port 5 goes into STOP mode when the INT input is enabled, inputs are always accepted.

To inhibit inputs, switch to PORT using the function register.

- **Port: Inputs are accepted only during a read.**
- **INT: Inputs are always accepted.**

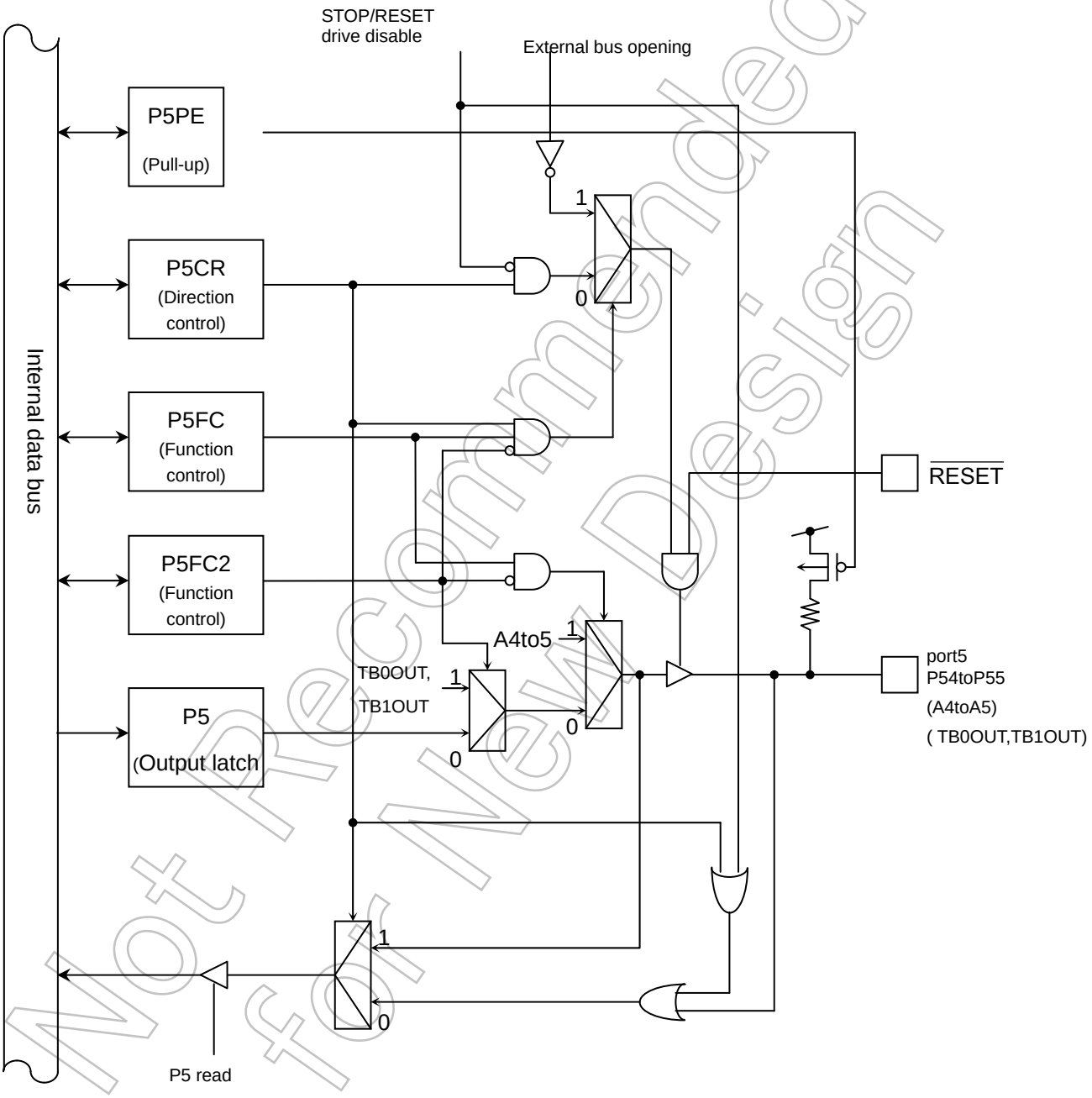


Fig. 7-15 Port 5 (P54 and P55)

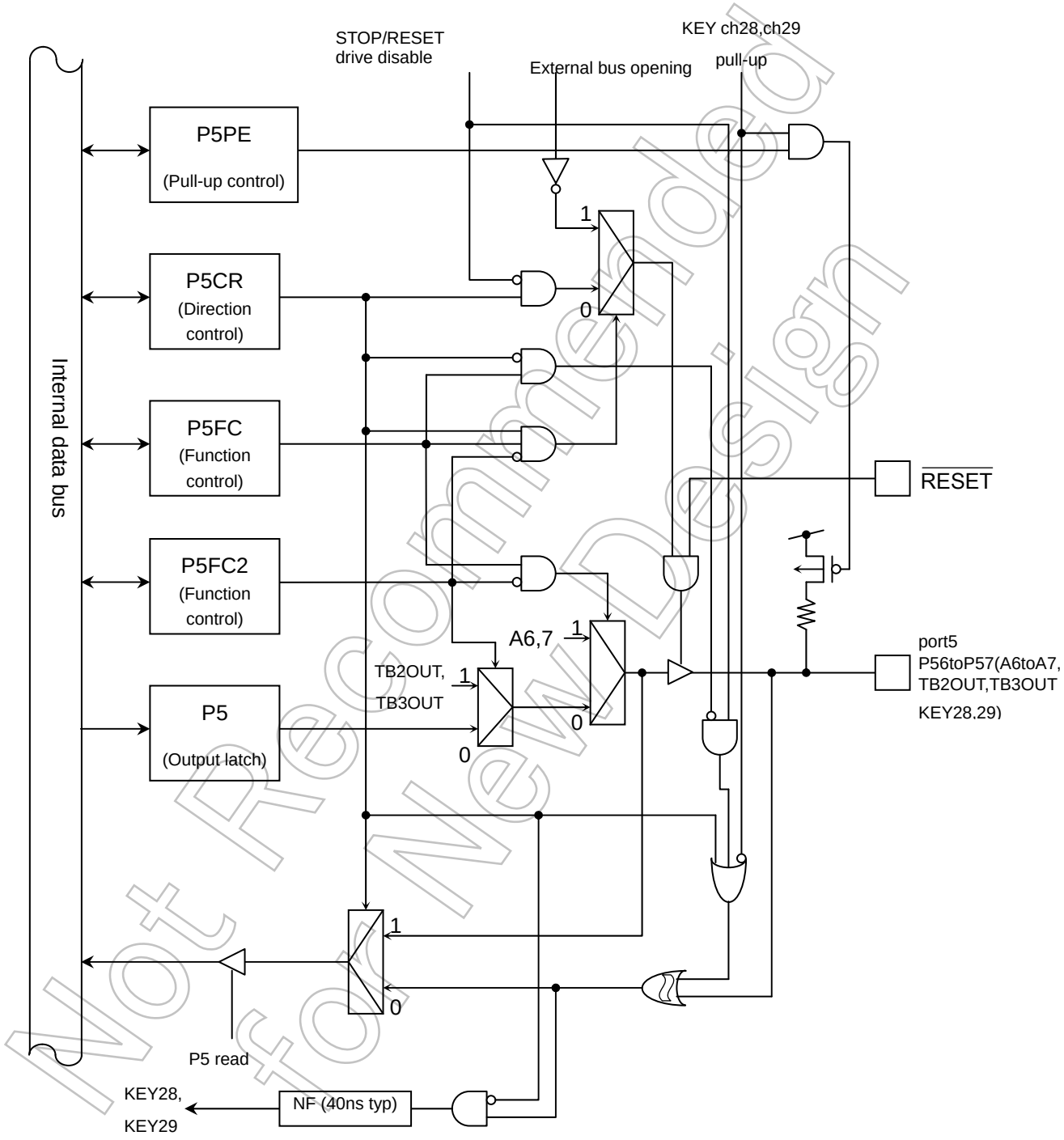


Fig. 7-16 Port 5 (P56 to P57)

If the port 5 goes into STOP mode when the KEY input is enabled, inputs are always accepted. To inhibit inputs, switch to PORT using the function register.

• Port: Inputs are accepted only during a read. • KEY: Inputs are always accepted.

Port 5 register

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

Port 5 control register

		7	6	5	4	3	2	1	0
P5CR	Bit Symbol	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
(0xFFFF_F02C)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	<<See P5FC>>							

Port 5 function register 1

		7	6	5	4	3	2	1	0
P5FC	Bit Symbol	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
(0xFFFF_F02D)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	P5FC/P5CR = 00: Input, 01: Output, 10: Input, 11: A7 through 0							

Port 5 function register 2

		7	6	5	4	3	2	1	0
P5FC2	Bit Symbol	P57F2	P56F2	P55F2	P54F2	–			
(0xFFFF_F03C)	Read/Write	R/W				R			
	After reset	0	0	0	0	0			
	Function	0: Address/ PORT 1: TB3OUT	0: Address/ PORT 1: TB2OUT	0: Address/ PORT 1: TB1OUT	0: Address/ PORT 1TB0OUT	"0" is read.			

Note: If P5FC = "0" and P5FC2 = "1," TB3OUT through TB0OUT are selected.
To use the port 5 in the PORT setting, set both P5FC and P5FC2 to "0."

Port 5 pull-up control register

		7	6	5	4	3	2	1	0
P5PE	Bit Symbol	PE57	PE56	PE55	PE54	PE53	PE52	PE51	PE50
(0xFFFF_F026)	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 to be used	Function	Corresponding BIT of P5FC2	Corresponding BIT of P5FC	Corresponding BIT of P5CR
P50 to P53	P50toP53input setting	-	*	0
	P50toP53output setting	-	0	1
	INTCtoINTFinput setting	-	1	0
	A0toA3output setting	-	1	1
P54/P55	P54/P55input setting	*	*	0
	P54/P55output setting	0	0	1
	TB0OUT/TB1OUToutput setting	1	*	1
	A4/A5output setting	0	1	1
P56/P57	P56/P57input setting	*	0	0
	P56/P57output setting	0	0	1
	TB2OUT/TB3OUToutput setting	1	*	1
	A4/A5output setting	0	1	1
	KEY28/KEY29input setting	*	0	0

Table 7-5

7.7 Port 6 (P60 through P67)

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

The port 6 also functions as an address bus (A8 through A15). To access external memory, P6CR and P6FC must be provisioned to allow the port 6 to function as an address bus. The address bus function can be used only in separate bus mode. (To put the port 6 in separate bus mode, the BUSMD pin (port 45) must be set to "L" level during a reset.)

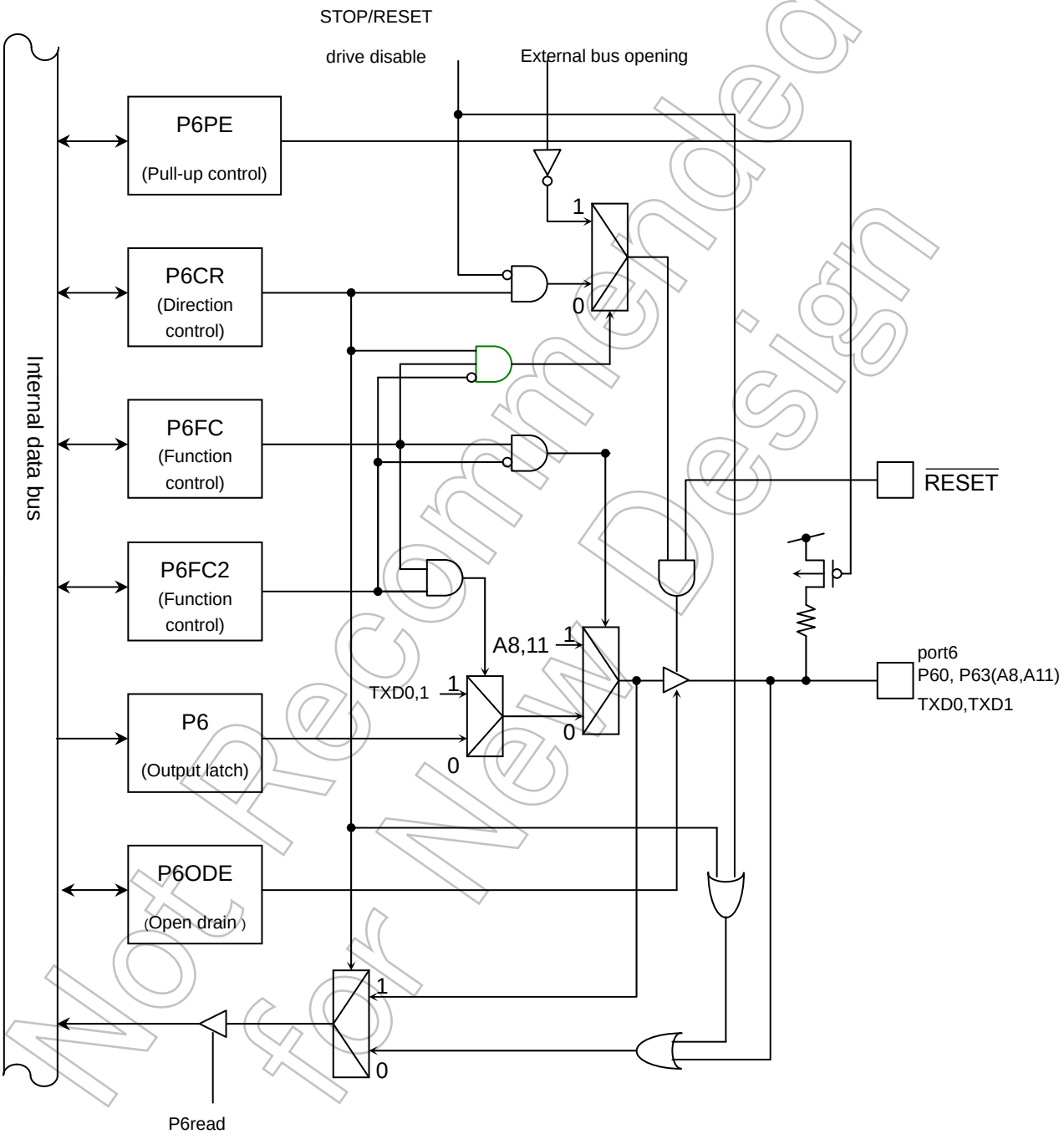


Fig. 7-17 Port 6 (P60, P63)

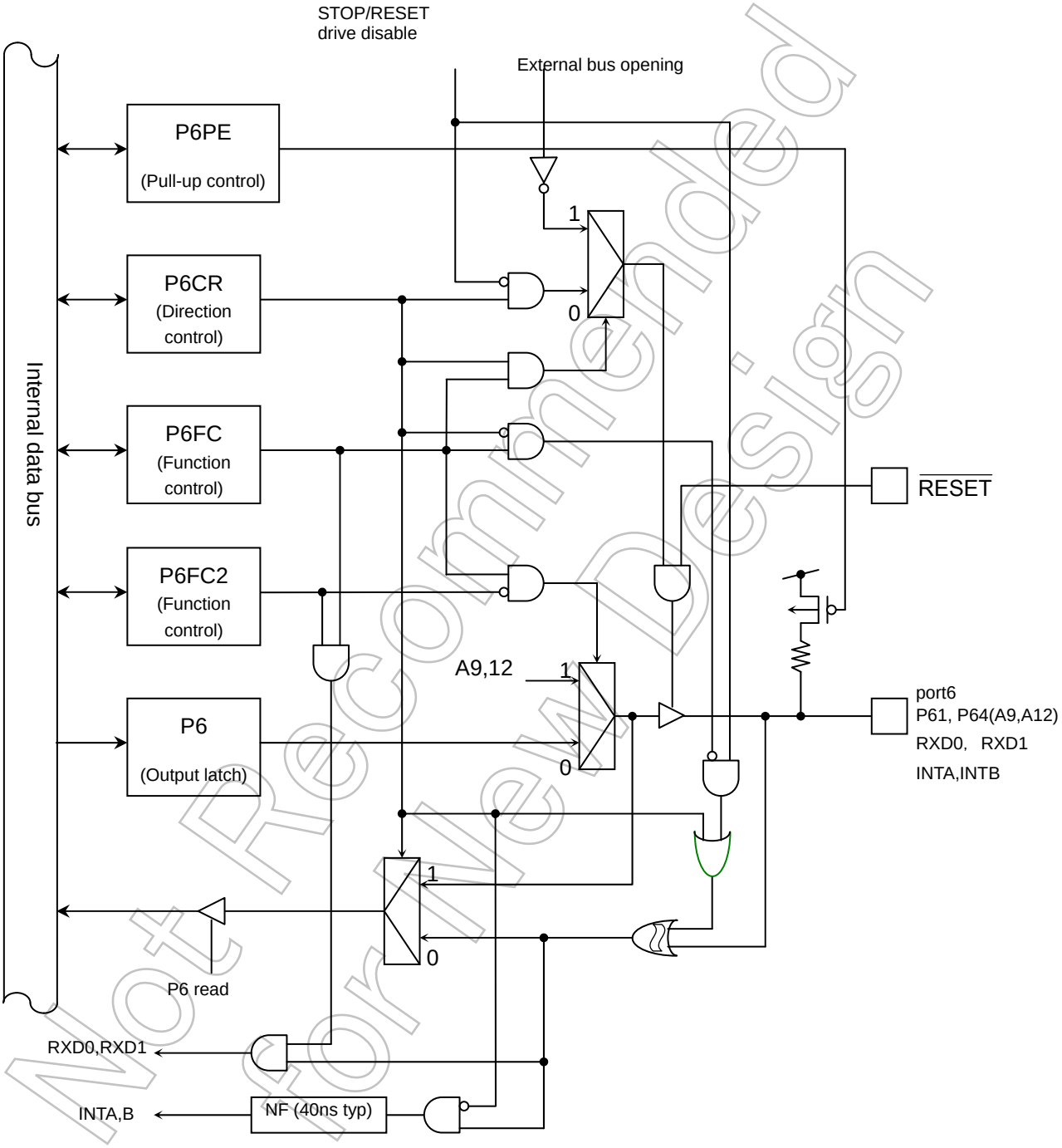


Fig. 7-18 Port 6 (P61, P64)

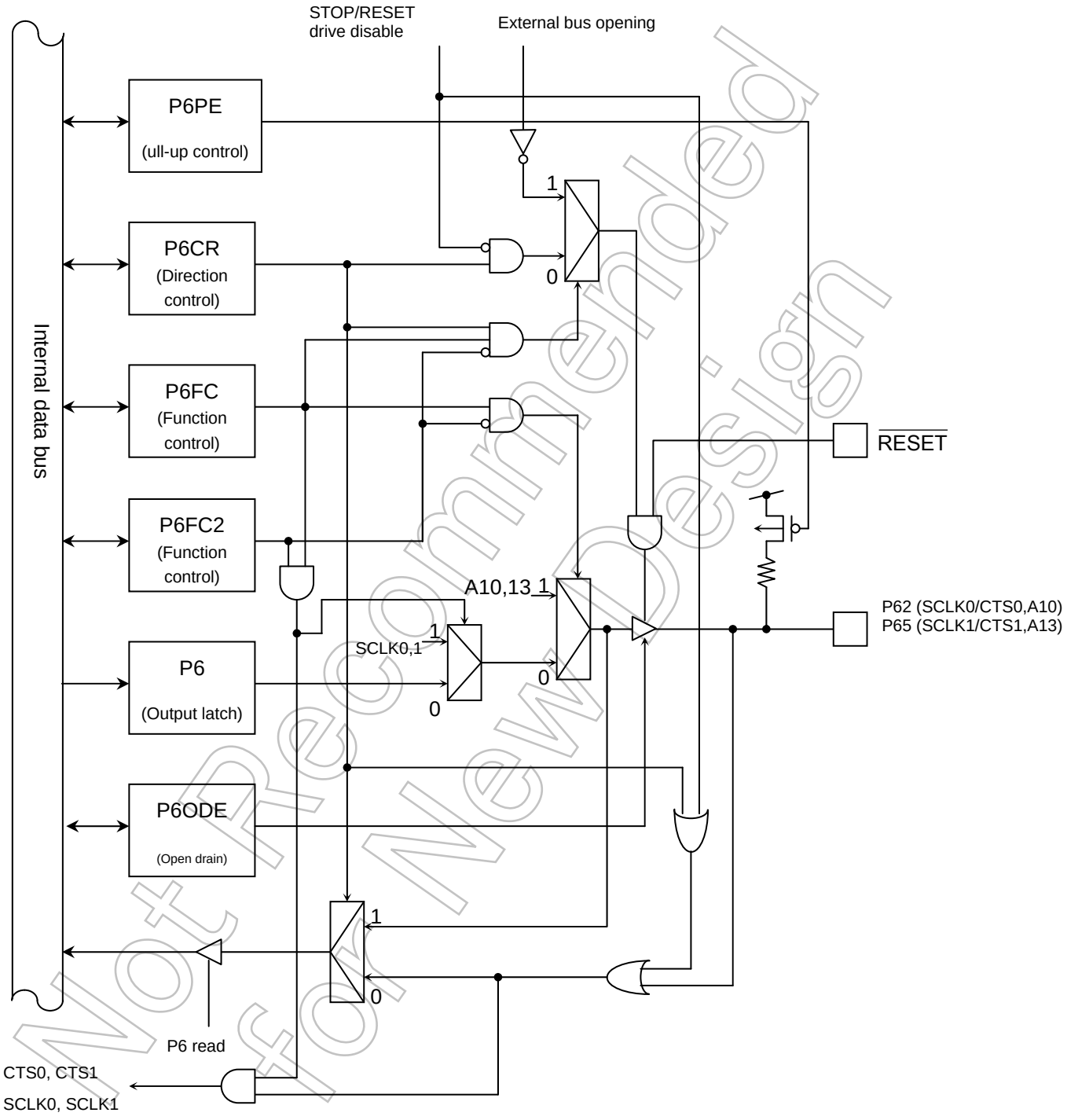


Fig. 7-19 Port 6 (P62, P65)

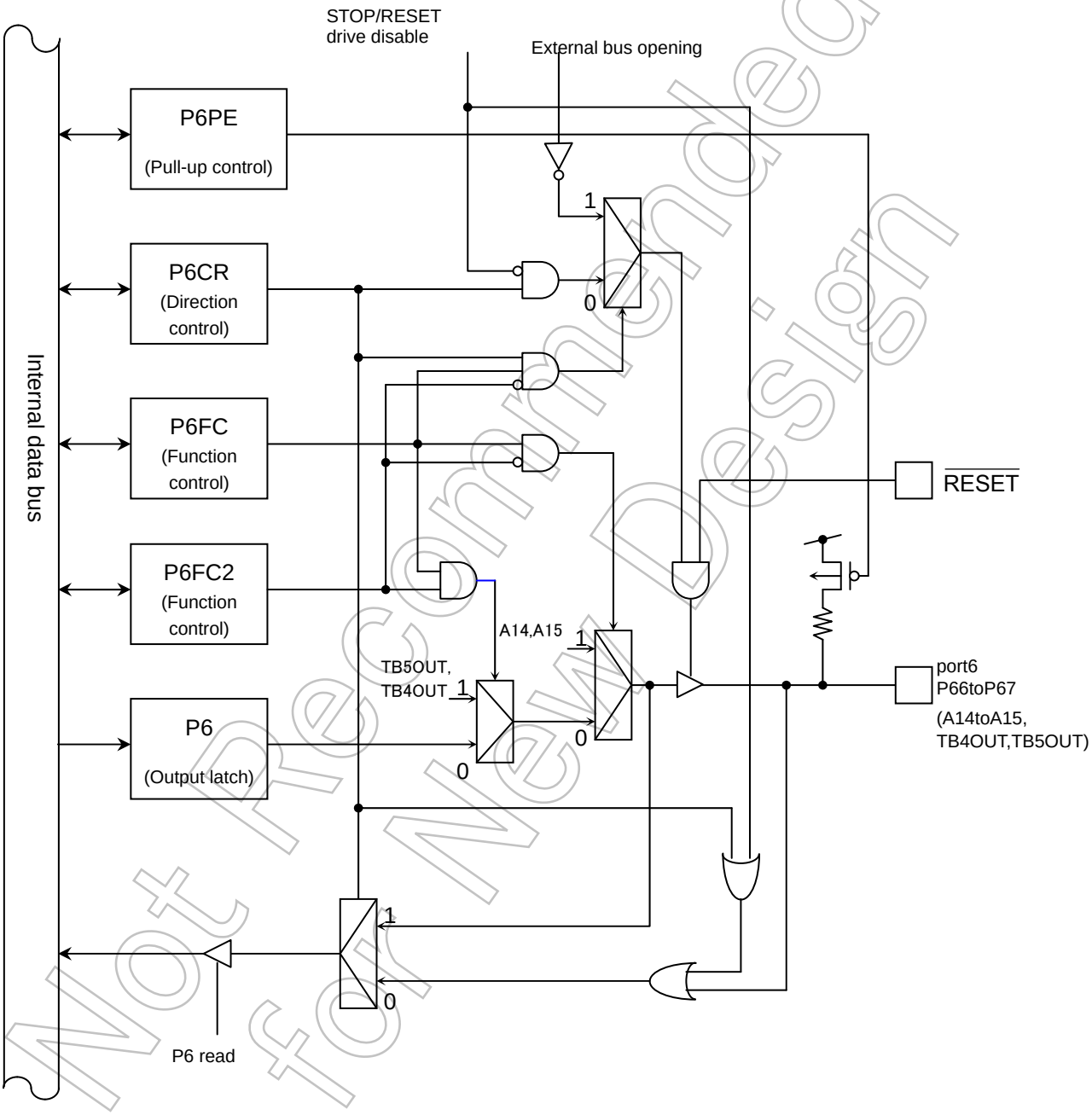


Fig. 7-20 Port 6 (P66, P67)

Port 6 register

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

Port 6 control register

		7	6	5	4	3	2	1	0
P6CR	Bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
(0xFFFF_F02E)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	<< See P6FC >>							

Port 6 function register

		7	6	5	4	3	2	1	0
P6FC	Bit Symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
(0xFFFF_F02F)	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	P6FC/P6CR = 00: Input, 01: Output, 10: Input, 11: A15 through 8							

Port 6 function settings

P60, P63

	P6CR <P6xC>	P6FC<P6xF> 0	P6FC<P6xF> 1
Separate bus mode (BUSMD=”L”)	0	Input port	
	1	Output port	Address bus (A11, 8) /TXD0, 1
Multiplexed bus mode (BUSMD=”H”)	0	Input port	
	1	Output port	Address bus (A11, 8) /TXD0, 1

P61, P64

	P6CR <P6xC>	P6FC<P6xF> 0	P6FC<P6xF> 1
Separate bus mode (BUSMD=”L”)	0	Input mode (INTA, B)	RXD0, 1/INTA, B
	1	Output port	Address bus (A12-9)
Multiplexed bus mode (BUSMD=”H”)	0	Input port (INTA, B)	RXD0, 1/INTA, B
	1	Output port	Address bus (A12-9)

P62, P65

	P6CR <P6xC>	P6FC<P6xF> 0	P6FC<P6xF> 1
Separate bus mode (BUSMD=”L”)	0	Input port	SCLK0, 1/CTS0, 1
	1	Output port	Address bus (A13-10) /SCLK0, 1
Multiplexed bus mode (BUSMD=”H”)	0	Input port	SCLK0, 1/CTS0, 1
	1	Output port	Address bus (A13-10) /SCLK0, 1

		P6CR <P6xC>	P6FC<P6xF>	
			0	1
P66, P67	Separate bus mode (BUSMD="L")	0	Input port	
		1	Output port	Address bus (A15, 14) /TB4OUT, TB5OUT
	Multiplexed bus mode (BUSMD="H")	0	Input port	
		1	Output port	Address bus (A15, 14) /TB4OUT, TB5OUT

Port 6 function register 2

P6FC2 (0xFFFF_F03D)

	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: A15 1: TB5OUT	0: A14 1: TB4OUT	0: A13 1: SCLK1/ CTS1	0: A12 1:RXD1	0: A11 1: TXD1	0: A10 1: SCLK0/ CTS0	0: A9 1: RXD0	0: A8 1: TXD0,

Note: If P6FC = "0" and P6FC2 = "1," PORT is selected.
To use this function register as a functional pin, set both P6FC and P6FC2 to "1."
(Set 1, 4bit to P6FC2 = "0.")

Port 6 pull-up control register

P6PE (0xFFFF_F027)

	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

Port 6 open drain control register

P6ODE (0xFFFF_ F030)

	7	6	5	4	3	2	1	0
Bit Symbol	–		P65ODE	–	P63ODE	P62ODE	–	P60ODE
Read/Write	R		R/W	R	R/W		R	R/W
After reset	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: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain

PORT to be used	Function	Corresponding BIT of P6FC2	Corresponding BIT of P6FC	Corresponding BIT of P6CR
P60/P63	P60/P63input setting	*	*	0
	P60/P63Output setting	*	0	1
	TXD0/TXD1Output setting	1	1	1
	A8/A11Output setting	0	1	1
P61/P64	P61/P64input setting	*	0	0
	P61/P64Output setting	*	0	1
	RXD0/RXD1input setting	1	1	0
	INTA/INTBinput setting	*	0	0
	A9,A12Output setting	0	1	1
P62/P65	P62/P65input setting	*	*	0
	P62/P65Output setting	*	0	1
	SCLK0/SCLK1Output setting	1	*	1
	CTS0/CTS1/SCLK0/SCLK1 input setting	1	1	0
	A10/A13Output setting	0	1	1
P66/P67	P66/P67input setting	*	*	0
	P66/P67Output setting	*	0	1
	TB4OUT/TB5OUTOutput setting	1	1	1
	A15/A16Output setting	0	1	1

Table 7-6

7.8 Port 7 (P70 through P77)

The port 7 is an 8-bit, analog input port for the A/D converter. Although P74 through P77 form part of the analog input port, they also perform another function of inputting the key-on wake-up.

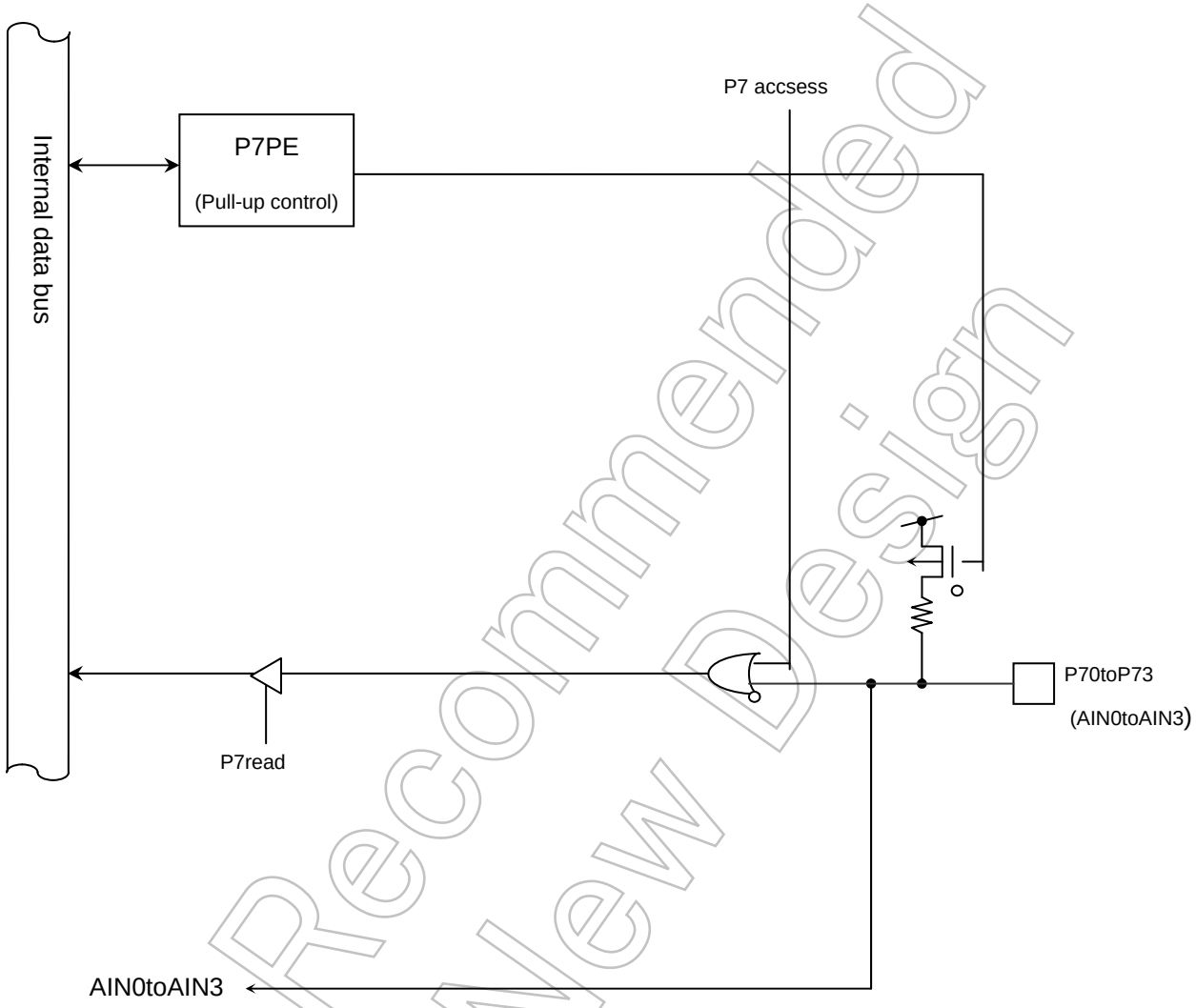


Fig. 7-21 Port 7 (P70 through P73)

- **If the port 7 goes into STOP mode when the PORT input is enabled, inputs are always accepted.**
  To inhibit inputs, switch to AD using the function register.
  - **AD: Inputs are accepted only during a read. Inputs are inhibited in STOP mode.**
  - **PORT: Inputs are always accepted.**

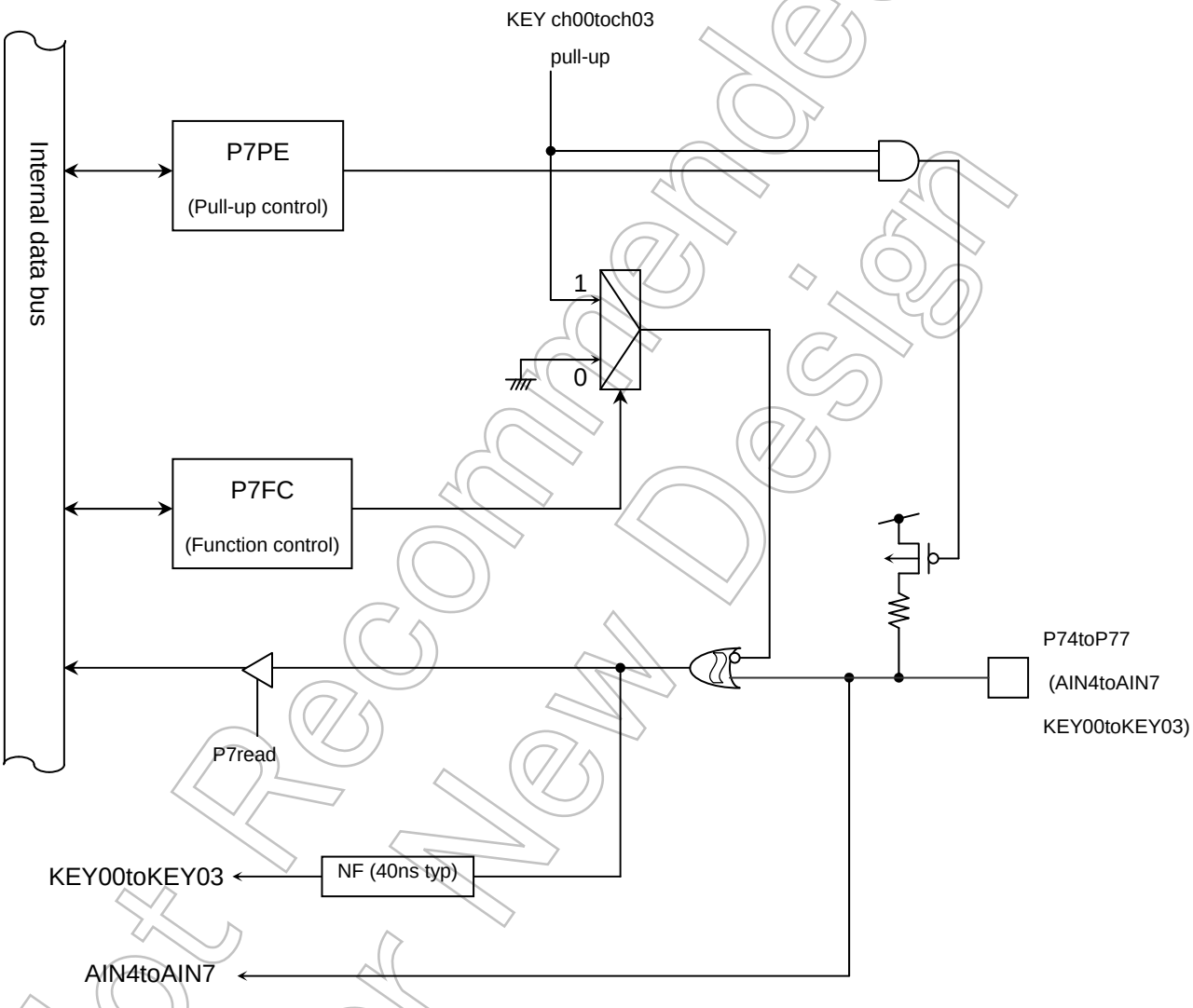


Fig. 7-22 Port 7 (P74 through P77)

- **If the port 7 goes into STOP mode when the KEY/PORT input is enabled, inputs are always accepted.**
  To inhibit inputs, switch to AD using the function register.
  - **AD: Inputs are accepted only during a read. Inputs are inhibited in STOP mode.**
  - **KEY/PORT: Inputs are always accepted.**

Port 7 register

P7 (0xFFFF_F040)

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

Port 7 function register

P7FC (0xFFFF_F048)

	7	6	5	4	3	2	1	0
Bit Symbol	P77F	P76F	P75F	P74F	–	–	–	–
Read/Write	R/W				R			
After reset	0	0	0	0	0			
Function	0: A/D 1: PORT /KEY03	0: A/D 1: PORT /KEY02	0: A/D 1: PORT /KEY01	0: A/D 1: PORT /KEY00	"0" is read.			

Port 7 pull-up control register

P7PE (0xFFFF_F04C)

	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

7.8 Port 8 (P80 through P87)

The port 8 is an 8-bit, analog input port for the A/D converter. Besides this analog input port function, P80 through P83 input the key-on wake-up, and P84 through P87 input external interrupts.

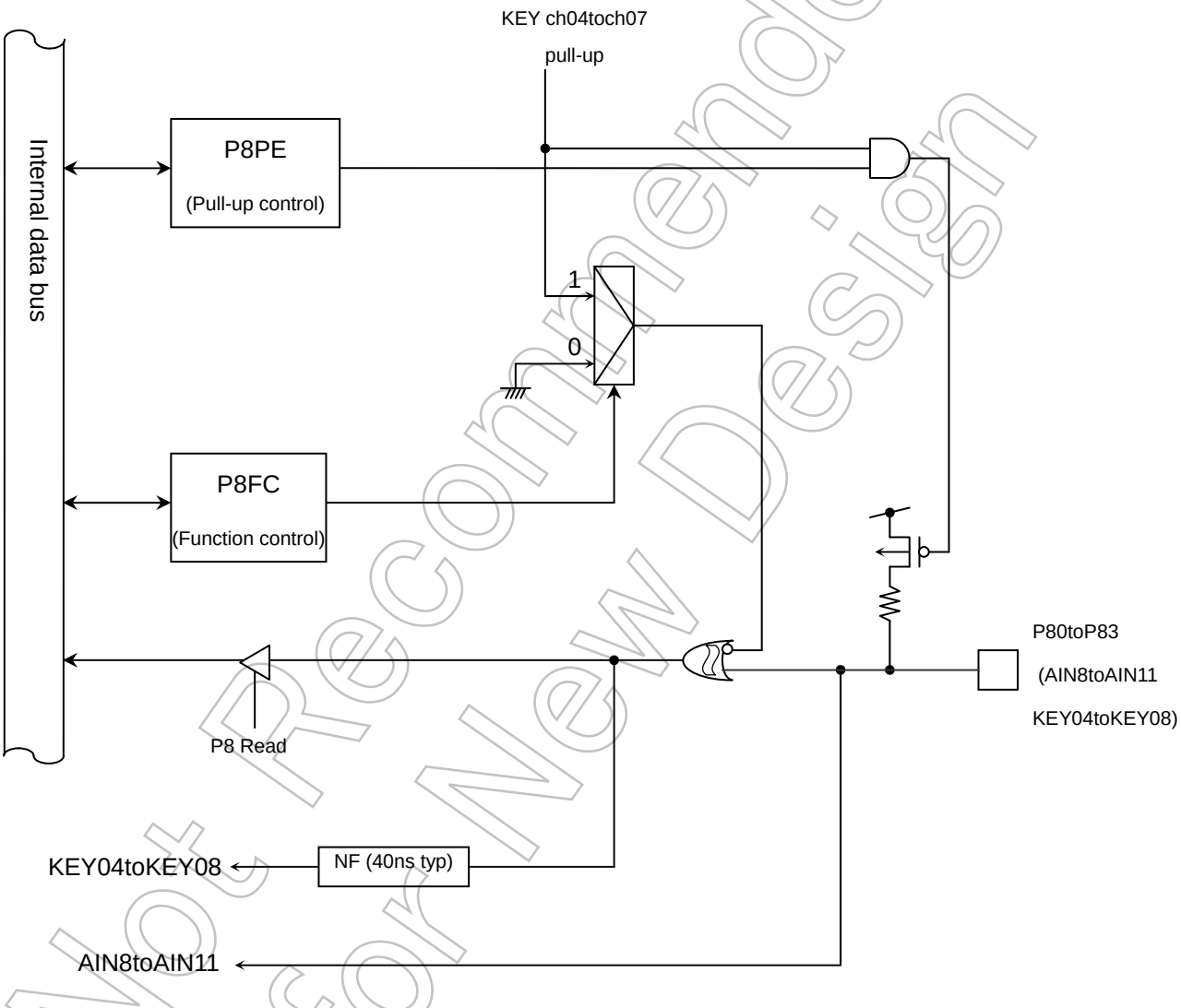


Fig. 7-23 Port 8 (P80 through P83)

If the port 8 goes into STOP mode when the KEY/PORT input is enabled, inputs are always accepted. To inhibit inputs, switch to PORT using the function register.

- **AD: Inputs are accepted only during a read. Inputs are inhibited in STOP mode.**
- **KEY/PORT: Inputs are always accepted.**

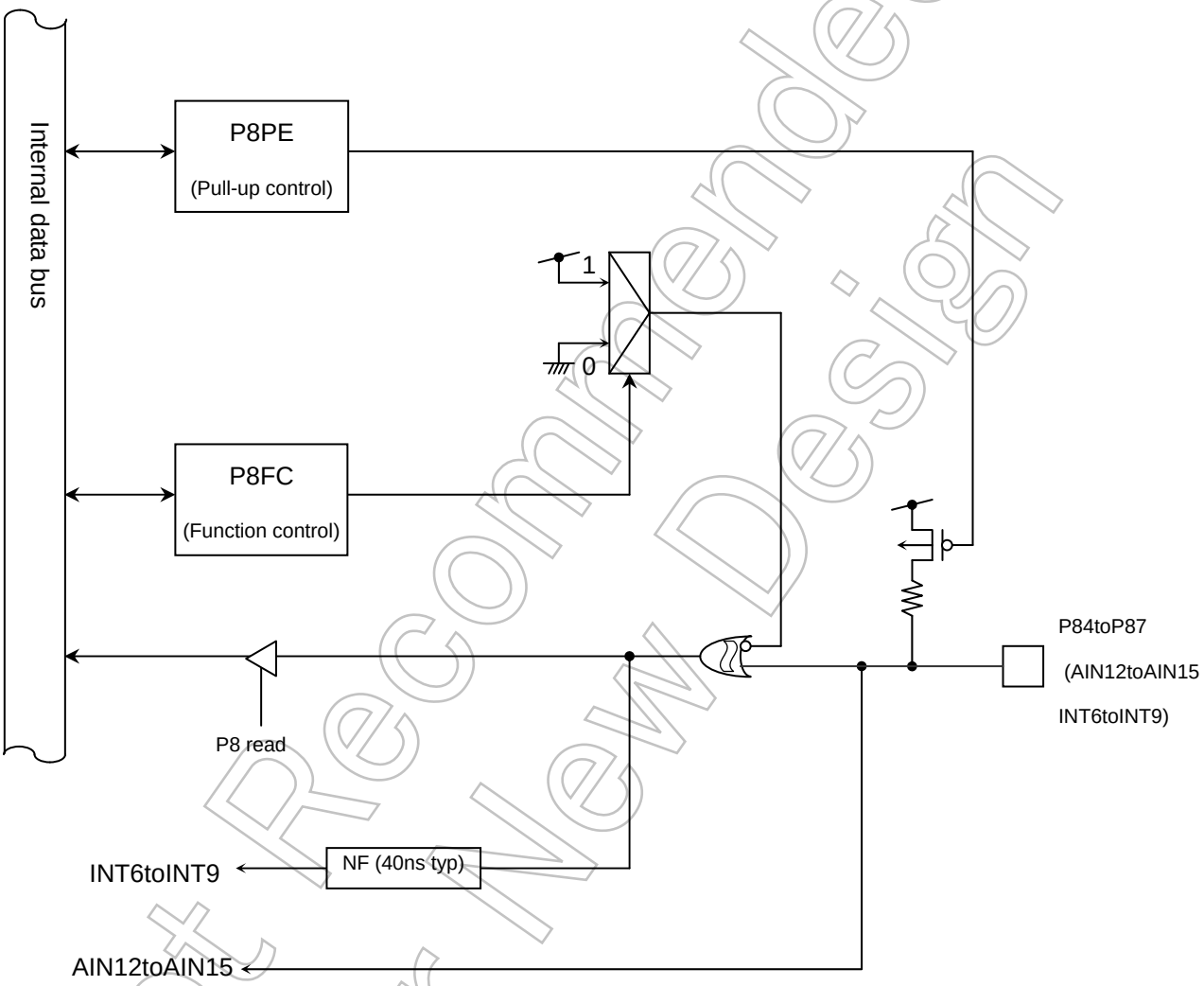


Fig. 7-24 Port 8 (P84 through P87)

- **If the port 8 goes into STOP mode when the KEY/PORT/INT input is enabled, inputs are always accepted.**
  To inhibit inputs, switch to AD using the function register.
  - **AD: Inputs are accepted only during a read. Inputs are inhibited in STOP mode.**
  - **KEY/PORT/INT: Inputs are always accepted.**

Port 8 register

P8 (0xFFFF_F041)

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

Port 8 function register

P8FC (0xFFFF_F049)

	7	6	5	4	3	2	1	0
Bit Symbol	P87F	P86F	P85F	P84F	P83F	P82F	P81F	P80F
Read/Write	RW							
After reset	0	0	0	0	0	0	0	0
Function	0: A/D 1: PORT /INT9	0: A/D 1: PORT / INT8	0: A/D 1: PORT / INT7	0: A/D 1: PORT / INT6	0: A/D 1: PORT / KEY07	0: A/D 1: PORT / KEY06	0: A/D 1: PORT / KEY05	0: A/D 1: PORT / KEY04

Port 8 pull-up control register

P8PE (0xFFFF_F04D)

	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

7.9 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 register P9FC. A reset allows all bits of the output latch P9 to be set to "1," all bits of P9CR and P9FC to be cleared to "0," and the port 9 to be put in input mode.

Besides the input/output port function, the port 9 performs other functions: P93 outputs SI0 data, P94 inputs SI0 data, P95 inputs and outputs SI0 CLK or inputs CTS, and P90, P91, P92, P96 and P97 output a 16-bit timer.

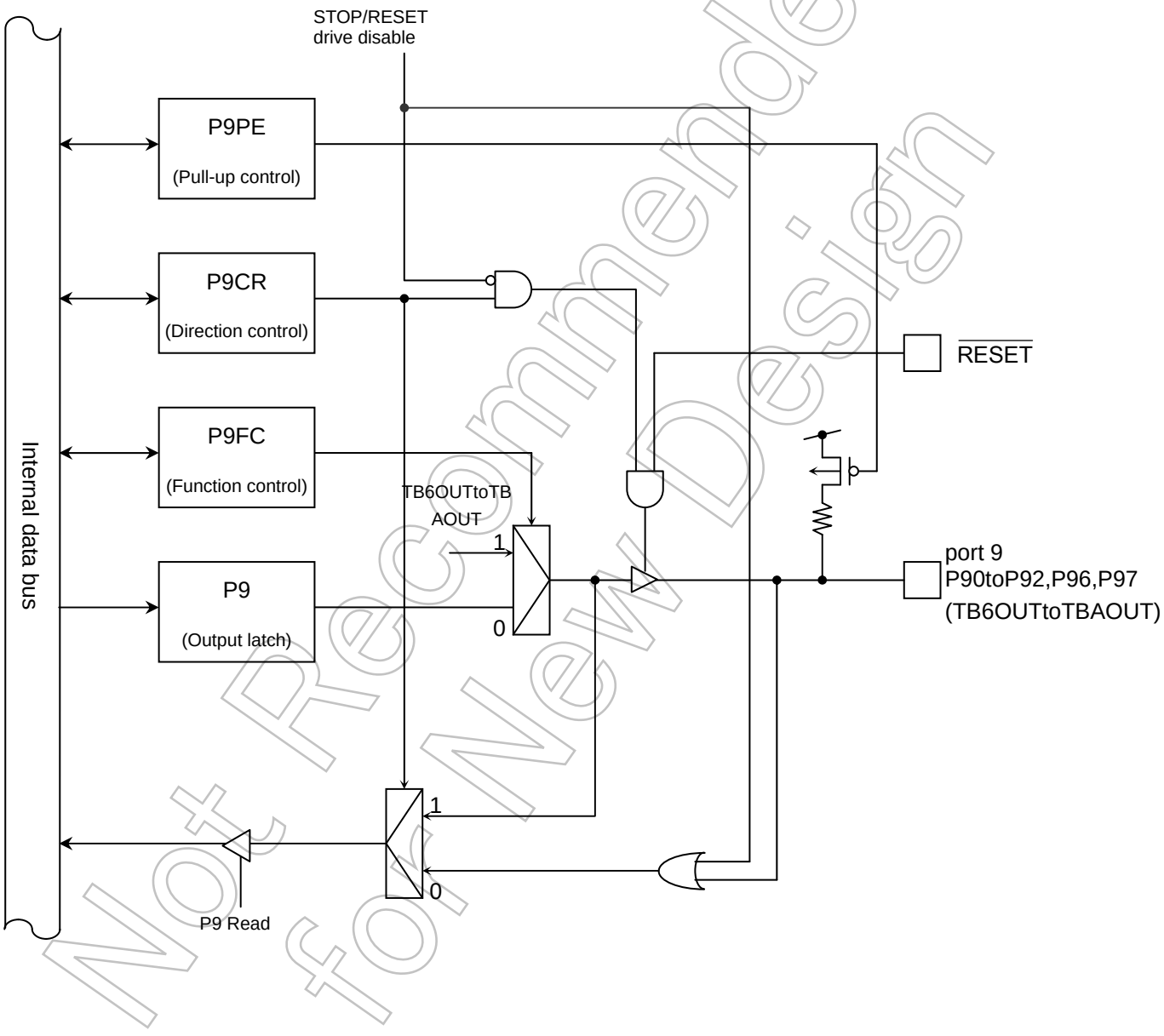


Fig. 7-25 Port 9 (P90, P91, P92, P96 and P97)

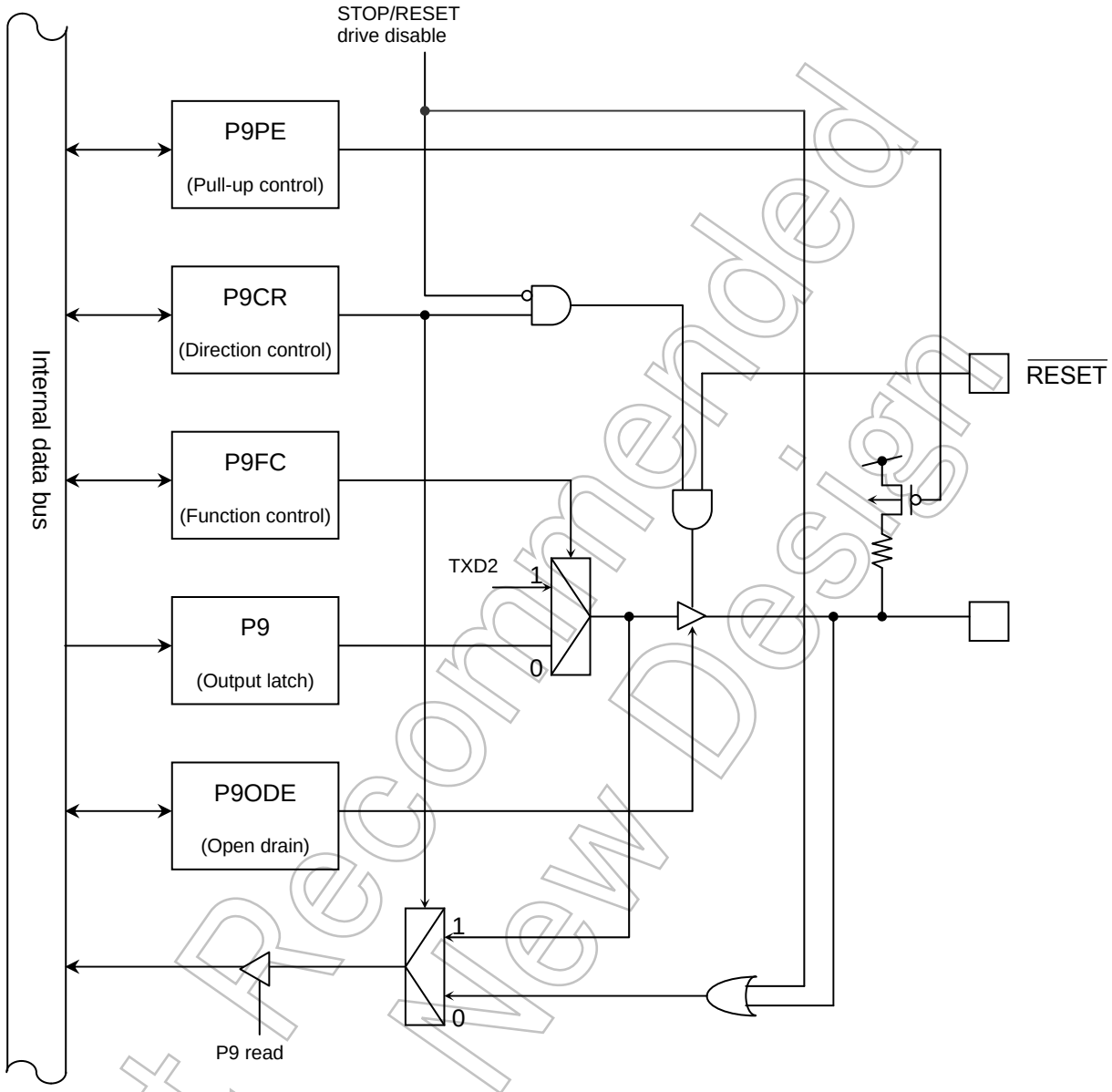


Fig. 7-26 Port 9 (P93)

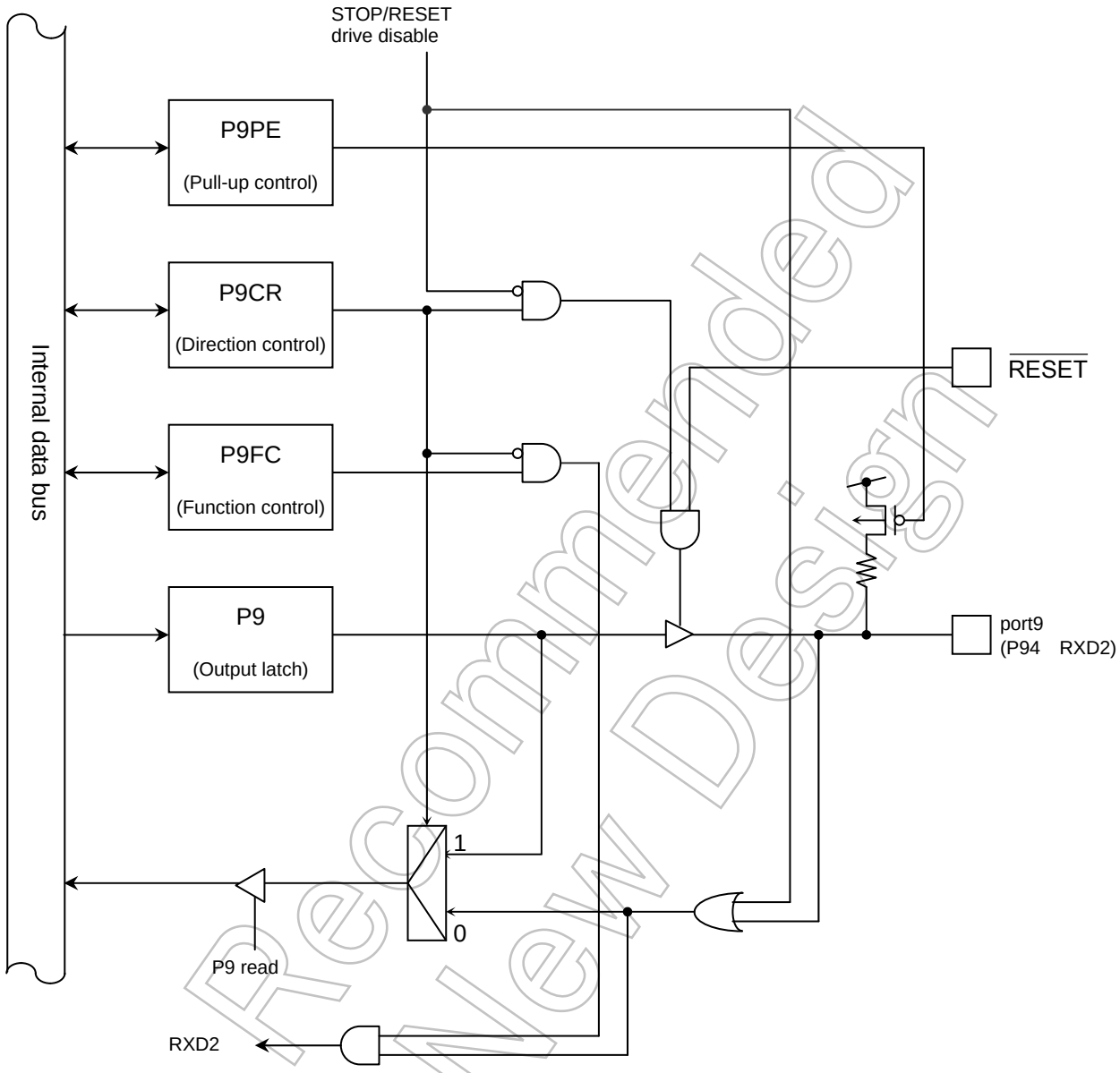


Fig. 7-27 Port 9 (P94)

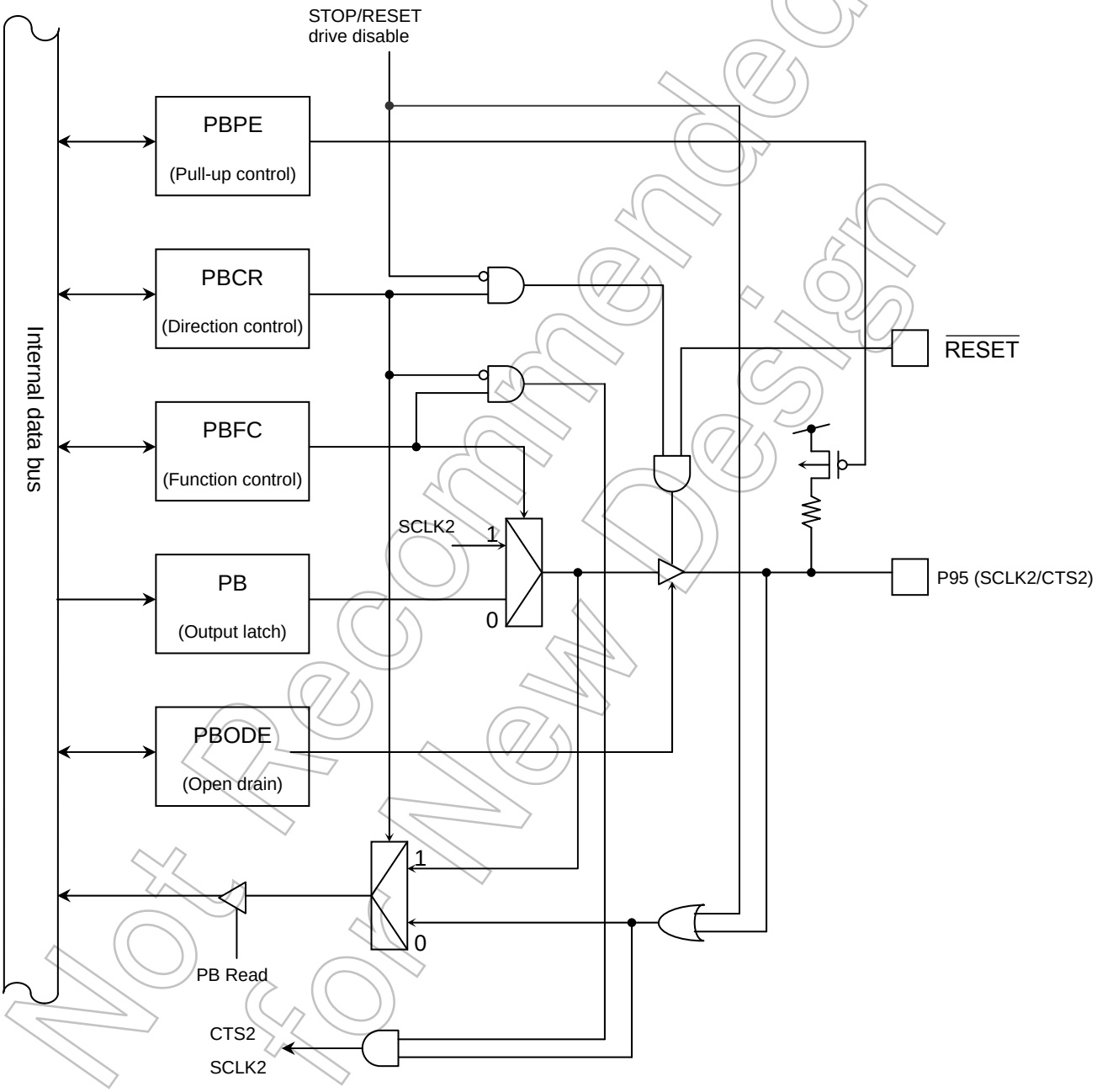


Fig. 7-28 Port 9 (P95)

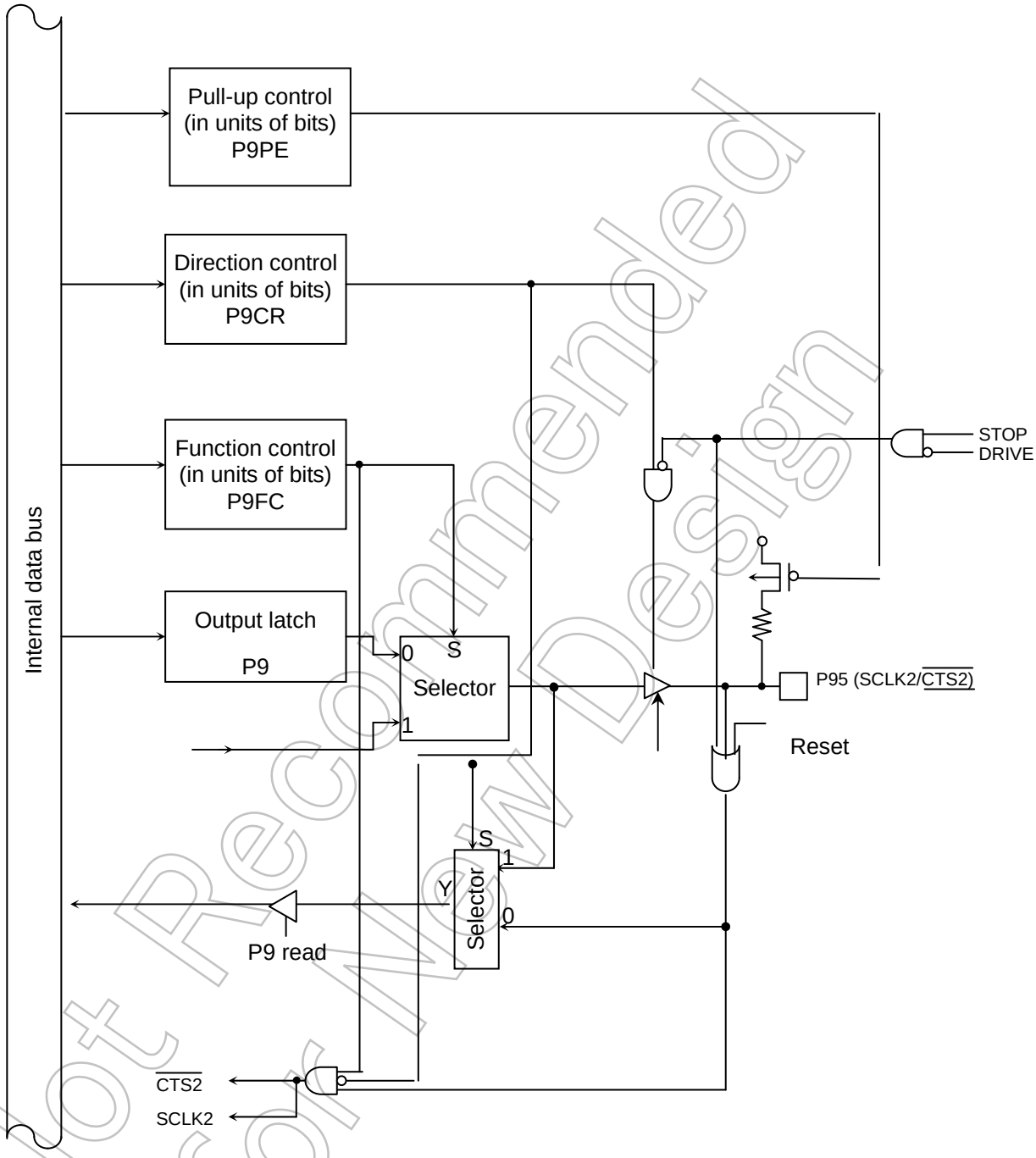


Fig. 7-29 Port 9 (P95)

Port 9 register

P9 (0xFFFF_F042)

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

Port 9 control register

P9CR (0xFFFF_F046)

	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

P9FC (0xFFFF_F04A)

	7	6	5	4	3	2	1	0
Bit Symbol	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: TBAOUT	0: PORT 1: TB9OUT	0: PORT 1: SCLK2/ CTS2	0: PORT 1: RXD2	0: PORT 1: TXD2	0: PORT 1: TB8OUT	0: PORT 1: TB7OUT	0: PORT 1: TB6OUT

Port 9 pull-up control register

P9PE (0xFFFF_F04E)

	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 open drain control register

P9ODE (0xFFFF_F031)

	7	6	5	4	3	2	1	0
Bit Symbol	–		P95ODE	–	P93ODE	–		
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.		

7.10 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 register PAFC. A reset allows all bits of the output latch PA to be set to "1," all bits of PACR and PAFC to be cleared to "0," and the port A to be put in input mode.

Besides the input/output port function, the port A performs other functions: PA0 through PA5 input external interrupts and a 16-bit timer, and PA6 and PA7 perform a dial input function.

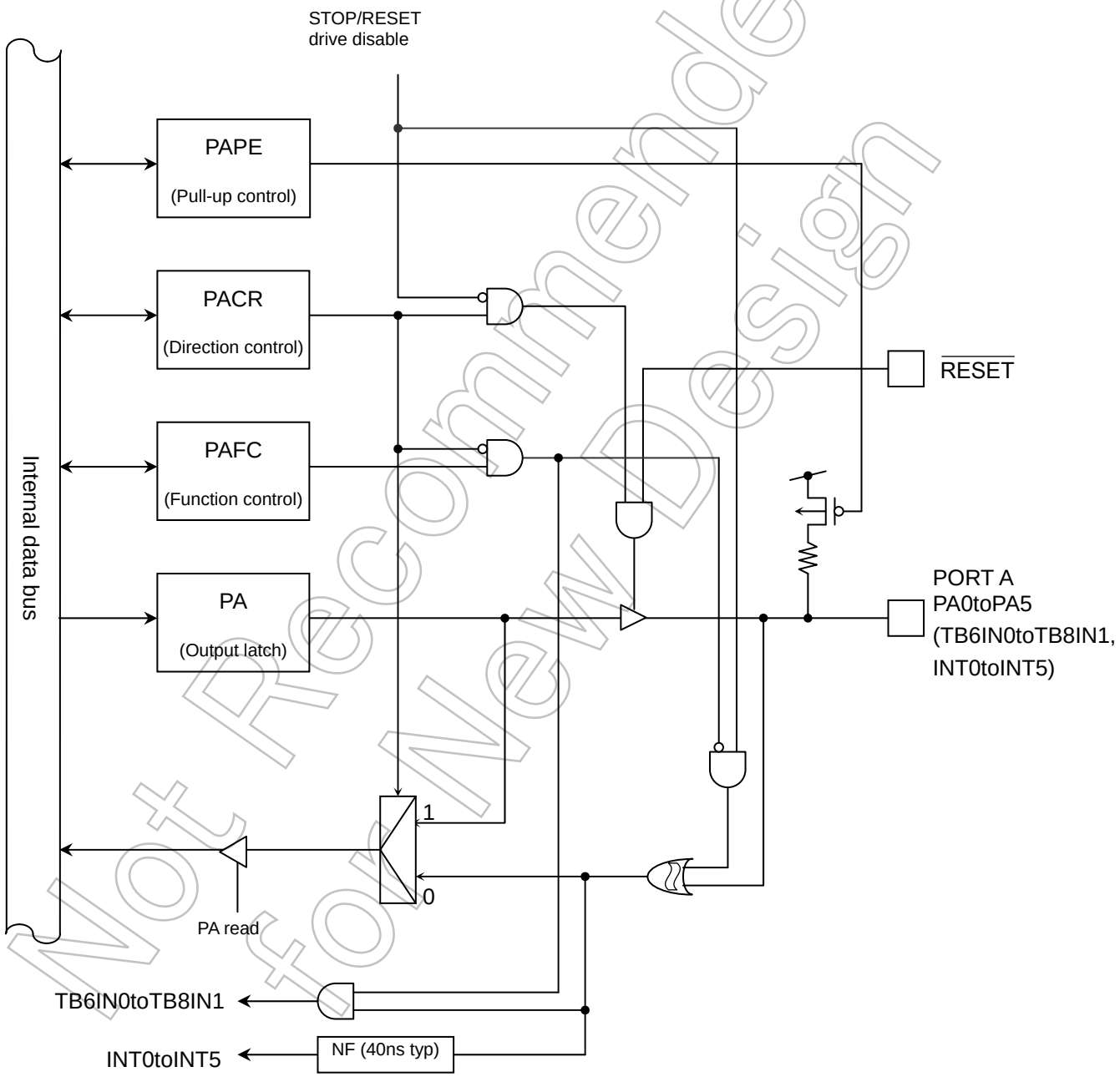


Fig. 7-30 Port A (PA0 through PA5)

If the port A goes into STOP mode when the INT input is enabled, inputs are always accepted. To inhibit inputs, switch to PORT using the function register.

- **Port: Inputs are accepted only during a read.**
- **INT: Inputs are always accepted.**

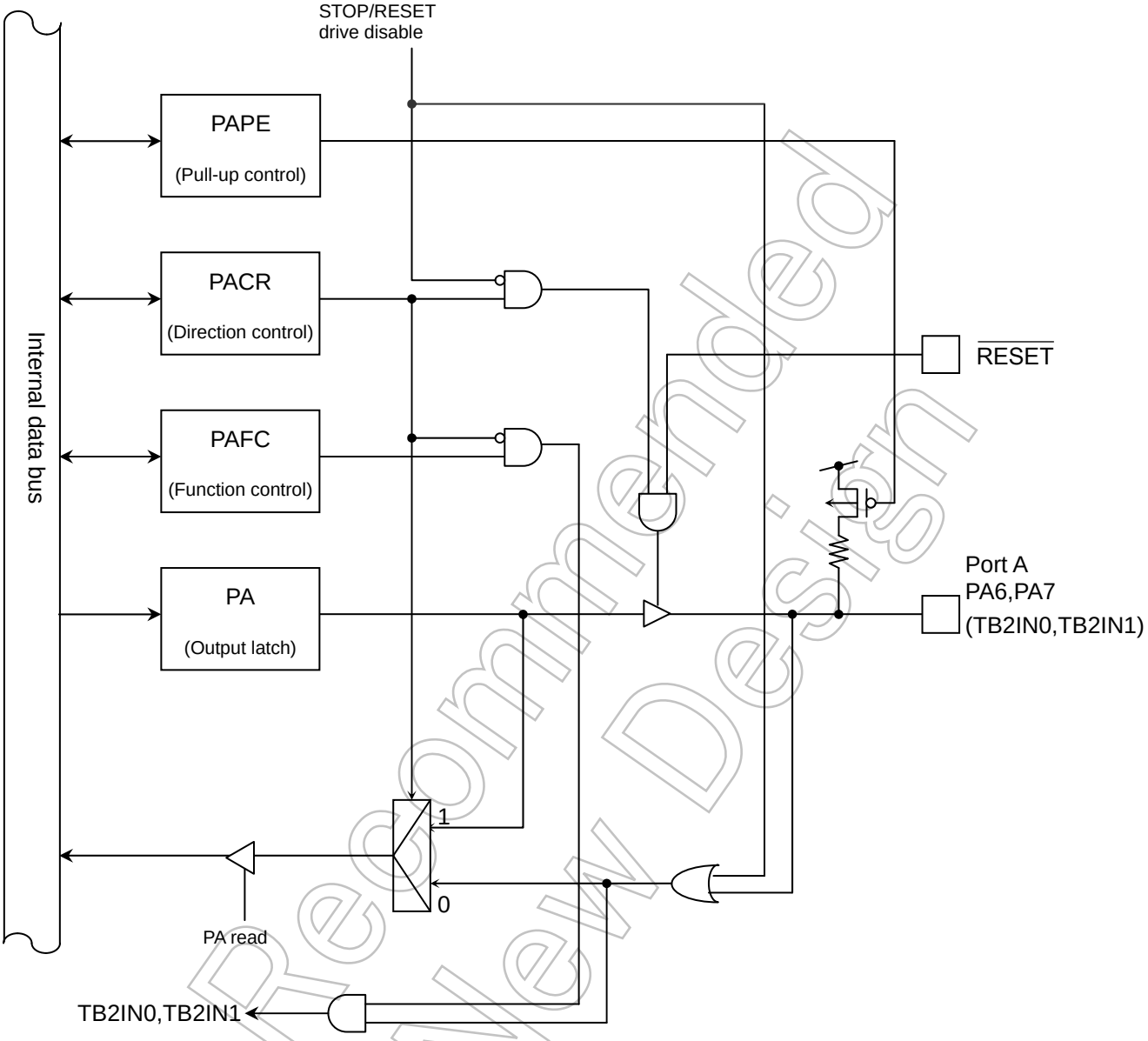


Fig. 7-31 Port A (PA6, PA7)

Port A register

PA (0xFFFF_F043)

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

Port A control register

PACR (0xFFFF_F047)

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

Port A function register

PAFC (0xFFFF_F04B)

	7	6	5	4	3	2	1	0
Bit Symbol	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: /TB2IN1	0: PORT 1: /TB2IN0	0: PORT / INT5 1: INT5 /TB8IN1	0: PORT / INT4 1: INT4 /TB8IN0	0: PORT / INT3 1: INT3 /TB7IN1	0: PORT / INT2 1: INT2 /TB7IN0	0: PORT / INT1 1: INT1 /TB6IN1	0: PORT / INT0 1: INT0 /TB6IN0

Port A pull-up control register

PAPE (0xFFFF_F04F)

	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

7.11 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 register PBFC. A reset allows all bits of the output latch PB to be set to "1," all bits of PBCR and PBFC to be cleared to "0," and the port B to be put in input mode.

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

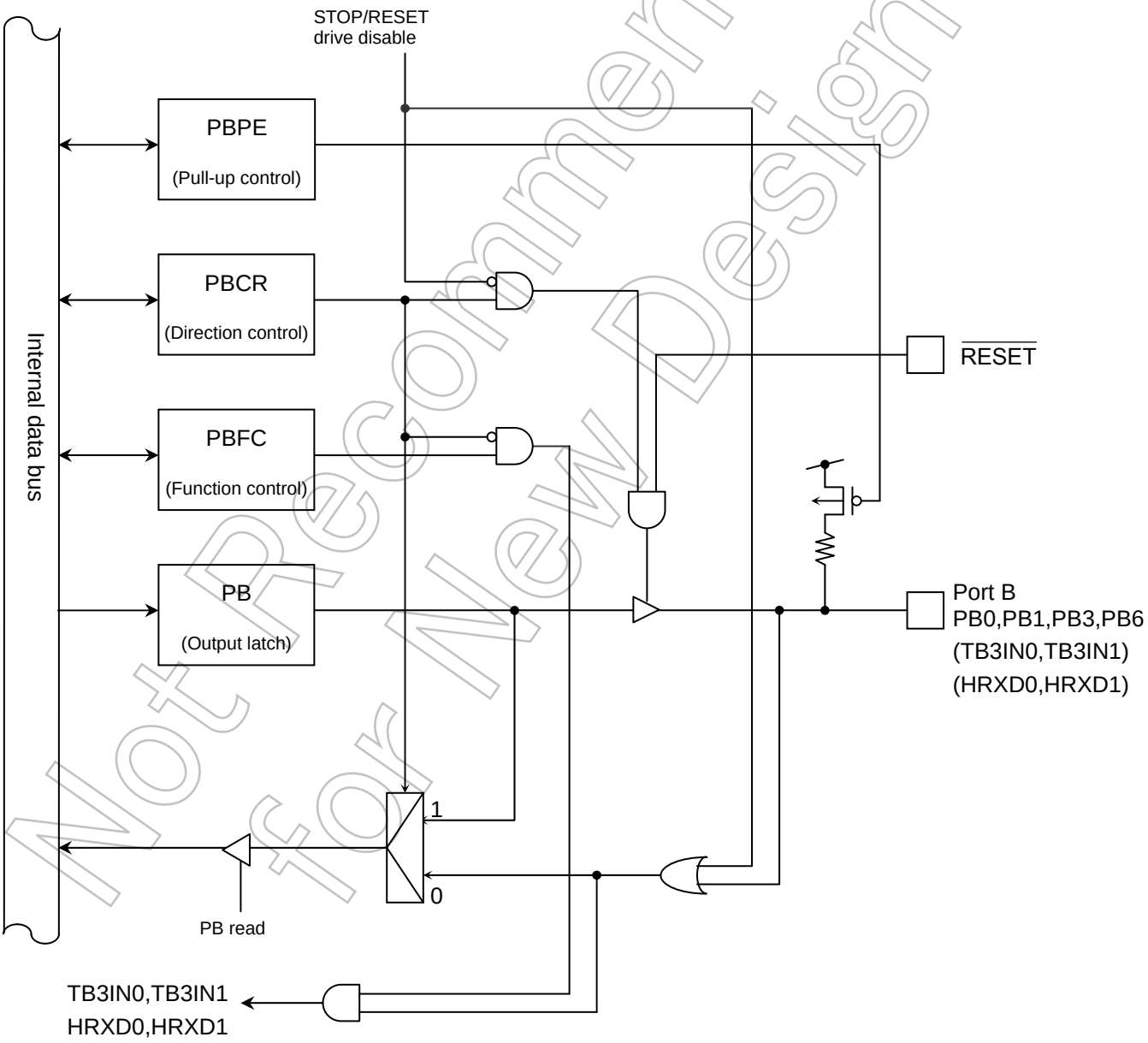


Fig. 7- 32 Port B (PB0, PB1)

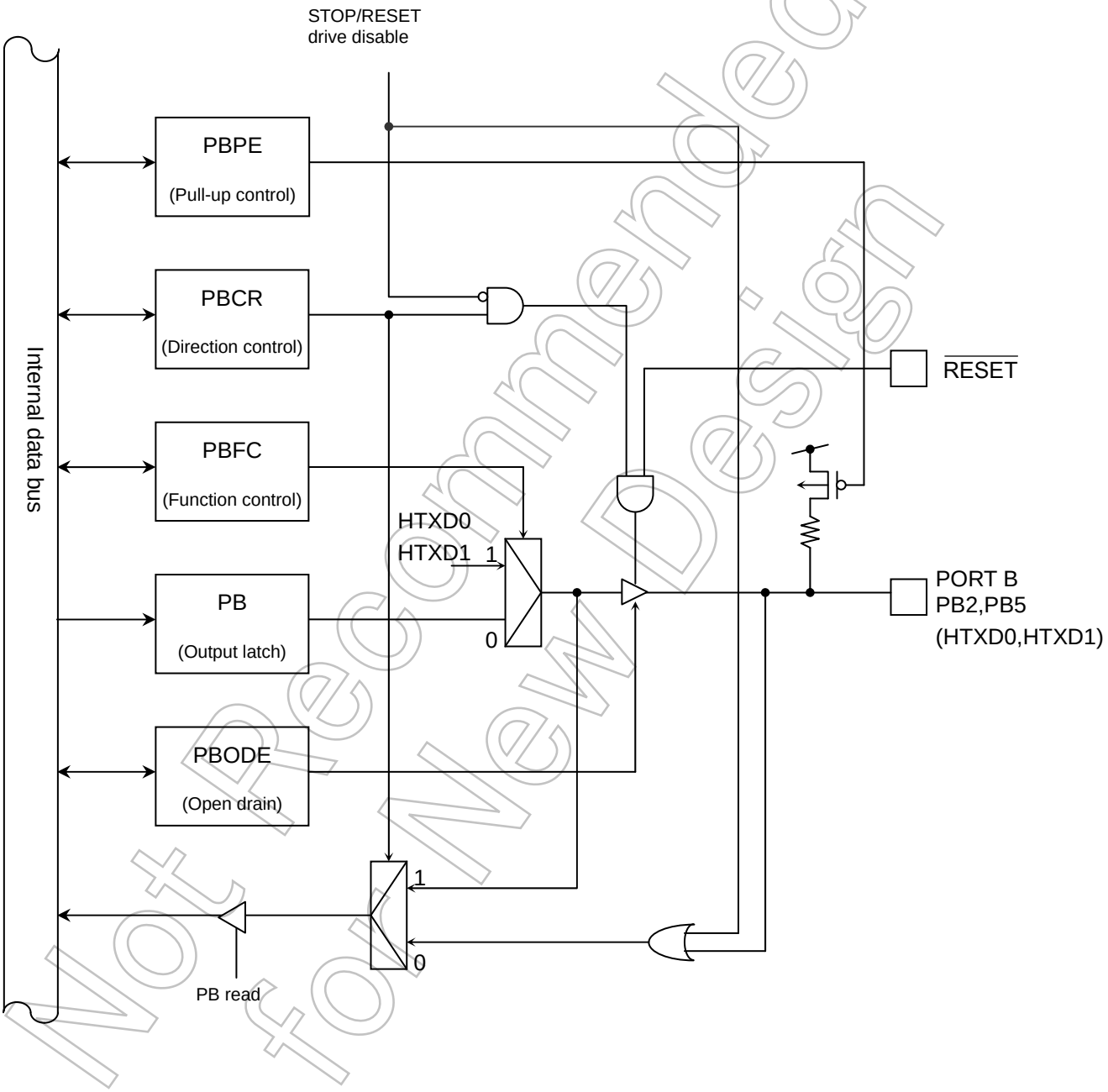


Fig. 7-33 Port B (PB2, PB5)

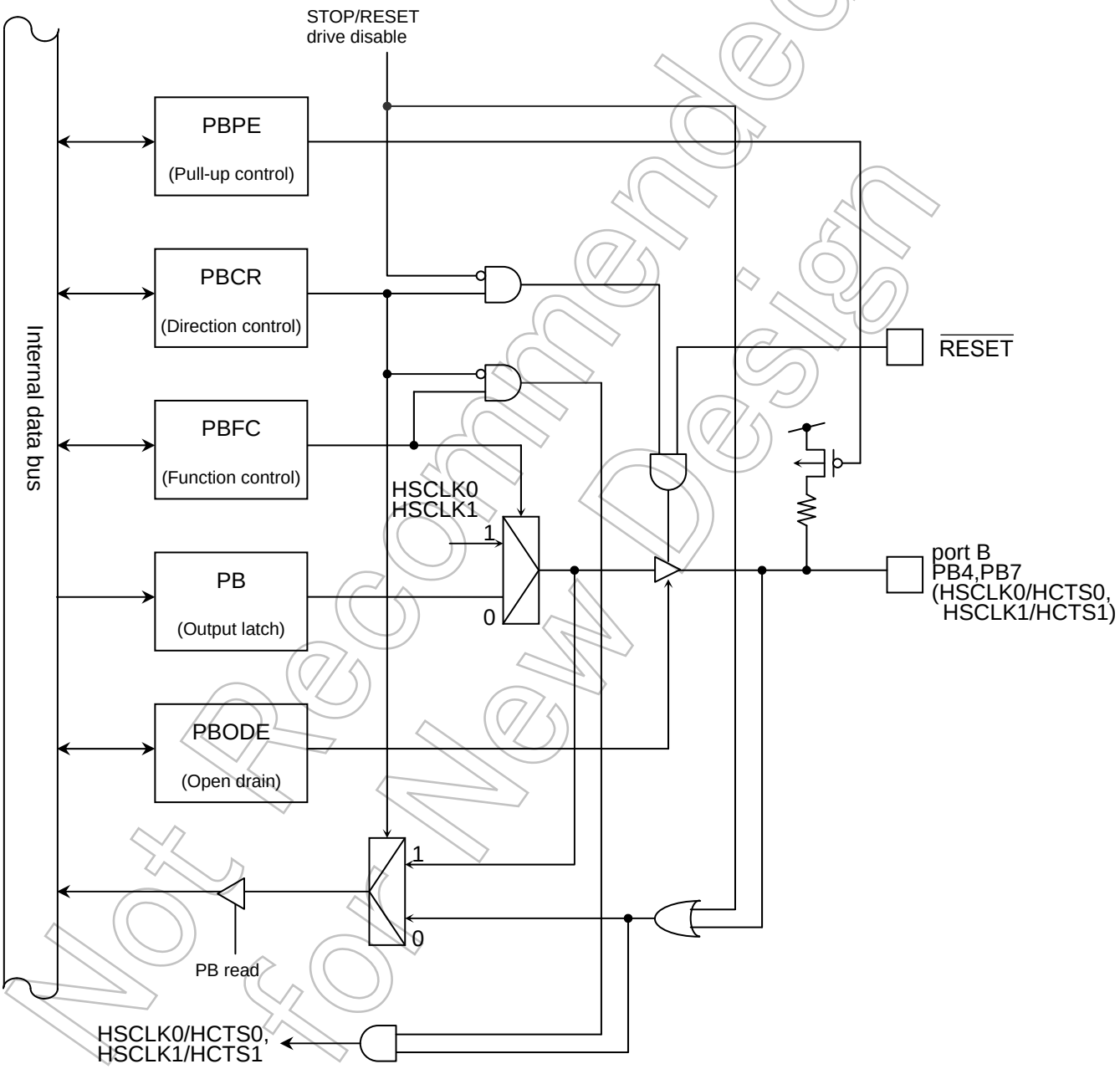


Fig. 7-344 Port B (PB4, PB7)

Port B register

PB (0xFFFF_F050)

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

Port B control register

PBCR (0xFFFF_F054)

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

Port B function register

PBFC (0xFFFF_F058)

	7	6	5	4	3	2	1	0
Bit Symbol	PB7F	PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	PB0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: HSCLK1 /HCTS1	0: PORT 1: HRXD1	0: PORT 1: HTXD1	0: PORT 1: HSCLK0 /HCTS0	0: PORT 1: HRXD0	0: PORT 1: HTXD0	0: PORT 1: TB3IN1	0: PORT 1: TB3IN0

Port B pull-up control register

PBPE (0xFFFF_F05C)

	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

Port B open drain control register

PBODE (0xFFFF_F034)

	7	6	5	4	3	2	1	0
Bit Symbol	PB7ODE	–	PB5ODE	PB4ODE	–	PB2ODE	–	
Read/Write	R/W	R	R/W		R	R/W	R	
After reset	0	0	0	0	0	0	0	
Function	0: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain	0: CMOS 1: Open drain	"0" is read.	0: CMOS 1: Open drain	"0" is read.	

7.12 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 register PCFC. A reset allows all bits of the output latch PC to be set to "1," all bits of PCCR and PCFC to be cleared to "0," and the port C to be put in input mode.

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 PC4 perform the 32-bit compare output function, and PC5 through PC7 input and output SBI.

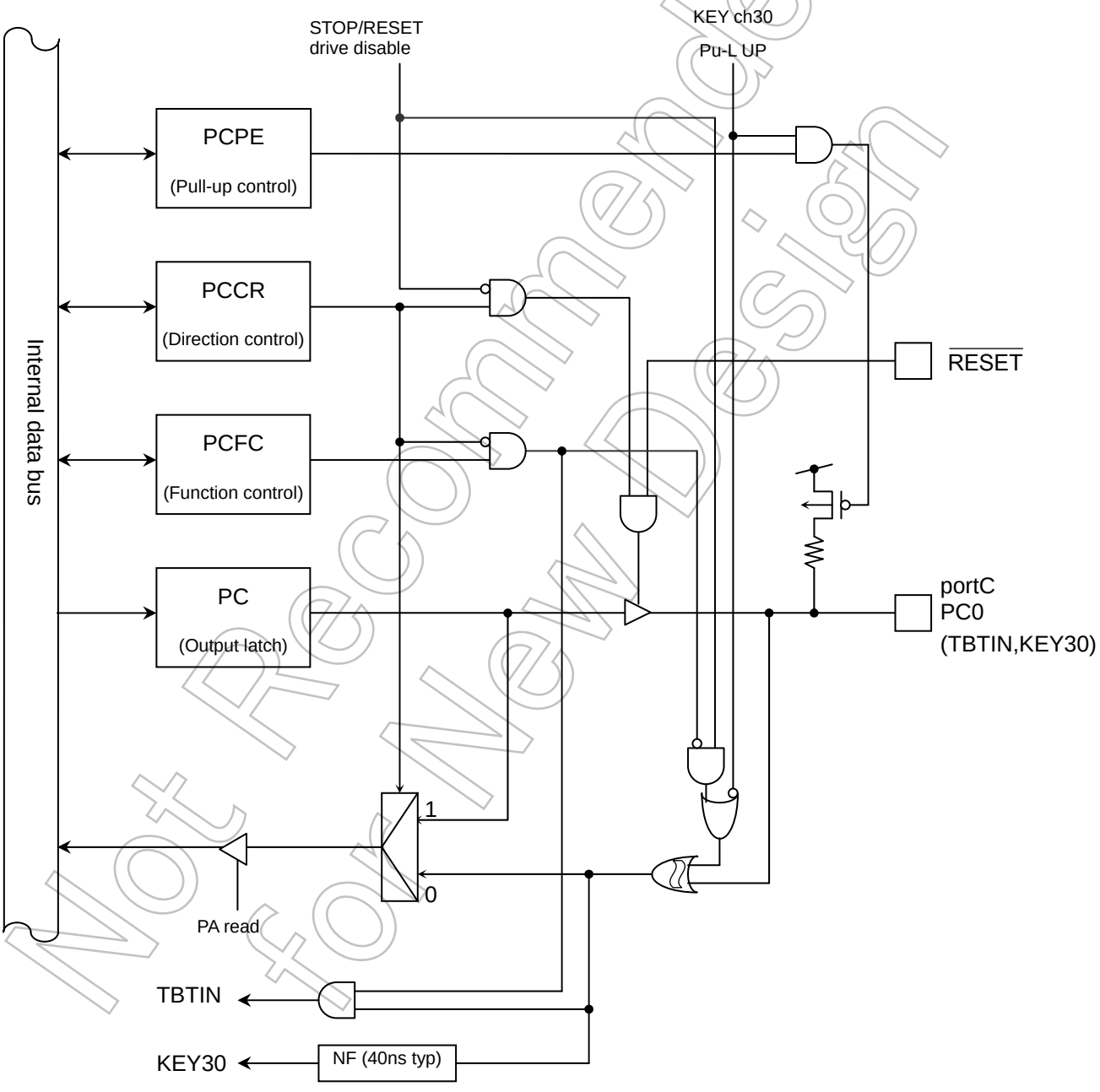


Fig. 7-35 Port C (PC0)

If the port C goes into STOP mode when the KEY/TBTIN input is enabled, inputs are always accepted. To inhibit inputs, switch to PORT using the function register.
• Port: Inputs are accepted only during a read. • KEY/TBTIN: Inputs are always accepted.

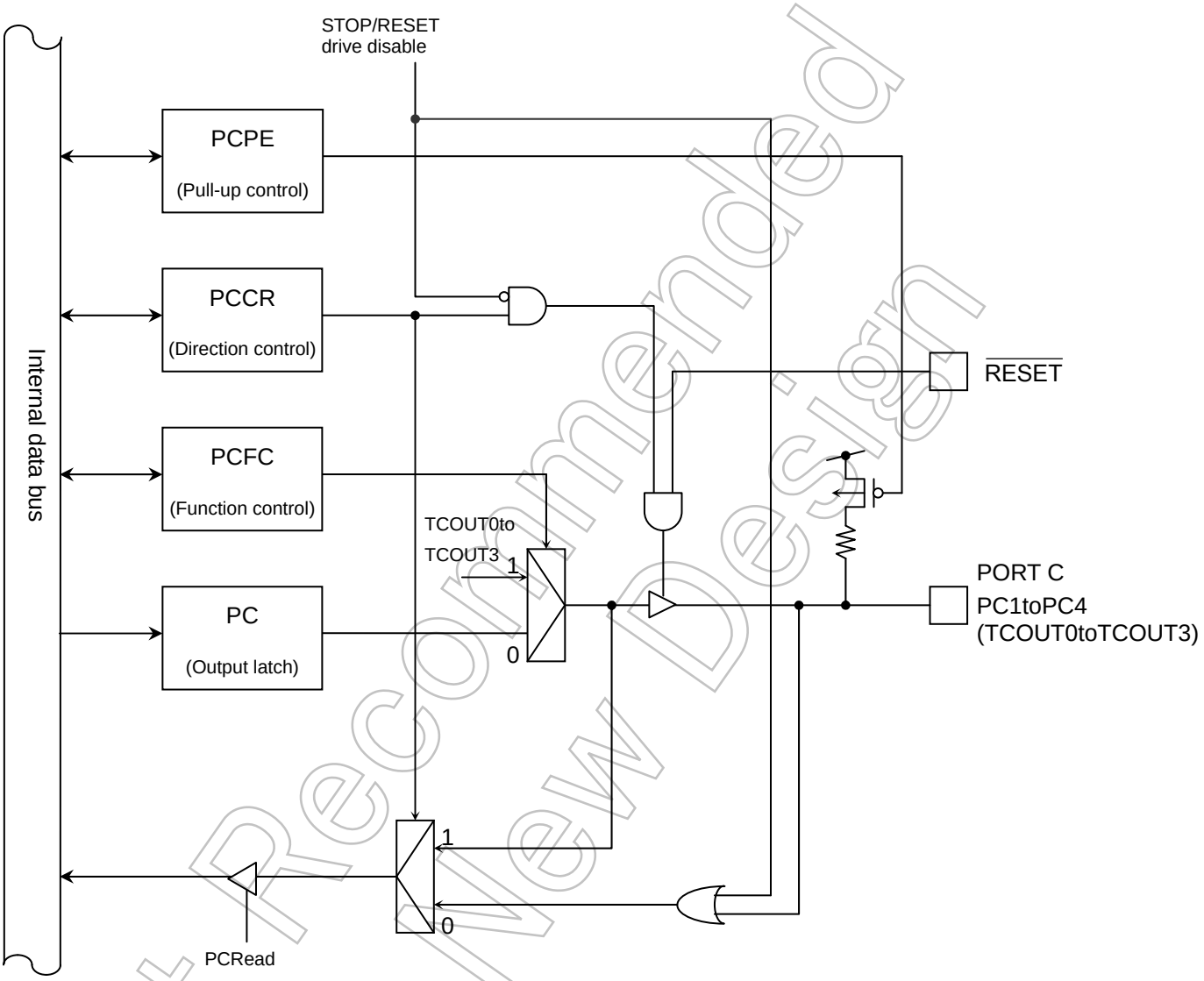


Fig. 7- 36 Port C (PC1 through PC4)

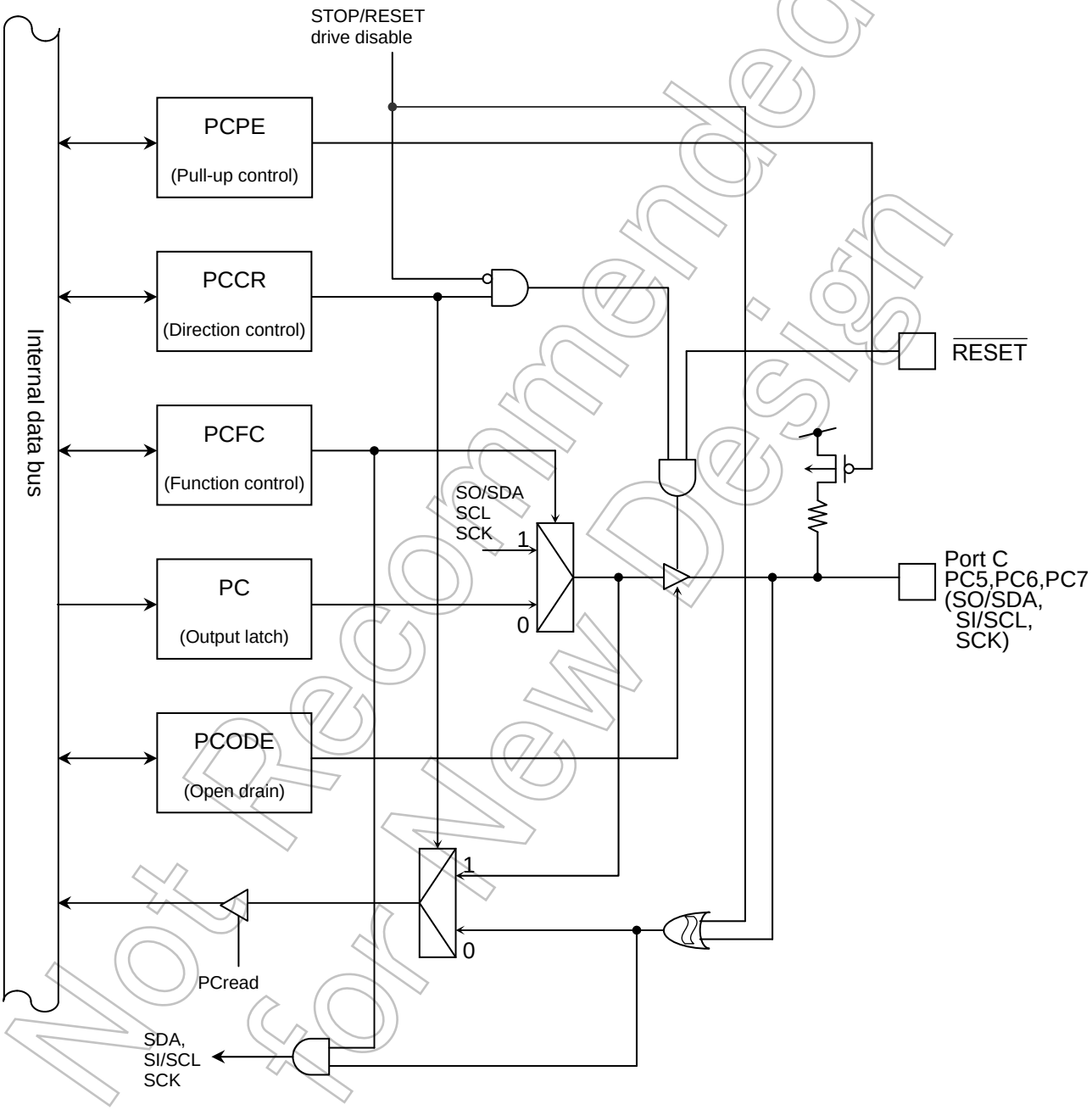


Fig. 7-37 Port C (PC5-PC7)

Port C register

PC (0xFFFF_F051)

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

Port C control register

PCCR (0xFFFF_F055)

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

Port C function register

PCFC (0xFFFF_F059)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7F	PC6F	PC5F	PC4F	PC3F	PC2F	PC1F	PC0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: SCK	0: PORT 1: SI /SCL	0: PORT 1: SO /SDA0	0: PORT 1: TCOUT3	0: PORT 1: TCOUT2	0: PORT 1: TCOUT1	0: PORT 1: TCOUT0	0: PORT /KEY30 1: TBTIN /KEY30

Port C pull-up control register

PCPE (0xFFFF_F05D)

	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 open drain control register

PCODE (0xFFFF_F035)

	7	6	5	4	3	2	1	0
Bit Symbol	PC7ODE	PC6ODE	PC5ODE	–				
Read/Write	R/W	R/W	R/W	R				
After reset	0	0	0	0				
Function	0: CMOS 1: Open drain	0: CMOS 1: Open drain	0: CMOS 1: Open drain	"0" is read.				

7.13 Port D (PD0 to PD6)

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 register PDFC. A reset allows all bits of the output latch PD to be set to "1," all bits of PDCR and PDFC to be cleared to "0," and the port D to be put in input mode.

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 and A/D triggers into the A/D converter.

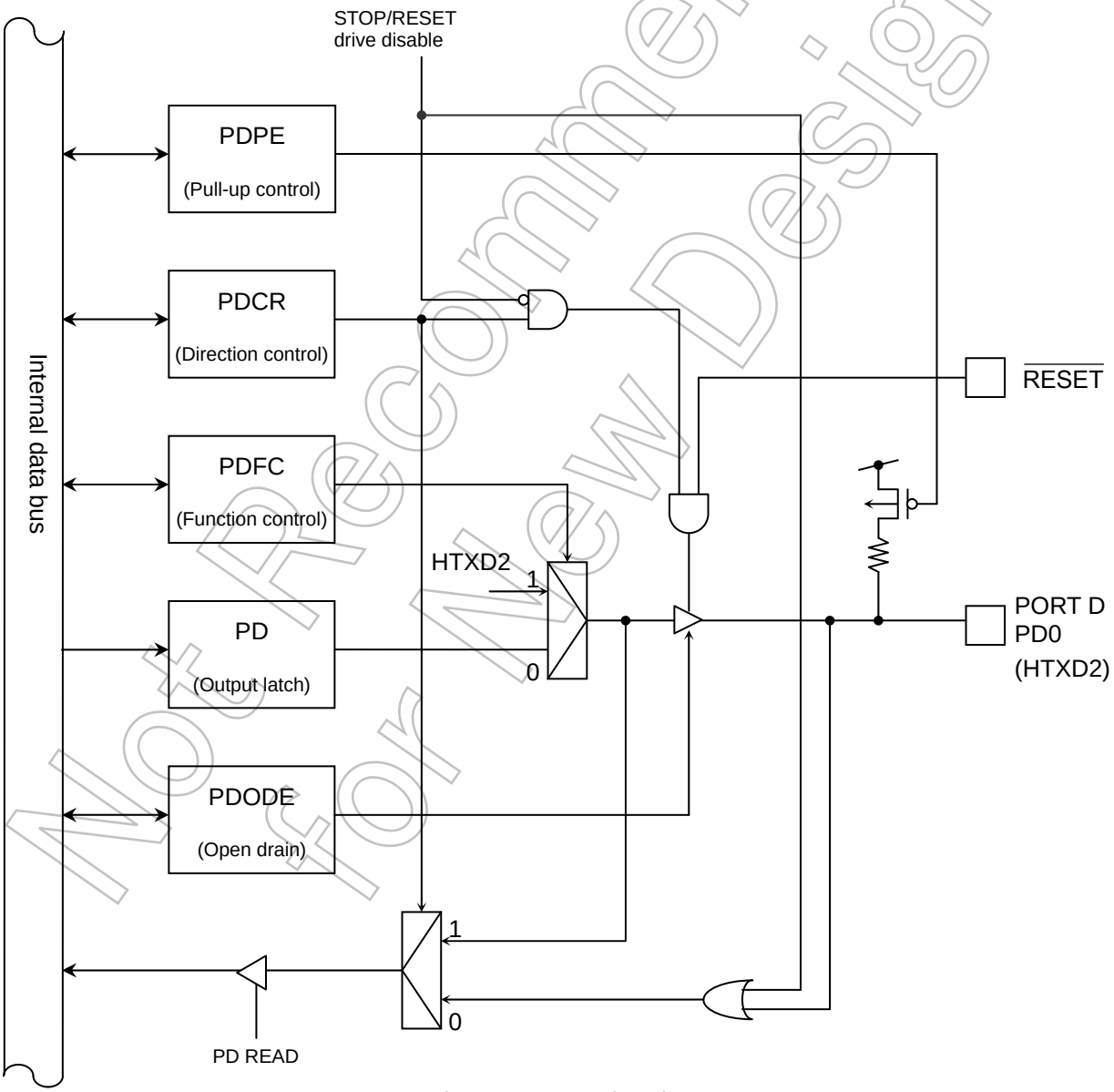


Fig. 7-38 Port D (PD0)

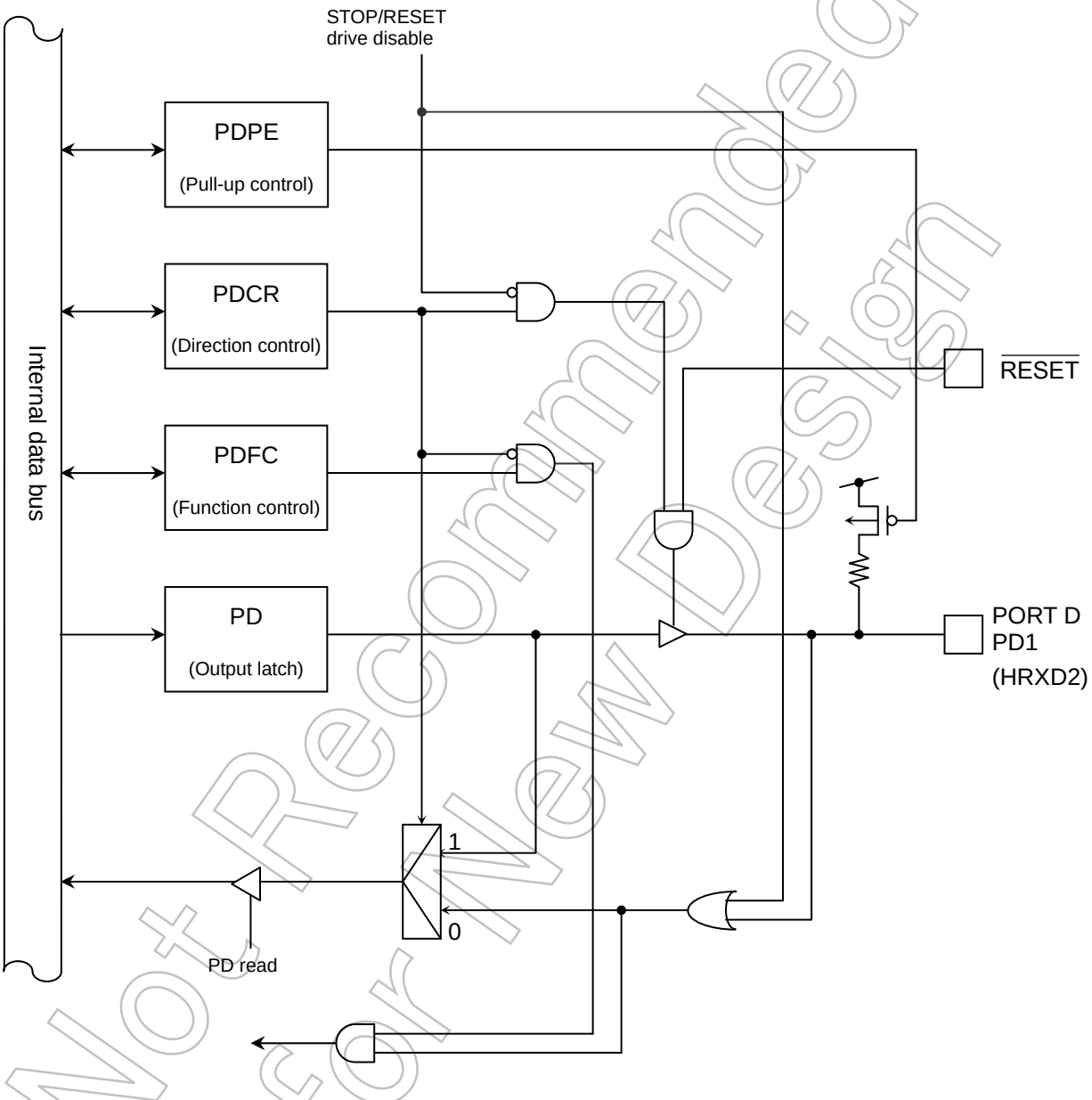


Fig. 7-39 Port D (PD1)

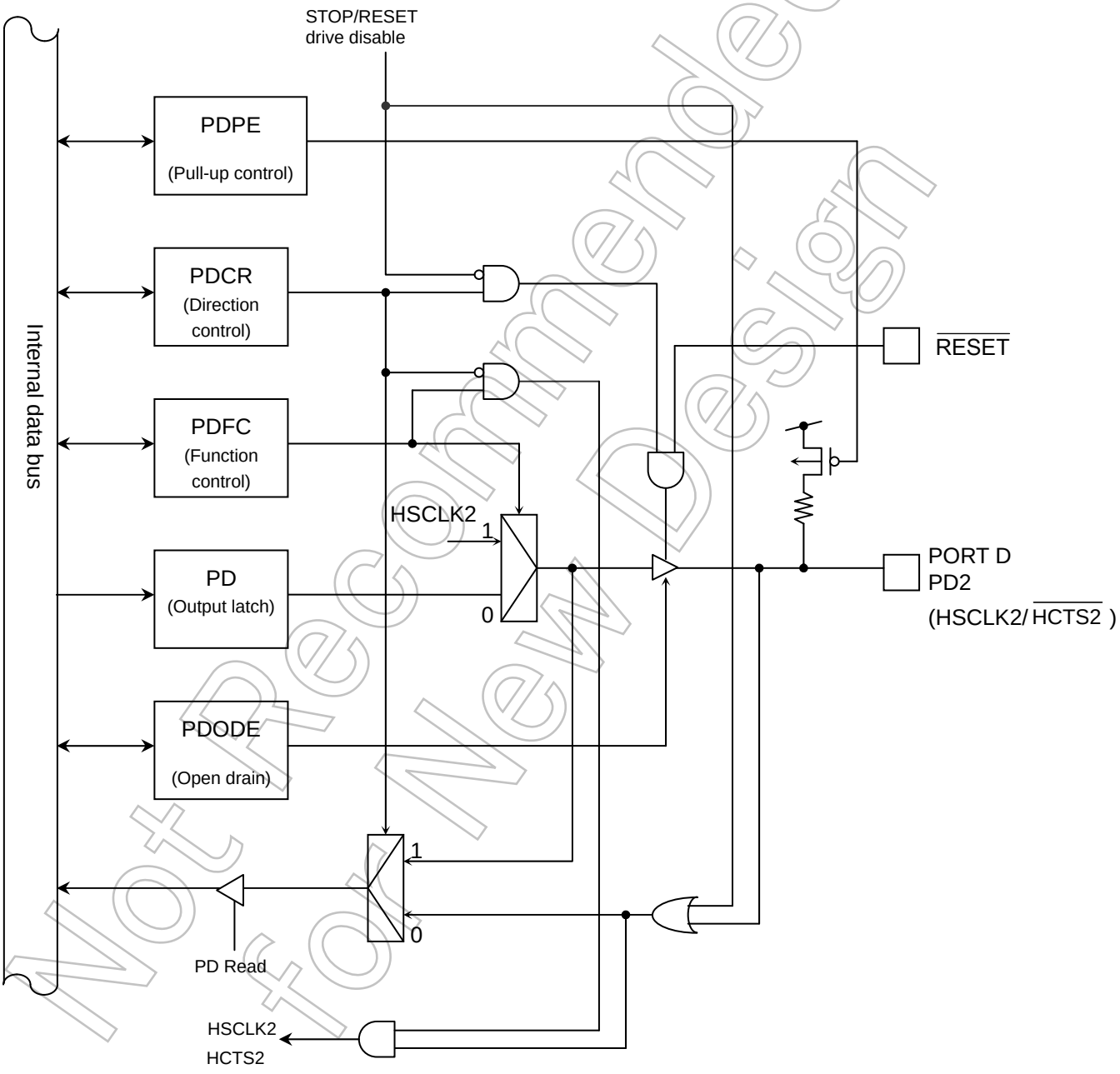


Fig. 7-40 Port D (PD2)

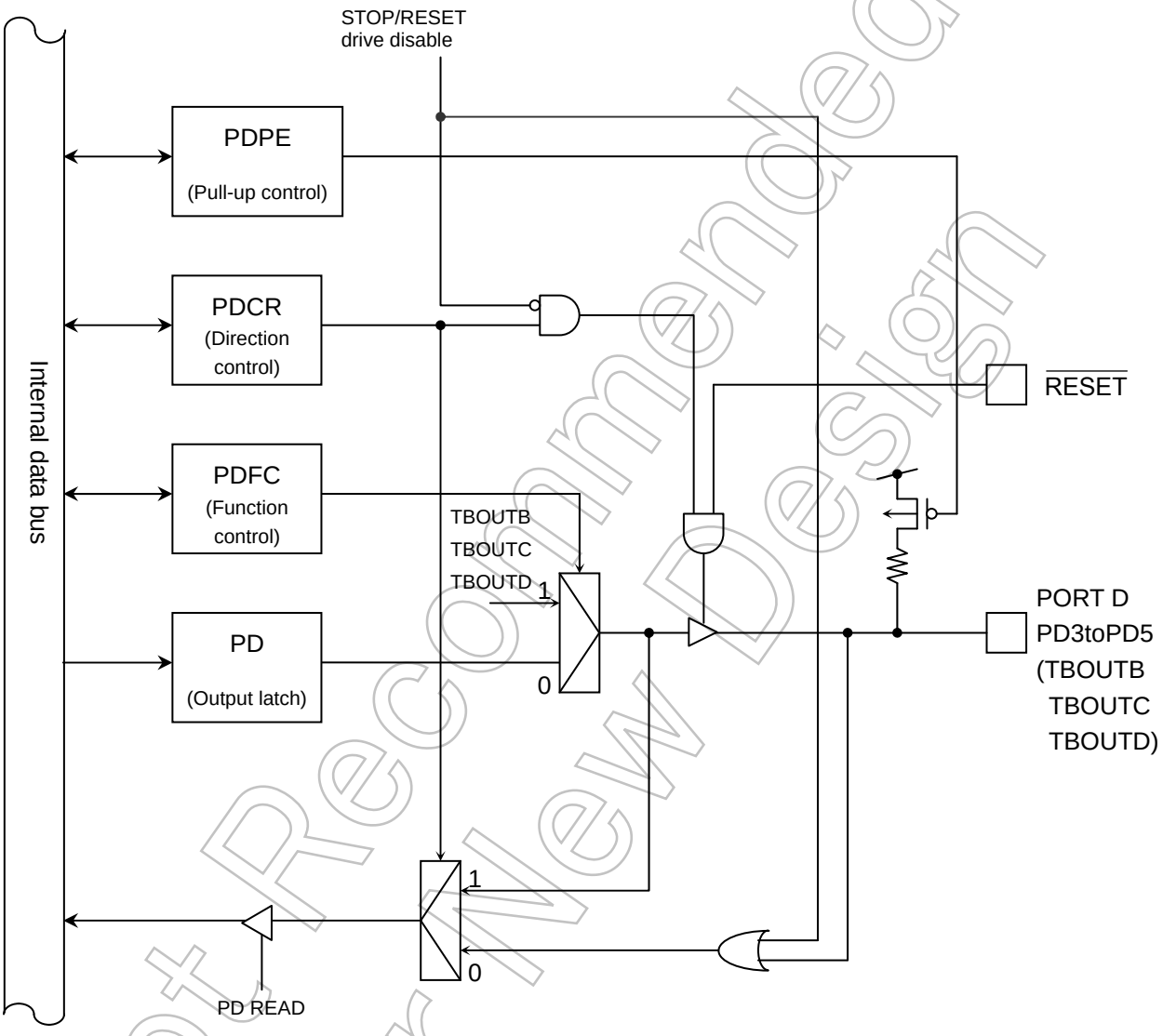


Fig. 7- 41 Port D (PD3 through PD5)

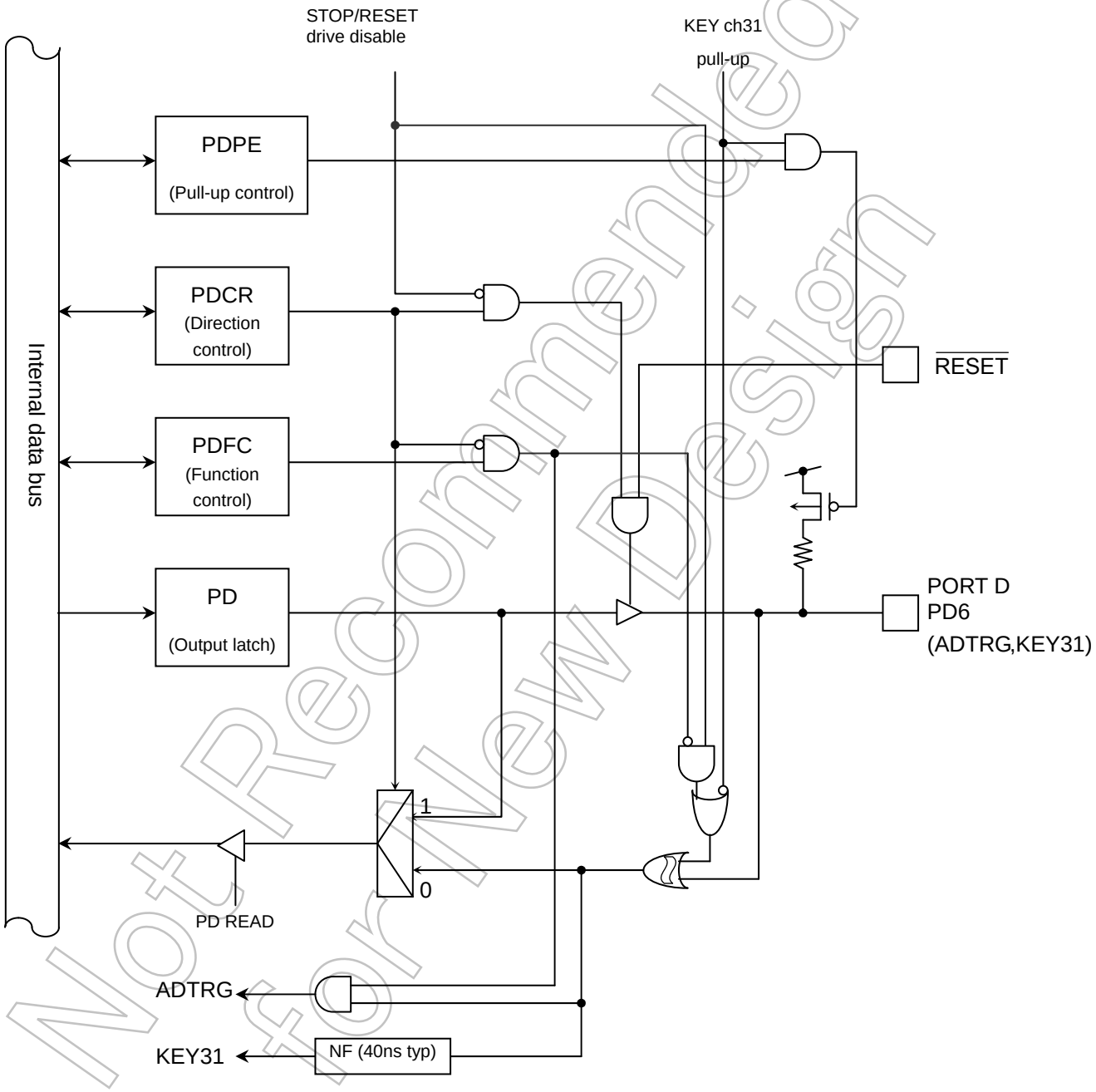


Fig. 7- 42 Port D (PD6)

If the port D goes into STOP mode when the KEY/ADTRG input is enabled, inputs are always accepted.
To inhibit inputs, switch to PORT using the function register.
• Port: Inputs are accepted only during a read. • KEY/ADTRG: Inputs are always accepted.

Port D register

PD (0xFFFF_F052)

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

Port D control register

PDCR (0xFFFF_F056)

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

Port D function register

PDFC (0xFFFF_F05A)

	7	6	5	4	3	2	1	0
Bit Symbol	–	PD6F	PD5F	PD4F	PD3F	PD2F	PD1F	PD0F
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	"0" is read.	0: PORT /KEY31 1: ADTRG /KEY31	0: PORT 1: TBDOUT	0: PORT 1: TBCOUT	0: PORT 1: TBBOUT	0: PORT 1: HSCLK2/ HCTS2	0: PORT 1: HRXD2	0: PORT 1: HTXD2

Port D pull-up control register

PDPE (0xFFFF_F05E)

	7	6	5	4	3	2	1	0
Bit Symbol		PED6	PED5	PED4	PED3	PED2	PED1	PED0
Read/Write	R	R/W						
After reset	0	0	0	0	0	0	0	0
Function	A written value can be read.	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 open drain control register

PDODE (0xFFFF_F036)

	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

7.14 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 PEFC. 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.

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

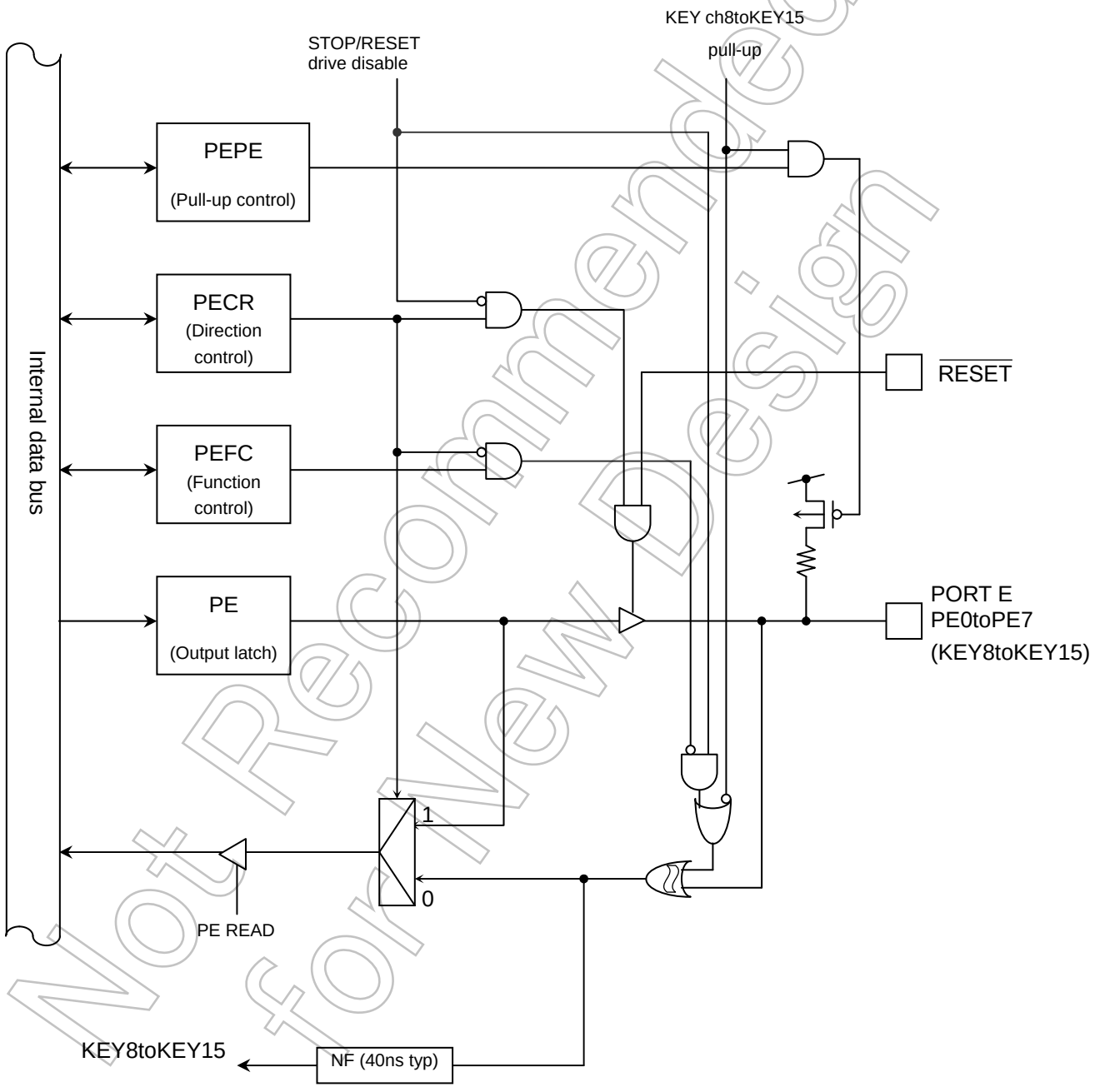


Fig. 7- 43 Port E (PE0 through PE7)

If the port E goes into STOP mode when the KEY input is enabled, inputs are always accepted.
To inhibit inputs, switch to PORT using the function register
• Port: Inputs are accepted only during a read. • KEY: Inputs are always accepted.

Port E register

PE (0xFFFF_F053)

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

Port E control register

PECR (0xFFFF_F057)

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

Port E function register

PEFC (0xFFFF_F05B)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7F	PE6F	PE5F	PE4F	PE3F	PE2F	PE1F	PE0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT / KEY15 1: KEY15	0: PORT / KEY14 1: KEY14	0: PORT / KEY13 1: KEY13	0: PORT / KEY12 1: KEY12	0: PORT / KEY11 1: KEY11	0: PORT / KEY10 1: KEY10	0: PORT / KEY09 1: KEY09	0: PORT / KEY08 1: KEY08

Port E pull-up control register

PEPE (0xFFFF_F05F)

	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

7.15 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 register PFFC. A reset allows all bits of the output latch PF to be set to "1," all bits of PFCR and PFFC to be cleared to "0," and the port F to be put in input mode. Besides the input/output port function, the port F performs the key-on wake-up input function, PF0 through PF3 perform a 32-bit timer capture input function, and PF4 through PF7 perform a 32-bit timer compare output function.

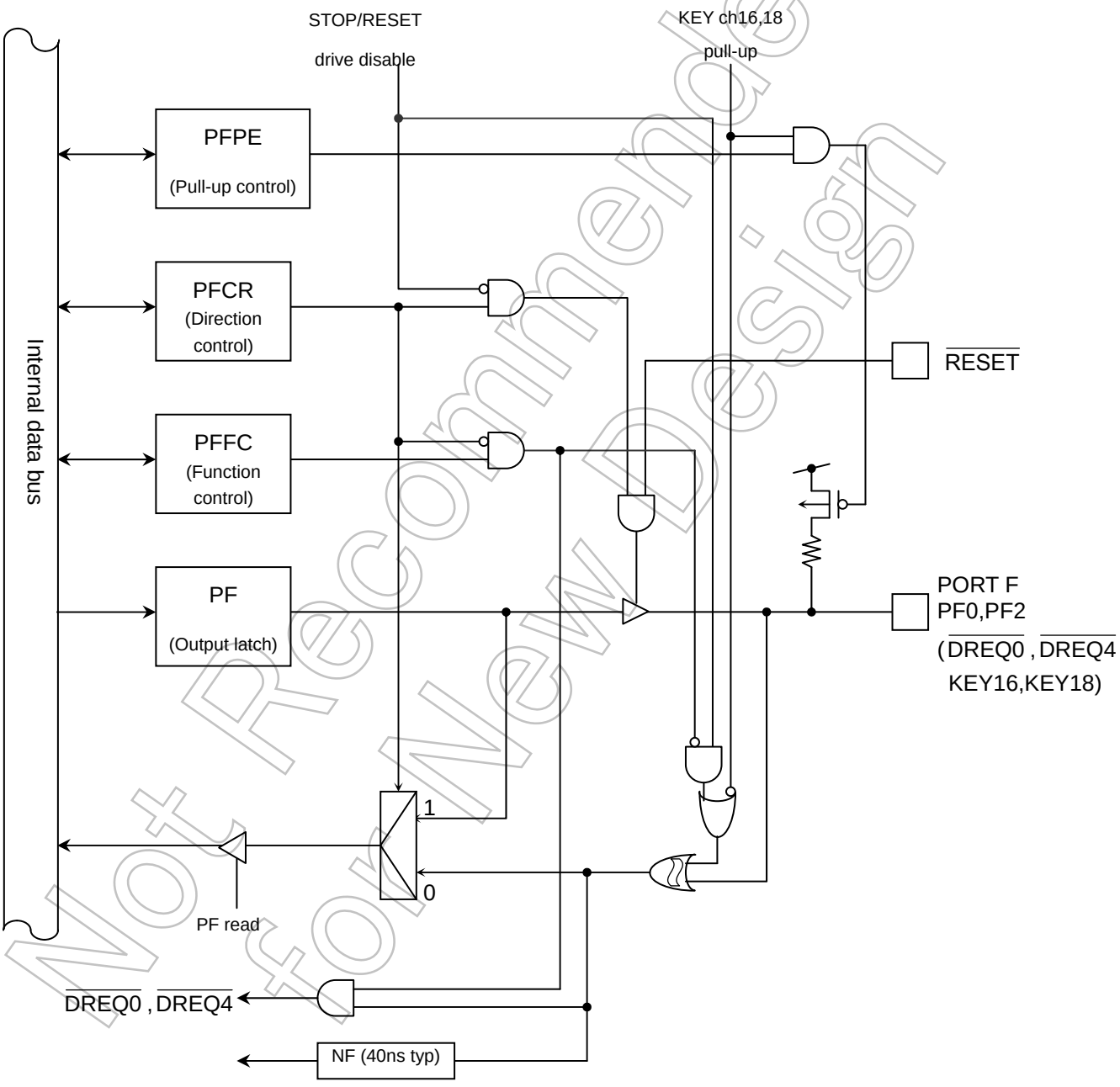


Fig. 7-44 Port F (PF0, PF2)

If the port F goes into STOP mode when the KEY/DREQ input is enabled, inputs are always accepted. To inhibit inputs, switch to PORT using the function register.

• Port: Inputs are accepted only during a read. • KEY/DREQ: Inputs are always accepted.

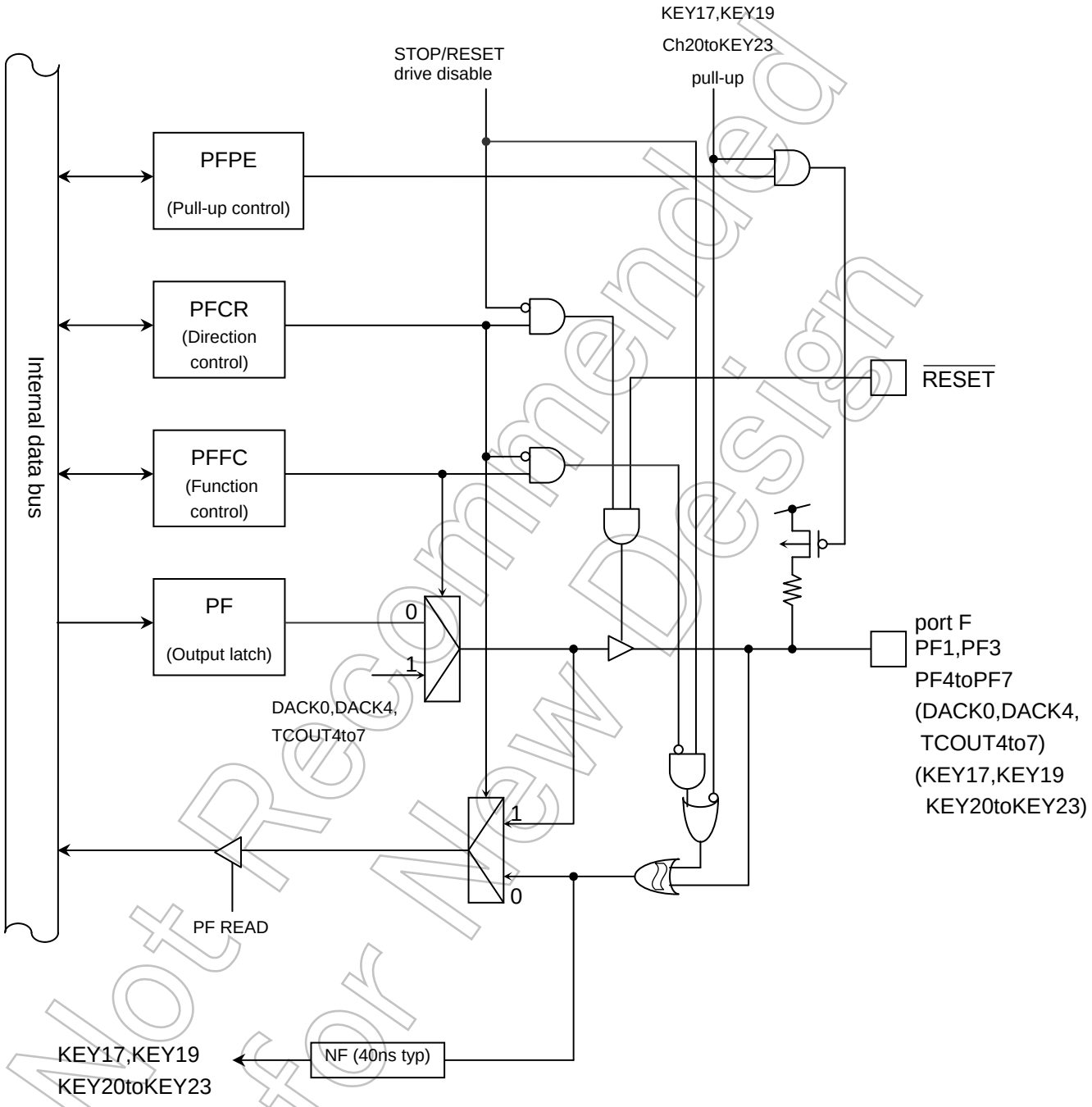


Fig. 7-45 Port F (PF1, PF3,PF4-PF7)

If the port F goes into STOP mode when the KEY input is enabled, inputs are always accepted.
To inhibit inputs, switch to PORT using the function register.

• Port: Inputs are accepted only during a read. • KEY: Inputs are always accepted.

Port F register

PF (0xFFFF_F060)

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

Port F control register

PFCR (0xFFFF_F064)

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

Port F function register

PFFC (0xFFFF_F068)

	7	6	5	4	3	2	1	0
Bit Symbol	PF7F	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT / KEY23 1: TCOUT7 / KEY23	0: PORT / KEY22 1: TCOUT6 / KEY22	0: PORT /KEY21 1: TCOUT5 /KEY21	0: PORT /KEY20 1: TCOUT4 /KEY20	0: PORT /KEY19 1: DACK4 /KEY19	0: PORT /KEY18 1: DREQ4 /KEY18	0: PORT /KEY17 1: DACK0 /KEY17	0: PORT /KEY16 1: DREQ0 /KEY16

Port F pull-up control register

PFPE (0xFFFF_F06C)

	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

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

Besides the input/output port function, the port G outputs data tracing signals for debugging. The port G gets ready to output data tracing signals according the debug level, independent of the register setting. If DSU-ICE is to be used for debugging, the port G outputs the signal for EJTAG. Therefore, it is recommended not to use the port G as an input/output port.

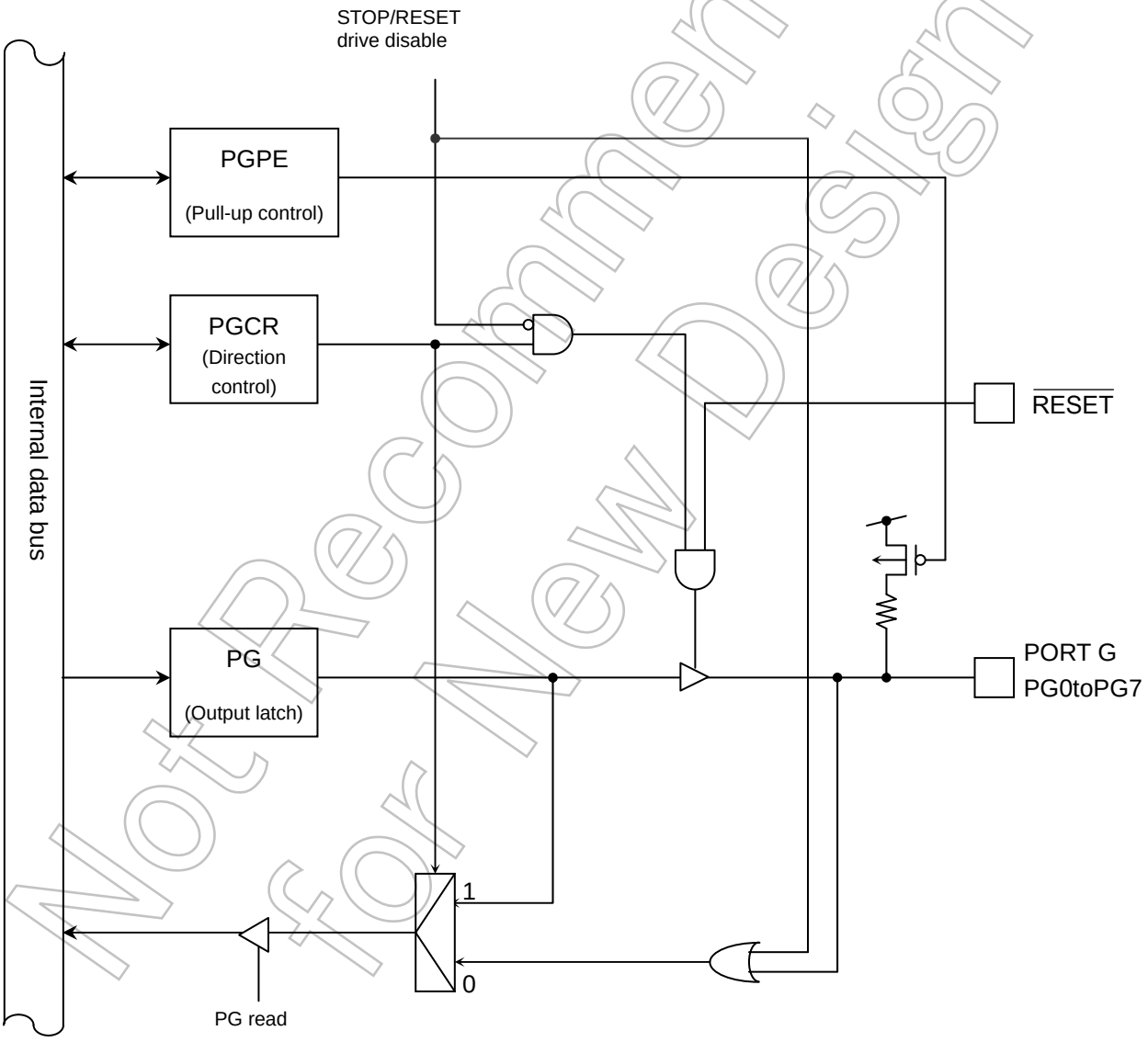


Fig. 7-46 Port G (PG0 through PG7)

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

Port G register

PG (0xFFFF_F061)

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

Port G control register

PGCR (0xFFFF_F065)

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

Port G pull-up control register

PGPE (0xFFFF_F06D)

	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

	Level 0	Level 1	Level 2		Level 3
PG Port	PORT	PORT	PGFC = 0	PORT	TPD
			PGFC = 1	TPD	
PH Port	PORT	TPC	PGFC = 0	TPD	TPC
			PGFC = 1	PORT	

Table 7-7 Pin states of the ports G and H relative to debug levels

* When EJTAG is used, the PGFC setting is made using a tool.

7.17 Port H (PH0 through PH7)

The port H is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits by using the control register PHCR. A reset allows 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.

Besides the port function, the port H outputs TPC/TPD of DSU, and is determined by the levels of PGFC and EJTAG. If DSU-ICE is used for debugging, the port H outputs the signal for EJTAG. Therefore, it is recommended not to use the port H as an input/output port.

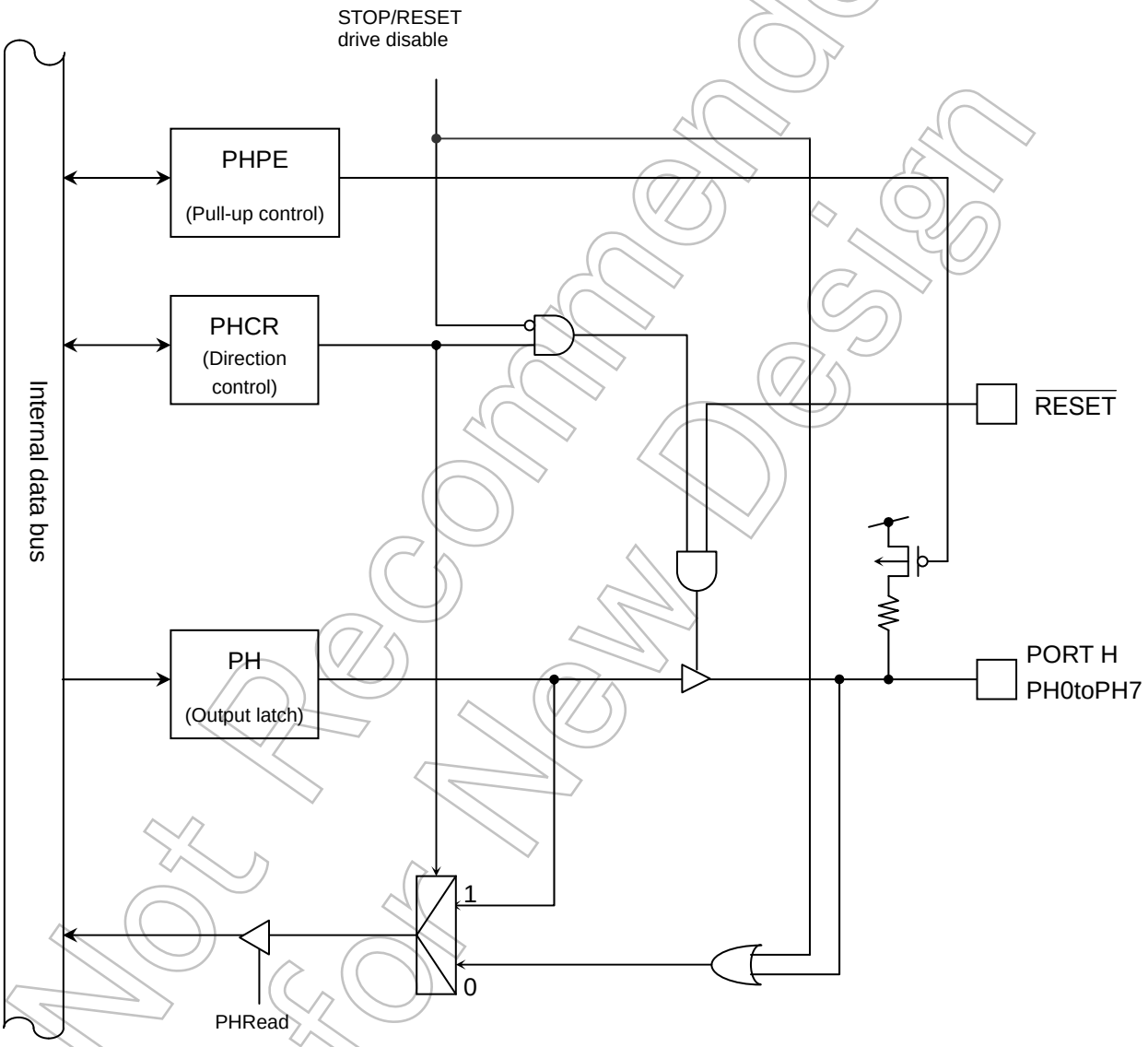


Fig. 7-47 Port H (PH0 through PH7)

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

Port H register

PH (0xFFFF_F062)

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

Port H control register

PHCR (0xFFFF_F066)

	7	6	5	4	3	2	1	0
Bit Symbol	PH7C	PH6C	PH5C	PH4C	PH3C	PH2C	PH1C	PH0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<<See P6FC>>							

Port H pull-up control register

PHPE (0xFFFF_F06E)

	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

8. External Bus Interface

The TMP19A43 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 7 clocks can be automatically inserted.
  - A wait can be inserted via the $\overline{\text{WAIT}}$ / $\overline{\text{RDY}}$ pin.
- Data bus width
  Either an 8- or 16-bit width can be set for each block.
- Recovery cycle (read/write)
  If an external bus cycle is in progress, a dummy cycle of up to 2 clocks can be inserted and this dummy cycle can be specified for each block.
- Recovery cycle (chip selector)
  When an external bus is selected, a dummy cycle of up to 1 clock can be inserted and this dummy cycle can be specified for each block.
- Bus arbitration function

8.1 Address and Data Pins

(1) Address and data pin settings

The TMP19A43 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.

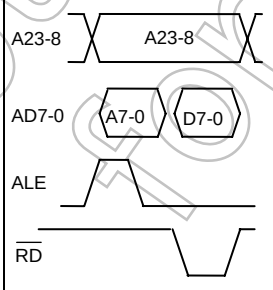
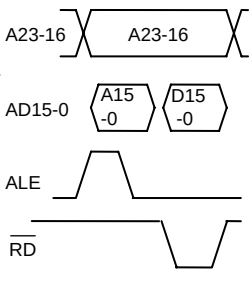
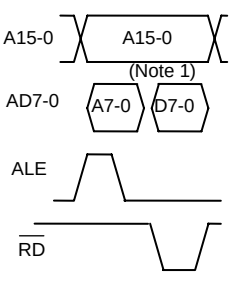
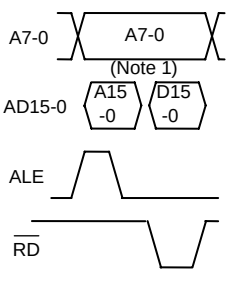
Table 8-1 Bus Mode, Address and Data Pins

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

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

In the multiplex mode, the four types of functions can be selected, as shown in Table 8-2, by setting the port registers (PnCR and PnFC).

Table 8-2 Address and Data Pins in the Multiplex Mode

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

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

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

(Note 3) Port 0 automatically becomes a data and address/data bus pin when an external memory is being accessed.

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

(2) Address HOLD when an internal area is accessed

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

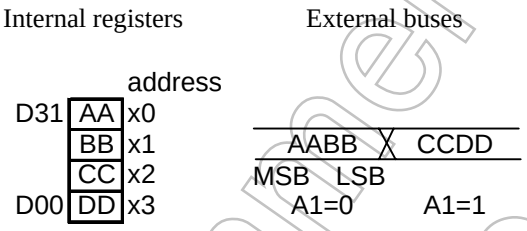
8.2 Data Format

Internal registers and external bus interfaces of the TMP19A43 are configured as described below.

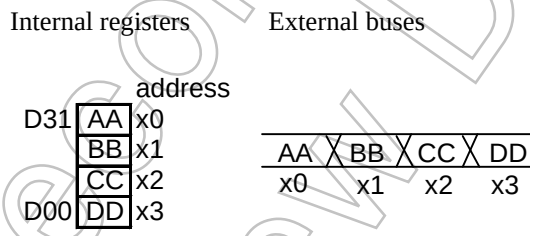
(1) Big-endian mode

① Word access

- 16-bit bus width

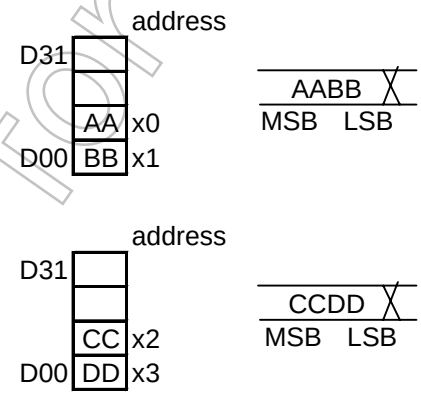


- 8-bit bus width

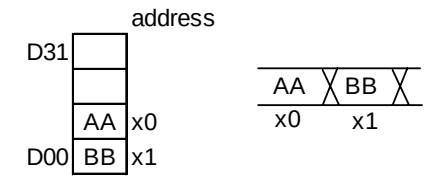


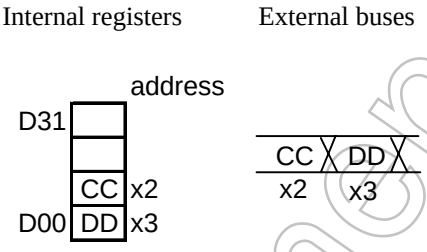
② Half word access

- 16-bit bus width



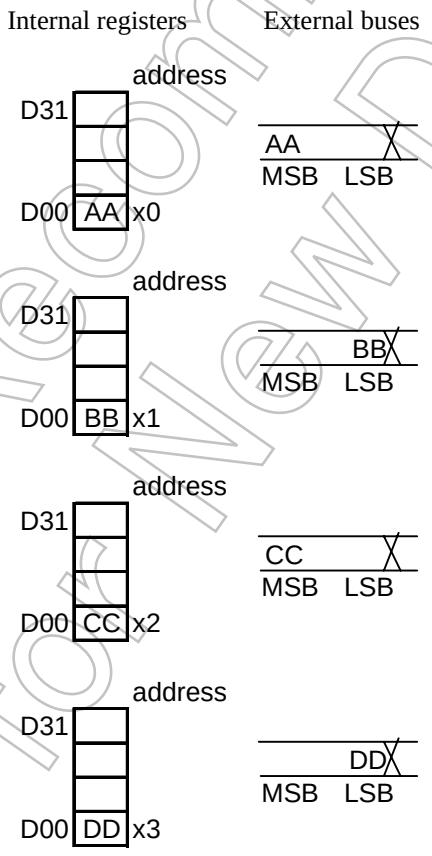
- 8-bit bus width



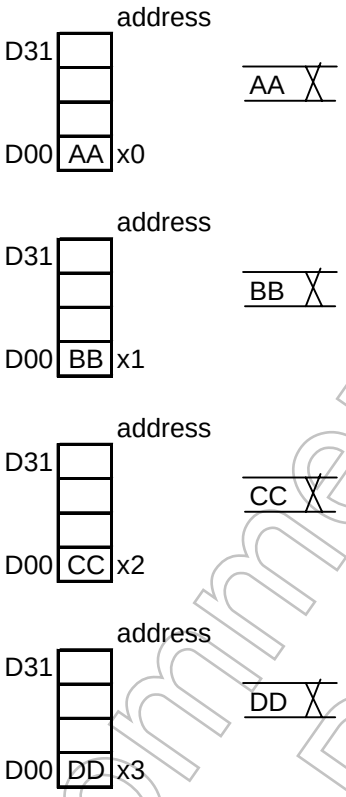


③ Byte access

- 16-bit bus width



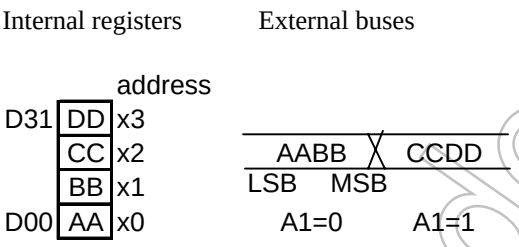
- 8-bit bus width



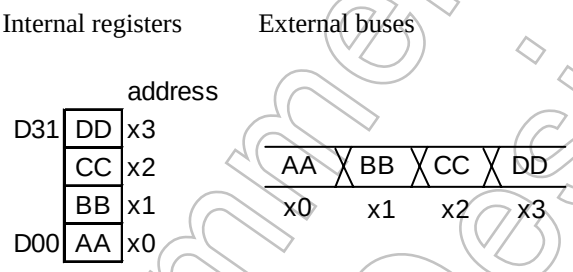
(2) Little-endian mode

① Word access

- 16-bit bus width

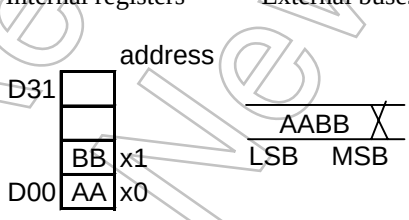


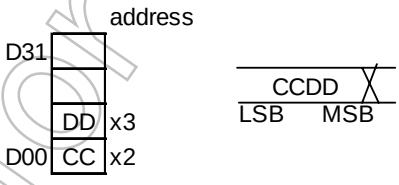
- 8-bit bus width



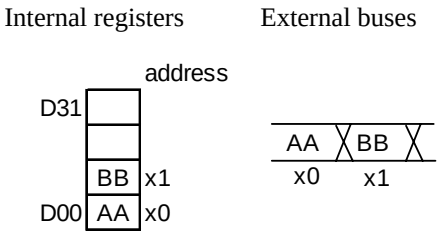
② Half word access

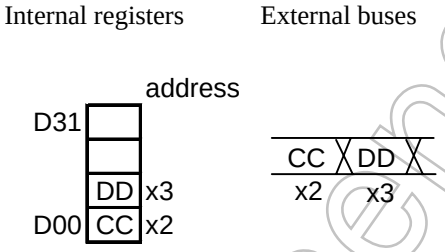
- 16-bit bus width





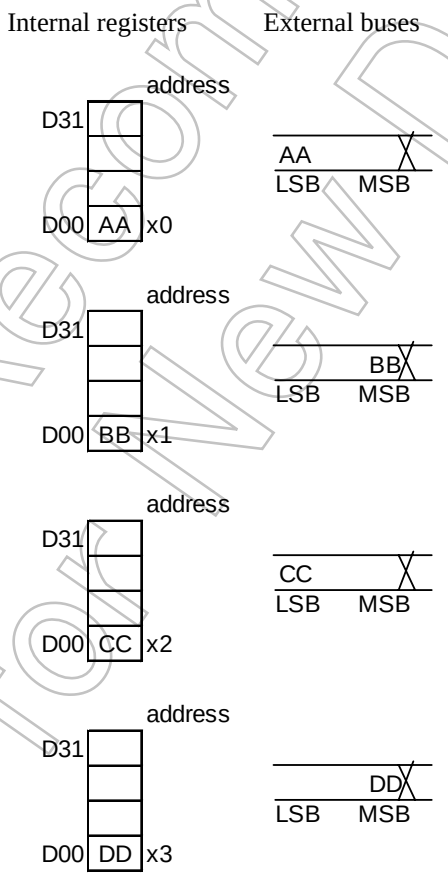
- 8-bit bus width



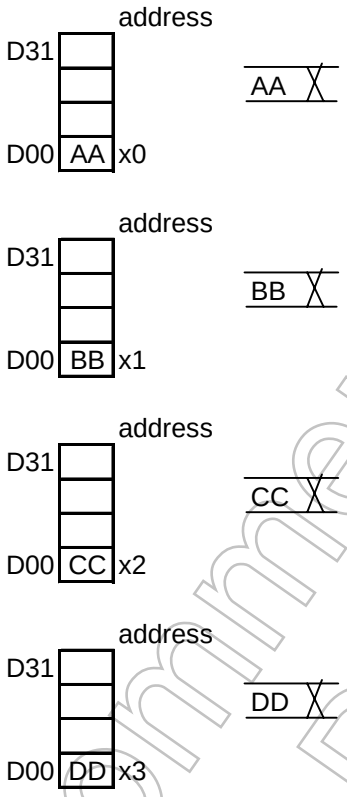


③ Byte access

- 16-bit bus width



- 8-bit bus width



8.3 External Bus Operations (Separate Bus Mode)

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

(1) Basic bus operation

The external bus cycle of the TMP19A43 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 $\overline{RD}$ and $\overline{WR}$ do not become active.

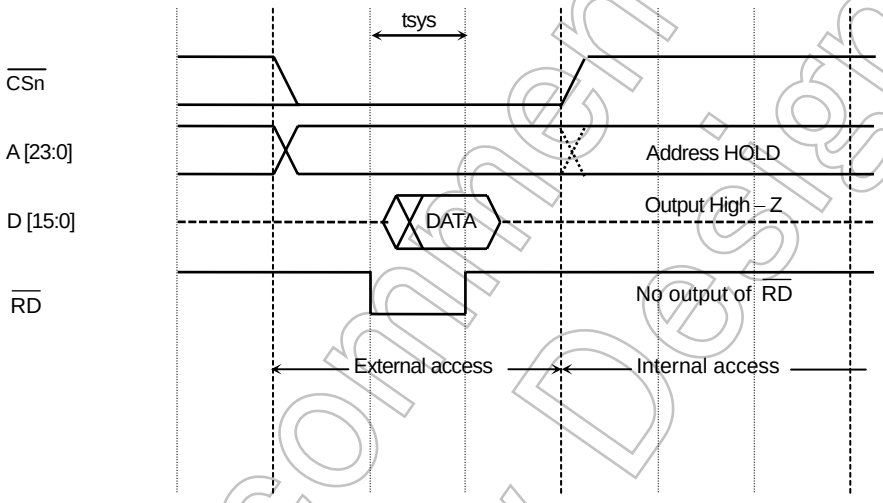


Fig. 8-1 Read Operation Timing Diagram

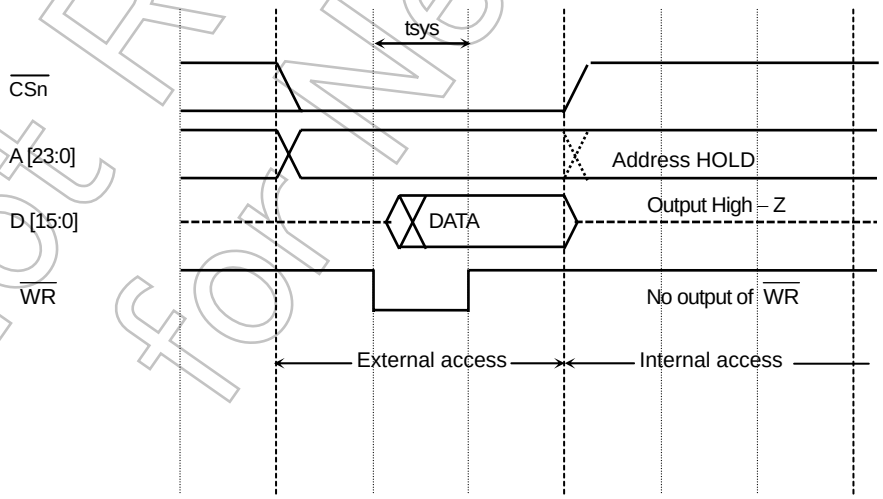


Fig. 8-2 Write Operation Timing Diagram

(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 7 clocks can be automatically inserted.

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

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

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

Fig. 8-3 through Fig. 8-10 show the timing diagrams in which waits have been inserted.

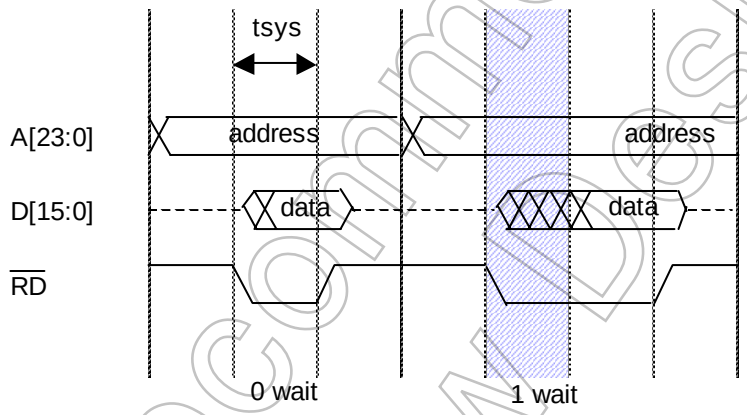


Fig. 8-3 Read Operation Timing Diagram (0 Wait and 1 Wait Automatically Inserted)

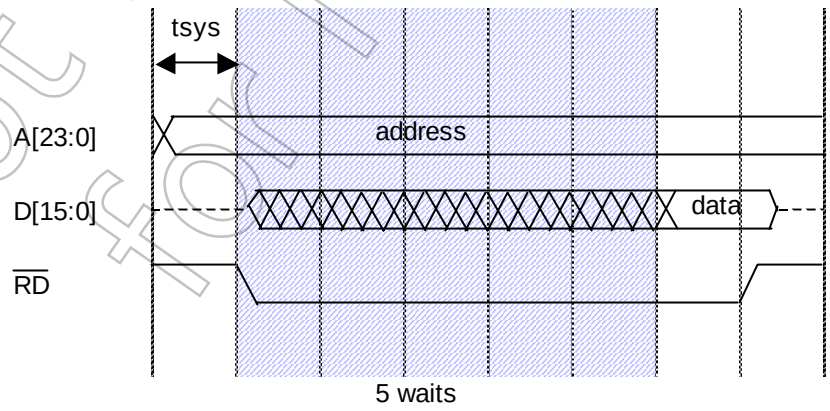


Fig. 8-4 Read Operation Timing Diagram (5 Waits Automatically Inserted)

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.

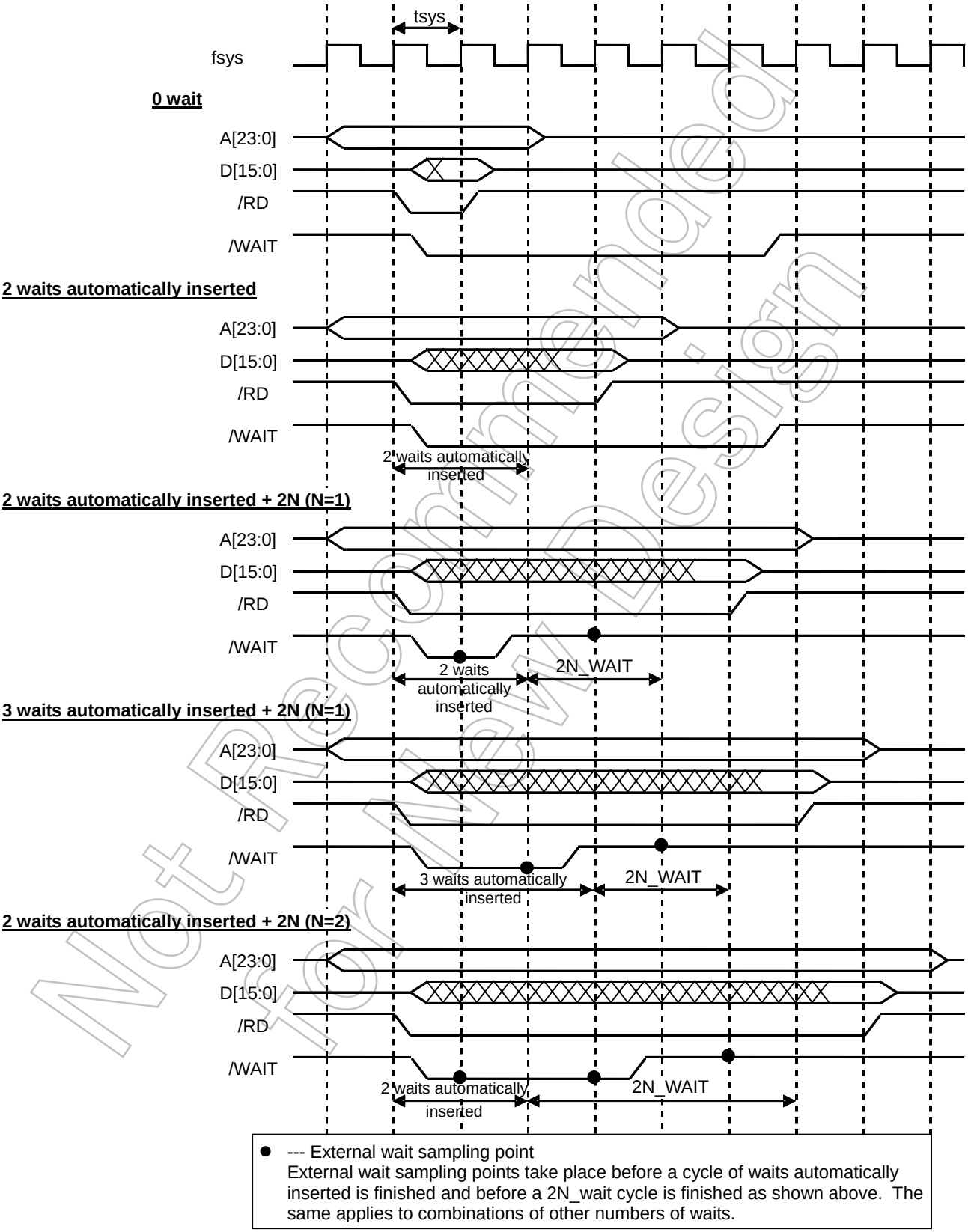


Fig. 8-5 Read Operation Timing Diagram

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.

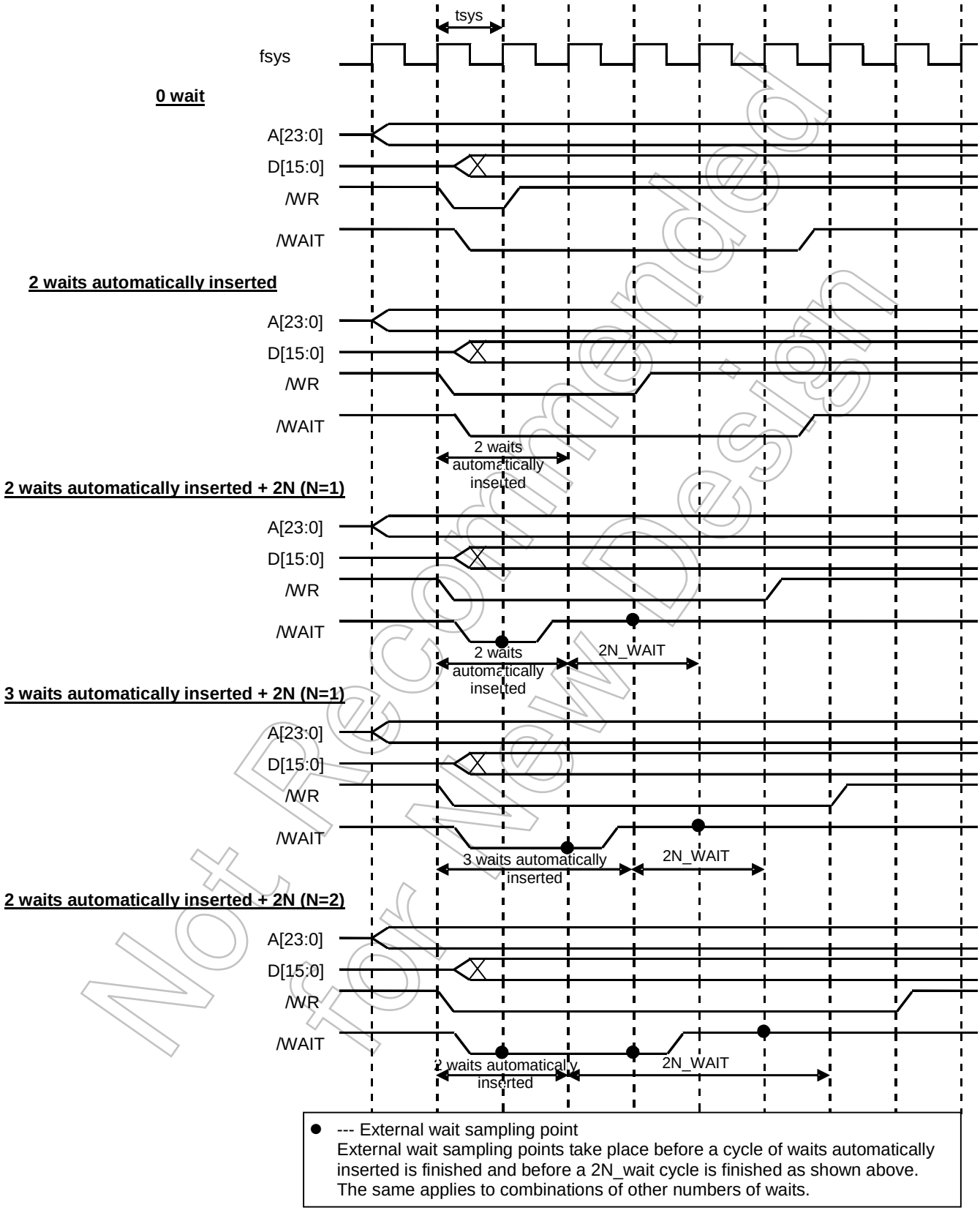


Fig. 8-6 Write Operation Timing Diagram

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.

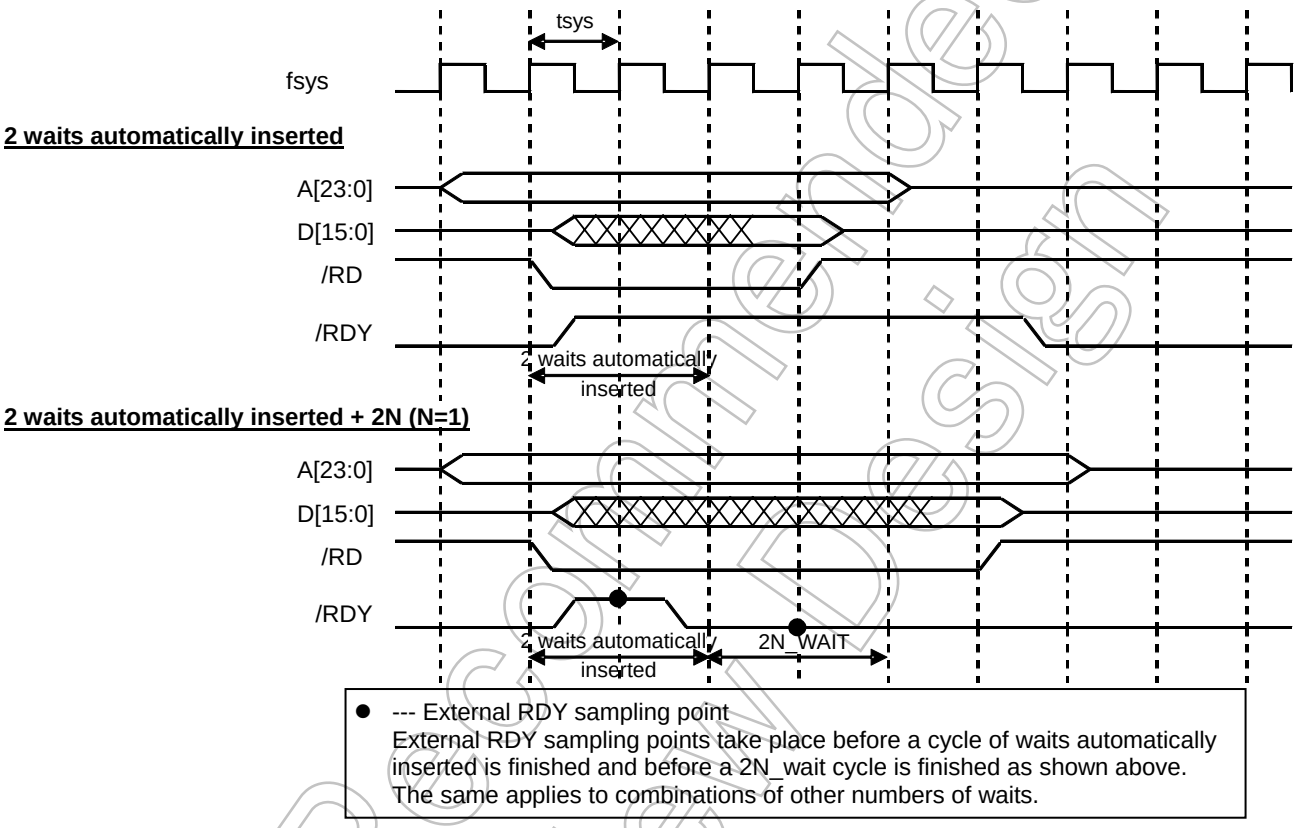


Fig. 8-7 $\overline{\text{RDY}}$ Input and Wait Operation Timing Diagram

(3) Time that it takes before ALE is asserted

When the external bus of the TMP19A43 is used as a multiplexed bus, the ALE width (assert time) can be specified by using the system control register SYSCR3 <ALESEL> in the CG. In the case of a separate bus mode, ALE is not output, but the time from when an address is established to the assertion of the $\overline{RD}$ or $\overline{WR}$ signal is different depending on the SYSCR3<ALESEL>.

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

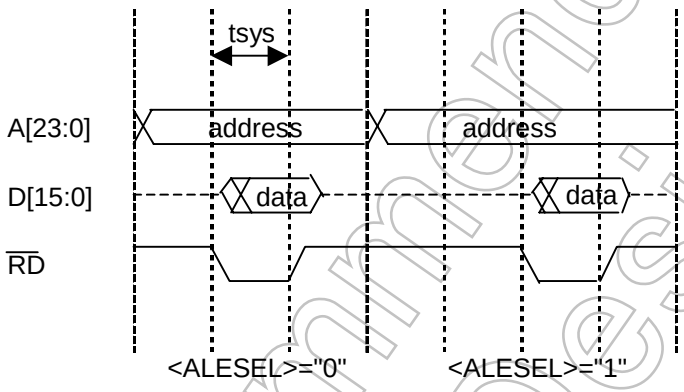


Fig. 8-8 SYSCR3<ALESEL> Set Value and External Bus Operation

(4) Recovery time

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

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip selector and wait controller registers, BmnCS<BnWCV> (write recovery cycle) and <BnRCV> (read recovery cycle). As for the number of dummy cycles, one or two system clocks (internal) can be specified for each block. Fig. 8-9 shows the timing of recovery time insertion.

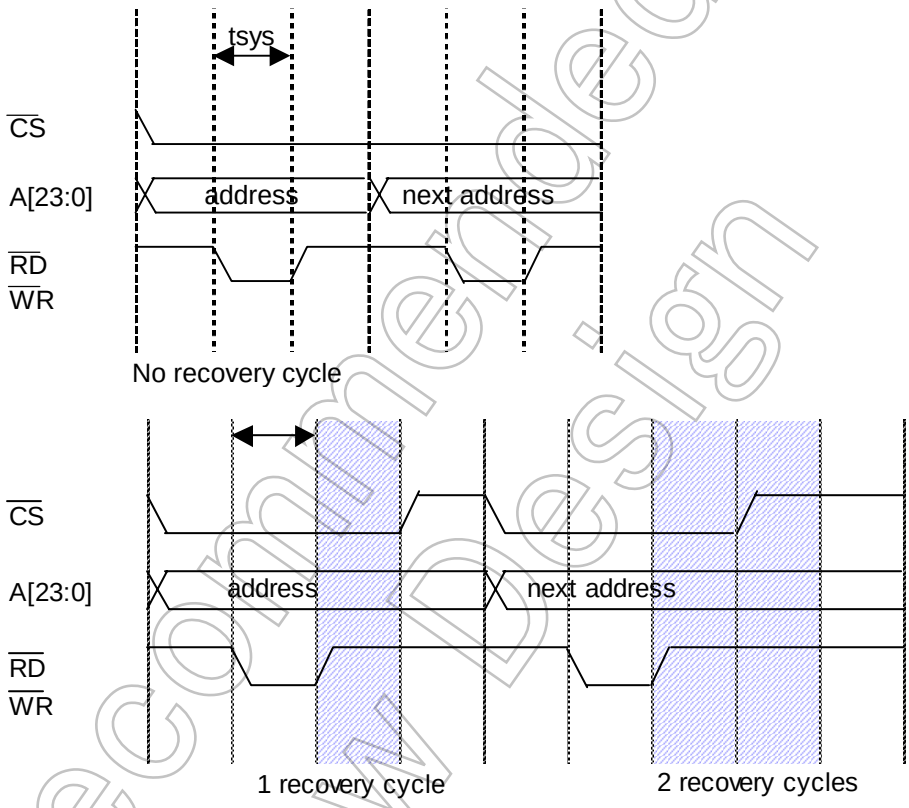


Fig. 8-9 Timing of Recovery Time Insertion

(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. 8-10 shows the timing of recovery time insertion.

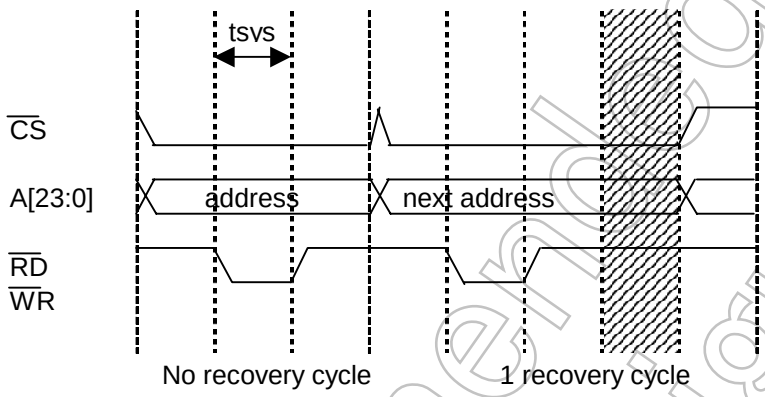


Fig. 8-10 Timing of Recovery Time Insertion

8.4 External Bus Operations (Multiplexed Bus Mode)

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

(1) Basic bus operation

The external bus cycle of the TMP19A43 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 $\overline{RD}$ and $\overline{WR}$ do not become active.

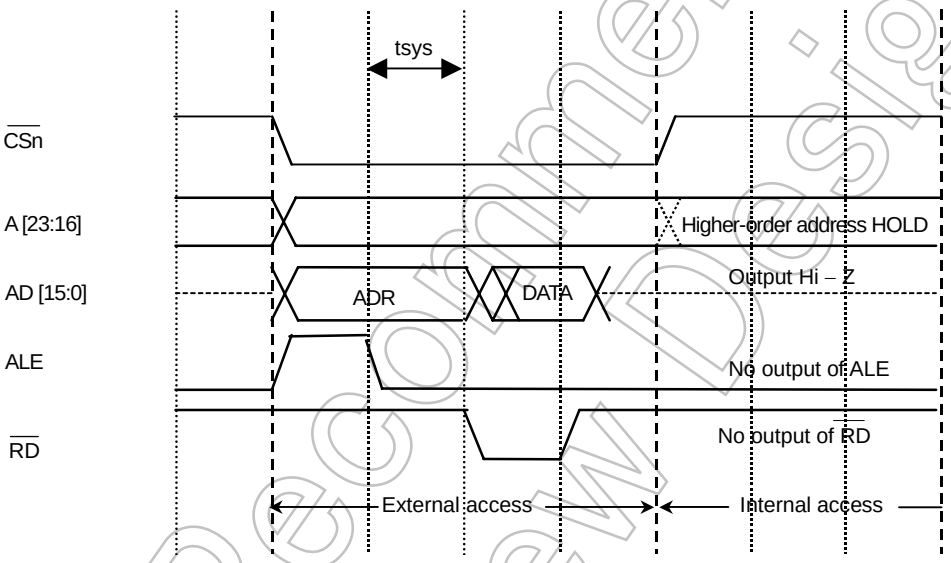


Fig. 8-11 Read Operation Timing Diagram

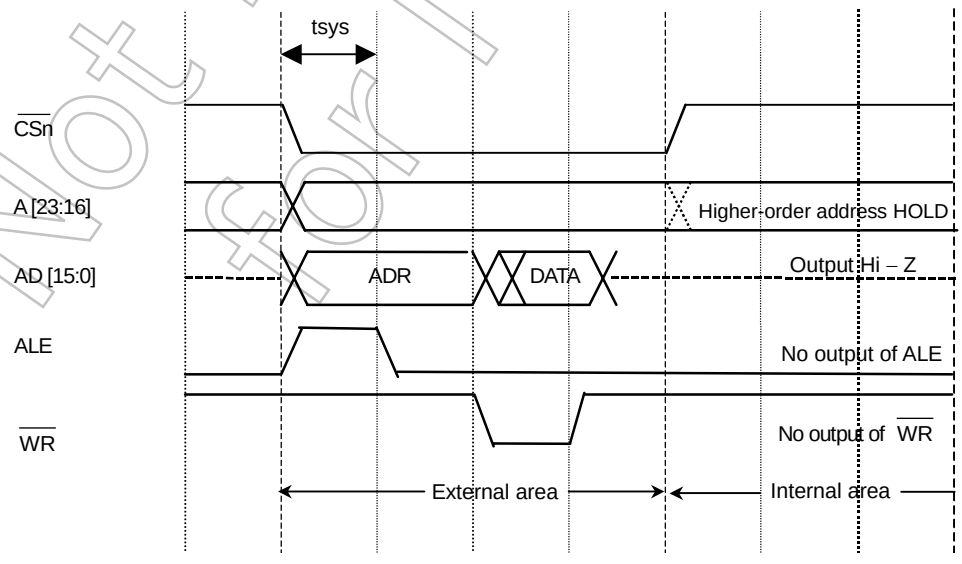


Fig. 8-12 Write Operation Timing Diagram

(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 7 clocks can be automatically inserted.

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

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

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

Fig. 8-13 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the multiplexed bus mode.

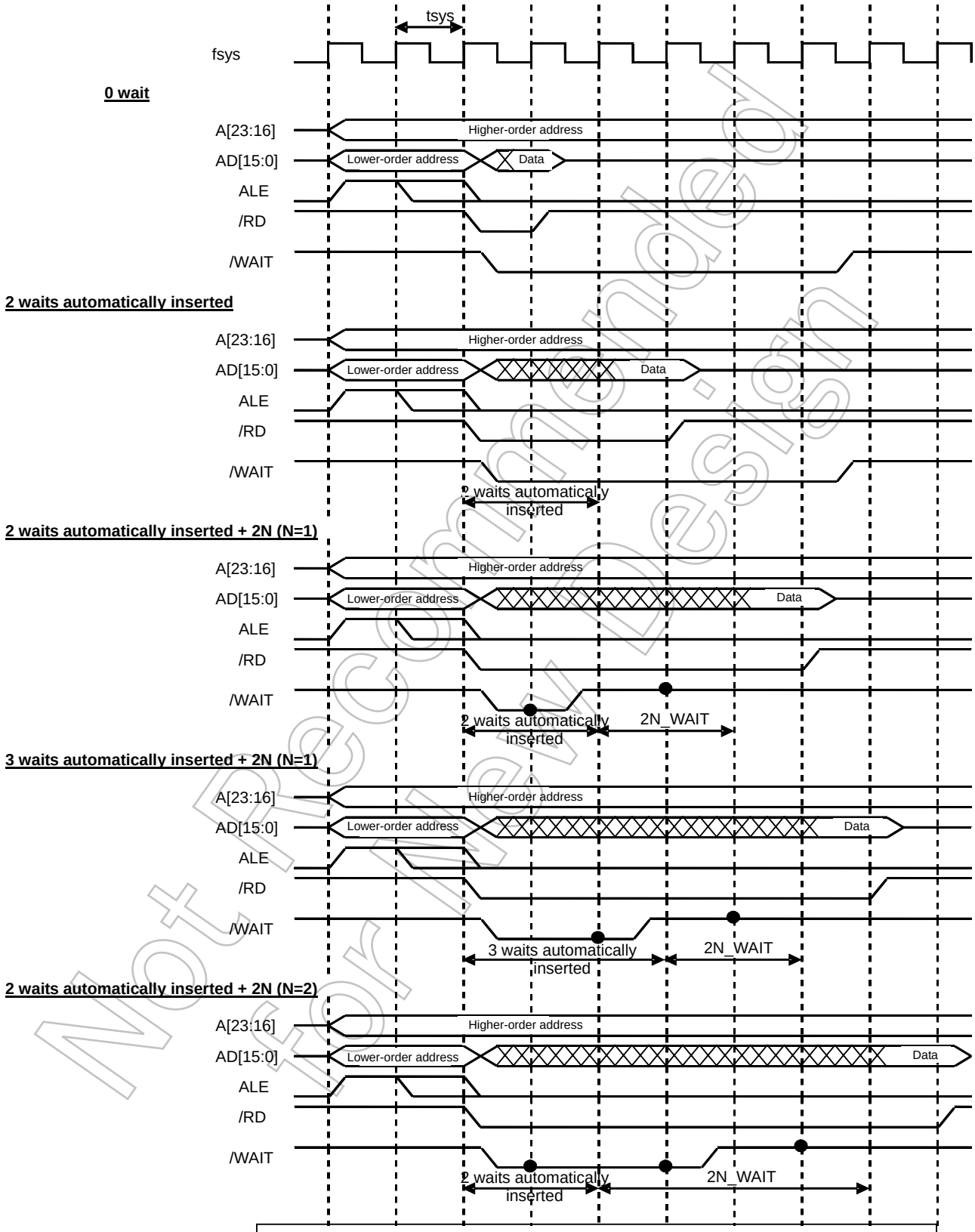
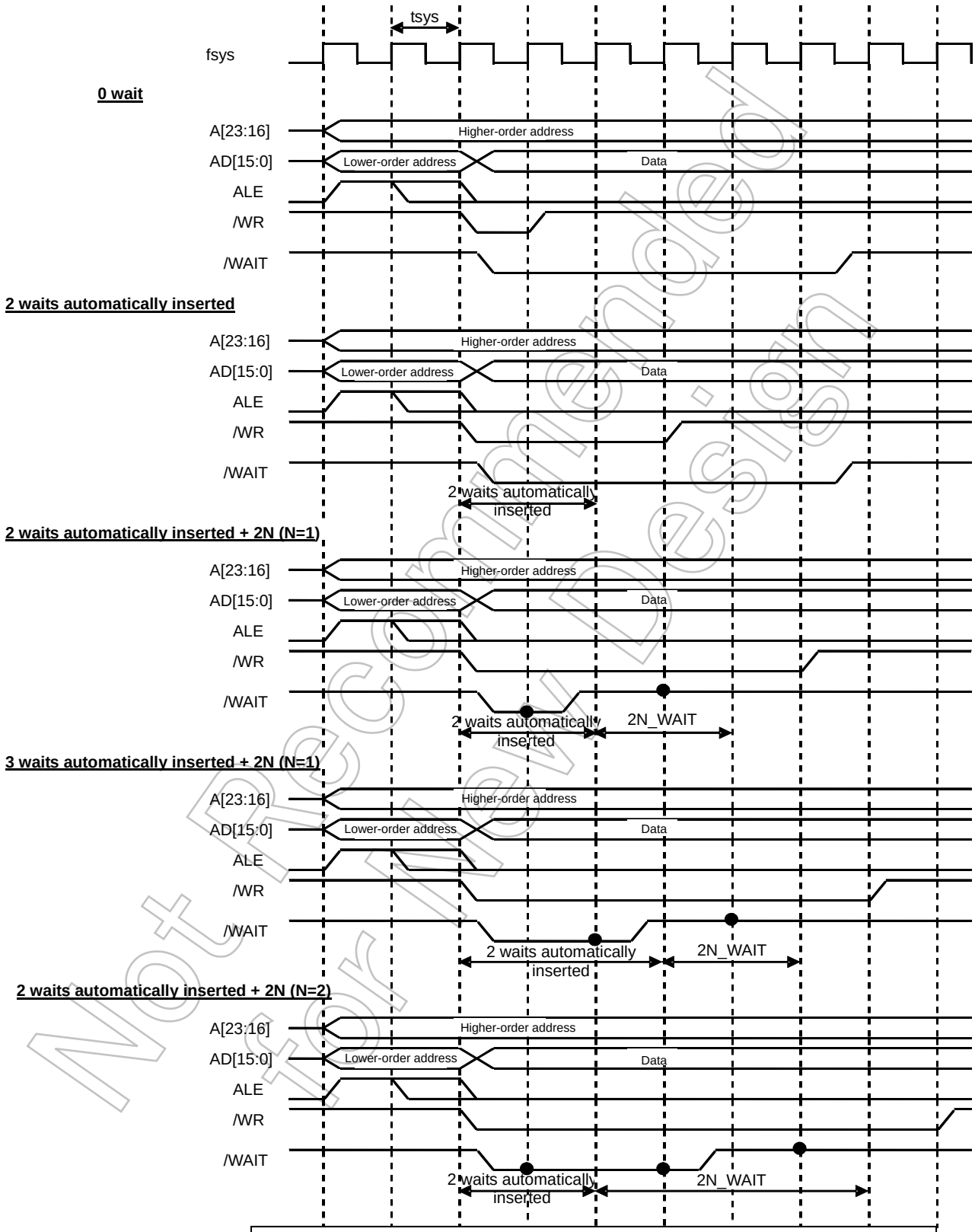


Fig. 8-13 Read Operation Timing Diagram

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

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

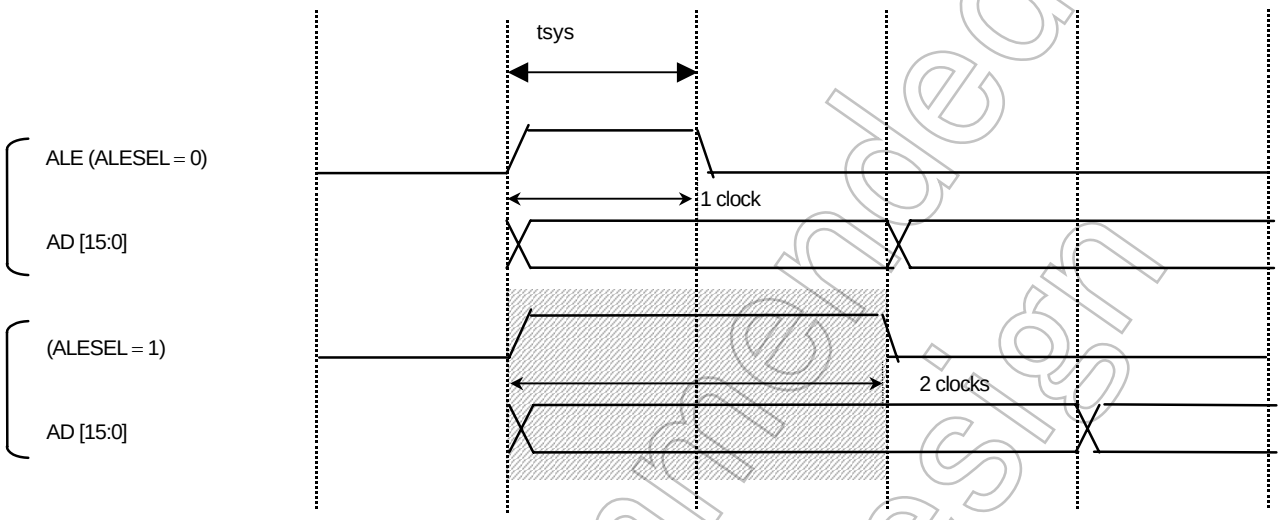


Fig. 8-15 Time That It Takes Before ALE Is Asserted

Fig. 8-16 shows the timing when the ALE is 1 clock or 2 clocks.

When the ALE is 1 clock or 2 clocks

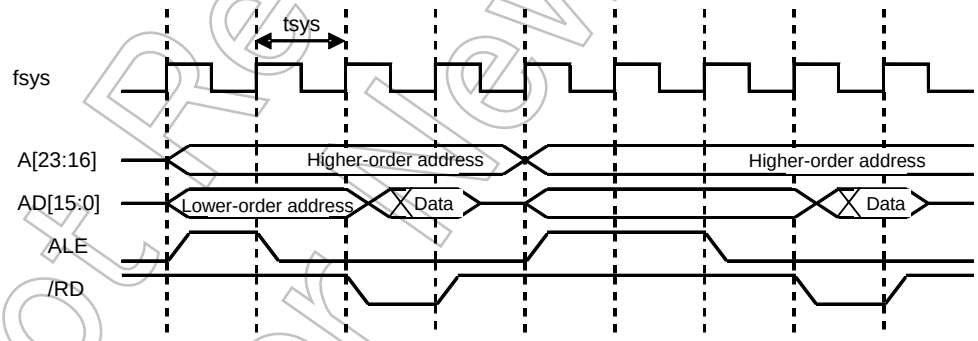


Fig. 8-16 Read Operation Timing Diagram (When the ALE is 1 Clock or 2 Clocks)

(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, one or two system clocks (internal) can be specified for each block. Fig. 8-17 shows the timing of recovery time insertion.

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

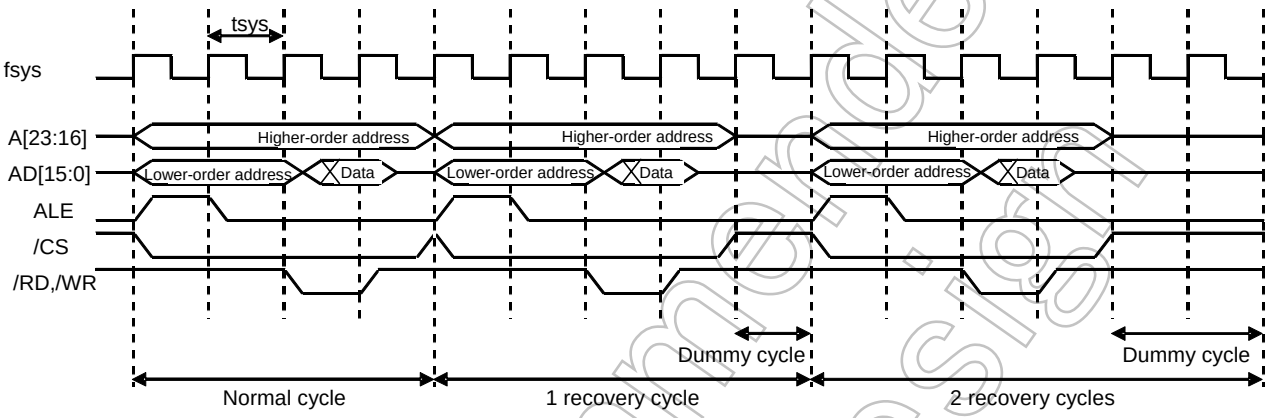


Fig. 8-17 Timing of Recovery Time Insertion

(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)

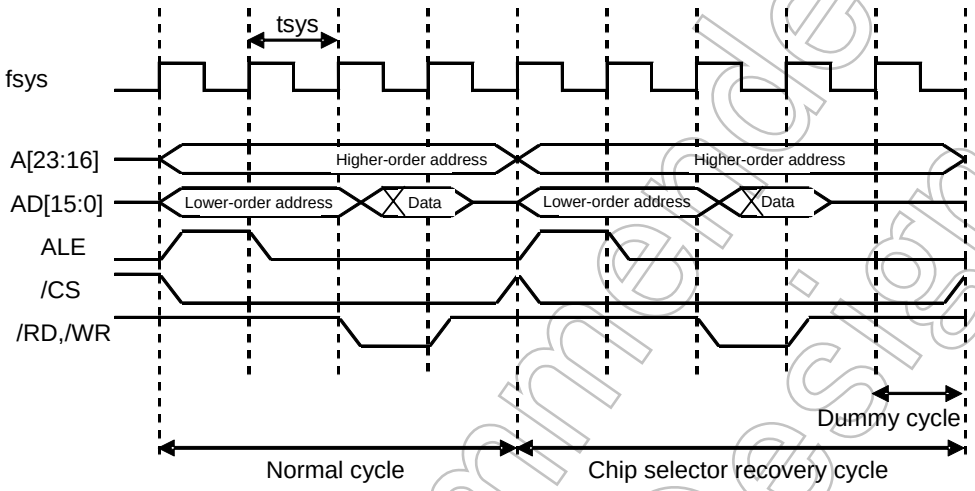


Fig. 8-18 Timing of Recovery Time Insertion

8.5 Bus Arbitration

The TMP19A43 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 TMP19A43 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 TMP19A43 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 TMP19A43 can lock.

(2) Acquisition of bus control authority

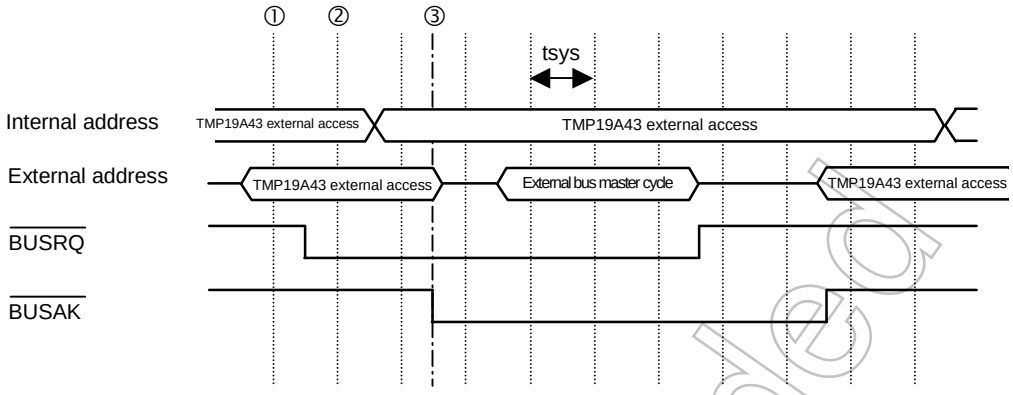
The external bus master requests the TMP19A43 for bus control authority by asserting the $\overline{\text{BUSRQ}}$ signal. The TMP19A43 samples the $\overline{\text{BUSRQ}}$ signal at the break of external bus cycles on the internal buses (G-BUS) and determines whether or not to give the bus control authority to the external bus master. When it gives the bus control authority to the external bus master, it asserts the $\overline{\text{BUSAK}}$ signal. At the same time, it makes address buses, data buses and bus control signals ($\overline{\text{RD}}$ and $\overline{\text{WR}}$) in a state of high impedance. (The internal pull-up is enabled for the R/$\overline{\text{W}}$, $\overline{\text{HWR}}$ and $\overline{\text{CSx}}$.)

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

If access to external areas occurs consecutively on the TMP19A43, 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 TMP19A43 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 TMP19A43 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.

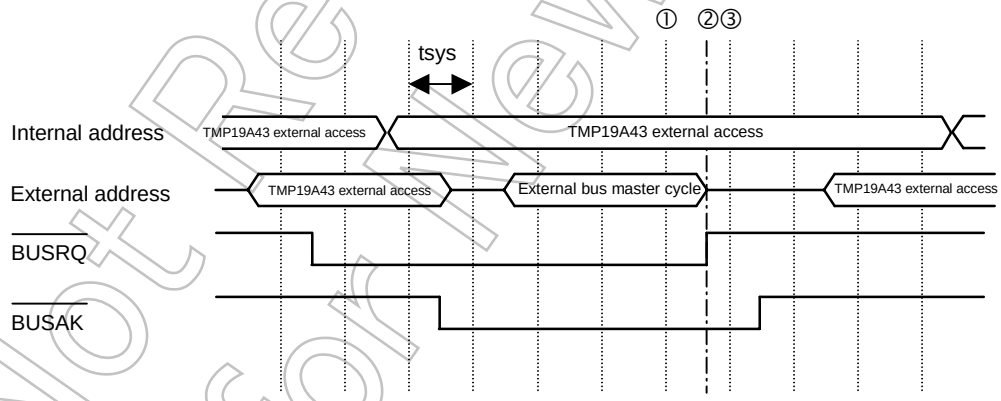
Fig. 8-19 Bus Control Authority Acquisition Timing

(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 deasserts the $\overline{\text{BUSRQ}}$ signal and returns the bus control authority to the TMP19A43.

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 TMP19A43 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 TMP19A43 can be connected to external devices (I/O devices, ROM and SRAM).

4-block address spaces (CS0 through CS3) can be established in the TMP19A43 and three parameters can be specified for each 4-block address and other address spaces: data bus width, the number of waits and the number of dummy cycles.

$\overline{CS0}$ through $\overline{CS3}$ (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 register (P4FC) 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, and BEXCS).

A bus wait request pin ($\overline{WAIT}$ /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 $\overline{CS0}$ through $\overline{CS3}$ 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 and Fig. 9-2 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.

Fig. 9-3 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) Address settings must be made using physical addresses.

Base and mask address setting registers BMA0 (0xFFFF_E400) to BMA3 (0xFFFF_E40C)

BMA0 (0xFFFF_E400)

	7	6	5	4	3	2	1	0
Bit symbol	MA0							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS0 space size setting 0: Address for comparison							
	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							

BMA1 (0xFFFF_E404)

	7	6	5	4	3	2	1	0
Bit symbol	MA1							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS1 space size setting 0: Address for comparison							
	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 TMP19A43 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 (0xFFFF_E408)

	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 (0xFFFF_E40C)

	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 TMP19A43 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)

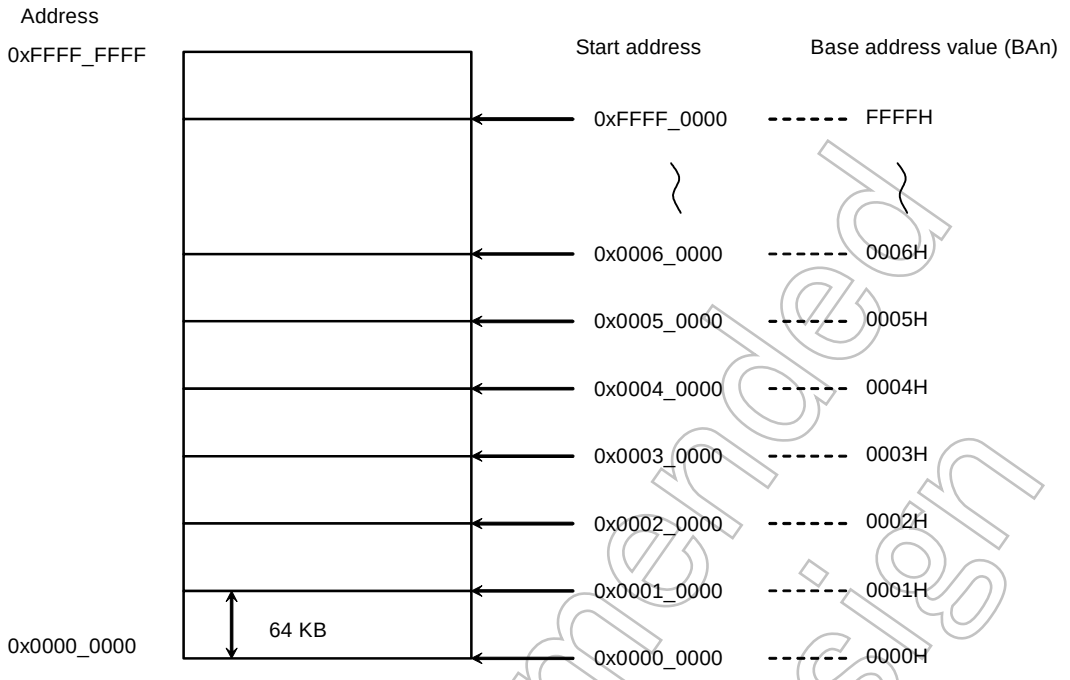
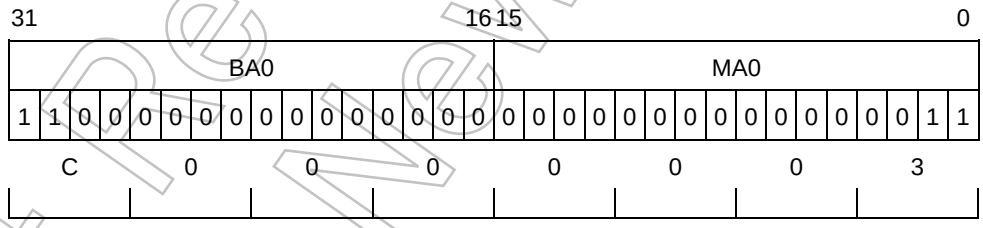


Fig. 9-3 Start and Base Address Register Values

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.

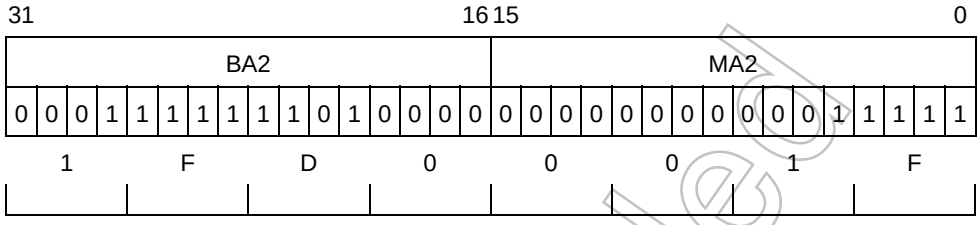


Values to be set in the base and mask address registers (BMA0)

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.



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, CS1 and 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

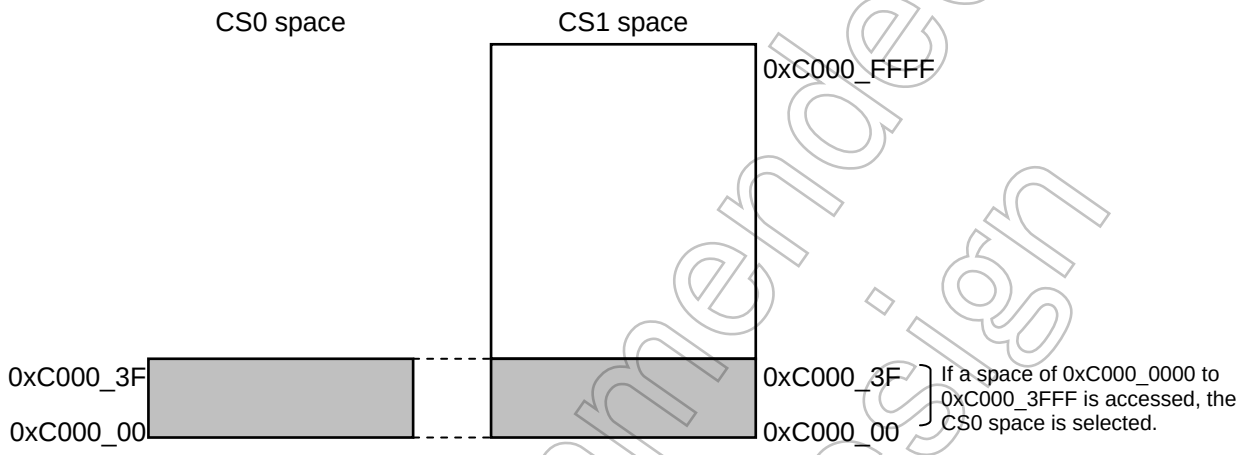


Table 9-1 CS Space and Space Sizes

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

Fig. 9-4 to Fig. 9-6 show the chip selector and wait controller registers. For each address space (spaces CS0 through CS3 and other address spaces), each chip selector and wait controller register (B01CS through B23CS, BEXCS) can be programmed to set master enable or disable, to select data bus width, to specify the number of waits and to insert dummy cycles.

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

B01CS (0xFFFF_E480), B23CS (0xFFFF_E484), BEXCS (0xFFFF_E488)

B01CS (0xFFFF_E480)

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

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

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

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

Fig. 9-4 Chip Selector and Wait Controller Registers

B23CS (0xFFFF_E484)

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

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

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

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

Fig. 9-5 Chip Selector and Wait Controller Registers

BEXCS
Little (0xFFFF_E48C)
Big (0xFFF,F_E48E)

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

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

Fig. 9-6 Chip Selector and Wait Controller Registers

A reset of the TMP19A43 allows the port 4 controller register (P4CR) and the port 4 function register (P4FC) to be cleared to "0," and the CS signal output is disabled. To output the CS signals, set the corresponding bits to "1" at the P4FC and the P4CR in that order.

The CS recovery time can be configured in any other areas than the CS setting areas, but CS signals will not be output.

10. DMA Controller (DMAC)

The TMP19A43 has a built-in 8-channel DMA Controller (DMAC).

10.1 Features

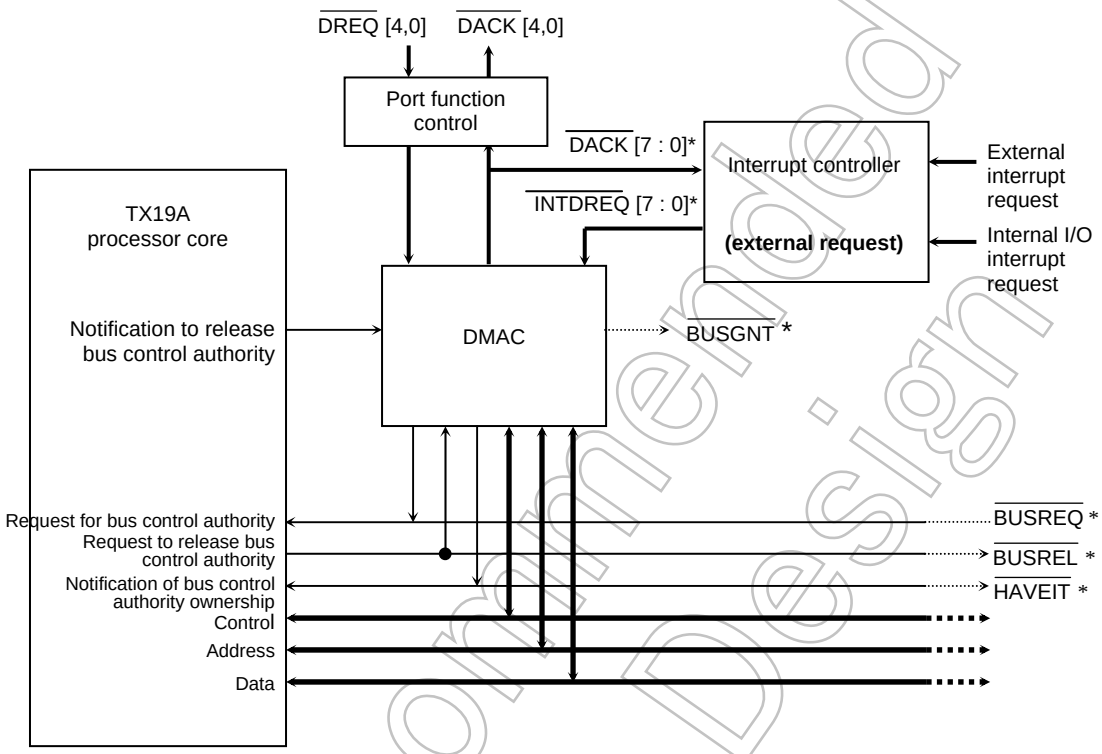
The DMAC of the TMP19A43 has the following features:

(1) DMA with 8 independent channels
(two interrupt factors, 0ch through 3ch: INTDMA0, 4ch through 7ch: INTDMA1)
(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 (CH0, 4): Level mode (memory → memory)
Edge mode (memory → I/O, I/O to memory)
(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 TMP19A43

Fig. 10-1 shows the internal connections with the DMAC in the TMP19A43.



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

Fig. 10-1 DMAC Connections in the TMP19A43

The DMAC has eight DMA channels. Each of these channels handles the data transfer request signal ($\overline{\text{INTDREO}}$n) from the interrupt controller and the acknowledgment signal ($\overline{\text{DACK}}$n) 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 processor core has a snoop function. Using the snoop function, the TX19A processor core opens the core's data bus to the DMAC, thus allowing the DMAC to access the internal ROM and RAM linked to the core. The DMAC is capable of determining whether or not to use this snoop function. For further information on the snoop function, refer to 10.2.3 "Snoop Function."

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

10.2.2 DMAC Internal Blocks

Fig. 10-2 shows the internal blocks of the DMAC.

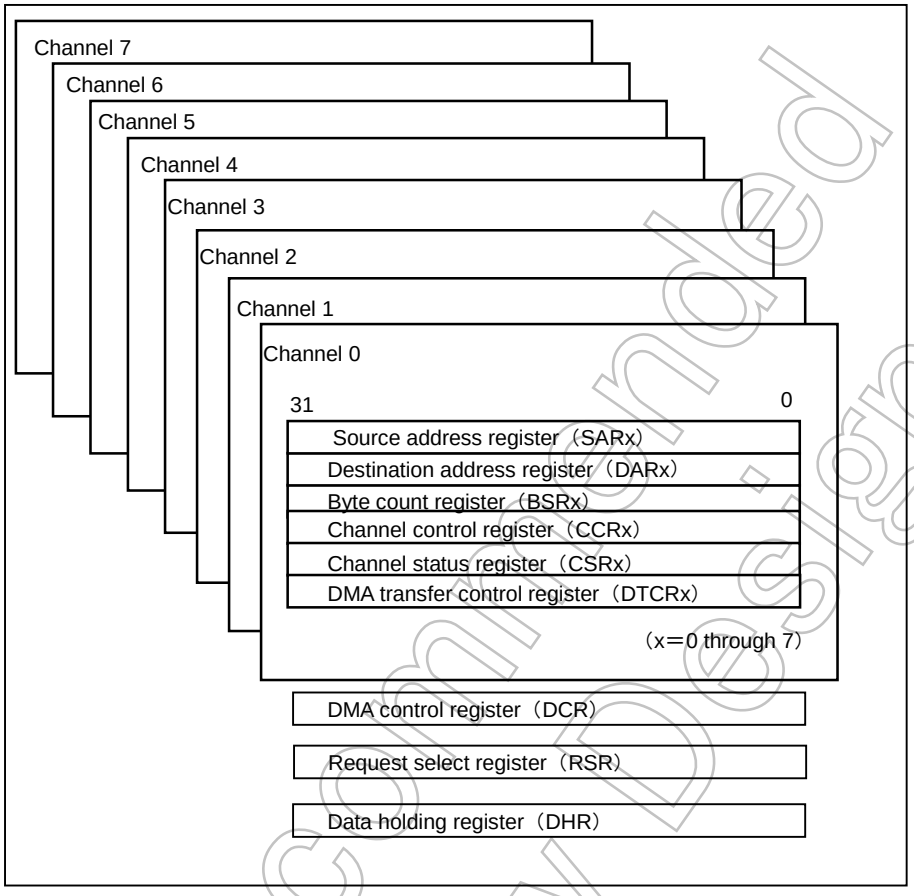


Fig. 10-2 DMAC Internal Blocks

10.2.3 Snoop Function

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

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

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

10.3 Registers

The DMAC has fifty-one 32-bit registers. Table 10.1 shows the register map of the DMAC.

Table 10.1 DMAC Registers

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

Table 10.2 DMAC Registers (continued)

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

10.3.1 DMA Control Register (DCR)

DCR (0xFFFF_E300H)

	7	6	5	4	3	2	1	0
bit Symbol	Rst7	Rst6	Rst5	Rst4	Rst3	Rst2	Rst1	Rst0
Read/Write	W							
After reset	0							
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)

(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)

		7	6	5	4	3	2	1	0
CCRn	bit Symbol	SAC	DIO	DAC		TrSiz		DPS	
(0xFFFF_E200H)	Read/Write	R/W	R/W	R/W		R/W		R/W	
(0xFFFF_E220H)	After reset	0							
(0xFFFF_E240H)	Function	See detailed description.	Always set this bit to “0”.	See detailed description					
(0xFFFF_E260H)		15	14	13	12	11	10	9	8
(0xFFFF_E280H)	bit Symbol		ExR	PosE	Lev	SReq	RelEn	SIO	SAC
(0xFFFF_E2A0H)	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
(0xFFFF_E2C0H)	After reset	0							
(0xFFFF_E2E0H)	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

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 its initial value is "1," always set this bit to "0."
20	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
19	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
18	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
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 processor core is enabled and the DMAC can use the data bus of the TX19A processor core. If the snoop function is not used, the snoop function of the TX19A processor core does not work. 1: Use snoop function (SREQ) 0: Do not use snoop function (GREQ)

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 processor core is specified. This function is valid only if GREQ is generated. If SREQ is generated, the TX19A processor core cannot make a bus control release request and, therefore, this function cannot be used. 1: The bus control release request is acknowledged if the DMAC has control of the bus. If the TX19A processor core issues a bus control release request, the DMAC relinquishes control of the bus to the TX19A processor core during a pause in bus operation. 0: The bus control release request is not acknowledged.
9	SIO	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	-	(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

Fig. 10-5 Channel Control Registers (CCRn) (2 of 3)

Bit	Mnemonic	Field name	Description
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) *Make sure to set the same size as 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) *Make sure to set the same size as the transfer unit (TrSiz)

Fig. 10-6 Channel Control Registers (CCRn) (3 of 3)

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

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

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

10.3.3 Request Select Register (RSR)

RSR (0xFFFF_E304H)

	7	6	5	4	3	2	1	0
bit Symbol				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								

Fig. 10-7

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.10-8 DMA Control Register (RSR)

10.3.4 Channel Status Registers (CSRn)

CSRn
(0xFFFF_E204H)
(0xFFFF_E224H)
(0xFFFF_E244H)
(0xFFFF_E264H)
(0xFFFF_E284H)
(0xFFFF_E2A4H)
(0xFFFF_E2C4H)
(0xFFFF_E2E4H)

	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. 10-9 Channel Status Registers (CSRn)

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. 10-10 Channel Status Registers (CSRn)

10.3.5 Source Address Registers (SARn)

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

Fig. 10-11

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. 10-12 Source Address Register (SARn)

10.3.6 Destination Address Register (DARn)

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

Fig. 10-13

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. 10-14 Destination Address Register (DARn)

10.3.7 Byte Count Registers (BCRn)

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

Fig. 10-15

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. 10-16 Byte Count Register (BCRn)

10.3.8 DMA Transfer Control Register (DTCRn)

		7	6	5	4	3	2	1	0
DTCRn	bit Symbol			DACM			SACM		
(0xFFFF_E218H)	Read/Write			R/W			R/W		
(0xFFFF_E238H)	After reset	0							
(0xFFFF_E258H)	Function			See detailed description.			See detailed description.		
(0xFFFF_E278H)		15	14	13	12	11	10	9	8
(0xFFFF_E298H)	bit Symbol								
(0xFFFF_E2B8H)	Read/Write								
(0xFFFF_E2D8H)	After reset	0							
(0xFFFF_E2F8H)	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								

Fig. 10-17

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. 10-18 DMA Transfer Control Register (DTCRn)

10.3.9 Data Holding Register (DHR)

DHR (0xFFFF_E30CH)

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

Fig. 10-19

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. 10-20 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 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 and between memory and an I/O device. A device from which data is transferred is called a source device and a device to which data is transferred is called a destination device. Both memory and I/O devices can be designated as a source or destination device. The DMAC supports data transfers from memory to I/O devices, from I/O devices to memory, and from memory to memory, but not between I/O devices.

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

An interrupt factor can be attached to a transfer request to be sent to the DMAC. If an interrupt factor is generated, the interrupt controller (INTC) issues a request to the DMAC (the TX19A processor core is not notified of the interrupt request. For details, see description on Interrupts.). The request issued by the INTC is cleared by the $\overline{\text{DACKn}}$ signal. Therefore, a request made to the DMAC is cleared after completion of each data transfer (transfer of the amount of data specified by TrSiz) if a single transfer is designated to select a transfer type (SIO BIT). 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 TMP19A43, a request made by the internal I/O to the DMAC is cleared after completion of each data transfer and the transfer operation is always put in a standby mode for the next transfer request if the number of bytes transferred (value set in the BCRn register) does not become "0." Therefore, the DMA transfer operation continues until the value of the BCRn register becomes "0."

(2) Bus control arbitration (bus arbitration)

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

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

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

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

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

(4) Address mode

For the DMAC of the TMP19A43, 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 $\overline{\text{DMAC}}$ 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. 10-21 shows the state transitions of channel operation.

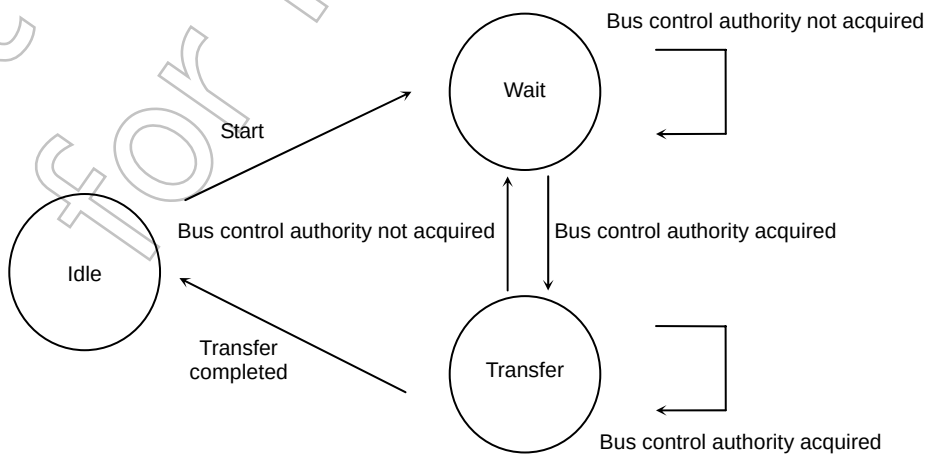


Fig. 10-21 Channel Operation State Transition

(6) Combinations of transfer modes

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

Transfer request	Edge/level	Address mode	Transfer devices
Internal	—	Dual	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

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

- External transfer request

  If the ExR bit of CCRn is "1," setting the Str bit of CCR to "1" allows a channel to go into a standby mode. The INTC or an external device then generates the $\overline{\text{INTDREQn}}$ or $\overline{\text{DREQn}}$ signal for this channel to notify the DMAC of a transfer request, and a transfer request is generated. This transfer request is called an external transfer request. The external transfer request is used for a single and a continuous transfer.

  The TMP19A43 recognizes the transfer request signal by detecting the "L" level of the $\overline{\text{INTDREQn}}$ signal or by detecting the falling edge or "L" level of the $\overline{\text{DREQn}}$ signal.

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

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

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

A transfer request made by the interrupt controller is cleared using the $\overline{\text{DACKn}}$ signal. This $\overline{\text{DACKn}}$ signal is asserted only if a bus cycle for a 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 $\overline{\text{INTDREQn}}$ 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 $\overline{\text{INTDREQn}}$ is not cleared until the number of bytes transferred (value set in the BCRn register) becomes "0."

This $\overline{\text{DACKn}}$ signal is asserted only if a bus cycle for a 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 $\overline{\text{INTDREQn}}$ 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 $\overline{\text{INTDREQn}}$ is not cleared until the number of bytes transferred (value set in the BCRn register) becomes "0."

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

② A transfer request made by an external device

External pins ($\overline{\text{DREQ0}}$ and $\overline{\text{DREQ4}}$) are internally wired to allow them to function as pins of the port F. These pins can be selected by setting the function control register PFFC to an appropriate setting.

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

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

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

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

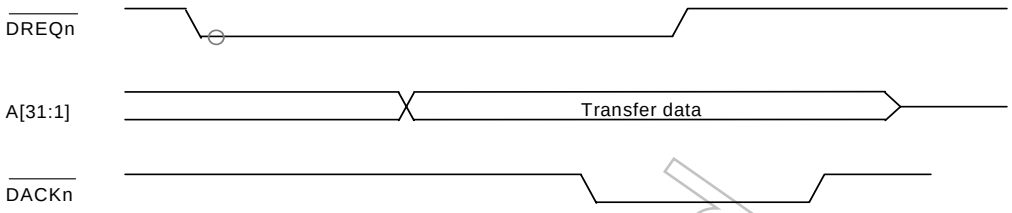


Fig. 10-22 Transfer Request Timing (Level Mode)

– 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.

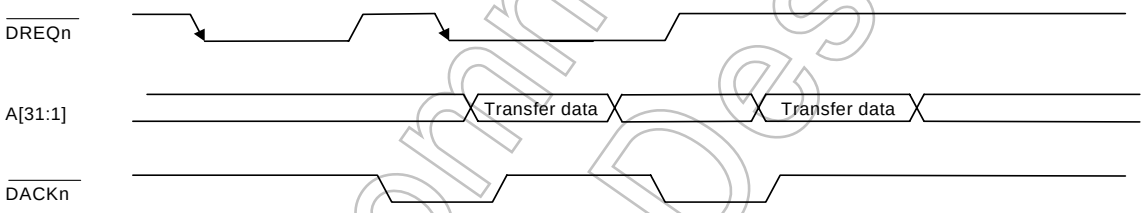


Fig. 10-23 Transfer Request Timing (Edge Mode)

Start factor is DMA interrupt. Instruction of the transmission demand cannot be done by the instruction. There is a possibility that the DMA start factor remains after the DMA forwarding ends last time. In DMA Interrupt, transmission the dummy.

```
CCRx    = Setting;
SARx    = (RAM address)
DARx    = (RAM address)
BCRx    = 0x01;                 } DMA Demand.    Transmission Dummy)
DTCRx   = 0x00;
CCRx    = (Channel start)
```

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 TMP19A43, 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.

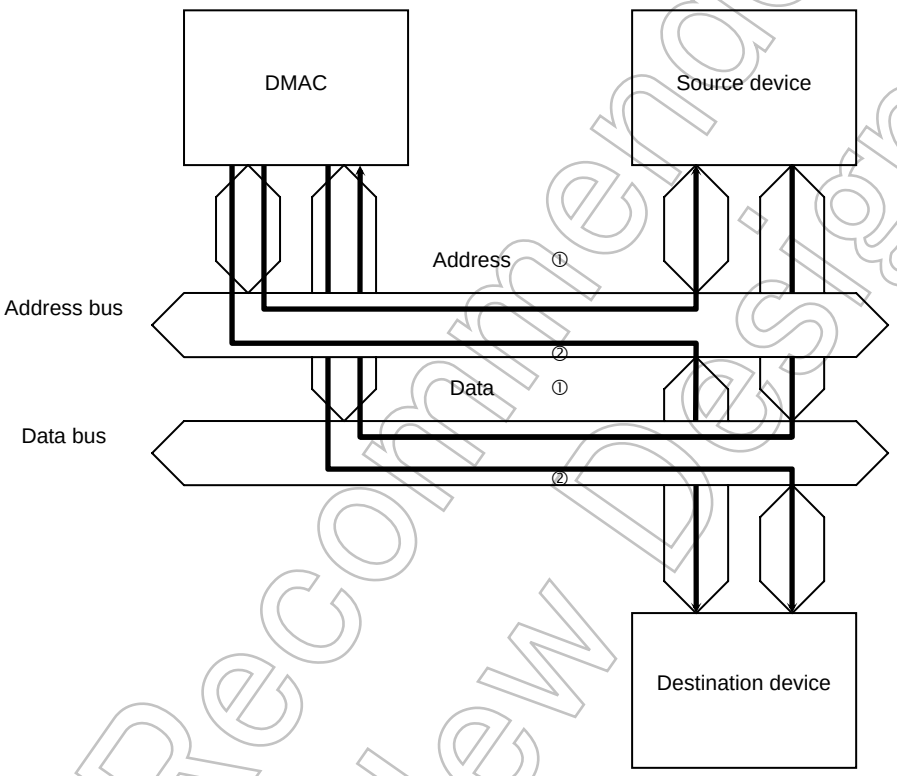


Fig. 10-24 Basic Concept of Data Transfer in the Dual Address Mode

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

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

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

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

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

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

An address change occurs by the amount defined as the unit of data to be transferred. The BCRn value also changes by the same amount. A device port size must not be larger than the unit of data to be transferred. The relationships between units of data to be transferred and device port sizes are summarized in Table 10.1.

Table 10.1 Units of Data to Be Transferred and Device Port Sizes (Dual Address Mode)

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

10.4.3 Channel Operation

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

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

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

Start of channel operation

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

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

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

Completion of channel operation

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

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

Normal completion

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

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

Abnormal completion

Cases of abnormal completion of DMAC operation are as follows:

- Completion due to a configuration error

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

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

– A prohibited combination of a device port size and a unit of data to be transferred was set.
– The Str bit of CCRn was set to "1" when the BCRn value was "0."

- Completion due to a bus error

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

  – A bus error was detected during data transfer.

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

10.4.4 Order of Priority of Channels

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

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

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

Interrupts

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

INTDMA0: 0ch through 3ch INTDMA1: 4ch through 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 processor core to authorize an interrupt.

- Abnormal completion interrupt

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

(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. 10-25 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."

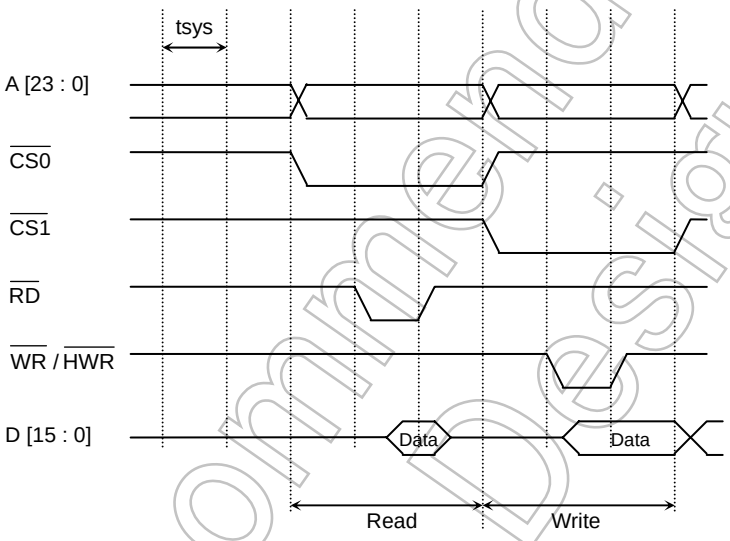


Fig. 10-25 Dual Address Mode (Memory-to-Memory)

- Memory-to-I/O device transfer

  Fig. 10-26 shows an example of the timing with which data is transferred from memory to an I/O device if the unit of data to be transferred is set to 16 bits and if the device port size is set to 8 bits.

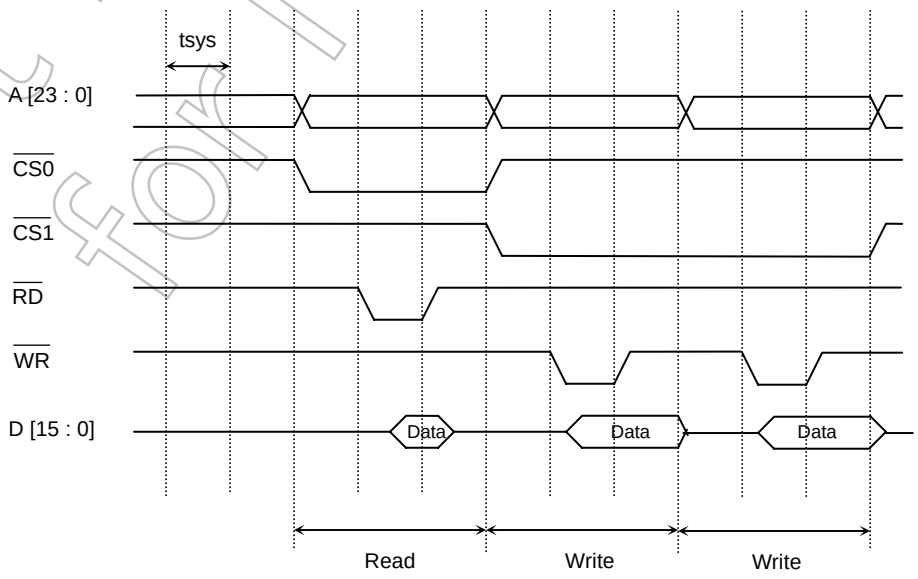


Fig. 10-26 Dual Address Mode (Memory-to-I/O Device)

- I/O device-to-memory transfer

  Fig. 10-27 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.

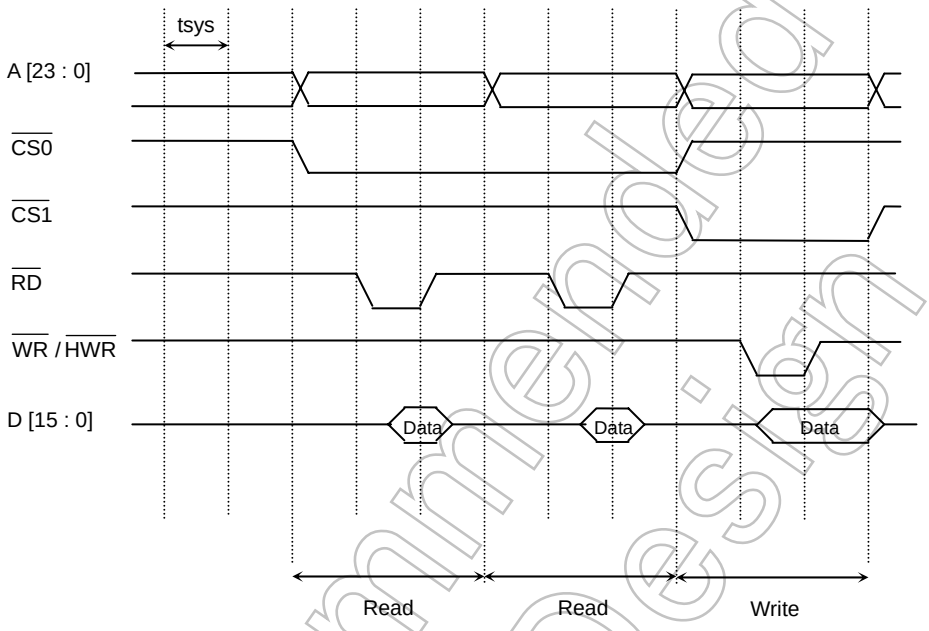


Fig. 10-27 Dual Address Mode (I/O Device-to-Memory)

10.5.2 DREQn-Initiated Transfer Mode

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

  Fig. 10-28 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

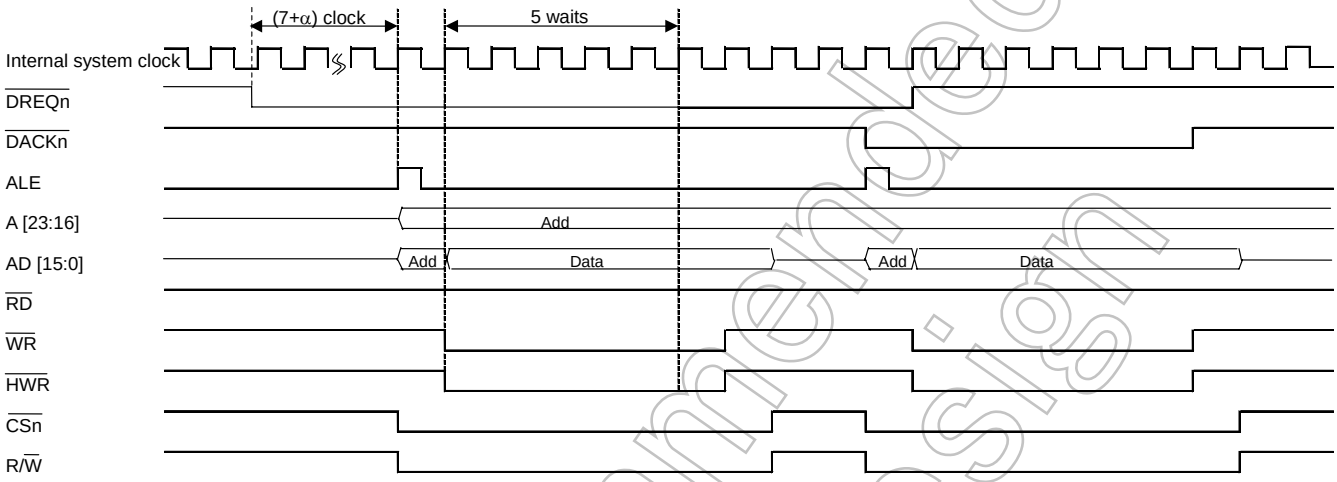


Fig. 10-28 Level Mode (from Internal RAM to External Memory)

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

  Fig. 10-29 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bit width) to internal RAM.

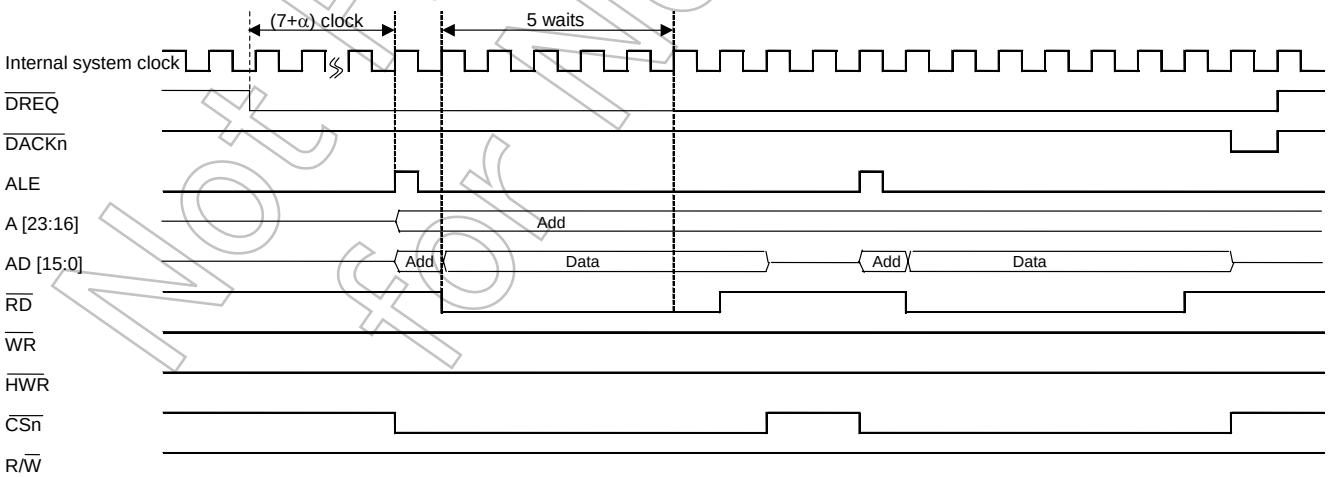


Fig. 10-29 Level Mode (from External Memory to Internal RAM)

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

  Fig. 10-30 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

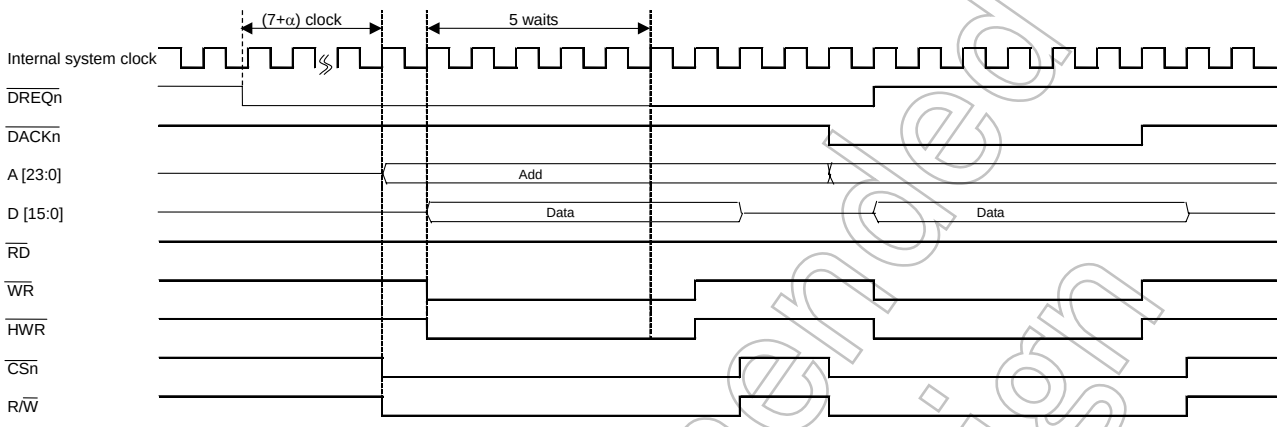


Fig. 10-30 Level Mode (Internal RAM to External Memory)

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

  Fig. 10-31 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bid width) to internal RAM.

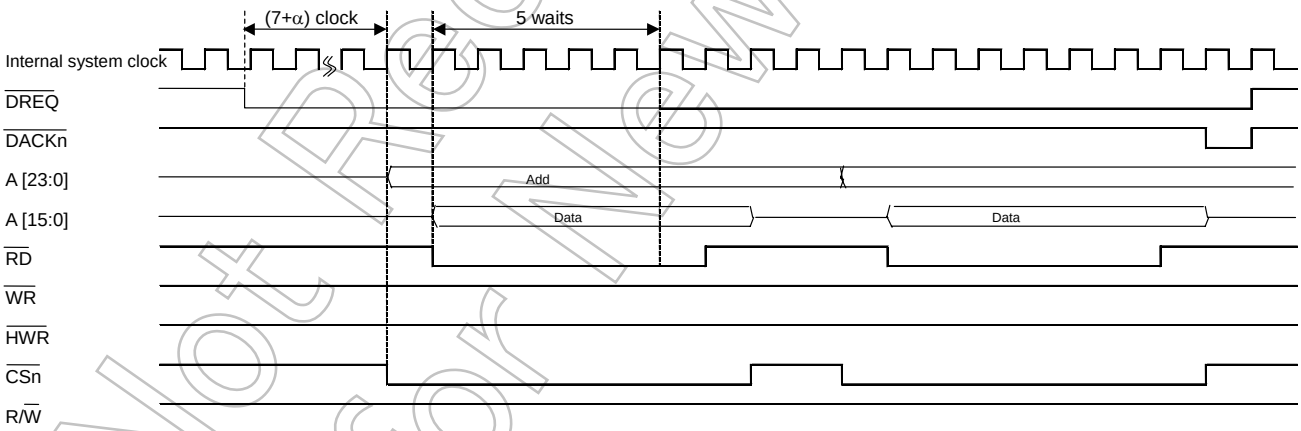


Fig. 10-31 Level Mode (from External Memory to Internal RAM)

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

  Fig. 10-32 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

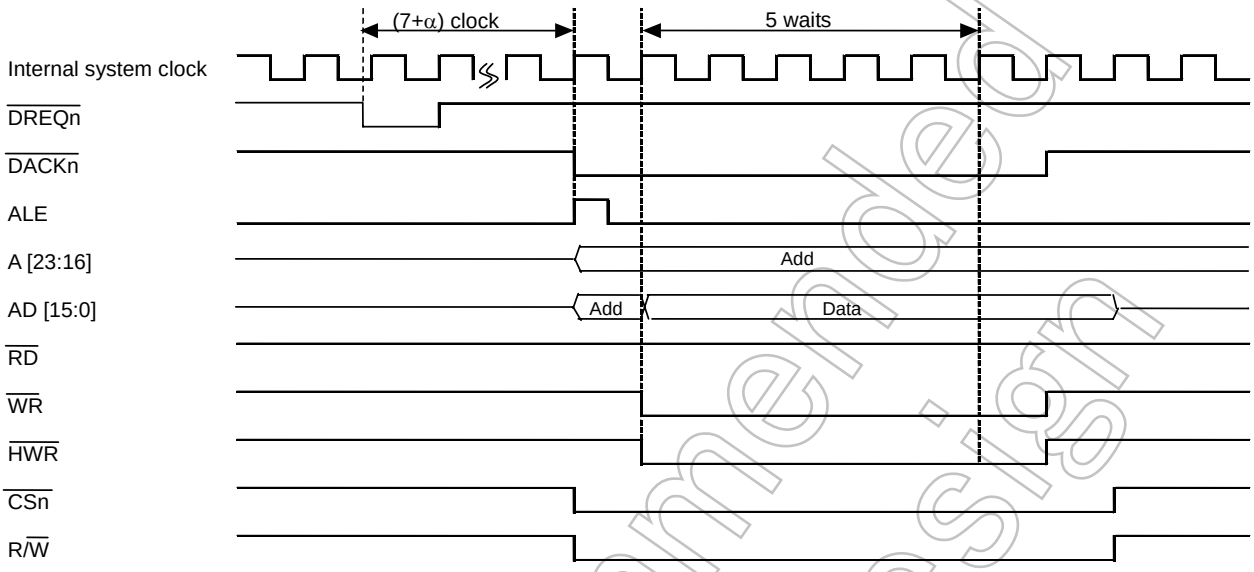


Fig. 10-32 Edge Mode (from Internal RAM to External Memory)

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

  Fig. 10-33 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

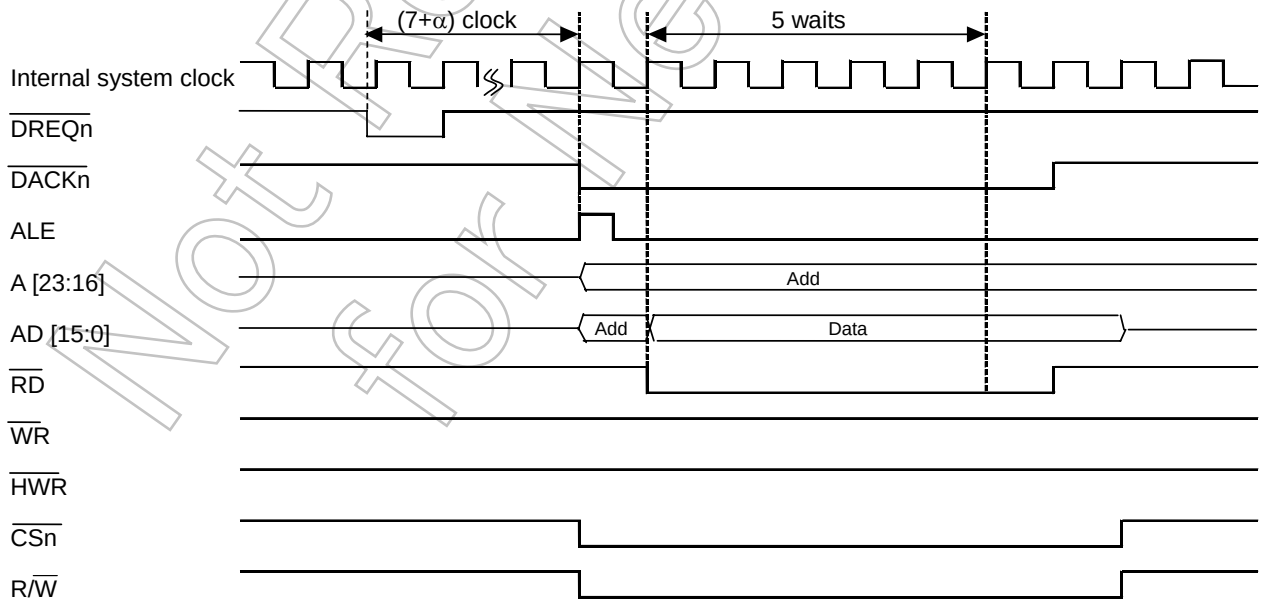


Fig. 10-33 Edge Mode (from External Memory to Internal RAM)

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

  Fig. 10-34 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

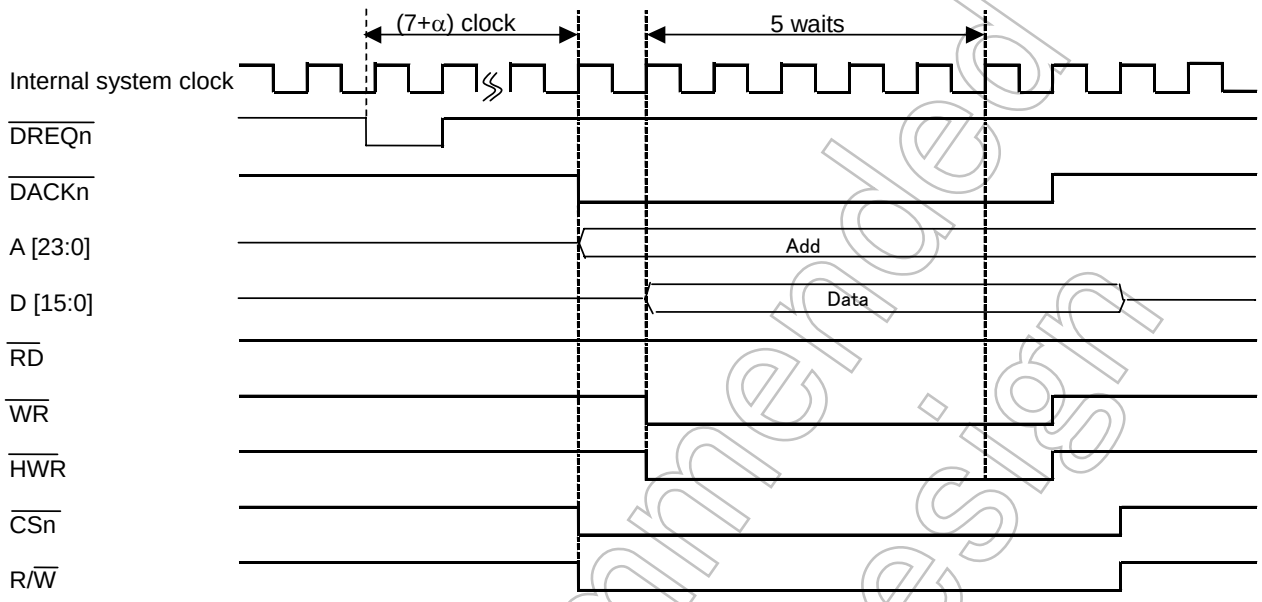


Fig. 10-34 Edge Mode (from Internal RAM to External Memory)

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

  Fig. 10-35 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

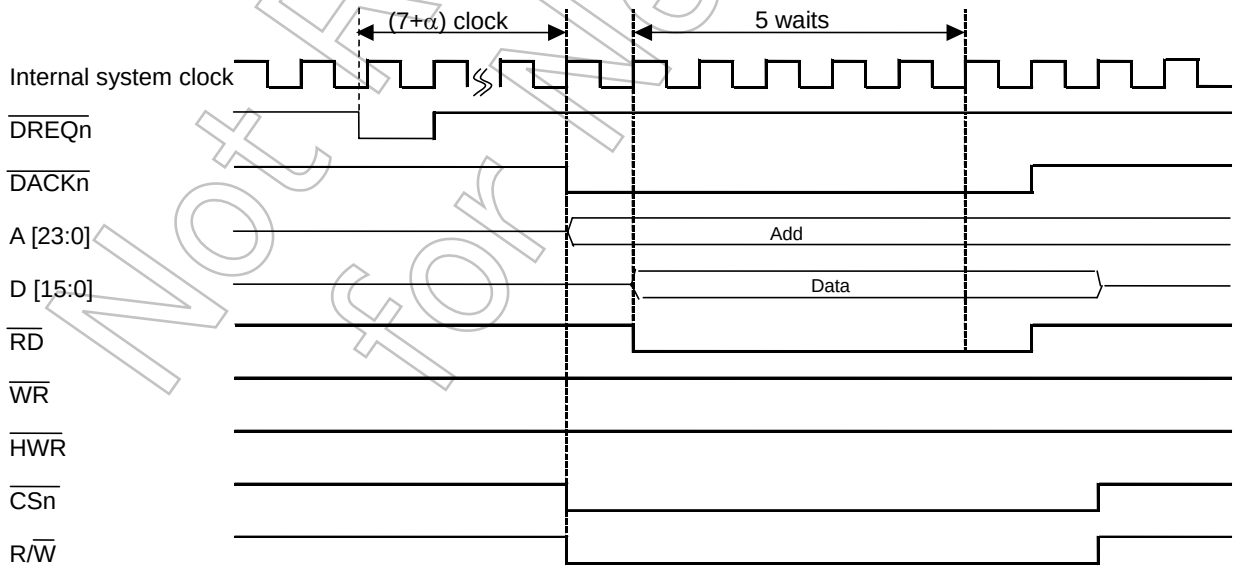


Fig. 10-35 Edge Mode (from External Memory to Internal RAM)

10.6 Case of Data Transfer

The settings described below relate to a case in which serial data received (SCnBUF) is transferred to the internal RAM by DMA transfer.

DMA (ch.0) is used to transfer data. The DMA0 is activated by a receive interrupt generated by SIO1.

<DMA setting>

- Channel used: 0
- Source address: SC1BUF
- Destination: (Physical address) 0xFFFF_9800
- Number of bytes transferred: 256 bytes

<Serial channel setting>

- Data length 8 bits: UART
- Serial channel: ch 1
- Transfer rate: 9600 bps

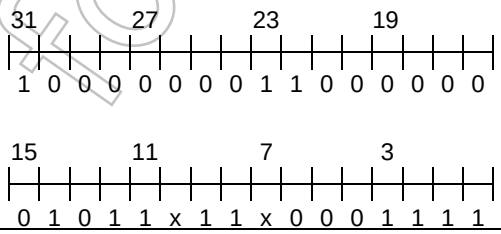
<SIO ch.1 setting>

```
IMC5LL      ←   x111, x100      /* assigned to DMC0 activation factor * /
INTCLR      ←   0x050           /* IVR [8:0], INTRX1 interrupt factor * /
SC1MOD0     ←   0x29            /* UART mode, 8-bit length, baud rate generator * /
SC1CR       ←   0x00
BR1CR       ←   0x1F            /* @fc = 40 MHz */
```

<DMA0 setting>

```
DCR         ←   0x8000_0000     /* DMA reset * /
IMCFHL      ←   x000, x000      /* disable interrupt * /
INTCLR      ←   0x0F8           /* IVR [8:0] value * /
IMCFHL      ←   x000, x100      /* level = 4 (any given value) */
DTCR0       ←   0x0000_0000     /* DACM = 000 * /
                                /* SACM = 000 * /
SAR0        ←   0xFFFF_F208     /* physical address of SC1BUF */
DAR0        ←   0xFFFF_9800     /* physical address of destination to which data is transferred */
BCR0        ←   0x0000_00FF     /* 256 (number of bytes transferred) /
CCR0        ←   0x80C0_5B0F     /* DMA ch.0 setting */
```

(Contents)



Start factor is DMA interrupt. Instruction of the transmission demand cannot be done by the instruction. There is a possibility that the DMA start factor remains after the DMA forwarding ends last time. In DMA Interrupt, transmission the dummy.

11. 16-bit Timer/Event Counters (TMRBs)

Each of the sixteen channels (TMRB0 through TMRF) 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)
- Two-phase pulse input counter mode (quad/nomal-speed, TMRB2, TMRB3, TMRB6 and TMRB7 only)
- 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 (one of which is double-buffered), two 16-bit capture registers, two comparators, a capture input control, a timer flip-flop and its associated control circuit. Timer operation modes and the timer flip-flop are controlled by a 13-byte register.

Each channel (TMRB0 through TMRBF) functions independently and while the channels operate in the same way, there are differences in their specifications as shown in Table 11-1 and the two-phase pulse count function. Therefore, the operational descriptions here are for TMRB0 only and for the two-phase pulse count function (TMRB2, TMRB3, TMRB6 and TMRB7) only.

Table 11-1 Differences in the Specifications of TMRB Modules

Specification \ Channel		TMRB0	TMRB1	TMRB2	TMRB3
External pins	External clock/ capture trigger input pins	TB0IN0 (shared with P20) TB0IN1 (shared with P21)	TB1IN0 (shared with P22) TB1IN1 (shared with P23)	TB2IN0 (shared with PA6) TB2IN1 (shared with PA7)	TB3IN0 (shared with PB0) TB3IN1 (shared with PB1)
	Timer flip-flop output pin	TB0OUT (shared with P54)	TB1OUT (shared with P55)	TB2OUT (shared with P56)	TB3OUT (shared with P57)
Internal signals	Timer for capture triggers	—	TB0OUT	TB0OUT	TB0OUT
Register names (addresses)	Timer RUN register	TB0RUN (0xFFFF_F140)	TB1RUN (0xFFFF_F150)	TB2RUN (0xFFFF_F160)	TB3RUN (0xFFFF_F170)
	Timer control register	TB0CR (0xFFFF_F141)	TB1CR (0xFFFF_F151)	TB2CR (0xFFFF_F161)	TB3CR (0xFFFF_F171)
	Timer mode register	TB0MOD (0xFFFF_F142)	TB1MOD (0xFFFF_F152)	TB2MOD (0xFFFF_F162)	TB3MOD (0xFFFF_F172)
	Timer flip-flop control register	TB0FFCR (0xFFFF_F143)	TB1FFCR (0xFFFF_F153)	TB2FFCR (0xFFFF_F163)	TB3FFCR (0xFFFF_F173)
	Timer status register	TB0ST (0xFFFF_F144)	TB1ST (0xFFFF_F154)	TB2ST (0xFFFF_F164)	TB3ST (0xFFFF_F174)
	Timer UC preset register	TB0UCL TB0UCH	TB1UCL TB1UCH	TB2UCL TB2UCH	TB3UCL TB3UCH
	Timer register	TB0RG0L (0xFFFF_F148) TB0RG0H (0xFFFF_F149) TB0RG1L (0xFFFF_F14A) TB0RG1H (0xFFFF_F14B)	TB1RG0L (0xFFFF_F158) TB1RG0H (0xFFFF_F159) TB1RG1L (0xFFFF_F15A) TB1RG1H (0xFFFF_F15B)	TB2RG0L (0xFFFF_F168) TB2RG0H (0xFFFF_F169) TB2RG1L (0xFFFF_F16A) TB2RG1H (0xFFFF_F16B)	TB3RG0L (0xFFFF_F178) TB3RG0H (0xFFFF_F179) TB3RG1L (0xFFFF_F17A) TB3RG1H (0xFFFF_F17B)
	Capture register	TB0CP0L (0xFFFF_F14C) TB0CP0H (0xFFFF_F14D) TB0CP1L (0xFFFF_F14E) TB0CP1H (0xFFFF_F14F)	TB1CP0L (0xFFFF_F15C) TB1CP0H (0xFFFF_F15D) TB1CP1L (0xFFFF_F15E) TB1CP1H (0xFFFF_F15F)	TB2CP0L (0xFFFF_F16C) TB2CP0H (0xFFFF_F16D) TB2CP1L (0xFFFF_F16E) TB2CP1H (0xFFFF_F16F)	TB3CP0L (0xFFFF_F17C) TB3CP0H (0xFFFF_F17D) TB3CP1L (0xFFFF_F17E) TB3CP1H (0xFFFF_F17F)

Specification \ Channel		TMRB4	TMRB5	TMRB6	TMRB7
External pins	External clock/ capture trigger input pins	TB4IN0 (shared with P24) TB4IN1 (shared with P25)	TB5IN0 (shared with P26) TB5IN1 (shared with P27)	TB6IN0 (shared with PA0) TB6IN1 (shared with PA1)	TB7IN0 (shared with PA2) TB7IN1 (shared with PA3)
	Timer flip-flop output pin	TB4OUT (shared with P66)	TB5OUT (shared with P67)	TB6OUT (shared with P90)	TB7OUT (shared with P91)
Internal signals	Timer for capture triggers	TB0OUT	TB0OUT	TB0OUT	TB0OUT
Register names (addresses)	Timer RUN register	TB4RUN (0xFFFF_F180)	TB5RUN (0xFFFF_F190)	TB6RUN (0xFFFF_F1A0)	TB7RUN (0xFFFF_F1B0)
	Timer control register	TB4CR (0xFFFF_F181)	TB5CR (0xFFFF_F191)	TB6CR (0xFFFF_F1A1)	TB7CR (0xFFFF_F1B1)
	Timer mode register	TB4MOD (0xFFFF_F182)	TB5MOD (0xFFFF_F192)	TB6MOD (0xFFFF_F1A2)	TB7MOD (0xFFFF_F1B2)
	Timer flip-flop control register	TB4FFCR (0xFFFF_F183)	TB5FFCR (0xFFFF_F193)	TB6FFCR (0xFFFF_F1A3)	TB7FFCR (0xFFFF_F1B3)
	Timer status register	TB4ST (0xFFFF_F184)	TB5ST (0xFFFF_F194)	TB6ST (0xFFFF_F1A4)	TB7ST (0xFFFF_F1B4)
	Timer UC preset register	TB4UCL TB4UCH	TB5UCL TB5UCH	TB6UCL TB6UCH	TB7UCL TB7UCH
	Timer register	TB4RG0L (0xFFFF_F188) TB4RG0H (0xFFFF_F189) TB4RG1L (0xFFFF_F18A) TB4RG1H (0xFFFF_F18B)	TB5RG0L (0xFFFF_F198) TB5RG0H (0xFFFF_F199) TB5RG1L (0xFFFF_F19A) TB5RG1H (0xFFFF_F19B)	TB6RG0L (0xFFFF_F1A8) TB6RG0H (0xFFFF_F1A9) TB6RG1L (0xFFFF_F1AA) TB6RG1H (0xFFFF_F1AB)	TB7RG0L (0xFFFF_F1B8) TB7RG0H (0xFFFF_F1B9) TB7RG1L (0xFFFF_F1BA) TB7RG1H (0xFFFF_F1BB)
	Capture register	TB4CP0L (0xFFFF_F18C) TB4CP0H (0xFFFF_F18D) TB4CP1L (0xFFFF_F18E) TB4CP1H (0xFFFF_F18F)	TB5CP0L (0xFFFF_F19C) TB5CP0H (0xFFFF_F19D) TB5CP1L (0xFFFF_F19E) TB5CP1H (0xFFFF_F19F)	TB6CP0L (0xFFFF_F1AC) TB6CP0H (0xFFFF_F1AD) TB6CP1L (0xFFFF_F1AE) TB6CP1H (0xFFFF_F1AF)	TB7CP0L (0xFFFF_F1BC) TB7CP0H (0xFFFF_F1BD) TB7CP1L (0xFFFF_F1BE) TB7CP1H (0xFFFF_F1BF)

Specification \ Channel		TMRB8	TMRB9	TMRBA	TMRBB
External pins	External clock/ capture trigger input pins	TB8IN0 (shared with PA4) TB8IN1 (shared with PA5x)	–	–	–
	Timer flip-flop output pin	TB8OUT (shared with P92)	TB9OUT (shared with P96)	TBAOUT (shared with P97)	TBBOUT (shared with PD3)
Internal signals	Timer for capture triggers	—	TB8OUT	TB8OUT	TB8OUT
Register names (addresses)	Timer RUN register	TB8RUN (0xFFFF_F1C0)	TB9RUN (0xFFFF_F1D0)	TBARUN (0xFFFF_F1E0)	TBBRUN (0xFFFF_F1F0)
	Timer control register	TB8CR (0xFFFF_F1C1)	TB9CR (0xFFFF_F1D1)	TBACR (0xFFFF_F1E1)	TBBCR (0xFFFF_F1F1)
	Timer mode register	TB8MOD (0xFFFF_F1C2)	TB9MOD (0xFFFF_F1D2)	TBAMOD (0xFFFF_F1E2)	TBBMOD (0xFFFF_F1F2)
	Timer flip-flop control register	TB8FFCR (0xFFFF_F1C3)	TB9FFCR (0xFFFF_F1D3)	TBAFFCR (0xFFFF_F1E3)	TBBFFCR (0xFFFF_F1F3)
	Timer status register	TB8ST (0xFFFF_F1C4)	TB9ST (0xFFFF_F1D4)	TBAST (0xFFFF_F1E4)	TBBST (0xFFFF_F1F4)
	Timer UC preset register	TB8UCL TB8UCH	TB9UCL TB9UCH	TBAUCL TBAUCH	TBBUCL TBBUCH
	Timer register	TB8RG0L (0xFFFF_F1C8) TB8RG0H (0xFFFF_F1C9) TB8RG1L (0xFFFF_F1CA) TB8RG1H (0xFFFF_F1CB)	TB9RG0L (0xFFFF_F1D8) TB9RG0H (0xFFFF_F1D9) TB9RG1L (0xFFFF_F1DA) TB9RG1H (0xFFFF_F1DB)	TBARG0L (0xFFFF_F1E8) TBARG0H (0xFFFF_F1E9) TBARG1L (0xFFFF_F1EA) TBARG1H (0xFFFF_F1EB)	TBBRG0L (0xFFFF_F1F8) TBBRG0H (0xFFFF_F1F9) TBBRG1L (0xFFFF_F1FA) TBBRG1H (0xFFFF_F1FB)
	Capture register	TB8CP0L (0xFFFF_F1CC) TB8CP0H (0xFFFF_F1CD) TB8CP1L (0xFFFF_F1CE) TB8CP1H (0xFFFF_F1CF)	TB9CP0L (0xFFFF_F1DC) TB9CP0H (0xFFFF_F1DD) TB9CP1L (0xFFFF_F1DE) TB9CP1H (0xFFFF_F1DF)	TBACP0L (0xFFFF_F1EC) TBACP0H (0xFFFF_F1ED) TBACP1L (0xFFFF_F1EE) TBACP1H (0xFFFF_F1EF)	TBBCP0L (0xFFFF_F1FC) TBBCP0H (0xFFFF_F1FD) TBBCP1L (0xFFFF_F1FE) TBBCP1H (0xFFFF_F1FF)

Specification \ Channel		TMRBC	TMRBD	TMRBE	TMRBF
External pins	External clock/ capture trigger input pins	–	–	–	–
	Timer flip-flop output pin	TBCOUT (shared with PD4)	TBDOUT (shared with PD5)	TBEOUT (shared with P32)	TBFOUT (shared with P47)
Internal signals	Timer for capture triggers	TB8OUT	TB8OUT	TB8OUT	TB8OUT
Register names (addresses)	Timer RUN register	TBCRUN (0xFFFF_F200)	TBDRUN (0xFFFF_F210)	TBERUN (0xFFFF_F220)	TBFRUN (0xFFFF_F230)
	Timer control register	TBCCR (0xFFFF_F201)	TBDCR (0xFFFF_F211)	TBECR (0xFFFF_F221)	TBFCR (0xFFFF_F231)
	Timer mode register	TBCMOD (0xFFFF_F202)	TBDMOD (0xFFFF_F212)	TBEMOD (0xFFFF_F222)	TBFMOD (0xFFFF_F232)
	Timer flip-flop control register	TBCFFCR (0xFFFF_F203)	TBDFFCR (0xFFFF_F213)	TBEFFCR (0xFFFF_F223)	TBFFFCR (0xFFFF_F233)
	Timer status register	TBCST (0xFFFF_F204)	TBDST (0xFFFF_F214)	TBEST (0xFFFF_F224)	TBFST (0xFFFF_F234)
	Timer UC preset register	TBCUCL TBCUCH	TBDUCL TBDUCH	TBEUCL TBEUCH	TBFUCL TBFUCH
	Timer register	TBCRG0L (0xFFFF_F208) TBCRG0H (0xFFFF_F209) TBCRG1L (0xFFFF_F20A) TBCRG1H (0xFFFF_F20B)	TBDRG0L (0xFFFF_F218) TBDRG0H (0xFFFF_F219) TBDRG1L (0xFFFF_F21A) TBDRG1H (0xFFFF_F21B)	TBERG0L (0xFFFF_F228) TBERG0H (0xFFFF_F229) TBERG1L (0xFFFF_F22A) TBERG1H (0xFFFF_F22B)	TBFRG0L (0xFFFF_F238) TBFRG0H (0xFFFF_F239) TBFRG1L (0xFFFF_F23A) TBFRG1H (0xFFFF_F23B)
	Capture register	TBCCP0L (0xFFFF_F20C) TBCCP0H (0xFFFF_F20D) TBCCP1L (0xFFFF_F20E) TBCCP1H (0xFFFF_F20F)	TBDCP0L (0xFFFF_F21C) TBDCP0H (0xFFFF_F21D) TBDCP1L (0xFFFF_F21E) TBDCP1H (0xFFFF_F21F)	TBECP0L (0xFFFF_F22C) TBECP0H (0xFFFF_F22D) TBECP1L (0xFFFF_F22E) TBECP1H (0xFFFF_F22F)	TBFCP0L (0xFFFF_F23C) TBFCP0H (0xFFFF_F23D) TBFCP1L (0xFFFF_F23E) TBFCP1H (0xFFFF_F23F)

11.1 Block Diagram of Each Channel

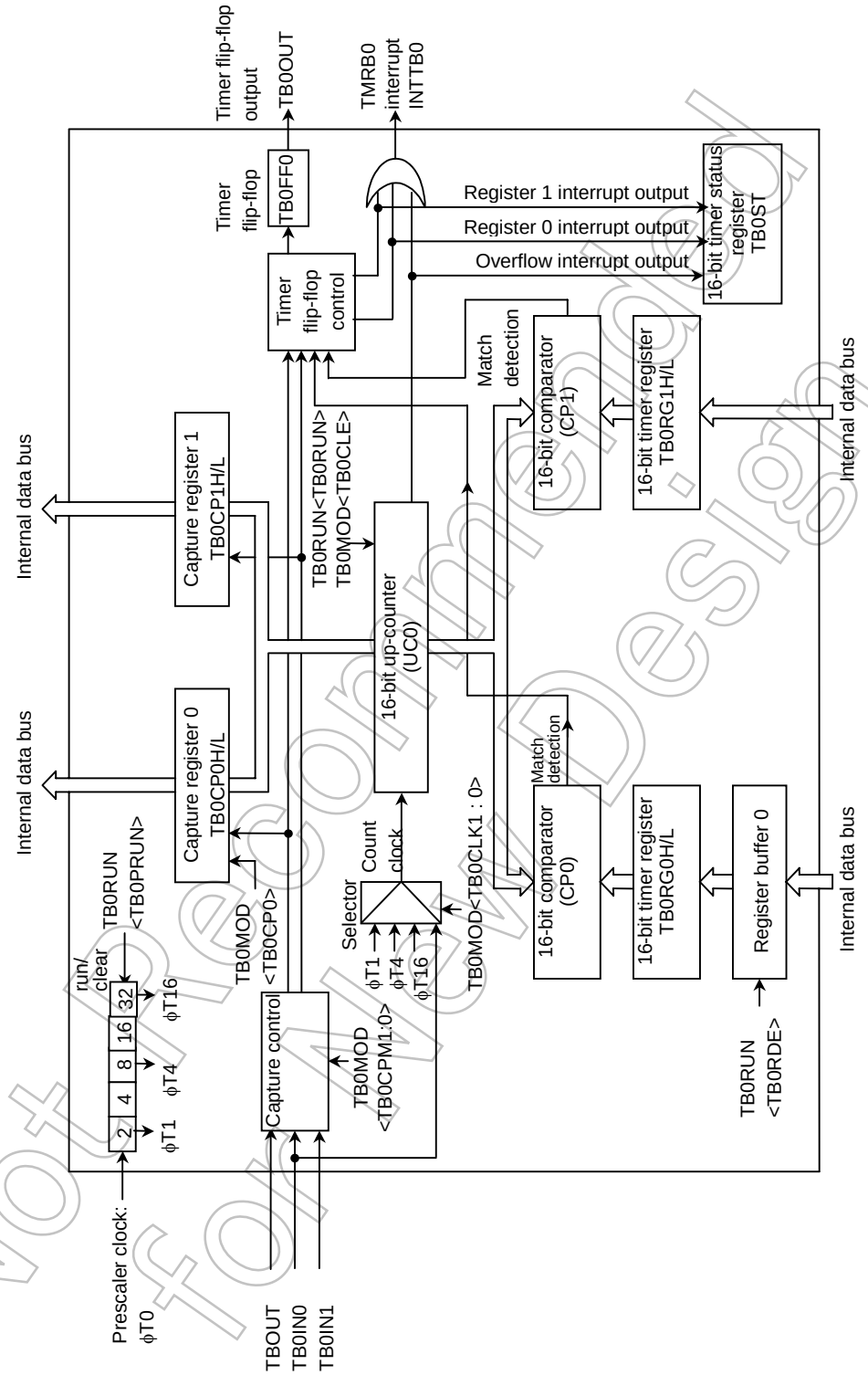


Fig. 11-1 TMRB0 Block Diagram (Same for Channels 1, 4, 5 and 8 through F)

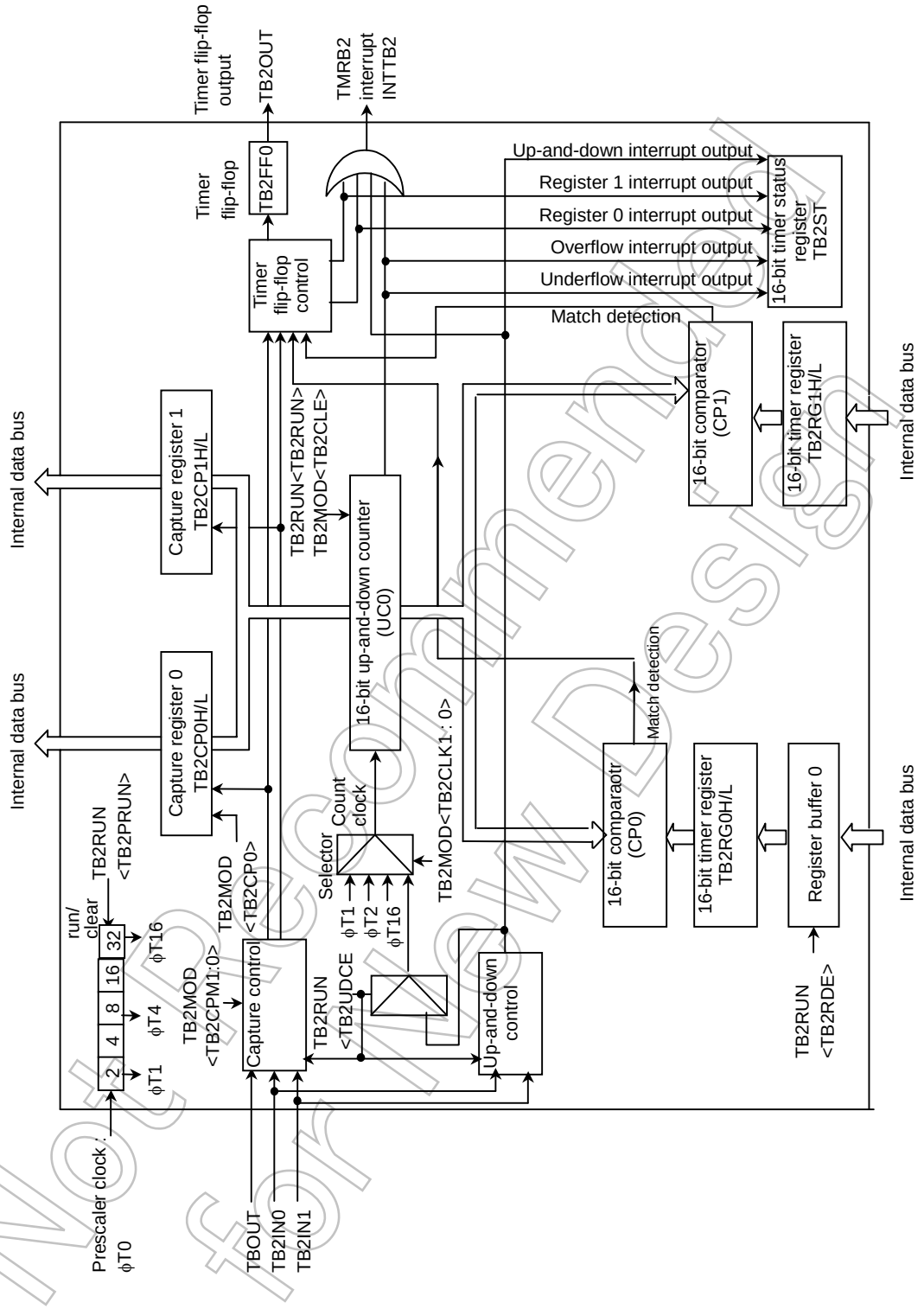


Fig. 11-2 TMRB2 Block Diagram (Same for Channels 3, 6 and 7)

11.2 Description of Operations for Each Circuit

11.2.1 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<PRCK1:0> in the CG. The peripheral clock, fperiph, is either fgear, a clock selected by SYSCR1<FPSEL> in the CG, or fc, which is a clock before it is divided by the clock gear.

The operation or the stoppage of a prescaler is set with TB0RUN<TB0PRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 11-2 shows prescaler output clock resolutions.

Table 11-2 Prescaler Output Clock Resolutions @ = 40MHz

Release peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolutions		
			ϕT1	ϕT4	ϕT16
0 (fgear)	000 (fc)	00(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		10(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		11(fperiph/2)	fc/2^2(0.1μs)	fc/2^4(0.4μs)	fc/2^6(1.6μs)
	100(fc/2)	00(fperiph/16)	fc/2^6(1.6μs)	fc/2^8 (6.4μs)	fc/2^{10}(25.6μs)
		01(fperiph/8)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		10(fperiph/4)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		11(fperiph/2)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
	110(fc/4)	00(fperiph/16)	fc/2^7(3.2μs)	fc/2^9(12.8μs)	fc/2^{11} (51.2μs)
		01(fperiph/8)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		10(fperiph/4)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		11(fperiph/2)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
	111(fc/8)	00(fperiph/16)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)	fc/2^{12}(102.4μs)
		01(fperiph/8)	fc/2^7(3.2μs)	fc/2^9 (12.8μs)	fc/2^{11} (51.2μs)
		10(fperiph/4)	fc/2^6(1.6μs)	fc/2^8(6.4μs)	fc/2^{10}(25.6μs)
		11(fperiph/2)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
1 (fc)	000 (fc)	00(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		10(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		11(fperiph/2)	fc/2^2(0.1μs)	fc/2^4(0.4μs)	fc/2^6(1.6μs)
	100(fc/2)	00(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		10(fperiph/4)	fc/2^3(0.2μs)	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		11(fperiph/2)	—	fc/2^4(0.4μs)	fc/2^6(1.6μs)
	110(fc/4)	00(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^4(0.4μs)	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		10(fperiph/4)	—	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		11(fperiph/2)	—	fc/2^4(0.4μs)	fc/2^6(1.6μs)
	111(fc/8)	00(fperiph/16)	fc/2^5(0.8μs)	fc/2^7(3.2μs)	fc/2^9(12.8μs)
		01(fperiph/8)	—	fc/2^6(1.6μs)	fc/2^8(6.4μs)
		10(fperiph/4)	—	fc/2^5(0.8μs)	fc/2^7(3.2μs)
		11(fperiph/2)	—	—	fc/2^6(1.6μs)

(Note 1) The prescaler output clock ϕTn must be selected so that ϕTn<fsys/2 is satisfied (so that ϕTn is slower than fsys/2).

(Note 2) Do not change the clock gear while the timer is operating.

(Note 3) "—" denotes a setting prohibited.

11.2.2 Up-counter (UC0) and Up-counter Capture Registers (TB0UCL, TB0UCH)

This is the 16-bit binary counter that counts up in response to the input clock specified by TB0MOD<TB0CLK1:0>.

UC0 input clock can be selected from either three types - 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 TB0RG1H/L timer register, it is cleared to "0" if the setting is "clear enable." Clear enable/disable is specified by TB0MOD<TB0CLE>.

If the setting is "clear disable," the counter operates as a free-running counter.

The current count value of the UC0 can be captured by reading the TB0UCL and TB0UCH registers.

Note Make sure that reading is performed in the order of low-order bits followed by high-order bits.

If UC0 overflow occurs, the INTTB01 overflow interrupt is generated.

TMRB2, TMRB3, TMRB6 and TMRB7 have the two-phase pulse input count function. The two-phase pulse count mode is activated by TB2RUN<TB2UDCE>. This counter serves as the up-and-down counter, and is initialized to 0x7FFF. If a counter overflow occurs, the initial value 0x0000 is reloaded. If a counter underflow occurs, the initial value 0xFFFF count is continued.. When the two-phase pulse count mode is not active, the counter counts up only.

11.2.3 Timer Registers (TB0RG0H/L, TB0RG1H/L)

These are 16-bit registers for specifying counter values and two registers are built into each channel. If a value set on this timer register matches that on a UC0 up-counter, the match detection signal of the comparator becomes active.

To write data to the TB0RG0H/L and TB0RG1H/L timer registers, either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits can be used.

TB0RG0 of this timer register is paired with register buffer 0 - a double-buffered configuration. TB0RG0 uses TB0RUN<TB0RDE> to control the enabling/disabling of double buffering so that if <TB0RDE> = "0," double buffering is disabled and if <TB0RDE> = "1," it is enabled. If double buffering is enabled, data is transferred from register buffer 0 to the TB0RG0 timer register when there is a match between UC0 and TB0RG1.

The values of TB0RG0H/L and TB0RG1H/L become undefined after a reset so to use a 16-bit timer, it is necessary to write data to them beforehand. A reset initializes TB0RUN <TB0RDE> to "0" and sets double buffering to "disable." To use double buffering, write data to the timer register, set <TB0RDE> to "1" and then write the following data to the register buffers.

TB0RG0 and the register buffers are assigned to the same address: 0xFFFF_F148/0xFFFF_F149. If <TB0RDE> = "0," the same value is written to TB0RG0 and each register buffer; if <TB0RDE> = "1," the value is only written to each register buffer. To write an initial value to the timer register, therefore, the register buffers must be set to "disable."

Note) Please rewrite neither TBxRG1 nor TBxRG0 a double buffer unused while the timer is working.
Note) When a double buffer is used, data is not updated while rewriting TBxREG0.

Capture Registers (TB0CP0H/L, TB0CP1H/L)

These are 16-bit registers for latching values from the UC0 up-counter. To read data from the capture register, use either a 2- or 1-byte data transfer instruction. Please read it in order of the title in subordinate position

11.2.4 Capture

This is a circuit that controls the timing of latching values from the UC0 up-counter into the TB0CP0 and TB0CP1 capture registers. The timing with which to latch data is specified by TB0MOD <TB0CPM1:0>.

Software can also be used to import values from the UC0 up-counter into the capture register; specifically, UC0 values are taken into the TB0CP0 capture register each time "0" is written to TB0MOD<TB0CP0>. To use this capability, the prescaler must be running (TB0RUN<TB0PRUN> = "1").

In the two-phase pulse count mode (for the TMRB2, TMRB3, TMRB6 and TMRB7 only), the counter value is captured by using software.

(Note 1) Although a read of low-order 8 bits in the capture register suspends the capture operation, it is resumed by successively reading high-order 8 bits.

(Note 2) If the timer stops after a read of low-order 8 bits, the capture operation remains suspended even after the timer restarts. Please ensure that the timer is not stopped after a read of low-order 8 bits.

11.2.5 Comparators (CP0, CP1)

These are 16-bit comparators for detecting a match by comparing set values of the UC0 up-counter with set values of the TB0RG0 and TB0RG1 timer registers. If a match is detected, INTTB0 is generated.

11.2.6 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.

11.3 Register Description

TMRBn RUN register (n=0, 1, 4, 5, 8 through F)

TBnRUN (0xFFFF_F1x0)

	7	6	5	4	3	2	1	0
Bit symbol	TBnRDE				I2TBn	TBnPRUN		TBnRUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffering 0: Disable 1: Enable	Write "0."	Write "0."	Write "0."	In the IDLE mode 0: Stop 1: Operate	Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as "0."		

<TBnRUN>: Controls the TMRB0 count operation.

<TBnPRUN>: Controls the TMRB0 prescaler operation.

<I2TBn>: Controls the operation in the IDLE mode.

<TBnRDE>: Controls enabling/disabling of double buffering.

(Note 1) The value read from bit 1 of TBnRUN is "0."

(Note 2) Do not set bits 7 to 3 (counter operating conditions) and bits 2 to 0 (count start) simultaneously

TMRBm RUN register (m=2, 3, 6, 7)

TBnRUN (0xFFFF_F1x0)

	7	6	5	4	3	2	1	0
Bit symbol	TBmRDE		UDmCK	TBmUDCE	I2TBm	TBmPRUN		TBmRUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffering 0: Disable 1: Enable	Write "0."	Sampling clock 0: fs 1: ϕT0/4	Enable/ disable two-phase counter 0: Disable 1: Enable	IDLE 0: Stop 1: Operate	Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as "0."		

<TBmRUN>: Controls the TMRB0 count operation.

<TBmPRUN>: Controls the TMRB0 prescaler operation.

<I2TBm>: Controls the operation in the IDLE mode.

<TBmUDCE>: Controls enabling/disabling of the two-phase pulse input count operation.

Enable: The counter counts up and counts down.

Disable: This is the normal timer mode and the counter counts up only.

<UdmCK>: Selects the two-phase pulse input sampling clock.

<TBmRDE>: Controls enabling/disabling of double buffering.

(Note 1) The value read from bit 1 of TBmRUN is "0."

(Note 2) Do not set bits 7 to 3 (counter operating conditions) and bits 2 to 0 (count start) simultaneously.

Fig. 11-3 TMRB-related Registers

TMRBn control register (n=2, 3, 6, 7)

TBnCR (0xFFFF_F1x1)

	7	6	5	4	3	2	1	0
Bit symbol	TBnEN				TBnSYC	UDnNF	UDnCNT	
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R
After reset	0	0	0	0	0	0	0	0
Function	TMRBn operation 0: Disable 1: Enable	Write "0."	This can be read as "0."	This can be read as "0."	Synchronization mode switch-over 0: Individual operation 1: Synchronous operation	Digital noise filter 0: No use 1: Use	Mode switch-over 0: Normal 1: Quadruple	This can be read as "0."

TMRBn control register (n=0, 1, 4, 5, 8 through F)

TBnCR (0xFFFF_F1x1)

	7	6	5	4	3	2	1	0
Bit symbol	TBnEN				TBnSYC			
Read/Write	R/W	R/W	R	R	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	TMRBn operation 0: Disable 1: Enable	Write "0."	This can be read as "0."	This can be read as "0."	Synchronization mode switch-over 0: Individual operation 1: Synchronous operation	This can be read as "0."	This can be read as "0."	This can be read as "0."

<TbnEN>: Specifies the TMRB operation. When the operation is disabled, no clock is supplied to the other registers in the TMRB module. This can reduce power dissipation. (This disables reading from and writing to the other registers.) To use the TMRB, enable the TMRB operation (set to "1") before programming each register in the TMRB module. If the TMRB operation is executed and then disabled, settings will be maintained in each register.

TMRBn mode register (n=0 through F)

TBnMOD (0xFFFF_F1x2)

	7	6	5	4	3	2	1	0
Bit symbol			TBnCP0	TBnCPM1	TBnCPM0	TBnCLE	TBnCLK1	TBnCLK0
Read/Write	R		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	This can be read as "00."		Capture control by software 0: Capture by software 1: Don't care	Capture timing 00: Disable 01: TBnIN0 ↑ TBnIN1 ↑ 10: TBnIN0 ↑ TBnIN0 ↓ 11: TB3OUT ↑ TB3OUT ↓		Up-counter control 0: Clear/disable 1: Clear/enable	Selects source clock 00: TB0IN0 pin input 01: ϕT1 10: ϕT4 11: ϕT16	

<TBnCLK1:0>: Selects the TMRBn timer count clock.

<TBnCLE>: Clears and controls the TMRBn up-counter.

"0": Disables clearing of the up-counter.

"1": Clears up-counter if there is a match with timer register 1 (TBnRG1).

<TBnCPM1:0>: Specifies TMRBn capture timing.

"00": Capture disable

"01": Takes count values into capture register 0 (TBnCP0) upon the rising of TBnIN0 pin input.
Takes count values into capture register 1 (TBnCP1) upon the rising of TBnIN1 pin input.

"10": Takes count values into capture register 0 (TBnCP0) upon the rising of TBnIN0 pin input.
Takes count values into capture register 1 (TBnCP1) upon the falling of TBnIN0 pin input.

"11": Takes count 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).

<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."

Fig. 11-4 TMRB-related Register

TMRBn flip-flop control register (n=0 through F)

TBnFFCR (0xFFFF_F1x3)

	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 as "11."	
			When the up-counter value is taken into TBnCP1	When the up-counter value is taken into TBnCP0	When the up-counter matches TBnRG1	When the up-counter matches TBnRG0		

<TBnFF0C1:0>: Controls the timer flip-flop.

"00": Reverses the value of TBnFF0 (reverse by using software).

"01": Sets TBnFF0 to "1."

"10": Clears TBnFF0 to "0."

"11": Don't care

<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).

Fig. 11-5 TMRB-related Register

TMRBn status register (n=0, 1, 4, 5, 8 through F)

TBnST (0xFFFF_F1x4)

	7	6	5	4	3	2	1	0
Bit symbol						INTTBOFn	INTTBn1	INTTBn0
Read/Write	R					R		
After reset	0					0	0	0
Function	This can be read as "0."					0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

<INTTBn0>: Interrupt generated if there is a match with timer register 0 (TBnRG0)
<INTTBn1>: Interrupt generated if there is a match with timer register 1 (TBnRG1)
<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.

TMRBm status register (m=2, 3, 6, 7)

① When TBmRUN <TBmUDCE> = 0: Normal timer mode

TBnST (0xFFFF_F1x4)

	7	6	5	4	3	2	1	0
Bit symbol						INTTBOFm	INTTBm1	INTTBm0
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

<INTTBm0>: Interrupt generated if there is a match with timer register 0 (TBmRG0)
<INTTBm1>: Interrupt generated if there is a match with timer register 1 (TBmRG1)
<INTTB0Fm>: Interrupt generated if an up-counter overflow occurs

(Note) If any interrupt is generated, the flag that corresponds to the interrupt is set to TBmST and the generation of interrupt is notified to INTC. The flag is cleared by reading the TBmST register.

② When TBmRUN <TBmUDCE> = 1: Two-phase pulse input count mode

TBnST (0xFFFF_F1x4)

	7	6	5	4	3	2	1	0
Bit symbol				INTTBUDm	INTTBUDFm	INTTBOUFm		
Read/Write	R			R			R	
After reset	0			0	0	0	0	
Function	This can be read as "0."			Up-and-down count 0: Not generated 1: Generated	Underflow 0: Not generated 1: Generated	Overflow 0: Not generated 1: Generated	This can be read as "0."	

INTTBUDF2: An up-and-down counter underflow occurs.
INTTBOUF2: An up-and-down counter overflow occurs.
INTTBUD2: An up- or down-count occurs.

<INTTBOUFm>: Interrupt generated if an up-and-down counter overflow occurs
<INTTBUDFm>: Interrupt generated if an up-and-down counter underflow occurs
<INTTBUDm>: Interrupt generated if an up- or down-count occurs

(Note) If any interrupt is generated, the flag that corresponds to the interrupt is set to TBmST and the generation of interrupt is notified to INTC. The flag is cleared by reading the TBmST register.

Fig. 11-6 TMRB-related Register

TBnRG0H/L and TBnRG1H/L timer registers

TBnRG0H/L timer registers (n=0 through F)

TBnRG0L (0xFFFF_F1x8)

	7	6	5	4	3	2	1	0
Bit symbol	TBnRG0L7	TBnRG0L6	TBnRG0L5	TBnRG0L4	TBnRG0L3	TBnRG0L2	TBnRG0L1	TBnRG0L0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of low-order 8 bits							

TBnRG0H (0xFFFF_F1x9)

	7	6	5	4	3	2	1	0
Bit symbol	TBnRG0H7	TBnRG0H6	TBnRG0H5	TBnRG0H4	TBnRG0H3	TBnRG0H2	TBnRG0H1	TBnRG0H0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of high-order 8 bits							

(Note) To write data to the timer registers, use either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.

TBnRG1H/L timer registers (n=0 through F)

TBnRG1L (0xFFFF_F1xA)

	7	6	5	4	3	2	1	0
Bit symbol	TBnRG1L7	TBnRG1L6	TBnRG1L5	TBnRG1L4	TBnRG1L3	TBnRG1L2	TBnRG1L1	TBnRG1L0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of low-order 8 bits							

TBnRG1H (0xFFFF_F1xB)

	7	6	5	4	3	2	1	0
Bit symbol	TBnRG1H7	TBnRG1H6	TBnRG1H5	TBnRG1H4	TBnRG1H3	TBnRG1H2	TBnRG1H1	TBnRG1H0
Read/Write	W							
After reset	Undefined							
Function	Timer count value, Data of high-order 8 bits							

(Note) To write data to the timer registers, use either a 2-byte data transfer instruction or a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.

TBnCP0H/L and TBnCP1H/L capture registers

TBnCP0H/L capture registers (n=0 through F)

TBnCP0L (0xFFFF_F1xC)

	7	6	5	4	3	2	1	0
Bit symbol	TBnCP0L7	TBnCP0L6	TBnCP0L5	TBnCP0L4	TBnCP0L3	TBnCP0L2	TBnCP0L1	TBnCP0L0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of low-order 8 bits							

TBnCP0H (0xFFFF_F1xD)

	7	6	5	4	3	2	1	0
Bit symbol	TBnCP0H7	TBnCP0H6	TBnCP0H5	TBnCP0H4	TBnCP0H3	TBnCP0H2	TBnCP0H1	TBnCP0H0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of high-order 8 bits							

(Note) To read data from the capture registers, use a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.
Don't use a 2-byte data transfer instruction.

TBnCP1H/L capture registers (n=0 through F)

TBnCP1L (0xFFFF_F1xE)

	7	6	5	4	3	2	1	0
Bit symbol	TBnCP1L7	TBnCP1L6	TBnCP1L5	TBnCP1L4	TBnCP1L3	TBnCP1L2	TBnCP1L1	TBnCP1L0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of low-order 8 bits							

TBnCP1H (0xFFFF_F1xF)

	7	6	5	4	3	2	1	0
Bit symbol	TBnCP1H7	TBnCP1H6	TBnCP1H5	TBnCP1H4	TBnCP1H3	TBnCP1H2	TBnCP1H1	TBnCP1H0
Read/Write	R							
After reset	Undefined							
Function	Timer capture value, Data of high-order 8 bits							

(Note) To read data from the capture registers, use a 1-byte data transfer instruction written twice in the order of low-order 8 bits followed by high-order 8 bits.
Don't use a 2-byte data transfer instruction.

11.4 Description of Operations for Each Mode

11.4.1 16-bit Interval Timer Mode

Generating interrupts at periodic cycles

To generate the INTTB0 interrupt, specify a time interval in the TB0RG1 timer register.

		7	6	5	4	3	2	1	0	
TB0CR		1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	0	0	0	0	–	0	X	0	Stops TMRB0.
IMC8	←	0	1	1	0	0	0	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.)
		0	1	1	0	0	0	0	0	
TB0FFCR	←	1	1	0	0	0	0	1	1	Disables the trigger.
TB0MOD	←	0	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock,
TB0RG1L	←	*	*	*	*	*	*	*	*	and specifies the time interval.
TB0RG1H		*	*	*	*	*	*	*	*	(16 bits)
TB0RUN	←	0	0	0	0	–	1	X	1	Starts TMRB0.

X; Don't care –; no change

11.4.2 16-bit Event Counter Mode

By using an input clock as an external clock (TB0IN0 pin input), it is possible to make it the event counter.

The up-counter counts up on the rising edge of TB0IN0 pin input. By capturing a value using software and reading the captured value, it is possible to read the count value.

		7	6	5	4	3	2	1	0	
TB0CR	←	1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	0	0	0	0	–	0	X	0	Stops TMRB0.
P2CR	←	–	–	–	–	–	–	–	0	Sets P20 to the input mode.
P2FC	←	–	–	–	–	–	–	–	1	
P2FC2	←	–	–	–	–	–	–	–	0	
IMC8	←	0	1	1	0	0	0	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.)
		0	1	1	0	0	0	0	0	
TB0FFCR	←	1	1	0	0	0	0	1	1	Disables the trigger.
TB0MOD	←	0	0	1	0	0	1	0	0	Designates the TB0IN0 pin input as the input clock.
TB0RUN	←	0	0	0	0	–	1	X	1	Starts TMRB0.
TB0MOD	←	X	X	0	0	0	1	0	0	Captures a value using software.
TB0RG1L	←	*	*	*	*	*	*	*	*	Specifies the time interval.
TB0RG1H		*	*	*	*	*	*	*	*	(16 bits)

X; Don't care –; no change
To be used as the event counter, put the prescaler in a "RUN" state (TB0RUN<TB0PRUN> = "1").

11.4.3 16-bit PPG (Programmable Square Wave) Output Mode

Square waves with any frequency and any duty (programmable square waves) can be output. The output pulse can be either low-active or high-active.

Programmable square waves can be output from the TB0OUT pin by triggering the timer flip-flop (TB0FF) to reverse when the set value of the up-counter matches the set values of the timer registers (TB0RG0H/L and TB0RG1H/L). Note that the set values of TB0RG0H/L and TB0RG1H/L must satisfy the following requirement:

(Set value of TB0RG0H/L) < (Set value of TB0RG1H/L)

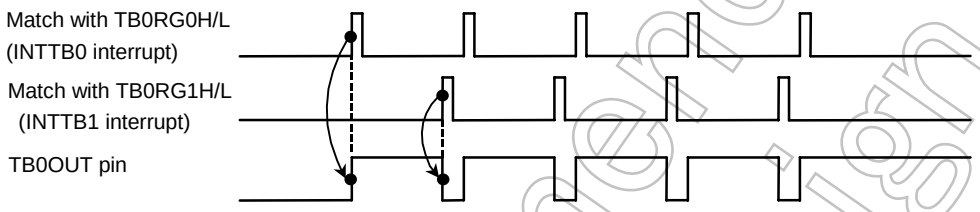


Fig. 11-7 Example of Output of Programmable Square Wave (PPG)

In this mode, by enabling the double buffering of TB0RG0H/L, the value of register buffer 0 is shifted into TB0RG0H/L when the set value of the up-counter matches the set value of TB0RG1H/L. This facilitates handling of small duties.

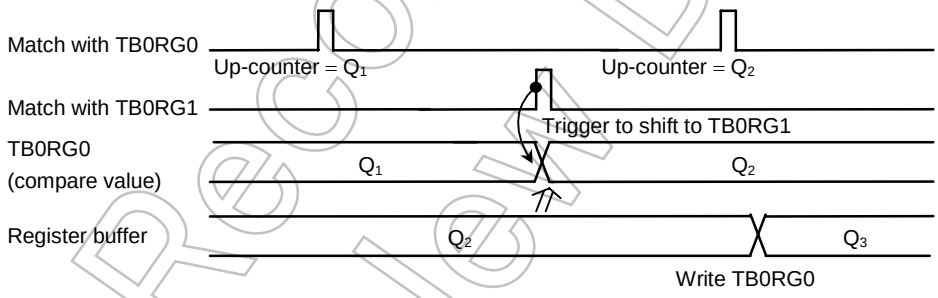


Fig. 11-8 Register Buffer Operation

Note: Double buffering is available for TB0RG0 only. Pay attention to changes to TB0RG1.

The block diagram of this mode is shown below.

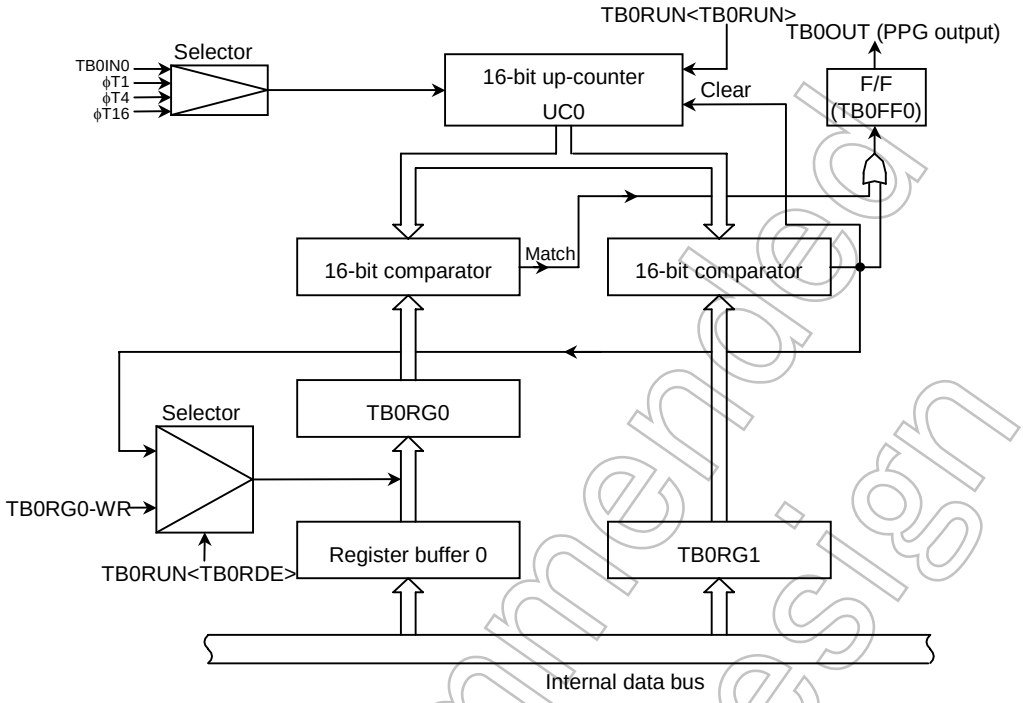


Fig. 11-9 Block Diagram of 16-bit PPG Mode

Each register in the 16-bit PPG output mode must be programmed as listed below.

		7	6	5	4	3	2	1	0	
TB0CR	←	1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB0RUN	←	0	0	0	0	–	0	X	0	Disables the TB0RG0 double buffering and stops TMRB0.
TB0RG0L	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
TB0RG0H	←	*	*	*	*	*	*	*	*	
TB0RG1L	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
TB0RG1H	←	*	*	*	*	*	*	*	*	
TB0RUN	←	1	0	0	0	–	0	X	0	Enables the TB0RG0 double buffering. (Changes the duty/cycle when the INTTB0 interrupt is generated)
TB0FFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TB0FF0 to reverse when a match with TB0RG0 or TB0RG1 is detected, and sets the initial value of TB0FF0 to "0."
TB0MOD	←	0	0	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function. (** = 01, 10, 11)
P5CR	←	–	–	–	1	–	–	–	–	Assigns P54 to TB0OUT.
P5FC	←	–	–	–	1	–	–	–	–	
P5FC2	←	–	–	–	0	–	–	–	–	
TB0RUN	←	1	0	0	0	–	1	X	1	Starts TMRB0.

X; Don't care –; no change

11.4.4 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 four blocks, TMRB0 through TMRB3, TMRB4 through TMRB7, TMRB8 through TMRBB and TMRBC through TMRBF.

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 or TMRBC.

Note:
- **For the channels to be output synchronously, set TBmRUN<TBmPRUN,TBmRUN> to "1,1" to enable simultaneous start before starting TMRB0, TMRB4, TMRB8 or TMRBC.**
- **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 and C start.**

note) Master : TMRB0, TMRB4, TMRB8, TMRBC write TBnSYC" 0"

TBnCR (0xFFFF_F1x1)

	7	6	5	4	3	2	1	0
Bit symbol	TBnEN				**TBnSYC**			
Read/Write	R/W	R/W	R	R	**R/W**	R	R	R
After reset	0	0	0	0	**0**	0	0	0
Function	TMRBn operation 0: Disable 1: Enable	Write "0."	This can be read as "0."	This can be read as "0."	**Write"0"**	This can be read as "0."	This can be read as "0."	This can be read as "0."

Slave : Write TBnSYC " 1"

TBnCR (0xFFFF_F1x1)

	7	6	5	4	3	2	1	0
Bit symbol	TBnEN				**TBnSYC**			
Read/Write	R/W	R/W	R	R	**R/W**	R	R	R
After reset	0	0	0	0	**0**	0	0	0
Function	TMRBn operation 0: Disable 1: Enable	Write "0."	This can be read as "0."	This can be read as "0."	Write "1"	This can be read as "0."	This can be read as "0."	This can be read as "0."

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 (TB6CP0H/L).

The INTC must be programmed so that an interrupt INT0 is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TB6RG0H/L) to the sum of the TB6CP0 value (c) and the delay time (d), (c + d), and set the timer registers (TB6RG1H/L) to the sum of the TB6RG0H/L 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 TB6RG0H/L and TB6RG1H/L. 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. 11-10 One-shot Pulse Output (With Delay)."

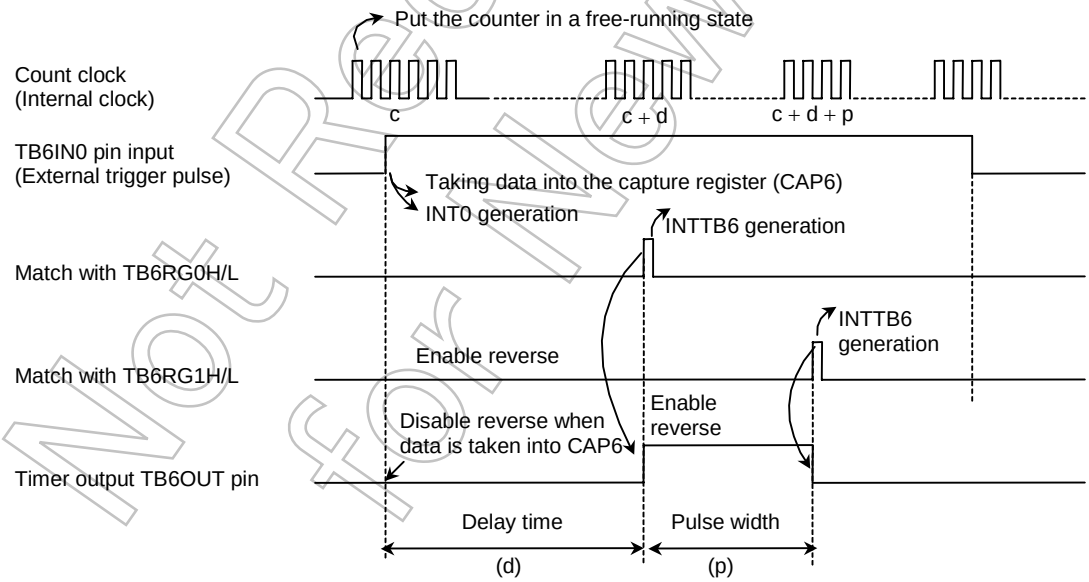


Fig. 11-10 One-shot Pulse Output (With Delay)

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/4 (fperiph fsys)

Main programming

		7	6	5	4	3	2	1	0	
										→ Puts to a free-running state
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
P9CR	←	–	–	–	–	–	–	–	1	Assigns P90 pin to TB6OUT
P9FC	←	–	–	–	–	–	–	–	1	
IMC0	←	X	X	0	0	0	0	0	0	
	←	X	X	1	1	0	1	0	0	Enables INT0 and disables INTTB6
	←	X	X	0	0	0	0	0	0	
	←	X	X	0	0	0	0	0	0	These are 32-bit registers and must be all processed.
IMC9	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	
TB6RUN	←	–	0	X	0	–	1	X	1	Starts TMRB6

INT0 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 TB2RG0, 1
IMC9	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	Enables INTTB6
	←	X	X	1	1	0	1	0	0	

INTTB6 programming

TB6FFCR	←	X	X	–	–	0	0	–	–	→ Disables TB6FF0 to reverse when there is a match with TB6RG0, 1
IMC9	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	
	←	X	X	1	1	0	0	0	0	
	←	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 INT0 interrupt. TB6FF0 is enabled to reverse when UC6 matches with TB6RG1, and is disabled by generating the INTTB6 interrupt.

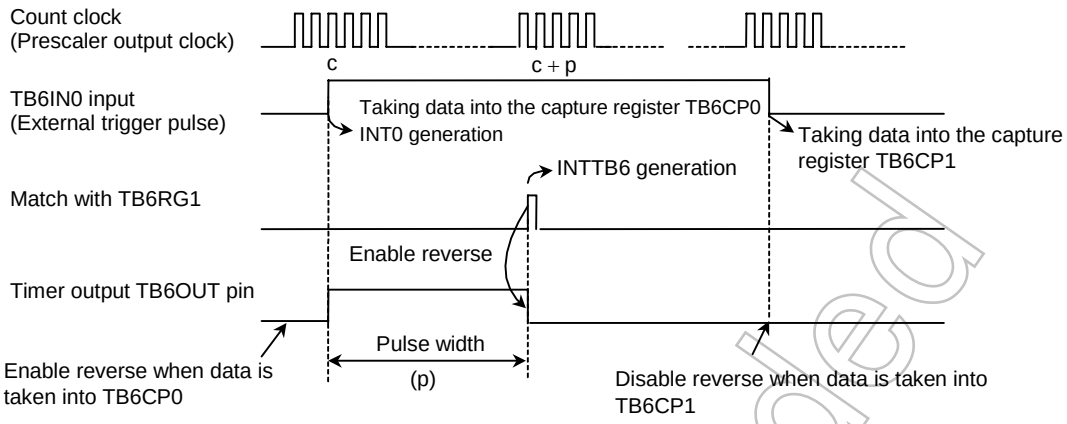


Fig. 11-11 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

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 (TB0CP0) upon the rising of a timer flip-flop (TB3FFCR) of the 16-bit timer (TMRB3), and an UC0 counter value to be taken into the capture register (TB0CP1) upon the falling of TB3FF of the 16-bit timer (TMRB3).

A frequency is then obtained from the difference between TB0CP0 and TB0CP1 based on the measurement, by generating the INTTB3 16-bit timer interrupt.

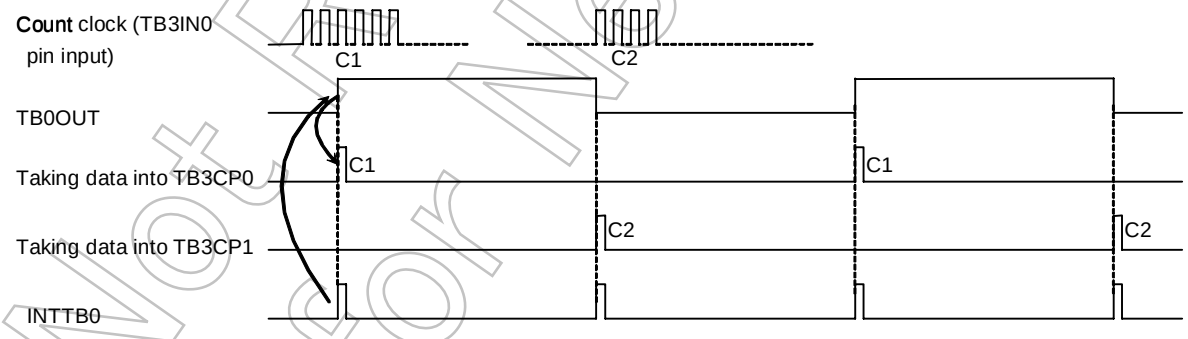


Fig. 11-12 Frequency Measurement

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 (TB0CP0, TB0CP1). The INTC must be programmed so that INT0 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 $100 \times 0.5\ \mu s = 50\ \mu s$.

Caution must be exercised when measuring pulse widths exceeding the UC2 maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.

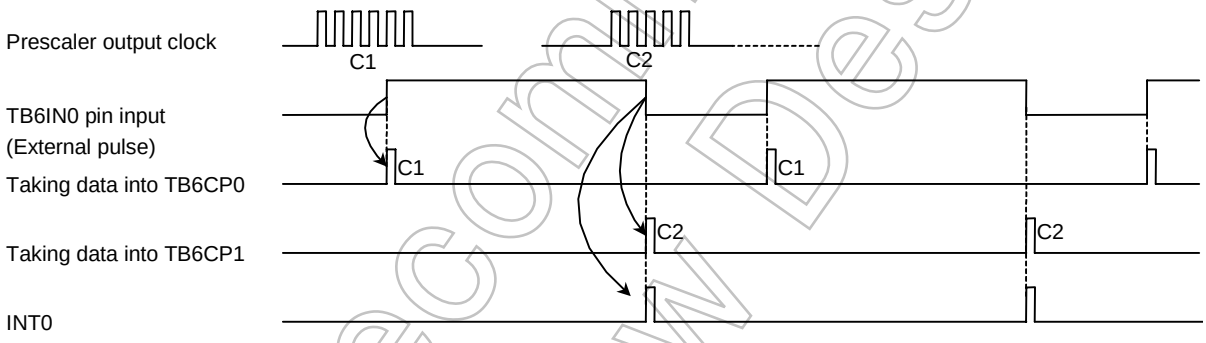


Fig. 11-13 Pulse Width Measurement

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 INT0 interrupt processing as shown in "Fig. 11-14 Time Difference Measurement," 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 INT0 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 INT1 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.

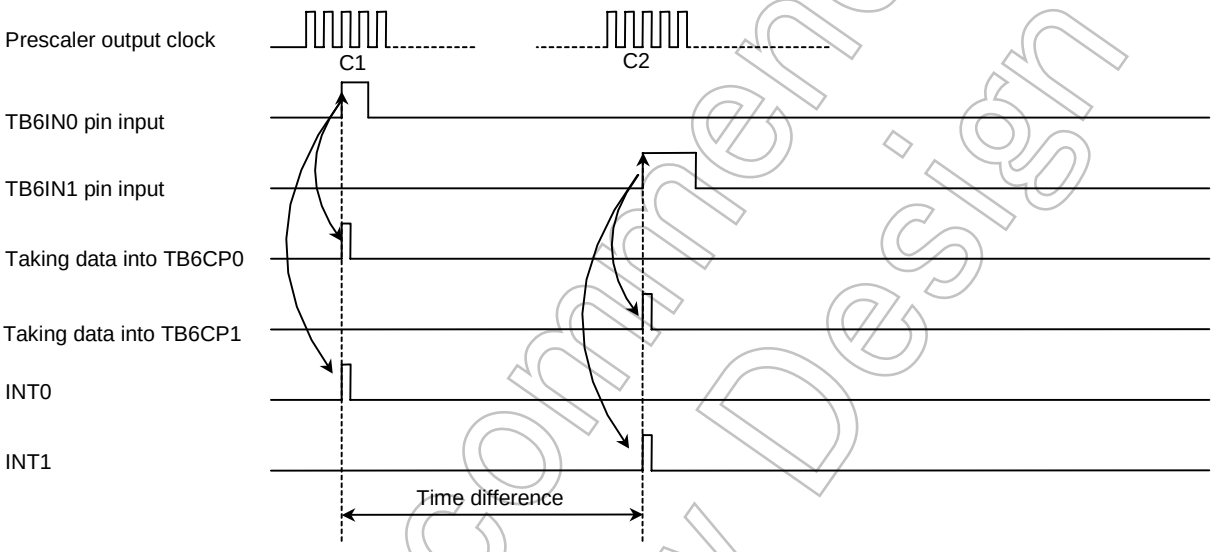


Fig. 11-14 Time Difference Measurement

11.5 Two-phase Pulse Input Count Mode (TMRB2, TMRB3, TMRB6, TMRB7)

(Operations are common to TMRB2, 3, 6 and 7. This section describes TMRB2 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 TB2IN0 and TB2IN1 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)

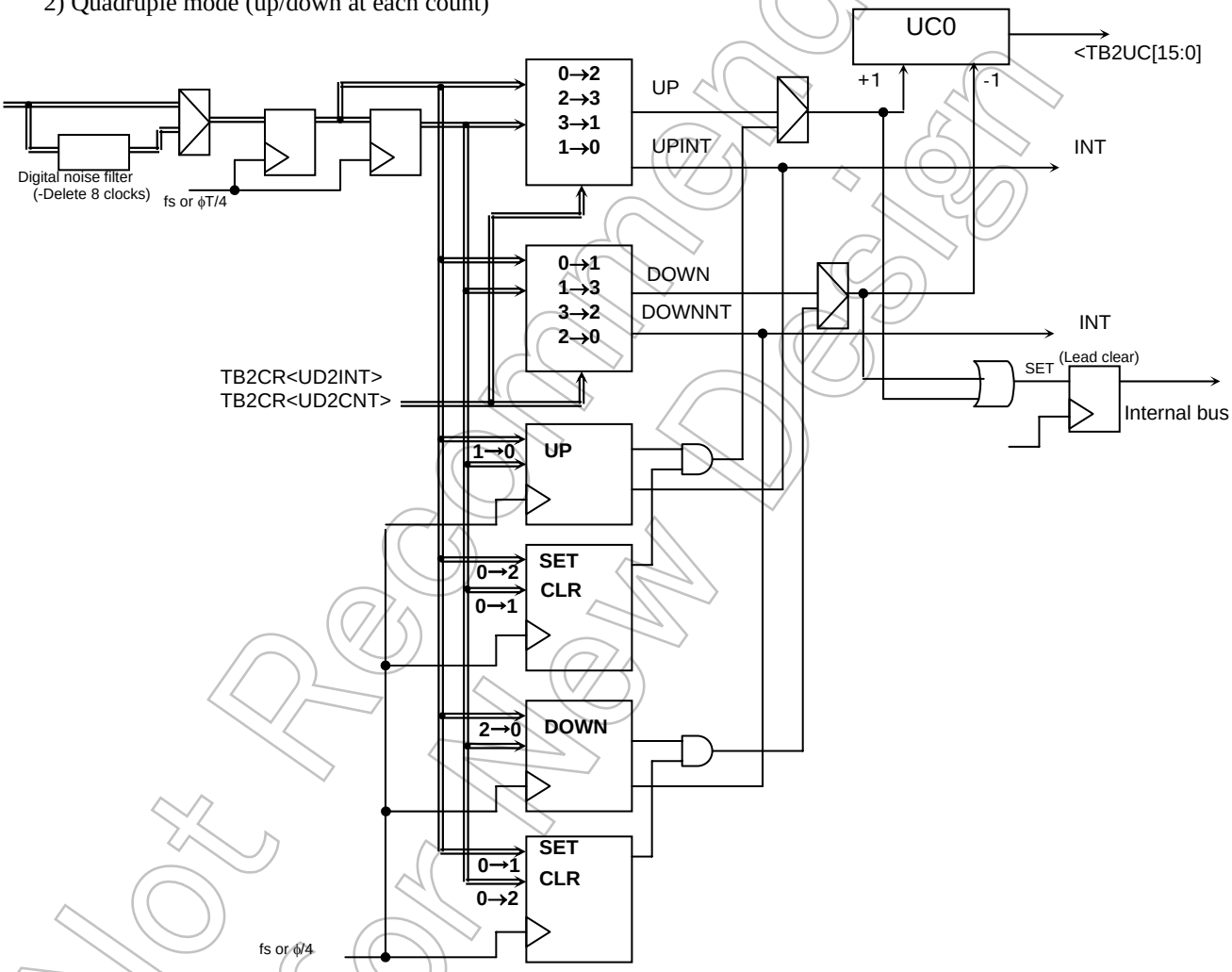
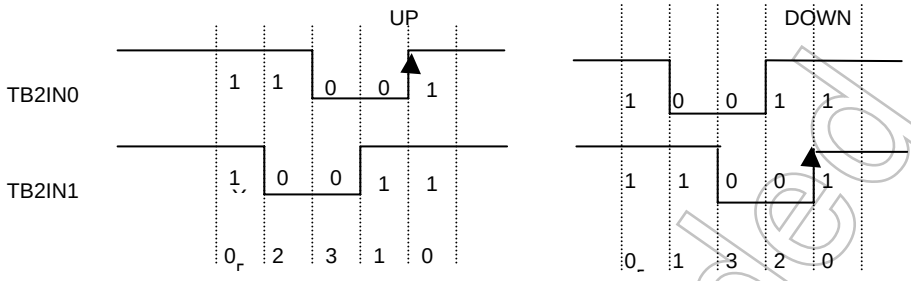


Fig. 11-15 Count Circuit of Two-phase Counter

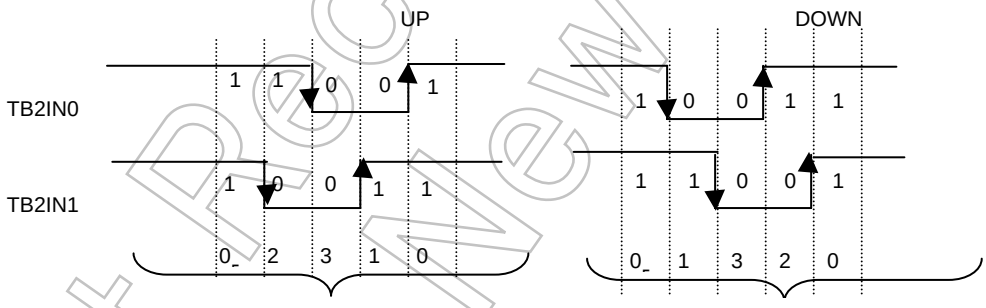
• Normal operation count mode



	Pin state						
Count condition	UP			DOWN			
TB2IN0, TB2IN1	0	→	2	0	→	1	← SET
		↓					
TB2IN0, TB2IN1	1	→	0	2	→	0	← Interrupt generated
TB2IN0, TB2IN1	1	→	0	2	→	0	← CLR

Note: Changes from 0 to 3 and from 3 to 0 are considered as irregular states and are not counted. Up and down state settings are cleared.

• Multiplication-by-4 operation count mode



① Count up at each edge **② Count down at each edge**

	Pin state					
Count condition	UP			DOWN		
TB2IN0, TB2IN1	0	→	2	0	→	1
	2		3	1		3
	3		1	3		2
	1		0	2		0

TMRB2RUN register (TB2RUN)

TB2RUN (0xFFFF_F160)

	7	6	5	4	3	2	1	0
bit Symbol	TB2RDE		UD2CK	TB2UDCE	I2TB2	TB2PRUN		TB2RUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffering 0: Disable 1: Enable	Write "0."	Select sampling clock 0: fs 1: ϕT0/4	Enable/ disable two-phase counter 0: Disable 1: Enable	IDLE 0: Stop 1: Operate	Timer Run/Stop Control 0: Stop & Clear 1: Run (Count Up)		

Fig. 11-16 Two-phase Pulse Input Count Mode Setting Register

For the sampling clock, the fifth bit <UD2CK> of the TB2RUN register is set to "1."

<< Recovery from the SLEEP mode >>

1) For TMRB2 and TMRB3

The two-phase counter counts up or down depending on the SLEEP release input state.

2) Recovery by using INT0 through INT3 for TMRB6 and TMRB7

The counter value does not change until the requirements are satisfied. To read the counter value after the SLEEP mode is released, it must be executed when an interrupt is generated due to counting up or down.

① Operation mode

Register setting determines whether the external input signals from the TB2IN0 and TB2IN1 input pins are input to the normal 16-bit timer (capture input) or the up-and-down counter.

- In the up-and-down counter mode, capture is executed by the software only. Capture at the external clock timing does not work.
- In the up-and-down counter mode, the comparator is disabled and it does not execute comparison with timer registers.
- The input clock sampling is executed by fs (32 KHz/16KHz) or the high-speed clock (system clock). The maximum input frequency is 4 kHz for fs and ϕT0/4 for the high-speed clock.

<< Recovery from the STOP mode >>

Recovery by using INT0 through INT3 for TMRB6 and TMRB7

The two-phase counter enters the STOP mode while it maintains the previous state. Therefore, when the relationship between the input state used for releasing the STOP mode and the maintained state satisfies the requirements for counting up or down, the counter value is incremented or decremented by one (+1 or -1) after the STOP mode is released. If it is necessary to keep a constant state after the STOP mode is released, the two-phase counter must be initialized to "0x7FFF" after the STOP mode is released (by setting TB7RUN <TB7UDCE> to "0" and turning it back to "1"). **This function is unavailable for TMRB2 and TMRB3.**

<< How to program the up-and-down counter >>

Set the TB2MOD register <TB2CLK0, TB2CLK1> to "00" (prescaler OFF). Then, program the fourth bit <TB2UDCE> of the TB2RUN register to determine whether to operate the counter as the up-and-down counter or as the conventional up-counter for external clock input.

TB2UDCE (Enable the up-and-down counter) = "0": Normal 16-bit timer operation
= "1": Up-and-down counter operation

② Interrupt

- In the NORMAL or SLOW mode

  The INTTB2 interrupt is enabled using the interrupt controller (INTC). The INTTB2 interrupt is generated by counting up or down. Reading the status register TB2ST during interrupt handling allows simultaneous check for occurrences of an overflow and an underflow. If TB2ST<INTTBOUF2> is "1," it indicates that an overflow has occurred. If <INTTBUDF2> is "1," it indicates that an underflow has occurred. This register is cleared after it is read. The counter becomes 0x0000 when an overflow occurs, and it becomes 0xFFFF when an underflow occurs. After that, the counter continues the counting operation.

TB2ST (0xFFFF_F164)

	7	6	5	4	3	2	1	0
bit Symbol				INTTBUD2	INTTBUDF2	INTTBOUF2		
Read/Write	R			R			R	
After reset	0			0	0	0	0	
Function	This can be read as "0."			Up-and-down count 0: Not occurred 1: Occurred	Underflow 0: Not occurred 1: Occurred	Overflow 0: Not occurred 1: Occurred	This can be read as "0."	

Fig. 11-17 TMRB2 Status Register

Note: The status is cleared after the register is read.

- In the SLEEP mode

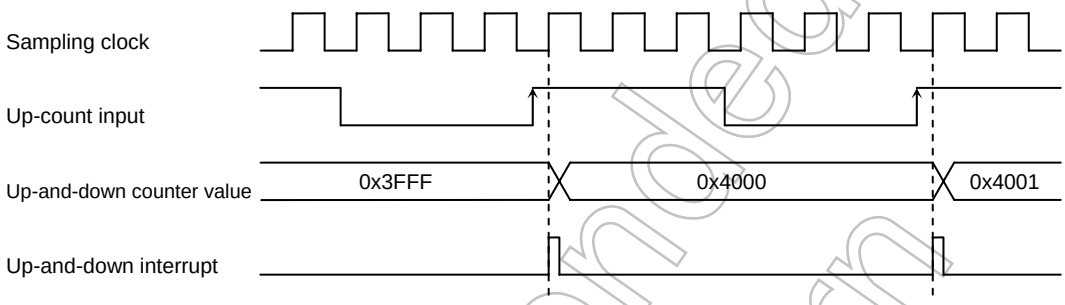
  The two-phase input pulse input counter operates. The INTTB2 interrupt is generated by the count-up or count-down input, and the system recovers from the SLEEP mode. Reading the status register TB2ST during interrupt handling allows simultaneous check for occurrences of an overflow and an underflow. If TB2ST<INTTBOUF2> is "1," it indicates that an overflow has occurred. If <INTTBUDF2> is "1," it indicates that an underflow has occurred. This register is cleared after it is read. The counter becomes 0x0000 when an overflow occurs, it and becomes 0xFFFF when an underflow occurs. After that, the counter continues the counting operation.

- In the STOP mode **(Recovery by using INT0 through INT3 for TMRB6 and TMRB7)**

  The two-phase input pulse input counter is stopped. After the release input and the elapse of warm-up time, the mode changes to the NORMAL mode and the counting operation restarts. When the relationship between the input state used for releasing the STOP mode and the maintained state satisfies the requirements for counting up or down, the counter value is incremented or decremented by one (+1 or -1) after the STOP mode is released.

③ Up-and-down counter

When the two-phase input count mode is selected (TB2RUN<TB2UDCE> = "1"), the up-counter becomes the up-and-down counter and it is initialized to 0x7FFF. If a counter overflow occurs, the counter returns to 0x0000. If a counter underflow occurs, the counter returns to 0xFFFF. After that, the counter continues the counting operation. Therefore, the state can be checked by reading the counter value and the status flag TB2ST 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 INTTB2 interrupt handling

TMRB2 control register

TB2CR (0xFFFF_F162)

	7	6	5	4	3	2	1	0
bit Symbol	TB2EN				TB2SYC	UD2NF	UD2CNT	
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R
After reset	0	0	0	0	0	0	0	0
Function	TMRB2 operation 0: Disable 1: Enable	Write "0."	This can be read as "0."	This can be read as "0."	Synchronization mode switch-over 0: Individual operation 1: Synchronous operation	Digital noise filter 0: No use 1: Use	Mode switch-over 0: Normal 1: Quadruple	This can be read as "0."

- Mode switch-over bit

  0: Normal mode

  1: Quadruple mode

UD2NF Controls noise removal.

If this is set to "1 (Use)," the TMRB2, TMRB3, TMRB6 and TMRB7 pin inputs are removed when they are shorter than 8 system clocks.

Pay close attention to the input signal frequency because an error of one system clock occurs due to synchronization with internal signals.

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

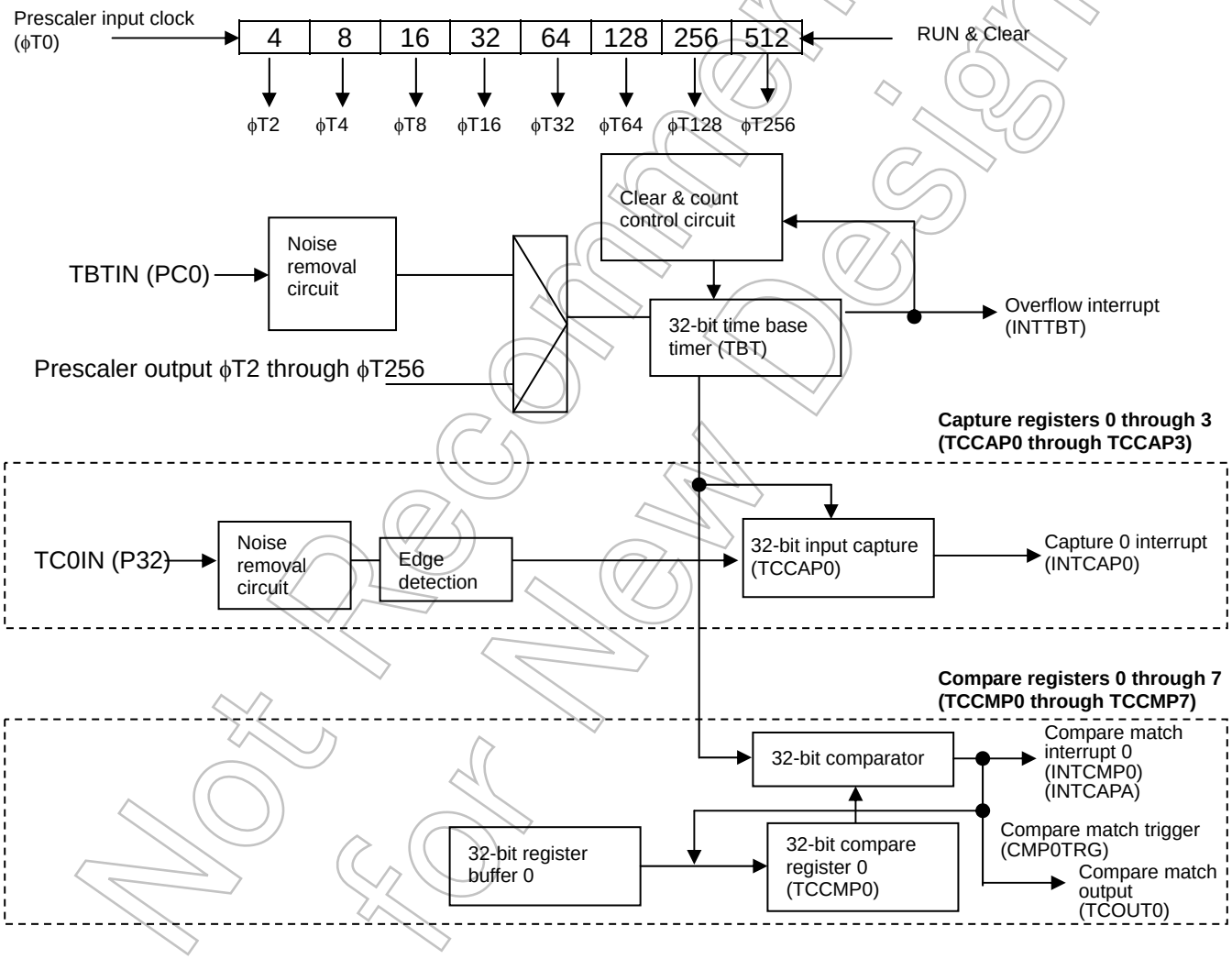


Fig. 12-1 Timer C Block Diagram

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 or fperiph/16 selected by SYSCR0<PRCK1:0> in the CG. ϕT2 through ϕT256 generated by dividing ϕT0 are available as TMRC prescaler input clocks and can be selected with TBTCR<TBTCLK3:0>.

Fperiph is either "fgear" which is a clock selected by SYSCR1<FPSEL> in the CG, or "fc" which is a clock before it is divided by the clock gear.

The operation or stoppage of the prescaler is set with TBTRUN<TBTPRUN> where writing "1" starts counting and writing "0" clears and stops counting. Table 12-1 shows the prescaler output clock resolutions.

Table 12-1 Prescaler Output Clock Resolutions

@fc = 40.0MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolution			
			ϕT2	ϕT4	ϕT8	ϕT16
0(fgear)	000(fc)	00(fperiph/16)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)
		10(fperiph/4)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)
		11(fperiph/2)	fc/2^3(0.20μs)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)
	100(fc/2)	00(fperiph/16)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)	fc/2^{10}(25.3μs)
		01(fperiph/8)	fc/2^6(1.60s)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.6μs)
		10(fperiph/4)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.32μs)
		11(fperiph/2)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)
	110(fc/4)	00(fperiph/16)	fc/2^8(6.40μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
		01(fperiph/8)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)
		10(fperiph/4)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
		11(fperiph/2)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)
	111(fc/8)	00(fperiph/16)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
		01(fperiph/8)	fc/2^8(6.40μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
		10(fperiph/4)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)	fc/2^{10}(25.6μs)
		11(fperiph/2)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
1(fc)	000(fc)	00(fperiph/16)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)
		10(fperiph/4)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)
		11(fperiph/2)	fc/2^3(0.20μs)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)
	100 (fc/2)	00(fperiph/16)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)
		10(fperiph/4)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)
		11(fperiph/2)	fc/2^3(0.20μs)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)
	110(fc/4)	00(fperiph/16)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)
		10(fperiph/4)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)
		11(fperiph/2)	fc/2^3(0.20μs)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)
	111(fc/8)	00(fperiph/16)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)	fc/2^9(12.8μs)
		01(fperiph/8)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)	fc/2^8(6.40μs)
		10(fperiph/4)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)	fc/2^7(3.20μs)
		11(fperiph/2)	fc/2^3(0.20μs)	fc/2^4(0.40μs)	fc/2^5(0.80μs)	fc/2^6(1.60μs)

@fc = 40MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR1:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolution			
			φT32	φT64	φT128	φT256
0(fgear)	000(fc)	00(fperiph/16)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
		01(fperiph/8)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
		10(fperiph/4)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
		11(fperiph/2)	fc/2^{7}(3.20μs)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)
	100(fc/2)	00(fperiph/16)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)	fc/2^{14}(409.6μs)
		01(fperiph/8)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
		10(fperiph/4)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
		11(fperiph/2)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
	110(fc/4)	00(fperiph/16)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)	fc/2^{14}(409.6μs)	fc/2^{15}(819.2μs)
		01(fperiph/8)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)	fc/2^{14}(409.6μs)
		10(fperiph/4)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
		11(fperiph/2)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
	111(fc/8)	00(fperiph/16)	fc/2^{13}(204.8μs)	fc/2^{14}(409.6μs)	fc/2^{15}(819.2μs)	fc/2^{16}(1638.4μs)
		01(fperiph/8)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)	fc/2^{14}(409.6μs)	fc/2^{15}(819.2μs)
		10(fperiph/4)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)	fc/2^{14}(409.6s)
		11(fperiph/2)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
1(fc)	000(fc)	00(fperiph/16)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
		01(fperiph/8)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
		10(fperiph/4)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
		11(fperiph/2)	fc/2^{7}(3.20μs)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)
	100(fc/2)	00(fperiph/16)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
		01(fperiph/8)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
		10(fperiph/4)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
		11(fperiph/2)	fc/2^{7}(3.20μs)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)
	110(fc/4)	00(fperiph/16)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)	fc/2^{13}(204.8μs)
		01(fperiph/8)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{12}(102.4μs)
		10(fperiph/4)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)
		11(fperiph/2)	fc/2^{7}(3.20μs)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)
	111 (fc/8)	00(fperiph/16)	fc/2^{10}(25.6μs)	fc/2^{11}(51.2μs)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)
		01(fperiph/8)	fc/2^{9}(12.8μs)	fc/2^{10}(25.6μs)	fc/2^{7}(3.20μs)	fc/2^{8}(6.40μs)
		10(fperiph/4)	fc/2^{8}(6.40μs)	fc/2^{9}(12.8μs)	fc/2^{6}(1.60μs)	fc/2^{7}(3.20μs)
		11(fperiph/2)	fc/2^{7}(3.20μs)	fc/2^{8}(6.40μs)	fc/2^{5}(0.80μs)	fc/2^{6}(1.60μs)

(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. When it is performing a read capture operation, it is possible to read a counter value by accessing the TBT read capture register (TBTRDCAP) in units of 32 bits.

However, a counter value cannot be read (captured) if the register is accessed in units of 8 or 16 bits.

12.2.4 Edge Detection Circuit

By performing sampling, this circuit detects the input edge of an external capture input (TcnIN). It can be set to "rising edge," "falling edge," "both edges" or "not capture" by provisioning the capture control register CAPnCR<CPnEG1:0>. Fig. 13.2 shows capture inputs, outputs (capture factor outputs) produced by the edge detection circuit, and specific detection circuit settings.

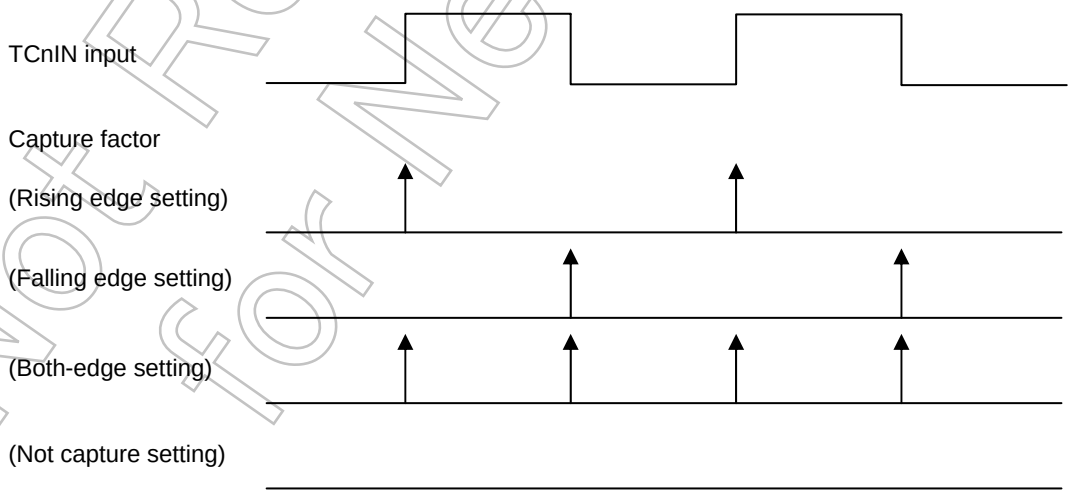


Fig. 12-2 Capture Inputs and Capture Factor Outputs (Outputs Produced by the Edge Detection Circuit)

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 grouped into one set of interrupt requests which are then notified to the interrupt controller. Which one of interrupt requests INTCAP0 through INTCAP3 must be processed can be identified by reading the status register TCG0ST during interrupt processing. Additionally, it is possible to mask unnecessary interrupts by setting the interrupt mask register TCG0IM to an appropriate bit setting. While a read of the capture register is ongoing, count values cannot be captured even if there are triggers.

Data is read in the order of **lower to higher bits** by using a word transfer instruction. If a half-word transfer instruction is used, data is read twice. If a byte data transfer instruction is used, data is read four times.

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<CMPEN1:0>.

To set TCCMPn to a specific value, data must be transferred to TCCMPn in the order of **lower to higher bits** by using a word transfer instruction. If a half-word transfer instruction is used, data is transferred twice to TCCMPn. If a byte data transfer instruction is used, data is transferred four times to TCCMPn.

Each compare register has a double-buffer structure, that is, TCCMPn forms a pair with a register buffer "n." "Enable" or "disable" of the double buffers is controlled by the compare control register CMPCTL <CMPRDEn>. If <CMPRDEn> is set to "0," the double buffers are disabled. If <CMPRDEn> is set to "1," they are enabled.

If the double buffers are enabled, data transfer from the register buffer "n" to the compare register TCCMPn takes place when the value of TBT matches that of TCCMPn.

Because TCCMPn is indeterminate when a reset is performed, it is necessary to prepare and write data in advance. A reset initializes CMPCTL <CMPRDEn> to "0" and disables the double buffers. To use the double buffers, data must be written to the compare register, <CMPRDEn> must be set to "1," and then the following data must be written to the register buffer.

TCCMPn and the register buffer are assigned to the same address. If <CMPRDEn> is "0," the same value is written to TCCMPn and each register buffer. If <CMPRDEn> is "1," data is written to each register buffer only. Therefore, to write an initial value to the compare register, it is necessary to set the double buffers to "disable."

12.3 Register Description

TMRC Control Register

TCCR (0xFFFF_F400)

	7	6	5	4	3	2	1	0
bit Symbol	TCEN	I2TBT						
Read/Write	R/W		R					
After reset	0	0	0					
Function	TMRC operation 0: Disable 1: Enable	IDLE 0: Stop 1: Run	"0" is read.					

<I2TBT>: Controls the operation in IDLE mode

<TCEN>: Specifies enabling/disabling of the TMRC operation. If set to "disable," a clock is not supplied to other registers of the TMRC module and, therefore, a reduction in power consumption is possible (a read of or a write to other registers cannot be executed). To use TMRC, the TMRC operation must be set to "enable" ("1") before making individual register settings of TMRC modules. If TMRC is operated and then set to "disable," individual register settings are retained.

TBTRUN Register

TBTRUN (0xFFFF_F401)

	7	6	5	4	3	2	1	0
bit Symbol						TBTCAP	TBTPRUN	TBTRUN
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	"0" is read.				Ensure this is set to "0."	TBT counter software capture 0: Don't Care 1: Software capture	TimerRun/Stop Control 0: Stop & clear 1: Count	

<TBTRUN>: Controls the TBT count operation

<TBTPRUN>: Controls the TBT prescaler operation

<TBTCAP>: If this is set to "1," the count value of the time base timer (TBT) is taken into the capture register TBTCAPn.

Fig. 12-3 TMRC-related Registers

TBT Control Register

TBTCR (0xFFFF_F402)

	7	6	5	4	3	2	1	0
bit Symbol	TBTNF				TBTCLK3	TBTCLK2	TBTCLK1	TBTCLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	TBTIN Input noise removal 0:2/fsys or more 1:6/fsys or more	Ensure this is set to "0."			TBT source clock 0000: φT2 0001: φT4 0010: φT8 0011: φT16 0100: φT32 0101: φT64 0110: φT128 0111: φT256 1111: TBTIN pin input			

<TBTCLK3:0>: This is an input clock for TBT. Clocks from "0000" to "0111" are available as prescaler output clocks. A clock "1111" is input through the TBTIN pin.

<TBTNF>: Controls the noise removal for the TBTIN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (50ns@fperiph=fc=40MHz) 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 (150ns@fperiph=fc=40MHz) is regarded as noise and removed, at whichever level the TBTIN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

TBT Capture Register (TBTCAP)

TBTCAP0 (0xFFFF_F404)

	7	6	5	4	3	2	1	0
bit Symbol	CAP07	CAP06	CAP05	CAP04	CAP03	CAP02	CAP01	CAP00
Read/Write	R							
After reset								
Function	Capture data							

TBTCAP1 (0xFFFF_F405)

	7	6	5	4	3	2	1	0
bit Symbol	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10	CAP09	CAP08
Read/Write	R							
After reset								
Function	Capture data							

TBTCAP2 (0xFFFF_F406)

	7	6	5	4	3	2	1	0
bit Symbol	CAP23	CAP22	CAP21	CAP20	CAP19	CAP18	CAP17	CAP16
Read/Write	R							
After reset								
Function	Capture data							

TBTCAP3 (0xFFFF_F407)

	7	6	5	4	3	2	1	0
bit Symbol	CAP31	CAP30	CAP29	CAP28	CAP27	CAP26	CAP25	CAP24
Read/Write	R							
After reset								
Function	Capture data							

Fig. 12-4 TMRC-related Registers

TBT Read Capture Register (TBTRDCAP)

		7	6	5	4	3	2	1	0
TBTRDCAPLL (0xFFFF_F408)	bit Symbol	RDCAP07	RDCAP06	RDCAP05	RDCAP04	RDCAP03	RDCAP02	RDCAP01	RDCAP00
	Read/Write	R							
	After reset								
	Function	Capture data							
		7	6	5	4	3	2	1	0
TBTRDCAPLH (0xFFFF_F409)	bit Symbol	RDCAP15	RDCAP14	RDCAP13	RDCAP12	RDCAP11	RDCAP10	RDCAP09	RDCAP08
	Read/Write	R							
	After reset								
	Function	Capture data							
		7	6	5	4	3	2	1	0
TBTRDCAPHL (0xFFFF_F40A)	bit Symbol	RDCAP23	RDCAP22	RDCAP21	RDCAP20	RDCAP19	RDCAP18	RDCAP17	RDCAP16
	Read/Write	R							
	After reset								
	Function	Capture data							
		7	6	5	4	3	2	1	0
TBTRDCAPHH (0xFFFF_F40B)	bit Symbol	RDCAP31	RDCAP30	RDCAP29	RDCAP28	RDCAP27	RDCAP26	RDCAP25	RDCAP24
	Read/Write	R							
	After reset								
	Function	Capture data							

Fig. 12-5 TMRC-related Registers

TMRC Capture 0 Control Register

CAP0CR (0xFFFF_F410)

	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 capture 01：Rising edge 10：Falling edge 11：Both edges	

<CP0EG1:0>: Selects the effective edge of an input to the trigger input pin TC0IN of the capture 0 register (TCCAP0). If this is set to "00," the capture operation is disabled.

<TC0NF>: Controls the noise removal for the TC0IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (50ns@fperiph=fc=40MHz) is accepted as a trigger input for TCCAP0, at whichever level the TC0IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (150ns@fperiph=fc=40MHz) is regarded as noise and removed, at whichever level the TC0IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

Fig. 12-6 TMRC-related Registers

TMRC Capture 0 Register (TCCAP0)

TCCAP0LL (0xFFFF_F414)

	7	6	5	4	3	2	1	0
bit Symbol	CAP007	CAP006	CAP005	CAP004	CAP003	CAP002	CAP001	CAP000
Read/Write	R							
After reset								
Function	Capture 0 data							

TCCAP0LH (0xFFFF_F415)

	7	6	5	4	3	2	1	0
bit Symbol	CAP015	CAP014	CAP013	CAP012	CAP011	CAP010	CAP009	CAP008
Read/Write	R							
After reset								
Function	Capture 0 data							

TCCAP0HL (0xFFFF_F416)

	7	6	5	4	3	2	1	0
bit Symbol	CAP023	CAP022	CAP021	CAP020	CAP019	CAP018	CAP017	CAP016
Read/Write	R							
After reset								
Function	Capture 0 data							

TCCAP0HH (0xFFFF_F417)

	7	6	5	4	3	2	1	0
bit Symbol	CAP031	CAP030	CAP029	CAP028	CAP027	CAP026	CAP025	CAP024
Read/Write	R							
After reset								
Function	Capture 0 data							

(Note 1) After a reset, the value of TCCAP0 is undefined.

(Note 2) Data is not captured during a read of the capture register.

TMRCG0 Interrupt Mask Register

TCG0IM (0xFFFF_F40C)

	7	6	5	4	3	2	1	0
bit Symbol					TCIM3	TCIM2	TCIM1	TCIM0
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	"0" is read.				Mask 1: INTCAP3	Mask 1: INTCAP2	Mask 1: INTCAP1	Mask 1: INTCAP0

TMRCG0 Status Register

TCG0ST (0xFFFF_F40D)

	7	6	5	4	3	2	1	0
bit Symbol					INTCAP3	INTCAP2	INTCAP1	INTCAP0
Read/Write	R							
After reset	0				0	0	0	0
Function	"0" is read.				0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

(Note 1) If TCG0ST is read, bits 0, 1, 2 and 3 are cleared.

Fig. 12-7 TMRC-related Registers

TMRC Capture 1 Control Register

CAP1CR (0xFFFF_F418)

	7	6	5	4	3	2	1	0
bit Symbol	TC1NF						CP1EG1	CP1EG0
Read/Write	R/W	R					R/W	
After reset	0	0					0	0
Function	TC1IN Input noise removal 0:2/fsys or more 1:6/fsys or more	"0" is read.					Select effective edge of TC1IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP1EG1:0>: Selects the effective edge of an input to the trigger input pin TC1IN of the capture 1 register (TCCAP1). If this is set to "00," the capture operation is disabled.

<TC1NF>: Controls the noise removal for the TC1NF pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (50ns@fperiph=fc=40MHz) 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 (150ns@fperiph=fc=40MHz) 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.

TMRC Capture 1 Register (TCCAP1)

		7	6	5	4	3	2	1	0
TCCAP1LL (0xFFFF_F41C)	bit Symbol	CAP107	CAP106	CAP105	CAP104	CAP103	CAP102	CAP101	CAP100
	Read/Write	R							
	After reset								
	Function	Capture 1 data							
		7	6	5	4	3	2	1	0
TCCAP1LH (0xFFFF_F41D)	bit Symbol	CAP115	CAP114	CAP113	CAP112	CAP111	CAP110	CAP109	CAP108
	Read/Write	R							
	After reset								
	Function	Capture 1 data							
		7	6	5	4	3	2	1	0
TCCAP1HL (0xFFFF_F41E)	bit Symbol	CAP123	CAP122	CAP121	CAP120	CAP119	CAP118	CAP117	CAP116
	Read/Write	R							
	After reset								
	Function	Capture 1 data							
		7	6	5	4	3	2	1	0
TCCAP1HH (0xFFFF_F41F)	bit Symbol	CAP131	CAP130	CAP129	CAP128	CAP127	CAP126	CAP125	CAP124
	Read/Write	R							
	After reset								
	Function	Capture 1 data							

(Note 1) After a reset, the value of TCCAP1 is undefined.

(Note 2) Data is not captured during a read of the capture register.

Fig. 12-8 TMRC-related Registers

TMRC Capture 2 Control Register

CAP2CR (0xFFFF_F420)

	7	6	5	4	3	2	1	0
bit Symbol	TC2NF						CP2EG1	CP2EG0
Read/Write	R/W	R					R/W	
After reset	0	0					0	0
Function	TC2IN Input noise removal 0: Disable 1: Enable	"0" is read.					Select effective edge of TC2IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP2EG1:0>: Selects the effective edge of an input to the trigger input pin TC2IN of the capture 2 register (TCCAP2). If this is set to "00," the capture operation is disabled.

<TC2NF>: Controls the noise removal for the TC2IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (50ns@fperiph=fc=40MHz) is accepted as a trigger input for TCCAP2, at whichever level the TC2IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (150ns@fperiph=fc=40MHz) is regarded as noise and removed, at whichever level the TC2IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

TMRC Capture 2 Register (TCCAP2)

TCCAP2LL (0xFFFF_F424)

	7	6	5	4	3	2	1	0
bit Symbol	CAP207	CAP206	CAP205	CAP204	CAP203	CAP202	CAP201	CAP200
Read/Write	R							
After reset								
Function	Capture 2 data							

TCCAP2LH (0xFFFF_F425)

	7	6	5	4	3	2	1	0
bit Symbol	CAP215	CAP214	CAP213	CAP212	CAP211	CAP210	CAP209	CAP208
Read/Write	R							
After reset								
Function	Capture 2 data							

TCCAP2HL (0xFFFF_F426)

	7	6	5	4	3	2	1	0
bit Symbol	CAP223	CAP222	CAP221	CAP220	CAP219	CAP218	CAP217	CAP216
Read/Write	R							
After reset								
Function	Capture 2 data							

TCCAP2HH (0xFFFF_F427)

	7	6	5	4	3	2	1	0
bit Symbol	CAP231	CAP230	CAP229	CAP228	CAP227	CAP226	CAP225	CAP224
Read/Write	R							
After reset								
Function	Capture 2 data							

(Note 1) After a reset, the value of TCCAP2 is undefined.

(Note 2) Data is not captured during a read of the capture register.

Fig. 12-9 TMRC-related Registers

TMRC Capture 3 Control Register

CAP3CR (0xFFFF_F428)

	7	6	5	4	3	2	1	0
bit Symbol	TC3NF						CP3EG1	CP3EG0
Read/Write	R/W	R					R/W	
After reset	0	0					0	0
Function	TC3IN Input noise removal 0: Disable 1: Enable	"0" is read.					Select effective edge of TC3IN input 00: Not capture 01: Rising edge 10: Falling edge 11: Both edges	

<CP3EG1:0>: Selects the effective edge of an input to the trigger input pin TC3IN of the capture 3 register (TCCAP3). If this is set to "00," the capture operation is disabled.

<TC3NF>: Controls the noise removal for the TC3IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (50ns@fperiph=fc=40MHz) is accepted as a trigger input for TCCAP3, at whichever level the TC3IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (150ns@fperiph=fc=40MHz) is regarded as noise and removed, at whichever level the TC3IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) Values read from bits 2 through 6 of CAP3CR are all "0."

TMRC Capture 3 Register (TCCAP3)

TCCAP3LL (0xFFFF_F42C)

	7	6	5	4	3	2	1	0
bit Symbol	CAP307	CAP306	CAP305	CAP304	CAP303	CAP302	CAP301	CAP300
Read/Write	R							
After reset								
Function	Capture 3 data							

TCCAP3LH (0xFFFF_F42D)

	7	6	5	4	3	2	1	0
bit Symbol	CAP315	CAP314	CAP313	CAP312	CAP311	CAP310	CAP309	CAP308
Read/Write	R							
After reset								
Function	Capture 3 data							

TCCAP3HL (0xFFFF_F42E)

	7	6	5	4	3	2	1	0
bit Symbol	CAP323	CAP322	CAP321	CAP320	CAP319	CAP318	CAP317	CAP316
Read/Write	R							
After reset								
Function	Capture 3 data							

TCCAP3HH (0xFFFF_F42F)

	7	6	5	4	3	2	1	0
bit Symbol	CAP331	CAP330	CAP329	CAP328	CAP327	CAP326	CAP325	CAP324
Read/Write	R							
After reset								
Function	Capture 3 data							

(Note 1) After a reset, the value of TCCAP3 is undefined.

(Note 2) Data is not captured during a read of the capture register.

Fig. 12-10 TMRC-related Registers

TMRC Compare Control Register (CMPCTL)

CMPCTL0 (0xFFFF_F470)

	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: Don't care		"0" is read.		Double buffers 0 0: Disable 1: Enable	Compare 0 enable 0: Disable 1: Enable

CMPCTL1 (0xFFFF_F471)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN1	TCFFC11	TCFFC10			CMPRDE1	CMPEN1
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF1 reversal 0: Disable 1: Enable	TCFF1 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 1 0: Disable 1: Enable	Compare 1 enable 0: Disable 1: Enable

CMPCTL2 (0xFFFF_F472)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN2	TCFFC21	TCFFC20			CMPRDE2	CMPEN2
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF2 reversal 0: Disable 1: Enable	TCFF2 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 2 0: Disable 1: Enable	Compare 2 enable 0: Disable 1: Enable

CMPCTL3 (0xFFFF_F473)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN3	TCFFC31	TCFFC30			CMPRDE3	CMPEN3
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF3 reversal 0: Disable 1: Enable	TCFF3 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 3 0: Disable 1: Enable	Compare 3 enable 0: Disable 1: Enable

Fig. 12-11 TMRC-related Registers

TMRC Compare Control Register (CMPCTL)

CMPCTL4 (0xFFFF_F474)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN4	TCFFC41	TCFFC40			CMPRDE4	CMPEN4
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF4 reversal 0: Disable 1: Enable	TCFF4 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 4 0: Disable 1: Enable	Compare 4 enable 0: Disable 1: Enable

CMPCTL5 (0xFFFF_F475)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN5	TCFFC51	TCFFC50			CMPRDE5	CMPEN5
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF5 reversal 0: Disable 1: Enable	TCFF5 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 5 0: Disable 1: Enable	Compare 5 enable 0: Disable 1: Enable

CMPCTL6 (0xFFFF_F476)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN6	TCFFC61	TCFFC60			CMPRDE6	CMPEN6
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF6 reversal 0: Disable 1: Enable	TCFF6 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 6 0: Disable 1: Enable	Compare 6 enable 0: Disable 1: Enable

CMPCTL7 (0xFFFF_F477)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN7	TCFFC71	TCFFC70			CMPRDE7	CMPEN7
Read/Write	R	R/W	R/W		R		R/W	
After reset	0	0	1	1	0		0	0
Function	"0" is read.	TCFF7 reversal 0: Disable 1: Enable	TCFF7 control 00: Reversal 01: Set 10: Clear 11: Don't care		"0" is read.		Double buffers 7 0: Disable 1: Enable	Compare 7 enable 0: Disable 1: Enable

<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. 12-12 TMRC-related Registers

TMRC Compare Register 0 (TCCMP0)

		7	6	5	4	3	2	1	0
TCCMP0LL (0xFFFF_F440)	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							
		7	6	5	4	3	2	1	0
TCCMP0LH (0xFFFF_F441)	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							
		7	6	5	4	3	2	1	0
TCCMP0HL (0xFFFF_F442)	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							
		7	6	5	4	3	2	1	0
TCCMP0HH (0xFFFF_F443)	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							

TMRC Compare Register 1 (TCCMP1)

		7	6	5	4	3	2	1	0
TCCMP1LL (0xFFFF_F444)	bit Symbol	CMP107	CMP106	CMP105	CMP104	CMP103	CMP102	CMP101	CMP100
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 1 data							
		7	6	5	4	3	2	1	0
TCCMP1LH (0xFFFF_F445)	bit Symbol	CMP115	CMP114	CMP113	CMP112	CMP111	CMP110	CMP109	CMP108
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 1 data							
		7	6	5	4	3	2	1	0
TCCMP1HL (0xFFFF_F446)	bit Symbol	CMP123	CMP122	CMP121	CMP120	CMP119	CMP118	CMP117	CMP116
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 1 data							
		7	6	5	4	3	2	1	0
TCCMP1HH (0xFFFF_F447)	bit Symbol	CMP131	CMP130	CMP129	CMP128	CMP127	CMP126	CMP125	CMP124
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 1 data							

Fig. 12-13 TMRC-related Registers

TMRC Compare Register 2 (TCCMP2)

		7	6	5	4	3	2	1	0
TCCMP2LL (0xFFFF_F448)	bit Symbol	CMP207	CMP206	CMP205	CMP204	CMP203	CMP202	CMP201	CMP200
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 2 data							
		7	6	5	4	3	2	1	0
TCCMP2LH (0xFFFF_F449)	bit Symbol	CMP215	CMP214	CMP213	CMP212	CMP211	CMP210	CMP209	CMP208
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 2 data							
		7	6	5	4	3	2	1	0
TCCMP2HL (0xFFFF_F44A)	bit Symbol	CMP223	CMP222	CMP221	CMP220	CMP219	CMP218	CMP217	CMP216
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 2 data							
		7	6	5	4	3	2	1	0
TCCMP2HH (0xFFFF_F44B)	bit Symbol	CMP231	CMP230	CMP229	CMP228	CMP227	CMP226	CMP225	CMP224
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 2 data							

TMRC Compare Register 3 (TCCMP3)

		7	6	5	4	3	2	1	0
TCCMP3LL (FFFF_F44C)	bit Symbol	CMP307	CMP306	CMP305	CMP304	CMP303	CMP302	CMP301	CMP300
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 3 data							
		7	6	5	4	3	2	1	0
TCCMP3LH (0xFFFF_F44D)	bit Symbol	CMP315	CMP314	CMP313	CMP312	CMP311	CMP310	CMP309	CMP308
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 3 data							
		7	6	5	4	3	2	1	0
TCCMP3HL (0xFFFF_F44E)	bit Symbol	CMP323	CMP322	CMP321	CMP320	CMP319	CMP318	CMP317	CMP316
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 3 data							
		7	6	5	4	3	2	1	0
TCCMP3HH (0xFFFF_F44F)	bit Symbol	CMP331	CMP330	CMP329	CMP328	CMP327	CMP326	CMP325	CMP324
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 3 data							

Fig. 12-14 TMRC-related Registers

TMRC Compare Register 4 (TCCMP4)

		7	6	5	4	3	2	1	0
TCCMP4LL (0xFFFF_F450)	bit Symbol	CMP407	CMP406	CMP405	CMP404	CMP403	CMP402	CMP401	CMP400
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 4 data							
		7	6	5	4	3	2	1	0
TCCMP4LH (0xFFFF_F451)	bit Symbol	CMP415	CMP414	CMP413	CMP412	CMP411	CMP410	CMP409	CMP408
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 4 data							
		7	6	5	4	3	2	1	0
TCCMP4HL (0xFFFF_F452)	bit Symbol	CMP423	CMP422	CMP421	CMP420	CMP419	CMP418	CMP417	CMP416
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 4 data							
		7	6	5	4	3	2	1	0
TCCMP4HH (0xFFFF_F453)	bit Symbol	CMP431	CMP430	CMP429	CMP428	CMP427	CMP426	CMP425	CMP424
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 4 data							

TMRC Compare Register 5 (TCCMP5)

		7	6	5	4	3	2	1	0
TCCMP5LL (0xFFFF_F454)	bit Symbol	CMP507	CMP506	CMP505	CMP504	CMP503	CMP502	CMP501	CMP500
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 5 data							
		7	6	5	4	3	2	1	0
TCCMP5LH (0xFFFF_F455)	bit Symbol	CMP515	CMP514	CMP513	CMP512	CMP511	CMP510	CMP509	CMP508
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 5 data							
		7	6	5	4	3	2	1	0
TCCMP5HL (0xFFFF_F456)	bit Symbol	CMP523	CMP522	CMP521	CMP520	CMP519	CMP518	CMP517	CMP516
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 5 data							
		7	6	5	4	3	2	1	0
TCCMP5HH (0xFFFF_F457)	bit Symbol	CMP531	CMP530	CMP529	CMP528	CMP527	CMP526	CMP525	CMP524
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 5 data							

Fig. 12-15 TMRC-related Registers

TMRC Compare Register 6 (TCCMP6)

		7	6	5	4	3	2	1	0
TCCMP6LL (0xFFFF_F458)	bit Symbol	CMP607	CMP606	CMP605	CMP604	CMP603	CMP602	CMP601	CMP600
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 6 data							
		7	6	5	4	3	2	1	0
TCCMP6LH (0xFFFF_F459)	bit Symbol	CMP615	CMP614	CMP613	CMP612	CMP611	CMP610	CMP609	CMP608
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 6 data							
		7	6	5	4	3	2	1	0
TCCMP6HL (0xFFFF_F45A)	bit Symbol	CMP623	CMP622	CMP621	CMP620	CMP619	CMP618	CMP617	CMP616
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 6 data							
		7	6	5	4	3	2	1	0
TCCMP6HH (0xFFFF_F45B)	bit Symbol	CMP631	CMP630	CMP629	CMP628	CMP627	CMP626	CMP625	CMP624
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 6 data							

TMRC Compare Register 7 (TCCMP7)

		7	6	5	4	3	2	1	0
TCCMP7LL (0xFFFF_F45C)	bit Symbol	CMP707	CMP706	CMP705	CMP704	CMP703	CMP702	CMP701	CMP700
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 7 data							
		7	6	5	4	3	2	1	0
TCCMP7LH (0xFFFF_F45D)	bit Symbol	CMP715	CMP714	CMP713	CMP712	CMP711	CMP710	CMP709	CMP708
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 7 data							
		7	6	5	4	3	2	1	0
TCCMP7HL (0xFFFF_F45E)	bit Symbol	CMP723	CMP722	CMP721	CMP720	CMP719	CMP718	CMP717	CMP716
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 7 data							
		7	6	5	4	3	2	1	0
TCCMP7HH (0xFFFF_F45F)	bit Symbol	CMP731	CMP730	CMP729	CMP728	CMP727	CMP726	CMP725	CMP724
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Compare register 7 data							

Fig. 12-16 TMRC-related Registers

13. Serial Channel (SIO)

13.1 Features

This device has three serial I/O channels: SIO0 to SIO2. Each channel operates in either the UART mode (asynchronous communication) or the I/O interface mode (synchronous communication) which is selected by the user.

I/O interface mode
- Mode 0: This is the mode to send and receive I/O data and associated synchronization signals (SCLK) to extend I/O.

Asynchronous (UART) mode:
- Mode 1: TX/RX Data Length: 7 bits
- Mode 2: TX/RX Data Length: 8 bits
- Mode 3: TX/RX Data Length: 9 bits

In the above modes 1 and 2, parity bits can be added. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). Fig. 13-2 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.

As the SIOs 0 to 2 operate in the same way, Only SIO0 is described here.

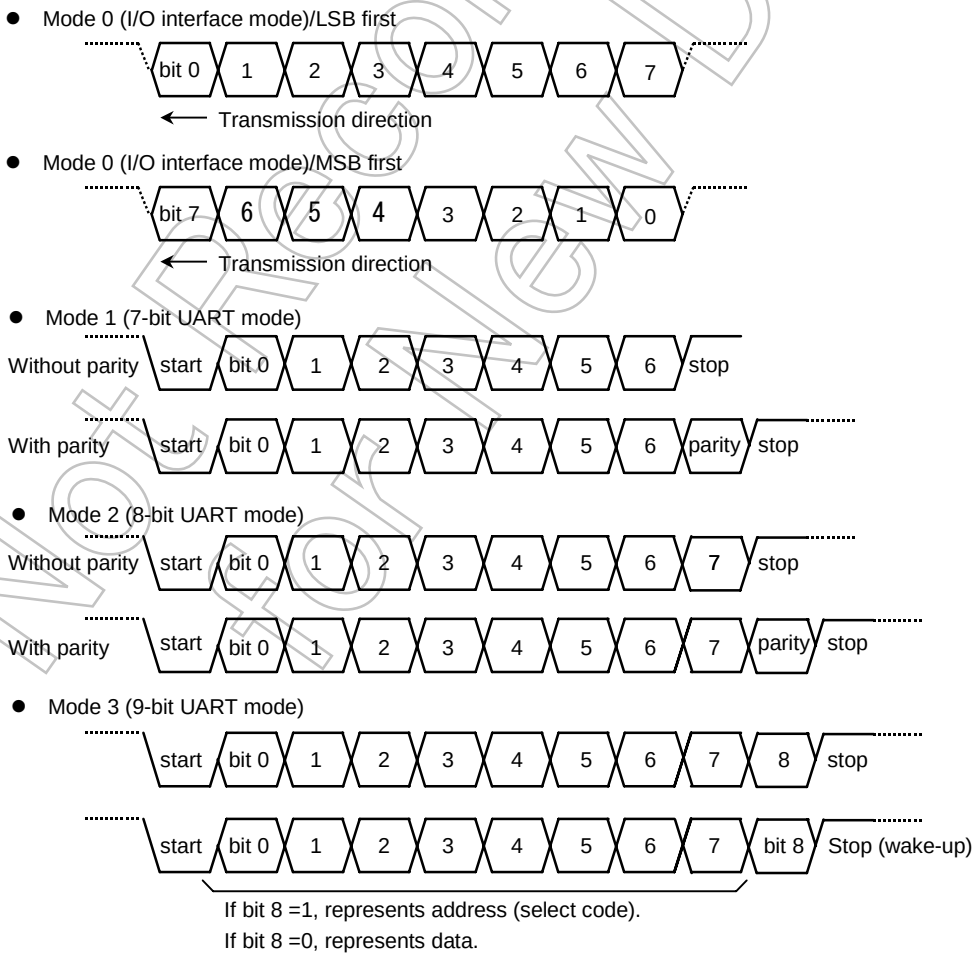


Fig. 13-1 Data Format

13.2 Block Diagram (Channel 0)

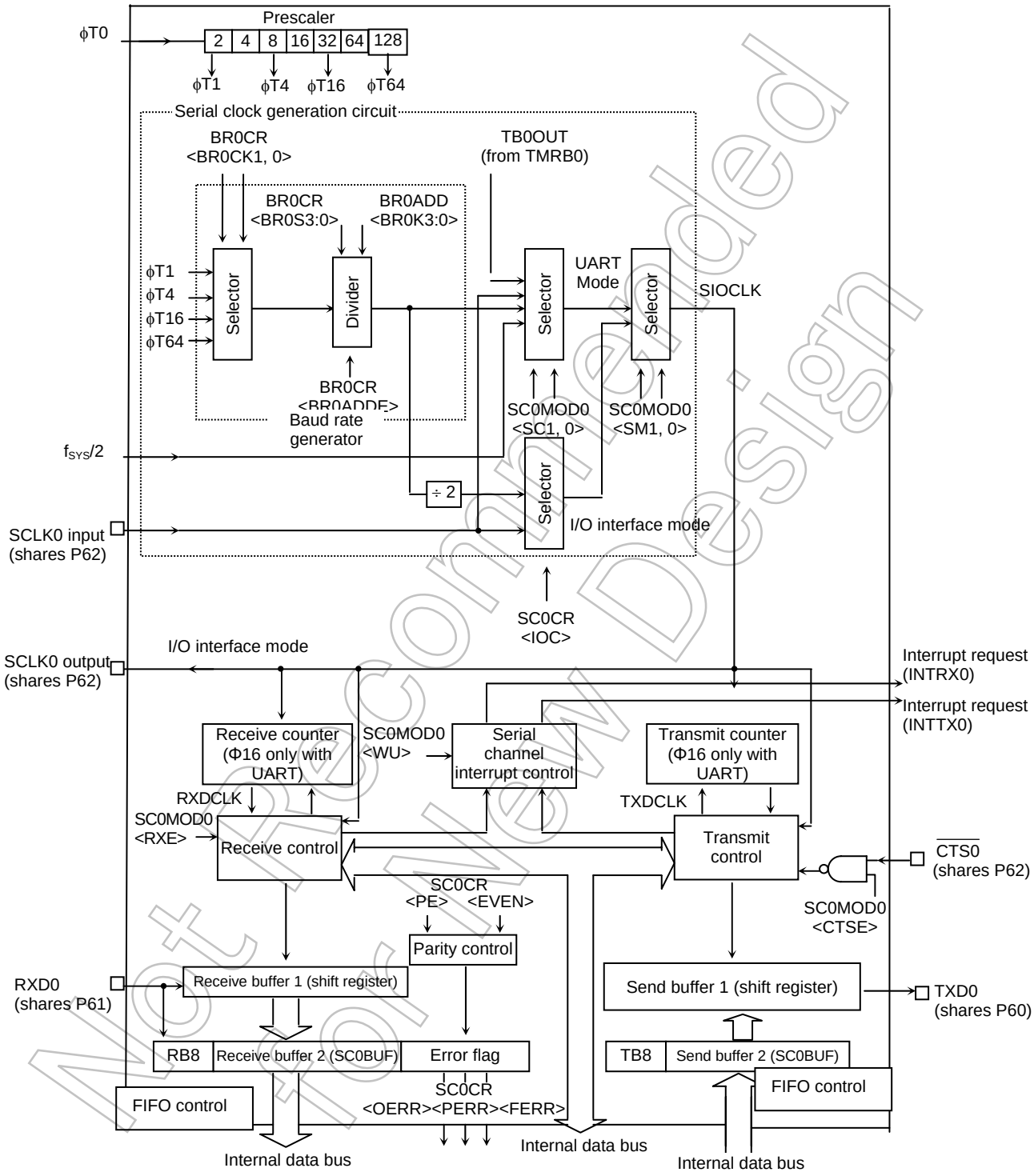


Fig. 13-2 SIO0 Block Diagram

13.3 Operation of Each Circuit (Channel 0)

13.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock ϕT0 to the prescaler is selected by SYSCR of CG <PRCK1:0> to provide the frequency of either fperiph/2, fperiph/4, fperiph/8, or fperiph/16.

The clock frequency fperiph is either the clock "fgear," to be selected by SYSCR1<FPSEL> of CG, or the clock "fc" before it is divided by the clock gear.

The prescaler becomes active only when the baud rate generator is selected for generating the serial transfer clock. Table 13-1 lists the prescaler output clock resolution.

Table 13-1 Clock Resolution to the Baud Rate Generator @ = 40MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR1:0>	Prescaler clock selection <PRCK1:0>	Prescaler output clock resolution			
			ϕT1	ϕT4	ϕT16	ϕT64
0 (fgear)	000 (fc)	00(fperiph/16)	fc/2^5(0.8 μs)	fc/2^7(3.2 μs)	fc/2^9(12.8 μs)	fc/2^{11}(51.2μs)
		01(fperiph/8)	fc/2^4(0.4 us)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
		10(fperiph/4)	fc/2^3(0.2us)	fc/2^5(0.8us)	fc/2^7(3.2us)	fc/2^9(12.8μs)
		11(fperiph/2)	fc/2^2(0.1us)	fc/2^4(0.4us)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)
	100(fc/2)	00(fperiph/16)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)	fc/2^{12} (102 μs)
		01(fperiph/8)	fc/2^5(0.8us)	fc/2^7(3.2us)	fc/2^9 (12.8μs)	fc/2^{11}(51.2μs)
		10(fperiph/4)	fc/2^4(0.4us)	fc/2^6(1.6 ms)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
		11(fperiph/2)	fc/2^3(0.2 us)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)
	110(fc/4)	00(fperiph/16)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)	fc/2^{13}(204μs)
		01(fperiph/8)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)	fc/2^{12} (102μs)
		10(fperiph/4)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		11(fperiph/2)	fc/2^4(0.4 us)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
	111(fc/8)	00(fperiph/16)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)	fc/2^{12} (102 μs)	fc/2^{14}(410us)
		01(fperiph/8)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)	fc/2^{13}(204μs)
		10(fperiph/4)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)	fc/2^{12} (102μs)
		11(fperiph/2)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
	000 (fc)	00(fperiph/16)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		01(fperiph/8)	fc/2^4(0.4 us)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
		10(fperiph/4)	fc/2^3(0.2 us)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)
		11(fperiph/2)	fc/2^2(0.1 us)	fc/2^4(0.4 us)	fc/2^6(1.6 us)	fc/2^8(6.4 μs)
1 (fc)	100(fc/2)	00(fperiph/16)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		01(fperiph/8)	fc/2^4(0.4 us)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
		10(fperiph/4)	fc/2^3(0.2 us)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)
		11(fperiph/2)	—	fc/2^4(0.4 us)	fc/2^6(1.6 us)	fc/2^8(6.4 μs)
	110(fc/4)	00(fperiph/16)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		01(fperiph/8)	—	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
		10(fperiph/4)	—	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)
		11(fperiph/2)	—	—	fc/2^6(1.6 us)	fc/2^8(6.4 μs)
	111(fc/8)	00(fperiph/16)	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)	fc/2^{11}(51.2μs)
		01(fperiph/8)	fc/2^4(0.4 us)	fc/2^6(1.6 μs)	fc/2^8(6.4 μs)	fc/2^{10}(25.6 μs)
		10(fperiph/4)	—	fc/2^5(0.8 us)	fc/2^7(3.2 us)	fc/2^9(12.8μs)
		11(fperiph/2)	—	—	fc/2^6(1.6 us)	fc/2^8(6.4 μs)

(Note 1) The prescaler output clock ϕTn must be selected so that the relationship "ϕTn < fsys/2" is satisfied (so that ϕTn is slower than fsys/2).
(Note 2) Do not change the clock gear while SIO is operating.
(Note 3) The horizontal lines in the above table indicate that the setting is prohibited.

The serial interface baud rate generator uses four different clocks, i.e., ϕT1, ϕT4, ϕT16 and ϕT64, supplied from the prescaler output clock.

13.3.2 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses either the ϕT1, ϕT4, ϕT16 or ϕT64 clock supplied from the 7-bit prescaler. This input clock selection is made by setting the baud rate setting register, BR0CR <BR0CK1:0>.

The baud rate generator contains built-in dividers for divide by 1, (N + m/16), and 16 where N is a

number from 2 to 15 and m is a number from 0 to 15. The division is performed according to the settings of the baud rate control registers BR0CR <BR0ADDE> <BR0S3:0> and BR0ADD <BR0K3:0> to determine the resulting transfer rate.

- UART Mode:

  1) If BR0CR <BR0ADDE> = 0,

     The setting of BR0ADD <BR0K3:0> is ignored and the counter is divided by N where N is the value set to BR0CR <BR0S3:0>. (N = 1 to 16).

  2) If BR0CR <BR0ADDE> = 1,

     The N + (16 - K)/16 division function is enabled and the division is made by using the values N (set in BR0CR <BR0S3:0>) and K (set in BR0ADD<BR0K3:0>). (N = 2 to 15, K = 1 to 15)

Note	**For the N values of 1 and 16, the above N+(16-K)/16 division function is inhibited. So, be sure to set BR0CR<BR0ADDE> to "0."**

- I/O interface mode:

  The N + (16 - K)/16 division function cannot be used in the I/O interface mode. Be sure to divide by N, by setting BR0CR <BR0ADDE> to "0."

- Baud rate calculation to use the baud rate generator:

  1) UART mode

     $$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} /16$$

     The highest baud rate out of the baud rate generator is 625 kbps when ϕT1 is 10 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 1.25 Mbps when fsys is 40 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 10 MHz. If double buffering is used, the divide ratio can be set to "1" and the resulting output baud rate will be 5 Mbps. (If double buffering is not used, the highest baud rate will be 2.5 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/16, using the baud rate generator input clock ϕT1, setting the divide ratio N (BR0CR<BR0S3:0>) = 4, and setting BR0CR<BR0ADDE> = "0," the resulting baud rate in the UART mode is calculated as follows:

* Clocking conditions
  - System clock : High-speed (fc)
  - High speed clock gear : x 1 (fc)
  - Prescaler clock : $f_{periph}/16$ ($f_{periph} = f_{sys}$)

$$\text{Baud rate} = \frac{fc/32}{4} /16$$

$$= 39.321 \text{ (bps)} \times 10^6 / 32 / 4 / 16 \approx 19200 \text{ (bps)}$$

(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/16, 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 (fc)
  - High-speed clock gear : x 1 (fc)
  - Prescaler clock : $f_{periph}/4$ ($f_{periph} = f_{sys}$)

$$\text{Baud rate} = \frac{fc/32}{7 + \frac{(16-3)}{16}} /16$$

$$= 19.2 \times 10^6 / 64 / (7 + \frac{13}{16}) / 16 = 4800 \text{ (bps)}$$

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 = 40 MHz, the highest baud rate will be 40 / 4 / 16 = 625 (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 = 40 MHz, the baud rate must be set to a rate lower than 40 / 12 = 3.3 (Mbps).

When double buffering is not used, it is necessary to satisfy the following relationship:

External clock input period > 16/fsys

Therefore, when fsys = 40 MHz, the baud rate must be set to a rate lower than 40 / 16 = 2.5 (Mbps).

Example baud rates for the UART mode are shown in Table 13-2 and Table 13-3.

Table 13-2 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 13-3 Selection of UART Baud Rate

(The TMRB0 timer output (internal TB0OUT) is used with the timer input clock set to ϕT0.)

Unit (kbps)

TB0REG \ fc	29.4912 MHz	24.576 MHz	24 MHz	19.6608 MHz	16 MHz	12.288 MHz
1H	230.4	192	187.5	153.6	125	96
2H	115.2	96	93.75	76.8	62.5	48
3H	76.8	64	62.5	51.2	41.67	32
4H	57.6	48	46.88	38.4	31.25	24
5H	46.08	38.4	37.5	30.72	25	19.2
6H	38.4	32	31.25	25.6	20.83	16
8H	28.8	24	23.44	19.2	15.63	12
AH	23.04	19.2	18.75	15.36	12.5	9.6
10H	14.4	12	11.72	9.6	7.81	6
14H	11.52	9.6	9.38	7.68	6.25	4.8

Baud rate calculation to use the TMRB0 timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by SYSCR0} < \text{PRCK1:0} >}{\text{TB0REG} \times \underline{2} \times 16}$$

↑ (When input clock to the timer TMRB0 is ϕT0)

(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.

13.3.3 Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

- I/O interface mode:

  In the SCLK output mode with the SC0CR <IOC> serial control register set to "0," the output of the previously mentioned baud rate generator is divided by 2 to generate the basic clock.

  In the SCLK input mode with SC0CR <IOC> set to "1," rising and falling edges are detected according to the SC0CR <SCLKS> setting to generate the basic clock.

- Asynchronous (UART) mode:

  According to the settings of the serial control mode register SC0MOD0 <SC1:0>, either the clock from the baud rate register, the system clock (f_{SYS}/2), the internal output signal of the TMRB0 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

13.3.4 Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by SIOCLK. Sixteen SIOCLK clock pulses are used in receiving a single data bit while the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

13.3.5 Receive Control Unit

- I/O interface mode:

  In the SCLK output mode with SC0CR <IOC> set to "0," the RXD0 pin is sampled on the rising edge of the shift clock output to the SCLK0 pin.

  In the SCLK input mode with SC0CR <IOC> set to "1," the serial receive data RXD0 pin is sampled on the rising or falling edge of SCLK input depending on the SC0CR <SCLKS> setting.

- Asynchronous (UART) mode:

  The receive control unit has a start bit detection circuit, which is used to initiate receive operation when a normal start bit is detected.

13.3.6 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), 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."

13.3.7 Receive FIFO Buffer

In addition to the double buffer function already described, data may be stored using the receive FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte receive buffer can be enabled. Also, in the UART mode or I/O interface mode, data may be stored up to a predefined fill level. When the receive FIFO buffer is to be used, be sure to enable the double buffer function.

If data with parity bit is to be received in the UART mode, parity check must be performed each time a data frame is received.

13.3.8 Receive FIFO Operation

① I/O interface mode with SCLK output:

The following example describes the case a 4-byte data stream is received in the half duplex mode:

SC0RFC<7:6>=01: Clears receive FIFO and sets the condition of interrupt generation.

SC0RFC<1:0>=00: Sets the interrupt to be generated at fill level 4.

SC0FCNF <1:0>=10111: Automatically inhibits continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

In this condition, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. After receiving 4 bytes, the RXE bit is automatically cleared and the receive operation is stopped (SCLK is stopped).

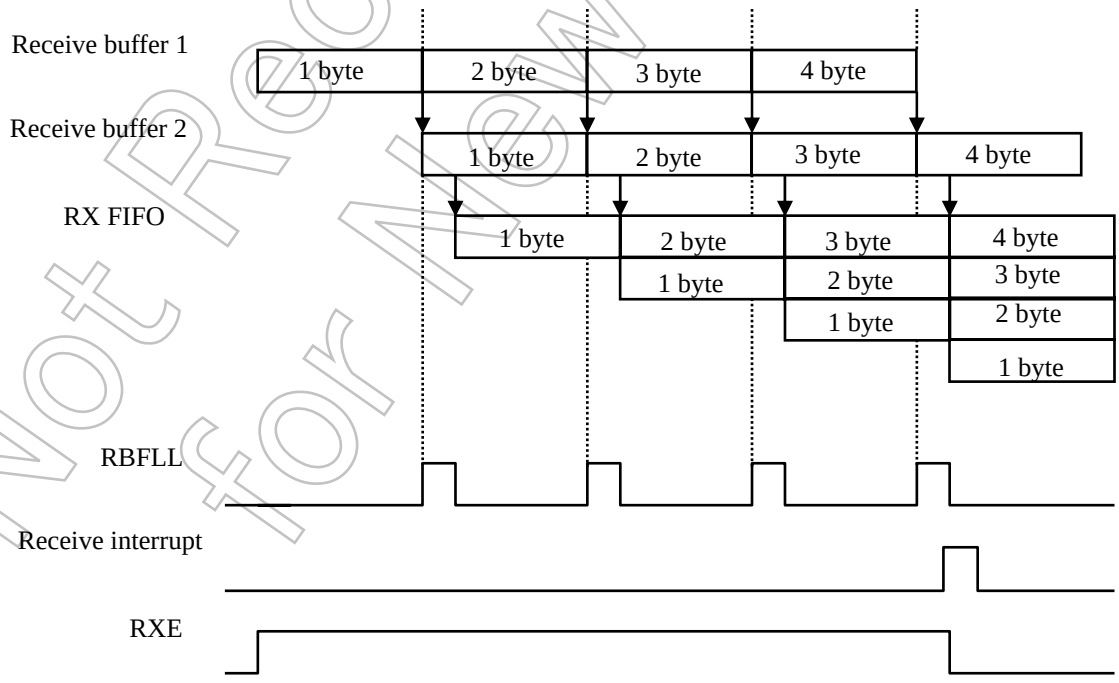


Fig. 13-3 Receive FIFO Operation

② I/O interface mode with SCLK input:

The following example describes the case a 4-byte data stream is received:

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

SC0RFC <1:0> = 00: Sets the interrupt to be generated at fill level 4.

SC0FCNF <1:0> = 10101: Automatically allows continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the maximum allowable number.

In this condition, 4-byte data reception can be initiated along with the input clock by setting the half duplex transmission mode and writing "1" to the RXE bit. When the 4-byte data reception is completed, the receive FIFO interrupt will be generated.

Note that preparation for the next data reception can be managed in this setting, i.e., the next 4-byte data can be received before data is fully read from the FIFO.

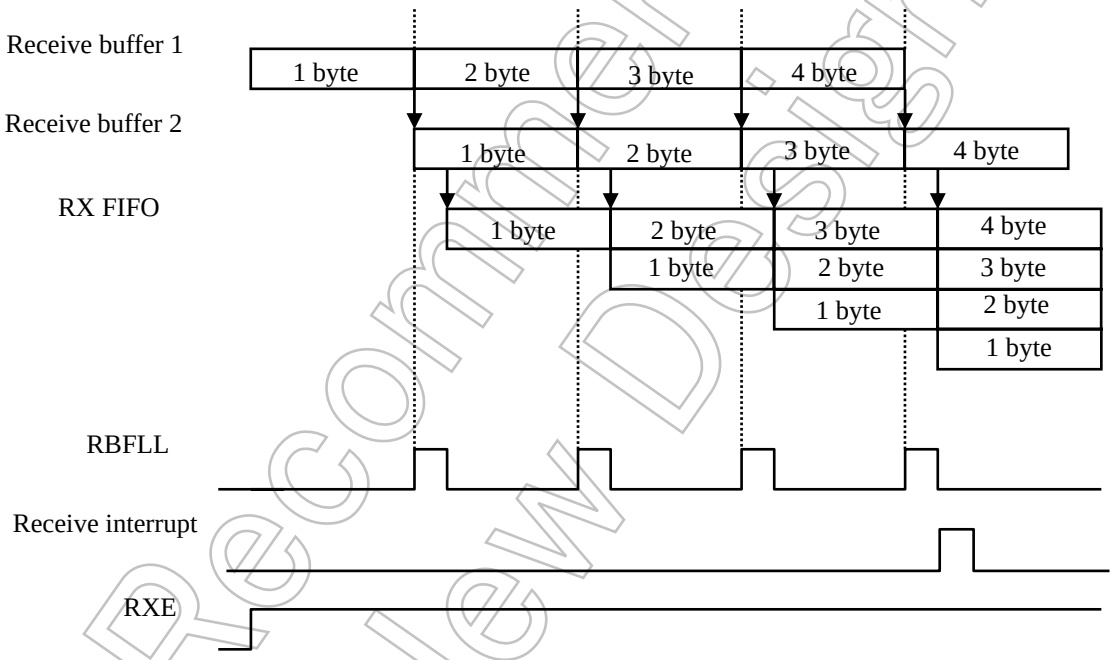


Fig. 13-4 Receive FIFO Operation

13.3.9 Transmit Counter

The transmit counter is a 4-bit binary counter used in the asynchronous communication (UART) mode. It is counted by SIOCLK as in the case of the receive counter and generates a transmit clock (TXDCLK) on every 16th clock pulse.

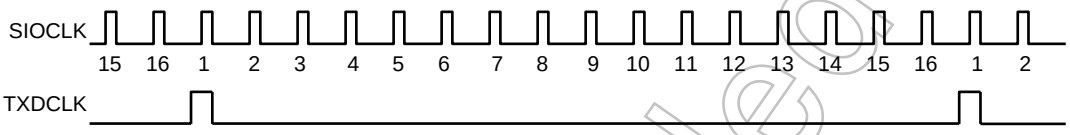


Fig. 13-5 Transmit Clock Generation

13.3.10 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.

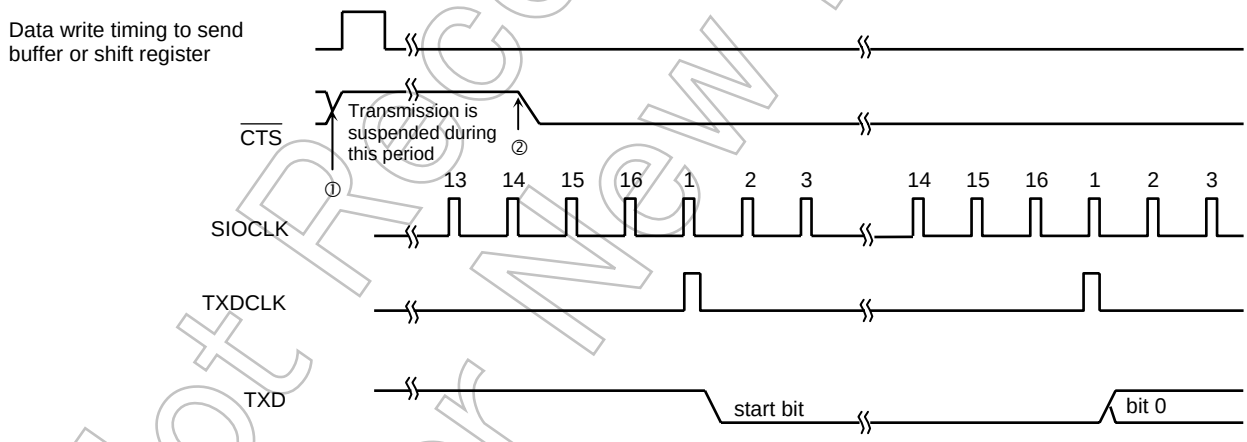
- Handshake function

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. 13-6 Handshake Function



(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. 13-7 $\overline{\text{CTS}}$ (Clear to Send) Signal Timing

13.3.11 Transmit Buffer

The send buffer (SC0BUF) is in a dual structure. The double buffering function may be enabled or disabled by setting the double buffer control bit <WBUF> in serial mode control register 2 (SC0MOD2). If double buffering is enabled, data written to send buffer 2 (SCOBUF) is moved to send buffer 1 (shift register).

If the transmit FIFO has been disabled (SCOFCNF <CNFG> = 0 or 1 and <FDPX1:0> = 01), the INTTX interrupt is generated at the same time and the send buffer empty flag <TBEMP> of SC0MOD2 is set to "1." This flag indicates that send buffer 2 is now empty and that the next transmit data can be written. When the next data is written to send buffer 2, the <TBEMP> flag is cleared to "0."

If the transmit FIFO has been enabled (SCNFCNF <CNFG> = 1 and <FDPX1:0> = 10/11), any data in the transmit FIFO is moved to the send buffer 2 and <TBEMP> flag is immediately cleared to "0." The CPU writes data to send buffer 2 or to the transmit FIFO.

If the transmit FIFO is disabled in the I/O interface SCLK input mode and if no data is set in send buffer 2 before the next frame clock input, which occurs upon completion of data transmission from send buffer 1, an under-run error occurs and a serial control register (SC0CR) <PERR> parity/under-run flag is set.

If the transmit FIFO is enabled in the I/O interface SCLK input mode, when data transmission from send buffer 1 is completed, the send buffer 2 data is moved to send buffer 1 and any data in transmit FIFO is moved to send buffer 2 at the same time.

If the transmit FIFO is disabled in the I/O interface SCLK output mode, when data in send buffer 2 is moved to send buffer 1 and the data transmission is completed, the SCLK output stops. So, no under-run errors can be generated.

If the transmit FIFO is enabled in the I/O interface SCLK output mode, the SCLK output stops upon completion of data transmission from send buffer 1 if there is no valid data in the transmit FIFO.

Note) In the I/O interface SCLK output mode, the SC0CR <PEER> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the SCLK output mode to another mode, SC0CR must be read in advance to initialize the flag.

If double buffering is disabled, the CPU writes data only to send buffer 1 and the transmit interrupt INTTX is generated upon completion of data transmission.

If handshaking with the other side is necessary, set the double buffer control bit <WBUF> to "0" (disable) to disable send buffer 2; any setting for the transmit FIFO should not be performed.

13.3.12 Transmit FIFO Buffer

In addition to the double buffer function already described, data may be stored using the transmit FIFO buffer. By setting <CNFG> of the SC0FCNF register and <FDPX1:0> of the SC0MOD1 register, the 4-byte send buffer can be enabled. In the UART mode or I/O interface mode, up to 4 bytes of data may be stored.

If data is to be transmitted with a parity bit in the UART mode, parity check must be performed on the receive side each time a data frame is received.

13.3.13 Transmit FIFO Operation

① I/O interface mode with SCLK output (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

SC0TFC <7:6> = 01: Clears transmit FIFO and sets the condition of interrupt generation

SC0TFC <1:0> = 00: Sets the interrupt to be generated at fill level 0.

SC0FCNF <1:0> = 01011: Inhibits continued transmission after reaching the fill level.

In this condition, data transmission can be initiated by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the send buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.

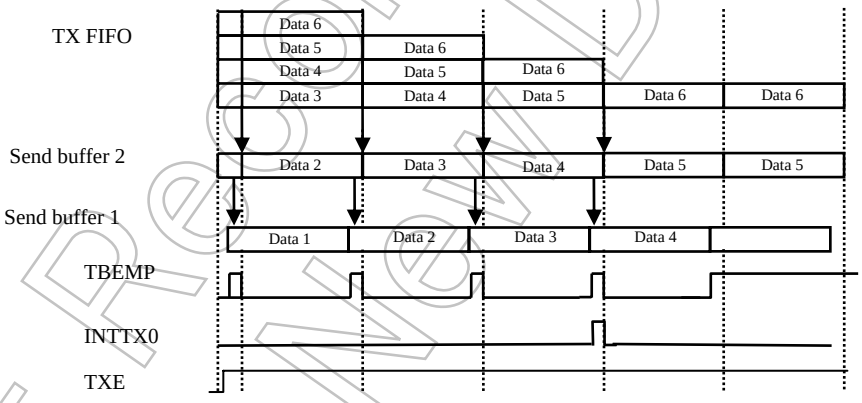


Fig. 13-8 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

SC0TFC <1:0> = 01: Clears the transmit FIFO and sets the condition of interrupt generation.

SC0TFC <7:2> = 000000: Sets the interrupt to be generated at fill level 0.

SC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the send buffer, the transmit FIFO interrupt is generated.

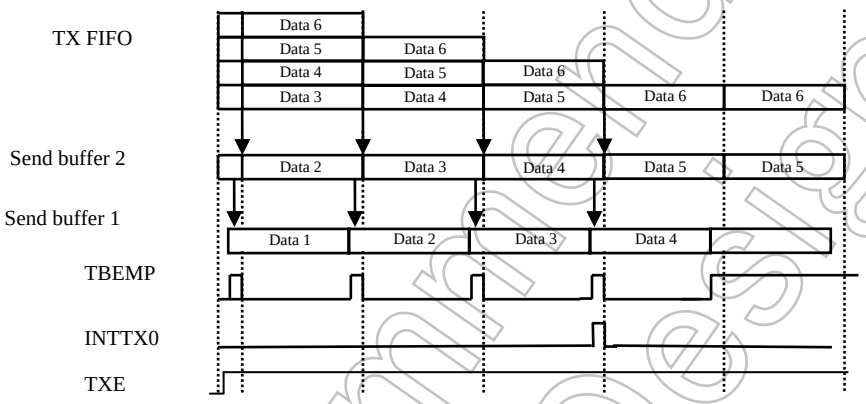


Fig. 13-9 Transmit FIFO Operation

13.3.14 Parity Control Circuit

If the parity addition bit <PE> of the serial control register SC0CR is set to "1," data is sent with the parity bit. Note that the parity bit may be used only in the 7- or 8-bit UART mode. The <EVEN> bit of SC0CR selects either even or odd parity.

Upon data transmission, the parity control circuit automatically generates the parity with the data written to the send buffer (SC0BUF). After data transmission is complete, the parity bit will be stored in SC0BUF bit 7 <TB7> in the 7-bit UART mode and in bit 7 <TB8> in the serial mode control register SC0MOD in the 8-bit UART mode. The <PE> and <EVEN> settings must be completed before data is written to the send buffer.

Upon data reception, the parity bit for the received data is automatically generated while the data is shifted to receive buffer 1 and moved to receive buffer 2 (SC0BUF). In the 7-bit UART mode, the parity generated is compared with the parity stored in SC0BUF <RB7>, while in the 8-bit UART mode, it is compared with the bit 7 <RB8> of the SC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the SC0CR register is set.

In the I/O interface mode, the SC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

13.3.15 Error Flag

Three error flags are provided to increase the reliability of received data.

1. Overrun error <OERR>: Bit 4 of the serial control register SC0CR

   In both UART and I/O interface modes, this bit is set to "1" when an error is generated by completing the reception of the next frame receive data before the receive buffer has been read. If the receive FIFO is enabled, the received data is automatically moved to the receive FIFO and no overrun error will be generated until the receive FIFO is full (or until the usable bytes are fully occupied). This flag is set to "0" when it is read. In the I/O interface SCLK output mode, no overrun error is generated and therefore, this flag is inoperative and the operation is undefined.

2. Parity error/under-run error <PERR>: Bit 3 of the SC0CR register

   In the UART mode, this bit is set to "1" when a parity error is generated. A parity error is generated when the parity generated from the received data is different from the parity received. This flag is set to "0" when it is read.

   In the I/O interface mode, this bit indicates an under-run error. When the double buffer control bit <WBUF> of the serial mode control register SC0MOD2 is set to "1" in the SCLK input mode, if no data is set to the transmit double buffer before the next data transfer clock after completing the transmission from the transmit shift register, this error flag is set to "1" indicating an under-run error. If the transmit FIFO is enabled, any data content in the transmit FIFO will be moved to the buffer. When the transmit FIFO and the double buffer are both empty, an under-run error will be generated. Because no under-run errors can be generated in the SCLK output mode, this flag is inoperative and the operation is undefined. If send buffer 2 is disabled, the under-run flag <PERR> will not be set. This flag is set to "0" when it is read.

3. Framing error <FERR>: Bit 2 of the SC0CR register

In the UART mode, this bit is set to "1" when a framing error is generated. This flag is set to "0" when it is read. A framing error is generated if the corresponding stop bit is determined to be "0" by sampling the bit at around the center. Regardless of the <SBLEN> (stop bit length) setting of the serial mode control register 2, SC0MOD2, the stop bit status is determined by only 1 bit on the receive side.

Operation mode	Error flag	Function
UART	OERR	Overrun error flag
	PERR	Parity error flag
	FERR	Framing error flag
I/O interface (SCLK input)	OERR	Overrun error flag
	PERR	Underrun error flag (WBUF = 1)
		Fixed to 0 (WBUF = 0)
	FERR	Fixed to 0
I/O interface (SCLK output)	OERR	Operation undefined
	PERR	Operation undefined
	FERR	Fixed to 0

13.3.16 Direction of Data Transfer

In the I/O interface mode, the direction of data transfer can be switched between "MSB first" and "LSB first" by the data transfer direction setting bit <DRCHG> of the SC0MOD2 serial mode control register 2. Don't switch the direction when data is being transferred.

13.3.17 Stop Bit Length

In the UART mode transmission, the stop bit length can be set to either 1 or 2 bits by bit 4 <SBLEN> of the SC0MOD2 register.

13.3.18 Status Flag

If the double buffer function is enabled (SC0MOD2 <WBUF> = "1"), the bit 6 flag <RBFLL> of the SC0MOD2 register indicates the condition of receive buffer full. When one frame of data has been received and transferred from buffer 1 to buffer 2, this bit is set to "1" to show that buffer 2 is full (data is stored in buffer 2). When the receive buffer is read by CPU/DMAC, it is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag. When double buffering is enabled (SC0MOD2 <WBUF> = "1"), the bit 7 flag <TBEMP> of the SC0MOD2 register indicates that send buffer 2 is empty. When data is moved from send buffer 2 to send buffer 1 (shift register), this bit is set to "1" indicating that send buffer 2 is now empty. When data is set to the send buffer by CPU/DMAC, the bit is cleared to "0." If <WBUF> is set to "0," this bit is insignificant and must not be used as a status flag.

13.3.19 Configurations of Send/Receive Buffers

		<WBUF> = 0	**<WBUF> = 1**
UART	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O interface (SCLK input)	Transmit buffer	Single	Double
	Receive buffer	Double	Double
I/O interface (SCLK output)	Transmit buffer	Single	Double
	Receive buffer	Single	Double

13.3.20 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.

13.3.21 Signal Generation Timing

① UART Mode:

Receive Side

Mode	9-bit	8-bit with parity	8-bit, 7-bit, and 7-bit with parity
Interrupt generation timing	Around the center of the 1st stop bit	Around the center of the 1st stop bit	Around the center of the 1st stop bit
Framing error timing	Around the center of the stop bit	Around the center of the stop bit	Around the center of the stop bit
Parity error generation timing	—	Around the center of the last (parity) bit	Around the center of the last (parity) bit
Overrun error generation timing	Around the center of the stop bit	Around the center of the stop bit	Around the center of the stop bit

Transmit Side

Mode	9-bit	8-bit with parity	8-bit, 7-bit, and 7-bit with parity
Interrupt generation timing (<WBUF> = 0)	Just before the stop bit is sent	Just before the stop bit is sent	Just before the stop bit is sent
Interrupt generation timing (<WBUF> = 1)	Immediately after data is moved to send buffer 1 (just before start bit transmission)	Immediately after data is moved to send buffer 1 (just before start bit transmission)	Immediately after data is moved to send buffer 1 (just before start bit transmission)

② I/O interface mode:

Receive Side

Interrupt generation timing (WBUF = 0)	SCLK output mode	Immediately after the rising edge of the last SCLK
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (WBUF = 1)	SCLK output mode	Immediately after the rising edge of the last SCLK (just after data transfer to receive buffer 2) or just after receive buffer 2 is read
	SCLK input mode	Immediately after the rising edge or falling edge of the last SCLK depending on the rising or falling edge triggering mode, respectively (right after data is moved to receive buffer 2)
Overrun error generation timing	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)

Transmit Side

Interrupt generation timing (WBUF = 0)	SCLK output mode	Immediately after the rising edge of the last SCLK
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for rising or falling edge mode, respectively)
Interrupt generation timing (WBUF = 1)	SCLK output mode	Immediately after the rising edge of the last SCLK or just after data is moved to send buffer 1
	SCLK input mode	Immediately after the rising or falling edge of the last SCLK (for the rising or falling edge mode, respectively) or just after data is moved to send buffer 1
Under-run error generation timing	SCLK input mode	Immediately after the falling or rising edge of the next SCLK (for the rising or falling edge triggering mode, respectively)

Note 1) Do not modify any control register when data is being sent or received (in a state ready to send or receive).
Note 2) Do not stop the receive operation (by setting SC0MOD0 <RXE> = "0") when data is being received.
Note 3) Do not stop the transmit operation (by setting SC0MOD1 <TXE> = "0") when data is being transmitted.

13.4 Register Description (Only for Channel 0)

SC0MOD0 (0xFFFF_F262)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Send data Bit 8	Handshake function control 0: Disables CTS 1: Enables CTS	Receive control 0: Disables reception 1: Enables reception	Wake-up function 0: Disable 1: Enable	Serial transfer mode 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode		Serial transfer clock (for UART) 00: Timer TB0OUT 01: Baud rate generator 10: Internal f_{SYS}/2 clock 11: External clock (SCLK0 input)	

Note) In the I/O interface mode, the serial control register (SC0CR) is used for clock

Wakeup function

	9-bit UART	Other mode
0	Interrupt when received	don't care
1	Interrupt at RB8=1	

Handshake function ($\overline{\text{CTS}}$ pin) enable

0	Disable (transmission is always allowed)
1	Enable

Note) With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2). Then set <RXE> to "1."

Fig. 13-10 Serial Mode Control Register 0 (for SIO0, SC0MOD0)

SC0MOD1 (0xFFFF_F265)

	7	6	5	4	3	2	1	0
bit Symbol	I2S0	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	–
Read/Write	R/W	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 000: None 100: 8SCLK 001: 1SCLK 101:16SCLK 010: 2SCLK 110: 32SCLK 011: 4SCLK 111: 64SCLK			Write "0."

Fig. 13-11 Serial Mode Control Register 1 (for SIO0, SC0MOD1)

<SINT2:0>: Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode. This parameter is invalid for the UART mode or when an external clock is used.

<TXE>: This bit enables transmission and is valid for all the transfer modes. If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<FDPX1:0>: Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<I2S0>: Specifies the Idle mode operation.

SC0MOD2 (0xFFFF_F266)

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R			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 0: 1-bit 1: 2-bit	Setting transfer direction 0: LSB first 1: MSB first	W-buffer 0: Disable 1: Enable	Soft reset Overwrite "01" on "10" to reset	

<SWRST1:0>: Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the mode register parameters SC0MOD0 <RXE>, SC0MOD1<TXE>, SC0MOD2 <TBEMP>, <RBFLL>, and <TXRUN>, control register parameters SC0CR <OERR>, <PERR>, and <FERR>, and their internal circuits are initialized.

<WBUF>: This parameter enables or disables the send/receive buffers to send (in both SCLK output/input modes) and receive (in SCLK output mode) data in the I/O interface mode and to transmit data in the UART. In all other modes, double buffering is enabled regardless of the <WBUF> setting.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, it is fixed to LSB first.

<TXRUN>: This is a status flag to show that data transmission is in progress.
When this bit is set to "1," it indicates that data transmission operation is in progress. If it is "0," the bit 7 <TBEMP> is set to "1" to indicate that the transmission has been fully completed and the same <TBEMP> is set to "0" to indicate that the send buffer contains some data waiting for the next transmission.

<RBFLL>: This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0."
If double buffering is disabled, this flag is insignificant.

<TBEMP>: This flag shows that the send double buffers are empty. When data in the send double buffers is moved to the send shift register and the double buffers are empty, this bit is set to "1." Writing data again to the double buffers sets this bit to "0."
If double buffering is disabled, this flag is insignificant.

<SBLEN>: This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

(Note) While data transmission is in progress, any software reset operation must be executed twice in succession.

Fig. 13-12 Serial Mode Control Register

SC0CR (0xFFFF_F261)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to "0" when read)			R/W	
After reset	0	0	0	0	0	0	0	0
Function	Receive data Bit 8	Parity 0: Odd 1: Even	Add parity 0: Disable 1: Enable	0: Normal operation 1: Error			0: SCLK0 ↑	0: Baud rate generator 1: SCLK0 pin input
				Overrun	Parity/ under-run	Framing	1: SCLK0 ↓	

I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK0 input operation

0	Data send/receive at rising edges of SCLK0 ↑
1	Data send/receive at falling edges of SCLK0 ↓

Framing error flag
Parity error/under-run error flag
Overrun error flag
} Cleared to "0" when read

Add/check even parity

0	Odd parity
1	Even parity

(Note) Any error flag is cleared when read.

Fig. 13-13 Serial Control Register (for SIO0, SC0CR)

BR0CR (0xFFFF_F263)

	7	6	5	4	3	2	1	0
bit Symbol	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Write "0."	N+(16-K)/16 divider function 0: Disable 1: Enable	00: φT1 01: φT4 10: φT16 11: φT64		Divide ratio "N"			

Select input clock to the baud rate generator

00	Internal clock φT1
01	Internal clock φT4
10	Internal clock φT16
11	Internal clock φT64

BR0ADD (0xFFFF_F264)

	7	6	5	4	3	2	1	0
bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	R				R/W			
After reset	0				0	0	0	0
Function	Always reads "0."				Specify K for the "N + (16 - K)/16" division			

Setting divide ratio of the baud rate generator

	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0
BR0ADD <BR0K3:0> \ BR0CR <BR0S3:0>	0000 (N = 16) ~ 0001 (N = 1)	0010 (N = 2) ~ 1111 (N = 15)	0001 (N = 1) (ONLY UART) ~ 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001 (K = 1) ~ 1111 (K = 15)	Disable	$N+\frac{(16-K)}{16}$ Division	Divide by N

(Note 1) In the UART mode, the division ratio "1" of the baud rate generator can be specified only when the "N + (16 - K)/16" division function is not used. In the I/O interface mode, the division ratio "1" of the baud rate generator can be specified only when double buffering is used.

(Note 2) To use the "N + (16 - K)/16" division function, be sure to set BR0CR <BR0ADDE> to "1" after setting the K value (K = 1 to 15) to BR0ADD <BR0K3:0>. However, don't use the "N + (16 - K)/16" division function when BR0CR <BR0S3:0> is set to either "0000" or "0001" (N = 16 or 1).

(Note 3) The "N + (16 - K)/16" division function can only be used in the UART mode. In the I/O interface mode, the "N + (16 - K)/16" division function must be disabled (prohibited) by setting BR0CR <BR0ADDE> to "0."

Fig. 13-14 Baud Rate Generator Control (for SIO0, BR0CR, BR0ADD)

SC0BUF (0xFFFF_F260)

	7	6	5	4	3	2	1	0
bit Symbol	TB7/RB7	TB6/RB6	TB5/RB5	TB4/RB4	TB3/RB3	TB2/RB2	TB1/RB1	TB0/RB0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	TB7 to TB0: Send buffer + FIFO RB7 to RB0: Receive buffer + FIFO							

Fig. 13-15

Note: HSC0BUF works as a send buffer for WR operation and as a receive buffer for RD operation.

Fig. 13-16 FIFO Configuration Register

SC0FCNF (0xFFFF_F26C)

	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

<CNFG>: If enabled, the SCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows:

<FDPX1:0> = 01 (Half duplex RX) ---- 4-byte RX FIFO

<FDPX1:0> = 10 (Half duplex TX) ---- 4-byte TX FIFO

<FDPX1:0> = 11 (Full duplex) ---- 2-Byte RX FIFO + 2-Byte TX FIFO

<RXTXCNT>:0 The function to automatically disable RXE/TXE bits is disabled.

1: If enabled, the SCOMOD1 <FDPX1:0> is used to set as follows:

<FDPX1:0> = 01 (Half duplex RX) ------ When the RX FIFO is filled up to the specified number of valid bytes, RXE is automatically set to "0" to inhibit further reception.

<FDPX1:0> = 10 (Half duplex TX) ------ When the TX FIFO is empty, TXE is automatically set to "0" to inhibit further transmission.

<FDPX1:0> = 11 (Full duplex) ---------- When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<RFIE>: When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<TFIE>: When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFST>: When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected.

0: The maximum number of bytes of the FIFO configured 4 bytes when <FDPX1:0> = 01 (Half duplex RX) and 2 bytes for <FDPX1:0> = 11 (Full duplex)

1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL1:0>.

(Note 1) Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.

Fig. 13-17 Receive FIFO Control Register

SC0RFC (0xFFFF_F268)

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS					RIL1	RIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0				0	0
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 Note: RIL1 is ignored when FDPX1:0 = 11 (full duplex)	

0: An interrupt is generated when the specified fill level is reached.
1: An interrupt is generated when the specified fill level is reached or if the specified fill level has been exceeded at the time data is read.

Transmit FIFO Configuration Register

SC0TFC (0xFFFF_F269)

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS					TIL1	TIL0
Read/Write	w	R/W	R				R/W	
After reset	0	0	0				0	0
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 when FDPX1:0 = 11 (full duplex).	

0: An interrupt is generated when the specified fill level is reached.
1: An interrupt is generated when the specified fill level is reached or if the level is lower than the specified fill level at the time new data is written.

Fig. 13-18 Receive FIFO Status Register

SC0RST (0xFFFF_F26A)

	7	6	5	4	3	2	1	0
bit Symbol	ROR					RLVL2	RLVL1	RLVL0
Read/Write	R	R				R		
After reset	0	0				0	0	0
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		

(Note) The <ROR> bit is cleared to "0" when receive data is read from the SC0BUF register.

Fig. 13-19 Transmit FIFO Status Register

SC0TST (0xFFFF_F26B)

	7	6	5	4	3	2	1	0
bit Symbol	TUR					TLVL2	TLVL1	TLVL0
Read/Write	R	R				R		
After reset	1	0				0	0	0
Function	TX FIFO Under run 1: Generated Cleared by writing to FIFO	Always reads "0."				Status of TX FIFO fill level 000: Empty 001: 1Byte 010: 2Byte 011: 3Byte 100: 4Byte		

(Note) The <TUR> bit is cleared to "0" when transmit data is written to the SC0BUF register.

SIO Enable Register

SC0EN (0xFFFF_F267)

	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 operation is disabled, the clock will not be supplied to the SIO module except for the register part and thus power dissipation can be reduced (other registers cannot be accessed for read/write operation). When SIO is to be used, be sure to enable SIO by setting "1" to this register before setting any other registers of the SIO module. If SIO is enabled once and then disabled, any register setting is maintained.

13.5 Operation in Each Mode

13.5.1 Mode 0 (I/O interface mode)

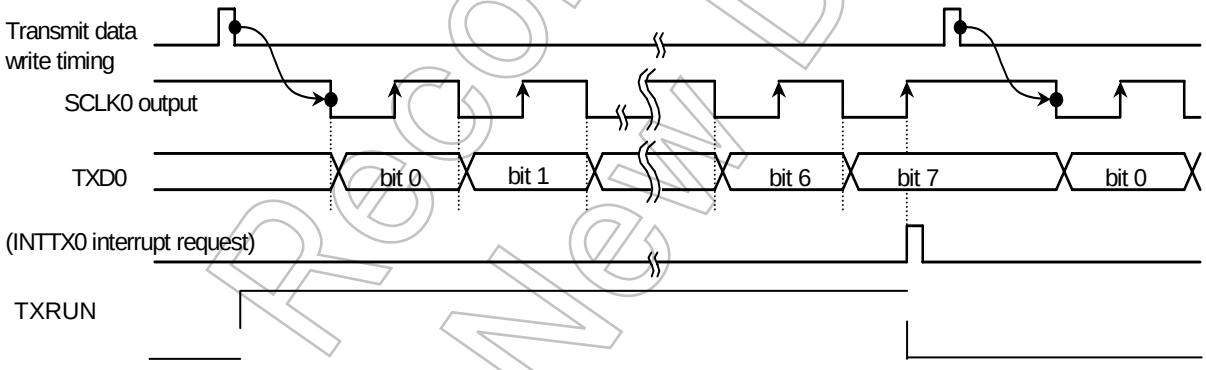
Mode 0 consists of two modes, i.e., the "SCLK output" mode to output synchronous clock and the "SCLK input" mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

① Sending data

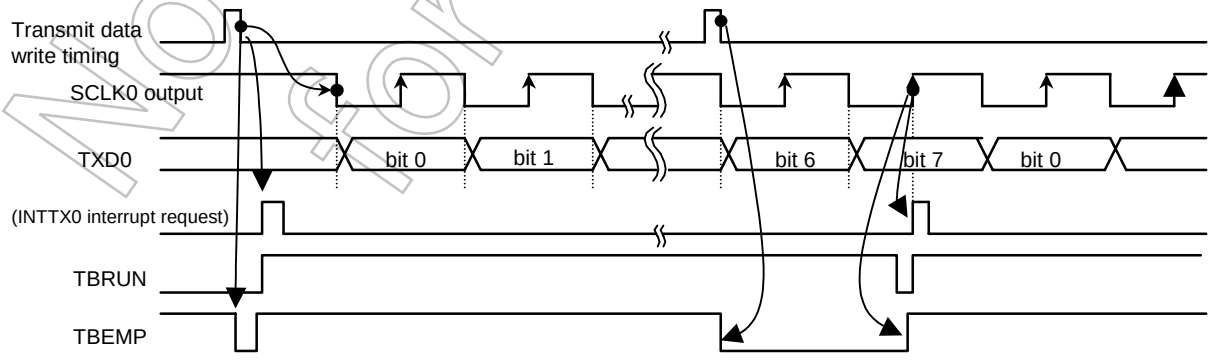
SCLK output mode

In the SCLK output mode, if SC0MOD2<WBUF> is set to "0" and the send double buffers are disabled, 8 bits of data are output from the TXD0 pin and the synchronous clock is output from the SCLK0 pin each time the CPU writes data to the send buffer. When all data is output, the INTTX0 interrupt is generated.

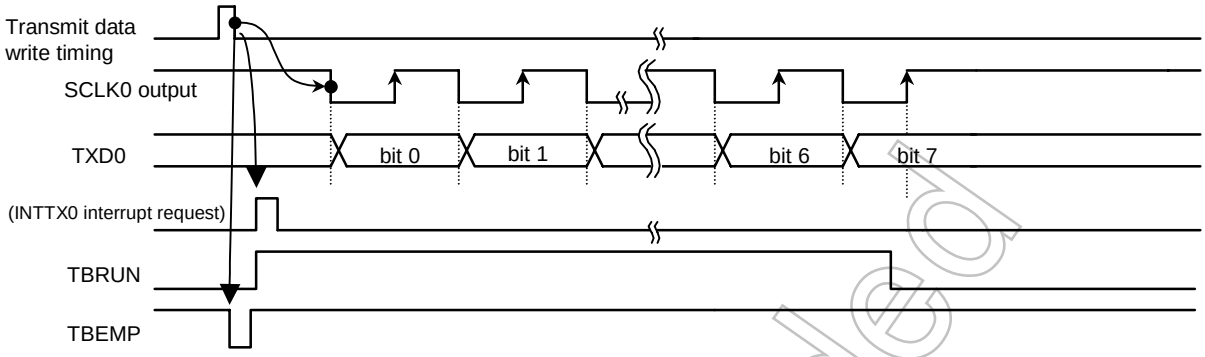
If SC0MOD2 <WBUF> is set to "1" and the send double buffers are enabled, data is moved from send buffer 2 to send buffer 1 when the CPU writes data to send buffer 2 while data transmission is halted or when data transmission from send buffer 1 (shift register) is completed. When data is moved from send buffer 2 to send buffer 1, the send buffer empty flag SC0MOD2 <TBEMP> is set to "1," and the INTTX0 interrupt is generated. If send buffer 2 has no data to be moved to send buffer 1, the INTTX0 interrupt is not generated and the SCLK0 output stops.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



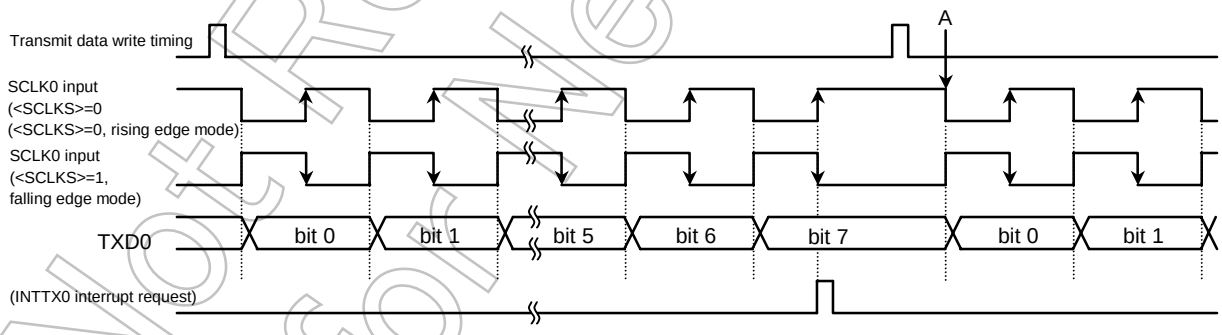
<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 13-20 Send Operation in the I/O Interface Mode (SCLK0 Output Mode)

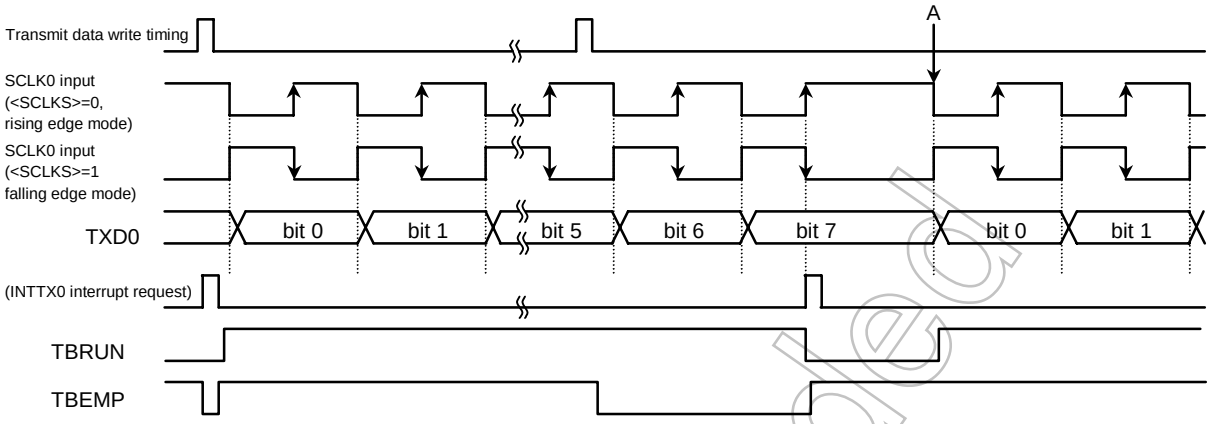
SCLK input mode

In the SCLK input mode, if SC0MOD2 <WBUF> is set to "0" and the send double buffers are disabled, 8-bit data that has been written in the send buffer is output from the TXD0 pin when the SCLK0 input becomes active. When all 8 bits are sent, the INTTX0 interrupt is generated. The next send data must be written before the timing point "A" as shown in Fig. 13-21.

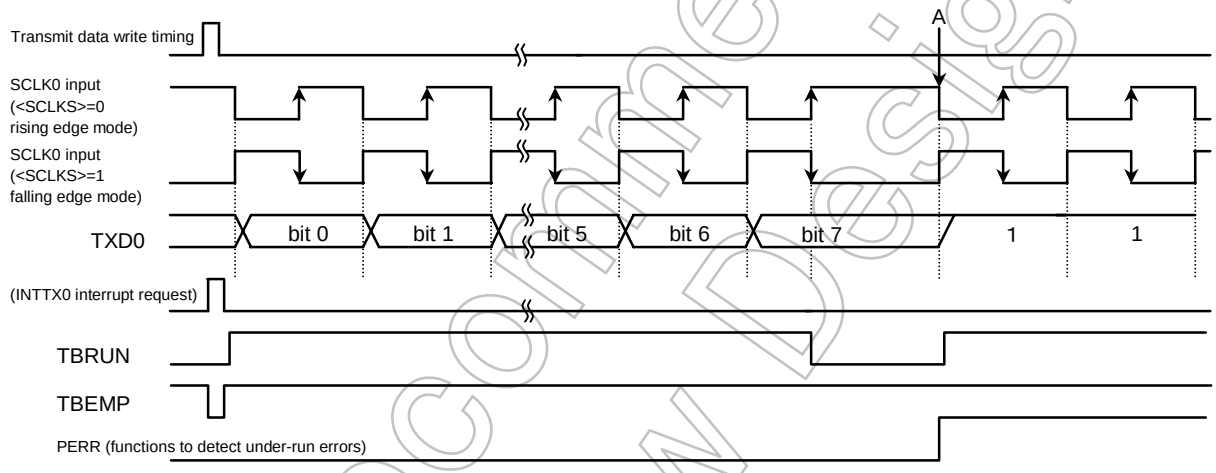
If SC0MOD2 <WBUF> is set to "1" and the send double buffers are enabled, data is moved from send buffer 2 to send buffer 1 when the CPU writes data to send buffer 2 before the SCLK0 becomes active or when data transmission from send buffer 1 (shift register) is completed. As data is moved from send buffer 2 to send buffer 1, the send buffer empty flag SC0MOD2 <TBEMP> is set to "1" and the INTTX0 interrupt is generated. If the SCLK0 input becomes active while no data is in send buffer 2, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is sent.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 13-21 Send Operation in the I/O Interface Mode (SCLK0 Input Mode)

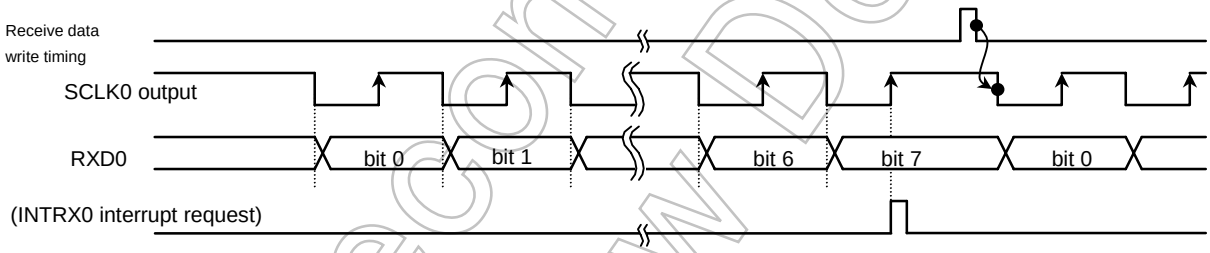
② Receiving data

SCLK output mode

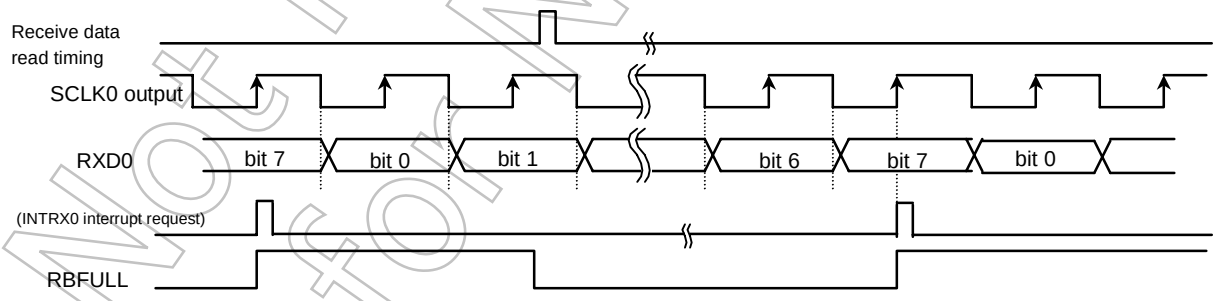
In the SCLK output mode, if SC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

The first SCLK output can be started by setting the receive enable bit SC0MOD0 <RXE> to "1." If the receive double buffering is enabled with SC0MOD2 <WBUF> set to "1," the first frame received is moved to receive buffer 2 and receive buffer 1 can receive the next frame successively. As data is moved from receive buffer 1 to receive buffer 2, the receive buffer full flag SC0MOD2 <RBFULL> is set to "1" and the INTRX0 interrupt is generated.

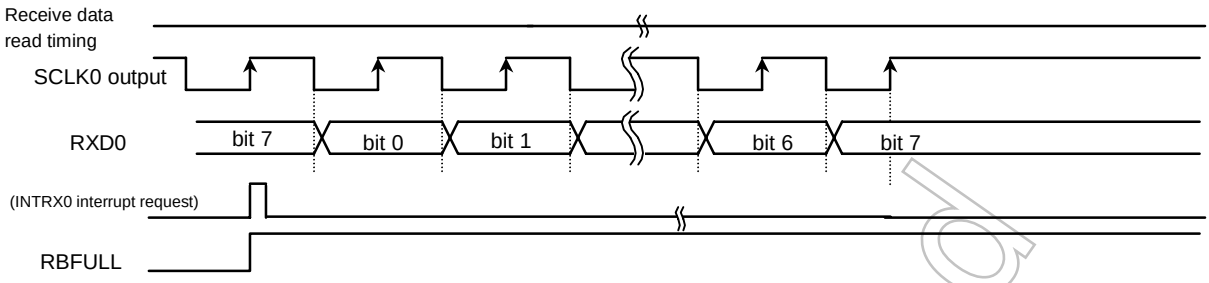
While data is in receive buffer 2, if CPU/DMAC cannot read data from receive buffer 2 in time before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 2 and thus the INTRX0 interrupt is generated and data reception resumes.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if data is read from buffer 2)



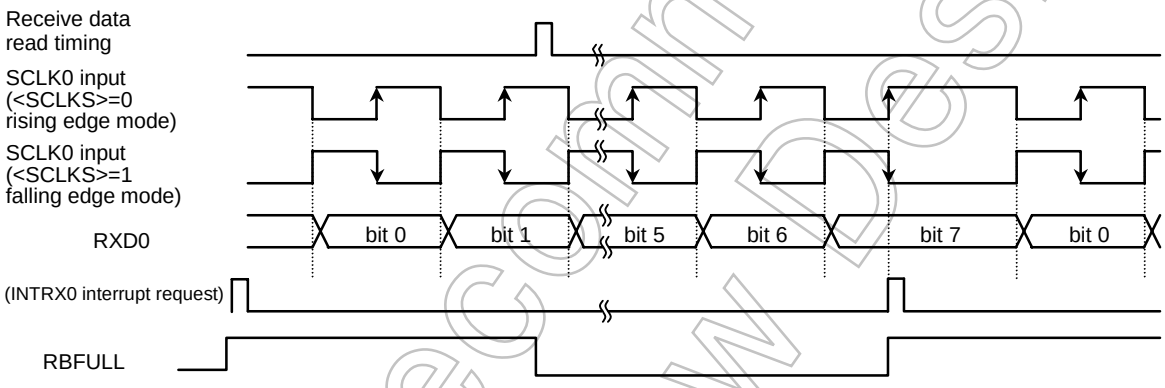
<WBUF> = "1" (if double buffering is enabled) (if data cannot be read from buffer 2)

Fig. 13-22 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

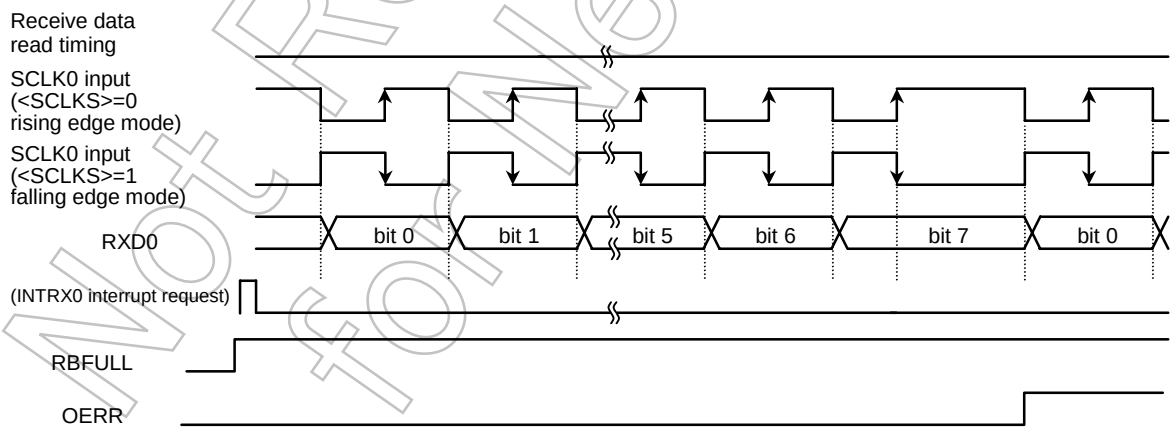
SCLK input mode

In the SCLK input mode, since receive double buffering is always enabled, the received frame can be moved to receive buffer 2 and receive buffer 1 can receive the next frame successively.

The INTRX receive interrupt is generated each time received data is moved to received buffer 2.



If data is read from buffer 2



If data cannot be read from buffer 2

Fig. 13-23 Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

(Note) To receive data, SC0MOD <RXE> must always be set to "1" (receive enable) regardless of the SCLK input or output mode.

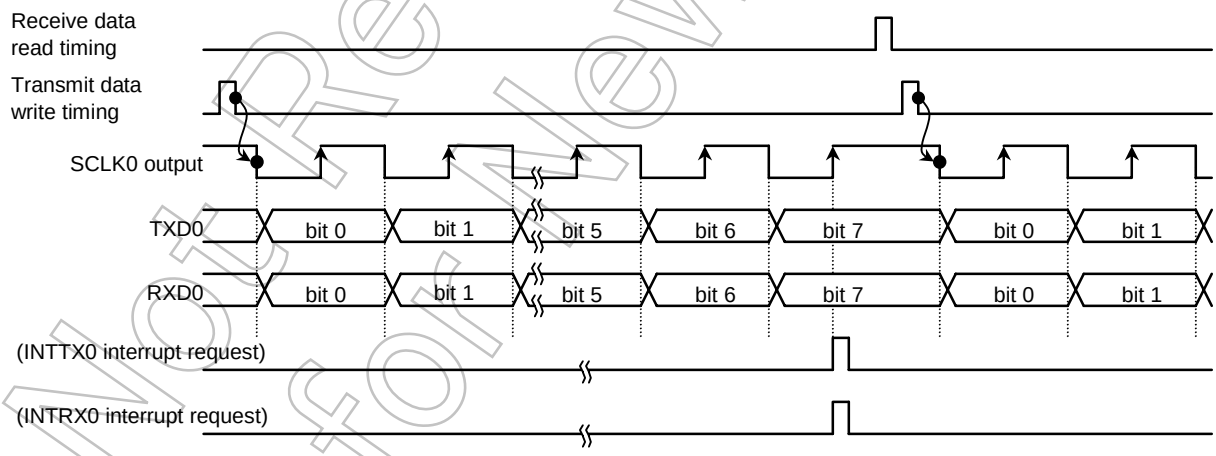
③ Send and receive (full-duplex)

The full-duplex mode is enabled by setting bit 6 <FDPX0> of the serial mode control register 1 (SC0MOD1) to "1."

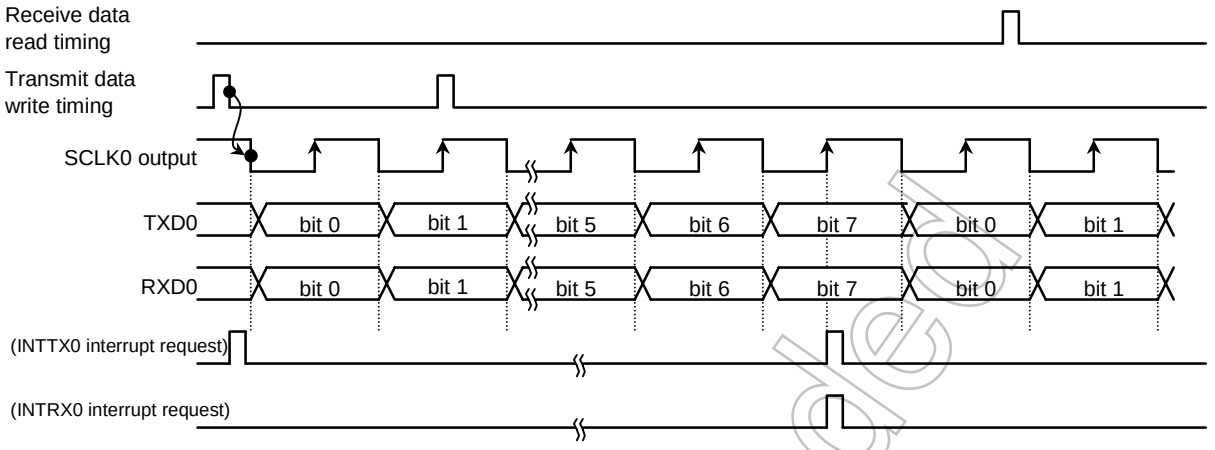
SCLK output mode

In the SCLK output mode, if SC0MOD2 <WBUF> is set to "0" and both the send and receive double buffers are disabled, SCLK is output when the CPU writes data to the send buffer. Subsequently, 8 bits of data are shifted into receive buffer 1 and the INTRX0 receive interrupt is generated. Concurrently, 8 bits of data written to the send buffer are output from the TXD0 pin, the INTTX0 send interrupt is generated when transmission of all data bits has been completed. Then, the SCLK output stops. In this, the next round of data transmission and reception starts when the data is read from the receive buffer and the next send data is written to the send buffer by the CPU. The order of reading the receive buffer and writing to the send buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

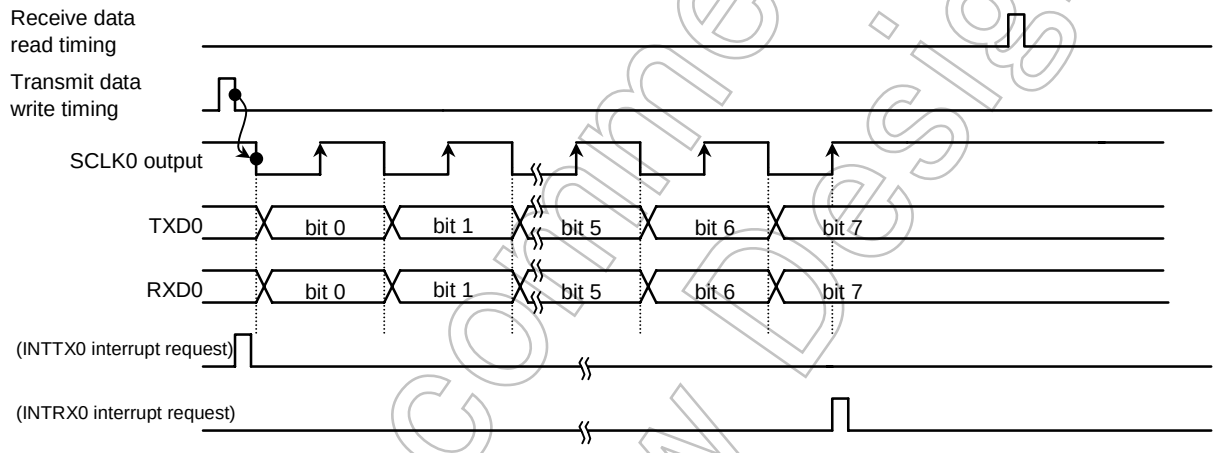
If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, SCLK is output when the CPU writes data to the send buffer. Subsequently, 8 bits of data are shifted into receive buffer 1, moved to receive buffer 2, and the INTRX0 interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is output from the TXD0 pin. When all data bits are sent out, the INTTX0 interrupt is generated and the next data is moved from the send buffer 2 to send buffer 1. If send buffer 2 has no data to be moved to send buffer 1 (SC0MOD2 <TBEMP> = 1) or when receive buffer 2 is full (SC0MOD2 <RBFULL> = 1), the SCLK clock is stopped. When both conditions are satisfied, i.e., receive data is read and send data is written, the SCLK output is resumed and the next round of data transmission is started.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled)



<WBUF> = "1" (if double buffering is enabled)

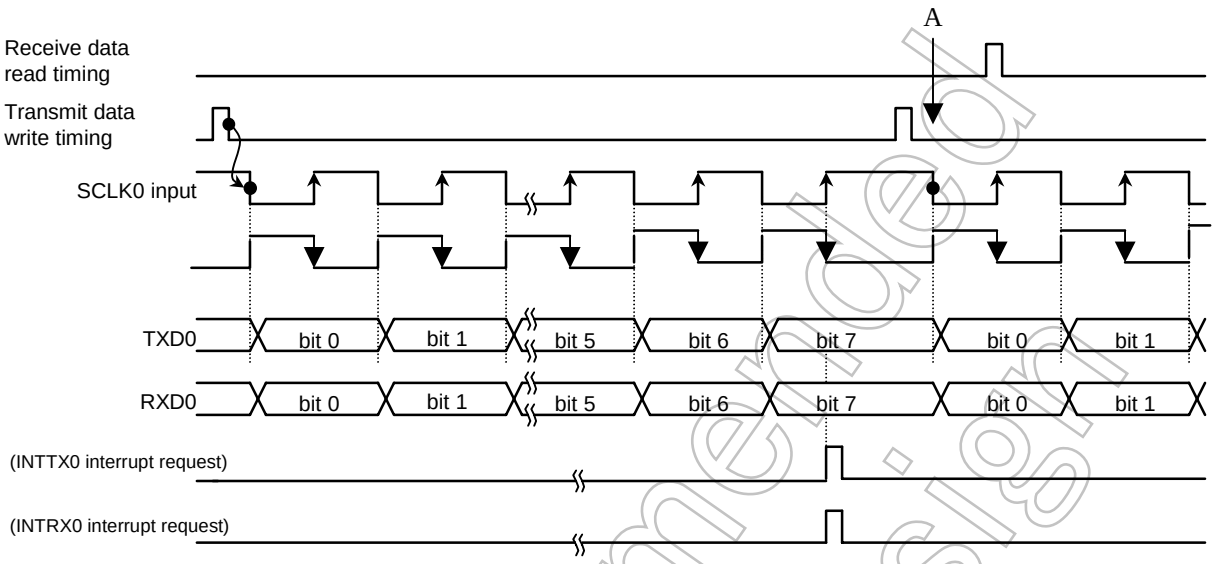
Fig. 13-24 Send/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

SCLK input mode

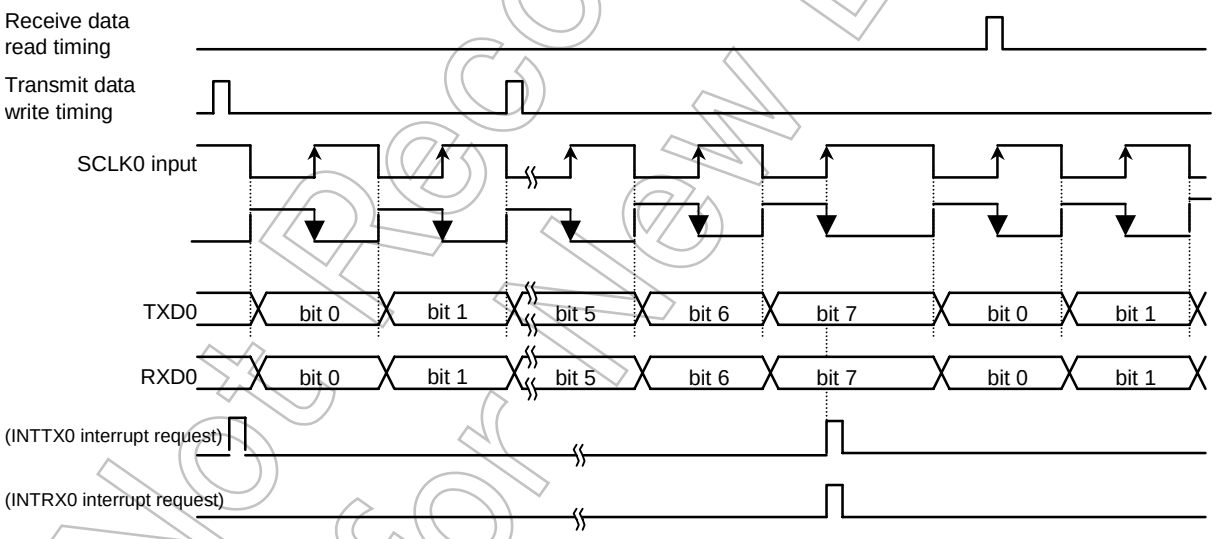
In the SCLK input mode with SC0MOD2 <WBUF> set to "0" and the send double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the send buffer is output from the TXD0 pin and 8 bits of data is shifted into the receive buffer when the SCLK input becomes active. The INTTX0 interrupt is generated upon completion of data transmission and the INTRX0 interrupt is generated at the instant the received data is moved from receive buffer 1 to receive buffer 2. Note that transmit data must be written into the send buffer before the SCLK input for the next frame (data must be written before the point A in Fig. 13-25). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing send buffer 2 data is moved to send buffer 1 after completing data transmission from send buffer 1. At the same time, the 8 bits of data received is shifted to buffer 1, moved to receive buffer 2, and the INTRX0 interrupt is generated. Upon the SCLK input for the next frame, transmission from send buffer 1 (in which data has been moved from send buffer 2) is started while receive data is shifted into receive buffer 1 simultaneously. If data in receive buffer 2 has not been read when the last bit of the frame is received, an overrun

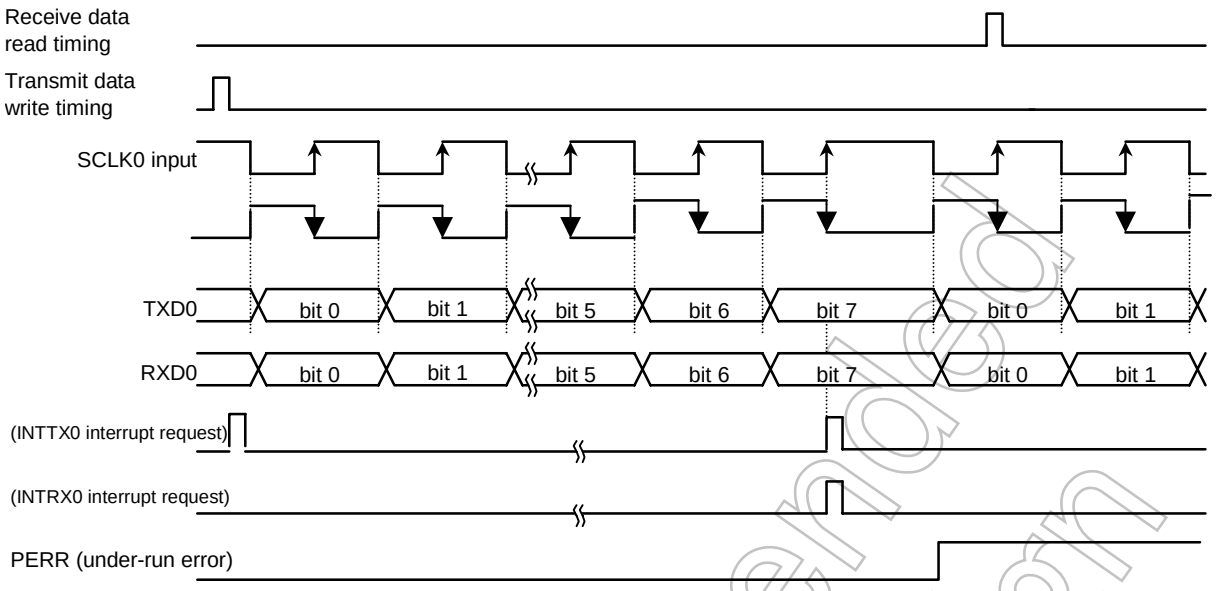
error occurs. Similarly, if there is no data written to send buffer 2 when SCLK for the next frame is input, an under-run error occurs.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (no errors)



<WBUF> = "1" (if double buffering is enabled) (error generation)

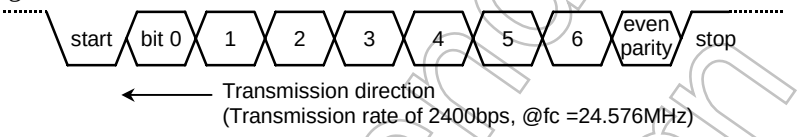
Fig. 13-25 Send/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

13.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SC0MOD <SM1, 0>) to "01."

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SC0CR <EVEN> bit. The length of the stop bit can be specified using SC0MOD2<SBLEN>.

Example: The control register settings for transmitting in the following data format are listed in the following table.



Transmission direction
(Transmission rate of 2400bps, @fc =24.576MHz)

* Clocking conditions

System clock	: High-speed (fc)
High-speed clock gear	: x 1 (fc)
Prescaler clock	: $f_{periph}/4$ ($f_{periph} = f_{sys}$)

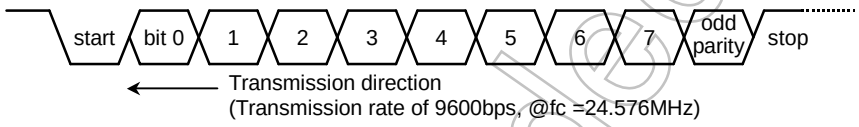
	7 6 5 4 3 2 1 0	
P6CR	← _ _ _ _ _ _ _ 1	Designates P60 as the TXD0 pin.
P6FC	← _ _ _ _ _ _ _ 1	
P6FC2	← _ _ _ _ _ _ _ 1	
SC0MOD	← X 0 _ X 0 1 0 1	Sets the 7-bit UART mode.
SC0CR	← X 1 1 X X X 0 0	Adds even parity.
BR0CR	← 0 0 1 0 1 0 1 0	Sets the data rate to 2400 bps.
IMC4	← _ 1 1 X 0 1 0 0	Enables the INTTX0 interrupt and sets to level 4 by the <31:24> bits of the 32 bit register.
SC0BUF	← * * * * * * * *	Sets the data to be sent.

Note:X: don't care - : no change

13.5.3 Mode 2 (8-bit UART Mode)

The 8-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using SC0CR <PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using SC0CR <EVEN>.

Example: The control register settings for receiving data in the following format are as follows:



Transmission direction
(Transmission rate of 9600bps, @fc =24.576MHz)

* Clocking conditions
- System clock : High-speed (fc)
- High-speed clock gear : x 1 (fc)
- Prescaler clock : $f_{perip}h/4$ ($f_{periph} = f_{sys}$)

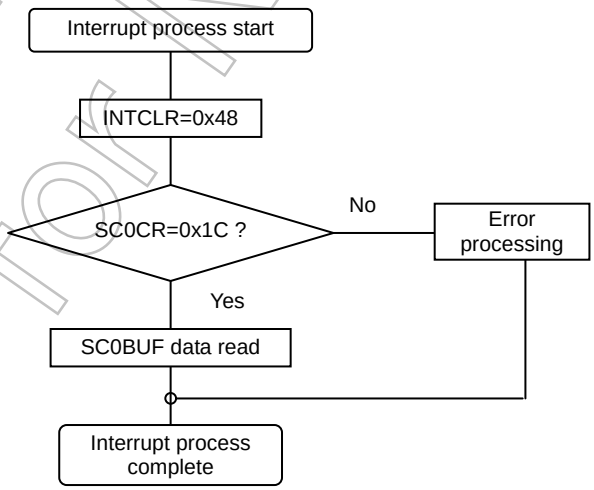
Main routine settings

		7	6	5	4	3	2	1	0	
P6CR	←	_	_	_	_	_	_	0	_	
P6FC	←	_	_	_	_	_	_	1	_	Designates P62 as the RXD0 pin.
P6FC2	←	_	_	_	_	_	_	1	_	
SC0MOD	←	_	0	0	X	1	0	0	1	Selects the 8-bit UART mode.
SC0CR	←	X	0	1	X	X	X	0	0	Sets odd parity.
BR0CR	←	0	0	0	1	0	1	0	1	Sets the data rate to 9600 bps.
IMC4	←	_	1	1	X	0	1	0	0	Enables the INTRX0 interrupt and sets to level 4 by the <23:16> bits of the 32 bit register.
SC0MOD	←	_	_	1	X	_	_	_	_	Enables reception of data.

An example interrupt routine process

INTCLR ← X 1 0 0 1 0 0 0	Clears the interrupt request. 0x0000_0048
Reg. ← SC0CR AND 0x1C	Performs error check
if reg. is not "0" then error processing	
Set SC0BUF to Reg.	Reads received data.
Interrupt processing is completed	

Note: X: don't care - : no change



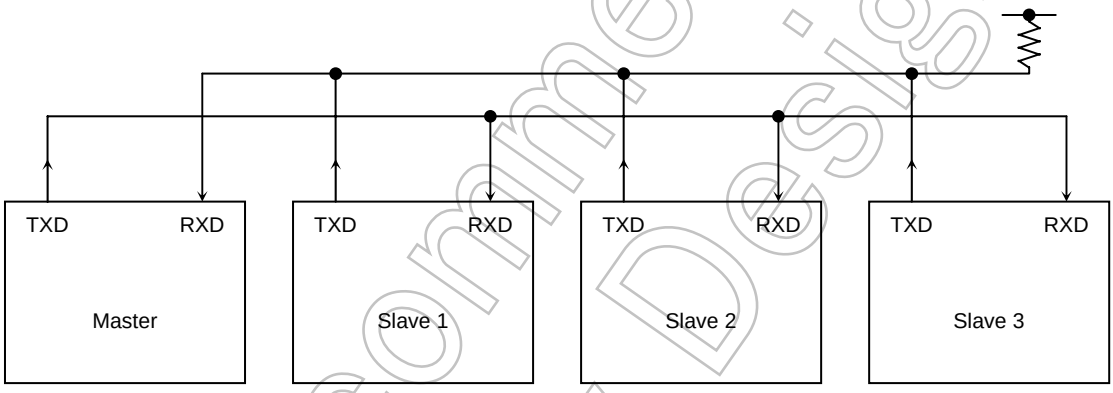
13.5.4 Mode 3 (9-bit UART)

The 9-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "11." In this mode, parity bits must be disabled (SC0CR <PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SC0MOD0) for transmit data and it is stored in bit 7 <RB8> of the serial control register SC0CR upon receiving data. When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SC0BUF. The stop bit length can be specified using SC0MOD2 <SBLEN>.

Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1."



(Note) The TXD pin of the slave controller must be set to the open drain output mode using the ODE register.

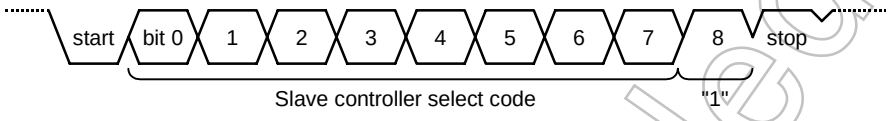
Fig. 13-26 Serial Links to Use Wake-up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

② Set SC0MOD <WU> to "1" for the slave controllers to make them ready to receive data.

③ The master controller is to send a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1."



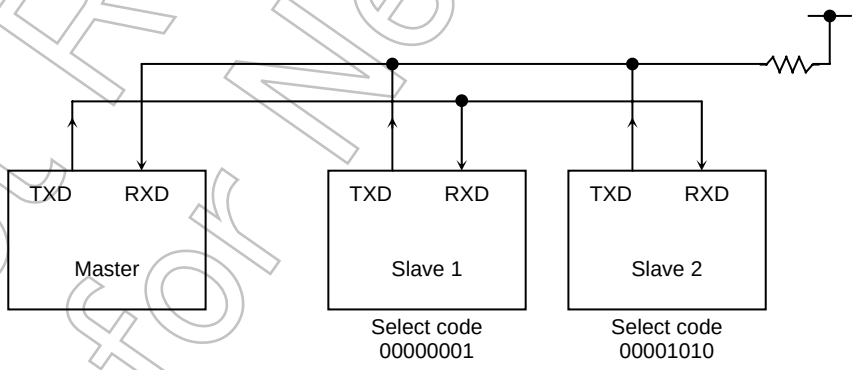
④ Every slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0."

⑤ The master controller transmits data to the designated slave controller (the controller of which SC0MOD <WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0."

⑥ The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (INTRX0) is generated.

Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

Example setting: Using the internal clock $f_{SYS}/2$ as the transfer clock, two slave controllers are serially linked as follows:



3) Master controller setting

Main routine

P6CR	← _ _ _ _ _ _ 0 1	Designates P60/P61 as the TXD0/RXD0 pins, respectively.
P6FC	← _ _ _ _ _ _ 1 1	
P6FC2	← _ _ _ _ _ _ 1 1	
IMCC4	← _ 1 1 _ 0 1 0 1	Enables the INTRX0 interrupt and sets to level 5 by the <23:16> bits of the 32 bit register.
IMCC4	← _ 1 1 _ 0 1 0 0	Enables the INTTX0 interrupt and sets to level 4 by the <31:24> bits of the 32 bit register.
SC0MOD0	← 1 0 1 0 1 1 1 0	Sets the 9-bit UART mode and f_{SYS}/2 transfer clock.
SC0BUF	← 0 0 0 0 0 0 0 1	Sets the select code of Slave 1.

Interrupt routine (INTTX0)

INTCLR	← X 1 0 0 1 1 0 0	Clears the interrupt request. (INTTX0)
SC0MOD0	← 0 _ _ _ _ _ _ _	Sets TB8 to "0."
SC0BUF	← * * * * * * * *	Sets the data to be sent.

Interrupt processing is completed.

4) Slave controller setting

Main routine

P6CR	← _ _ _ _ _ _ 0 1	Designates P60 as TXD (open drain output) and P61 as RXD.
P6FC	← _ _ _ _ _ _ 1 1	
P6FC2	← _ _ _ _ _ _ 1 1	
P6ODE	← _ _ _ _ _ _ 1 _	
IMC4	← _ _ 1 1 0 1 1 0	Enables INTTX0 and INTRX0.
IMC4	← _ _ 1 1 0 1 0 1	
SC0MOD0	← 0 0 1 1 1 1 1 0	Sets the 9-bit UART mode and f_{SYS}/2 transfer clock and sets <WU> to "1."

Interrupt routine (INTRX0)

INTCLR	← 0 1 0 0 1 0 0 0	Clears the interrupt request.
Reg.	← SC0BUF	
if Reg. = select code, Then		
SC0MOD0	← _ _ _ 0 _ _ _ _	Clears <WU> to "0."

14. Serial Channel (HSIO)

This device has three high-speed serial I/O channels: HSIO0 to HSIO2. Each channel operates in either the UART mode (asynchronous communication) or the I/O interface mode (synchronous communication) which is selected by the user.

I/O interface mode
- Mode 0: This is the mode to send and receive I/O data and associated synchronization signals (HSCLK) to extend I/O.

Asynchronous (UART) mode:
- Mode 1: TX/RX Data Length: 7 bits
- Mode 2: TX/RX Data Length: 8 bits
- Mode 3: TX/RX Data Length: 9 bits

In the above modes 1 and 2, parity bits can be added. The mode 3 has a wakeup function in which the master controller can start up slave controllers via the serial link (multi-controller system). Fig. 14-2 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.

As the HSIOs 0 to 2 operate in the same way, Only HSIO0 is described here.

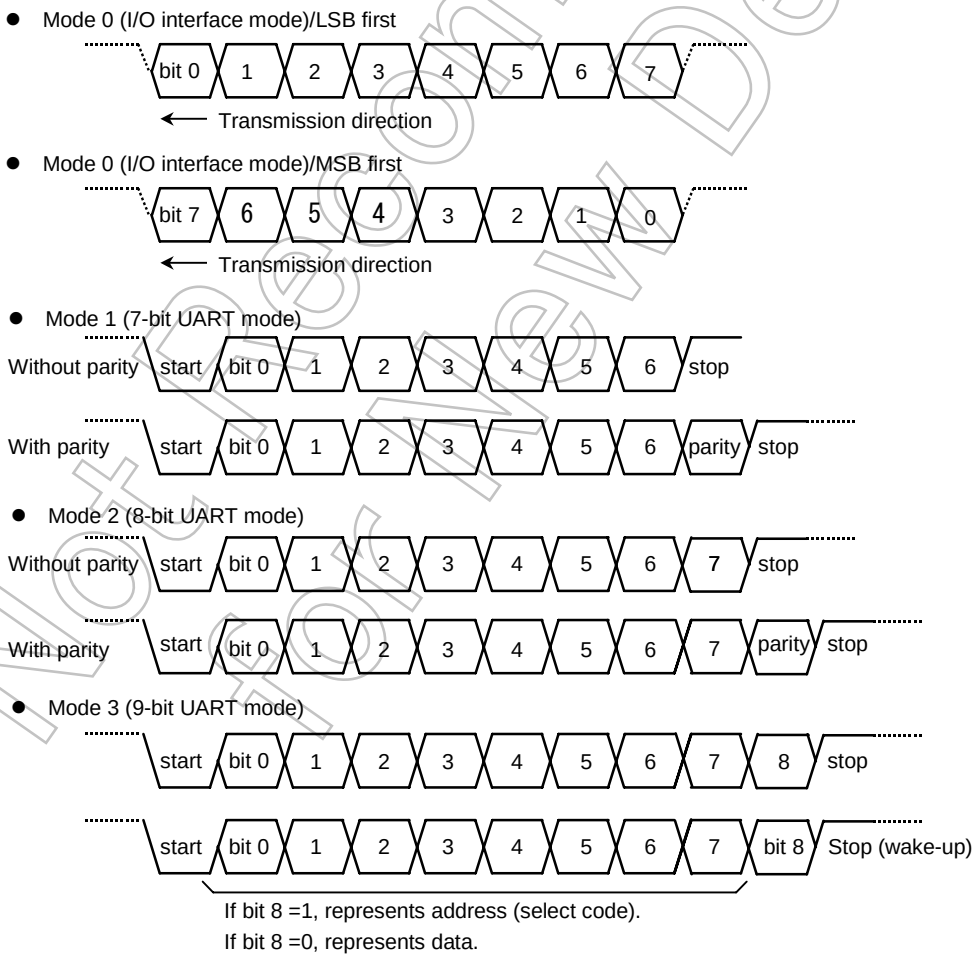


Fig. 14-1 Data Format

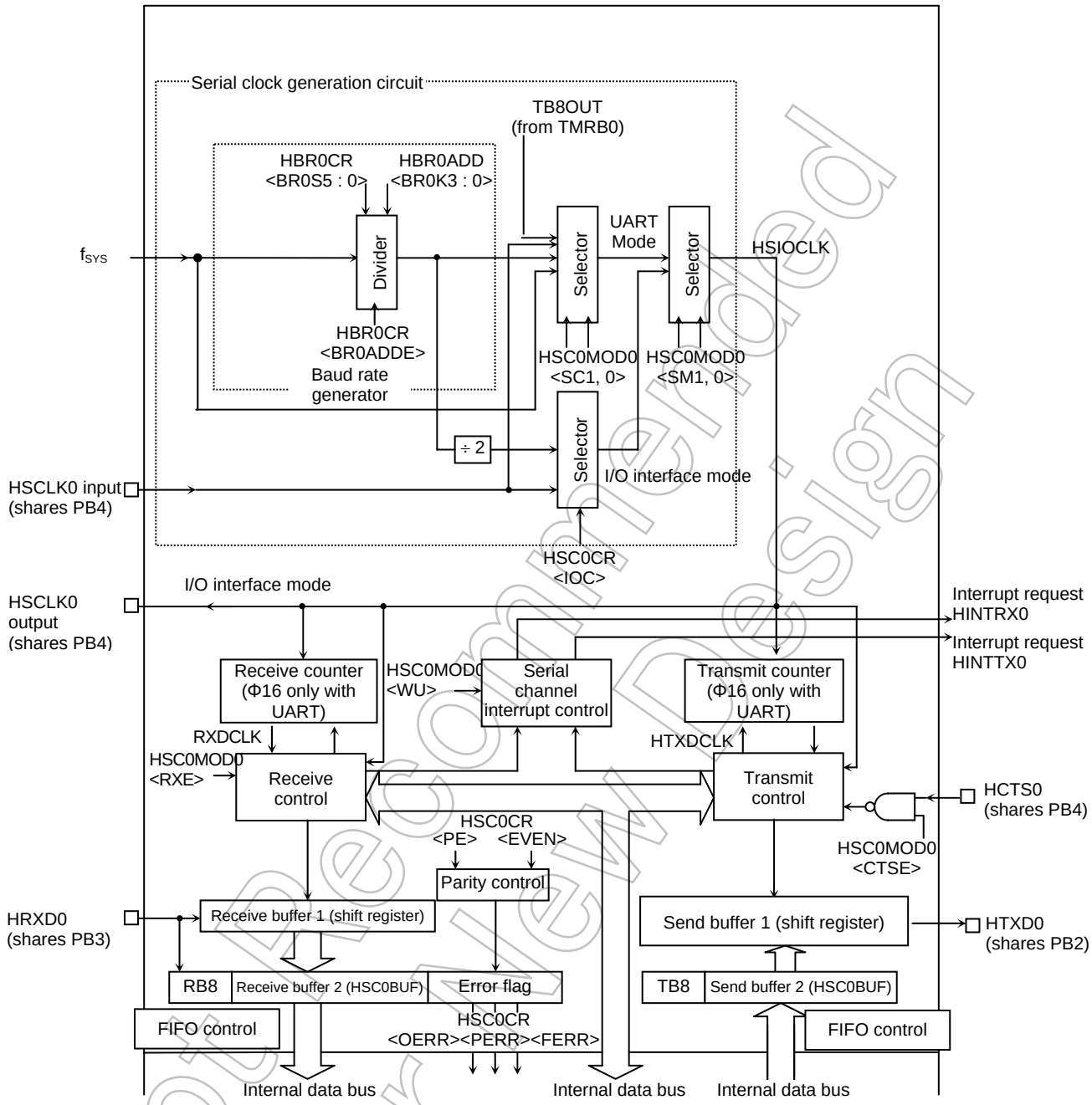


Fig. 14-2 HSIO0 Block Diagram

Note: The baud rate generator cannot be set for "divide by 1."

14.1 Operation of Each Circuit (HSIO Channel 0)

14.1.1 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses 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 2.5 Mbps when fsys is 40 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 40 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.")

Example baud rate setting:

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)

$$\text{Baud rate} = \frac{\text{fsys}}{4} /16$$

$= 40 \times 10^6 / 4 / 16 = 625$ 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)

$$\text{Baud rate} = \frac{\text{Fsys}}{4 + \frac{(16-14)}{16}} /16$$

$= 40 \times 10^6 / (4 + \frac{2}{16}) / 16 = 60.6$ k (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 2/fsys.

   If fsys = 40 MHz, the highest baud rate will be 40 / 4 / 16 = 625 (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 > 6/fsys

   Therefore, when fsys = 40 MHz, the baud rate must be set to a rate lower than 40 / 12 = 3.3 (Mbps).

   When double buffering is not used, it is necessary to satisfy the following relationship:

   External clock input period > 8/fsys

   Therefore, when fsys = 40 MHz, the baud rate must be set to a rate lower than 40 / 16 = 2.5 (Mbps).

14.1.2 High-speed Serial Clock Generation Circuit

This circuit generates basic transmit and receive clocks.

I/O interface mode:

In the HSCLK output mode with the HSC0CR <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 HSC0CR <IOC> set to "1," rising and falling edges are detected according to the HSC0CR <SCLKS> setting to generate the basic clock.

Asynchronous (UART) mode:

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 TMRB8 timer, or the external clock (HSCLKO pin) is selected to generate the basic clock, HSIOCLK.

14.1.3 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.

14.1.4 Receive Control Unit

I/O interface mode:

In the HSCLK output mode with HSC0CR <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 HSC0CR <IOC> set to "1," the serial receive data HRXD0 pin is sampled on the rising or falling edge of HSCLK input depending on the HSC0CR <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.

14.1.5 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 (HSC0BUF). At the same time, the receive buffer full flag (HSC0MOD2 "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), the HINTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (HSCNFCNF <CNFG> = 1 and SCOMOD1<FDPX1:0> = 01), 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 HSC0MOD2 <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 HSC0MOD2 <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 HSC0CR <OERR> flag is insignificant and the operation is undefined. Therefore, before switching from the HSCLK output mode to another mode, the HSC0CR 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 HSC0CR <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 HSC0CR <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 HSC0CR <RB8> is set to "1."

14.1.6 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 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.

14.1.7 Receive FIFO Operation

① I/O interface mode with HSCLK output:

The following example describes the case a 4-byte data stream is received in the half duplex mode:

HSC0RFC<7:6>=01: Clears receive FIFO and sets the condition of interrupt generation.

HSC0RFC<1:0>=00: Sets the interrupt to be generated at fill level 4.

HSC0FCNF <1:0>=10111: Automatically inhibits continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the same as the interrupt generation fill level.

In this condition, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. After receiving 4 bytes, the RXE bit is automatically cleared and the receive operation is stopped (HSCLK is stopped).

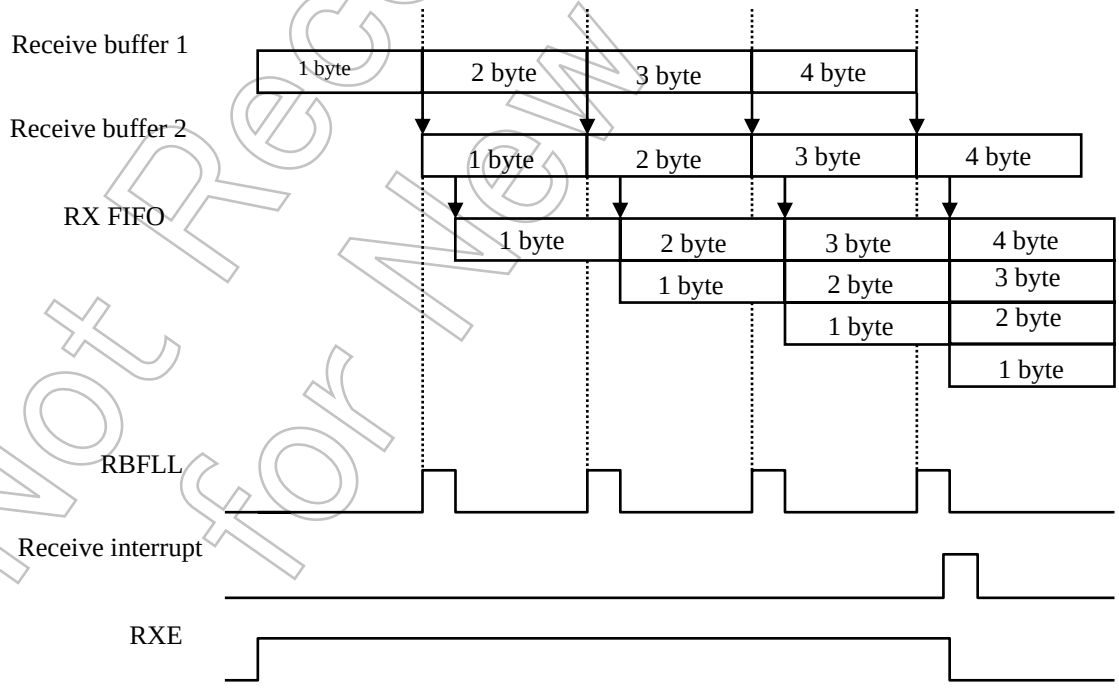


Fig. 14-3 Receive FIFO Operation

② I/O interface mode with HSCLK input:

The following example describes the case a 4-byte data stream is received:

HSC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

HSC0RFC <1:0> = 00: Sets the interrupt to be generated at fill level 4.

HSC0FCNF <1:0> = 10101: Automatically allows continued reception after reaching the fill level. The number of bytes to be used in the receive FIFO is the maximum allowable number.

In this condition, 4-byte data reception can be initiated along with the input clock by setting the half duplex transmission mode and writing "1" to the RXE bit. When the 4-byte data reception is completed, the receive FIFO interrupt will be generated.

Note that preparation for the next data reception can be managed in this setting, i.e., the next 4-byte data can be received before data is fully read from the FIFO.

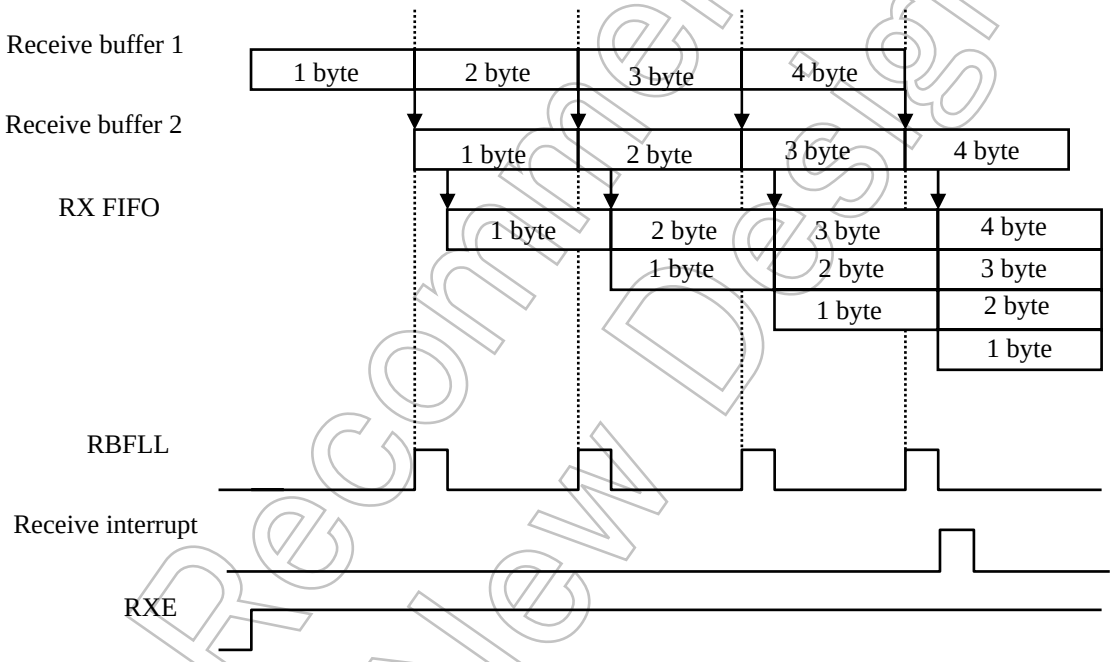


Fig. 14-4 Receive FIFO Operation

14.1.8 Transmit Counter

The transmit counter is a 4-bit binary counter used in the asynchronous communication (UART) mode. It is counted by HSIOCLK as in the case of the receive counter and generates a transmit clock (TXDCLK) on every 16th clock pulse.

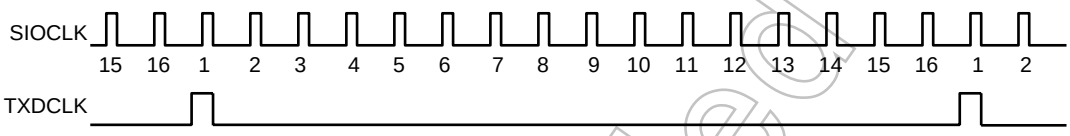


Fig. 14-5 Transmit Clock Generation

14.1.9 Transmit Control Unit

I/O interface mode:

In the HSCLK output mode with HSC0CR <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 HSC0CR <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 HSC0CR <SCLKS> setting.

Asynchronous (UART) mode:

When the CPU writes data to the send buffer, the sending of data begins on the rising edge of the next HTXDCLK and a send shift clock (HTXDSFT) is generated.

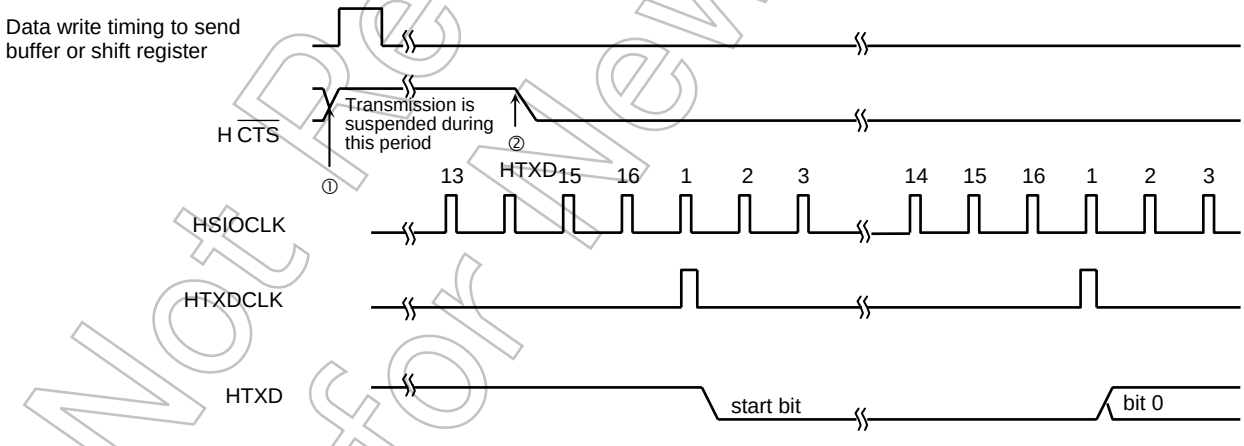
Handshake function

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. 14-6 Handshake Function



(Note) ① **If the H$\overline{\text{CTS}}$ 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 HTXDCLK clock after H$\overline{\text{CTS}}$ is set to "L."**

Fig. 14-7 H$\overline{\text{CTS}}$ (Clear to Send) Signal Timing

14.1.10 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 (HSC0MOD2). If double buffering is enabled, data written to send buffer 2 (HSCOBUF) is moved to send buffer 1 (shift register).

If the transmit FIFO has been disabled (HSCOFCNF <CNFG> = 0 or 1 and <FDPX1:0> = 01), the HINTTX interrupt is generated at the same time and the send buffer empty flag <TBEMP> of HSC0MOD2 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 (HSC0CR) <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 HSC0CR <PERR> flag is insignificant. In this case, the operation is undefined. Therefore, to switch from the HSCLK output mode to another mode, HSC0CR 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 HINTTX0 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 so the transmit FIFO is not configured.

14.1.11 Transmit FIFO Buffer

In addition to the double buffer function already described, data may be stored using the transmit FIFO buffer. By setting <HCNFG> of the HSC0FCNF register and <FDPX1:0> of the HSC0MOD1 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.

14.1.12 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:

HSC0TFC <7:6> = 01: Clears transmit FIFO and sets the condition of interrupt generation

HSC0TFC <1:0> = 00: Sets the interrupt to be generated at fill level 0.

HSC0FCNF <1:0> = 01011: Inhibits continued transmission after reaching the fill level.

In this condition, data transmission can be initiated by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the send buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.

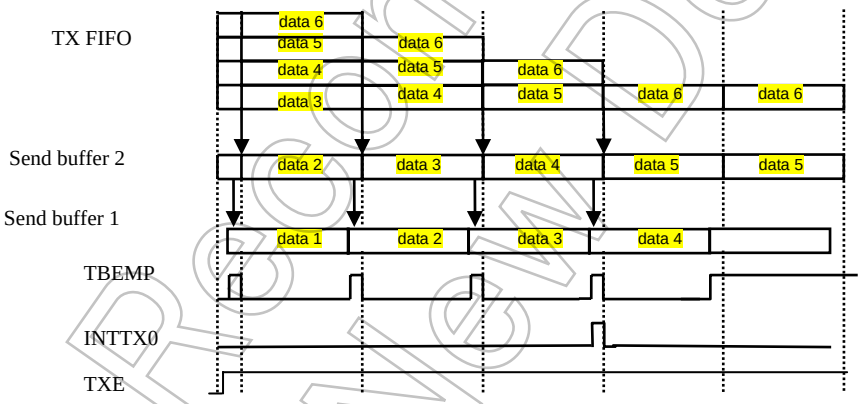


Fig. 14-8 Transmit FIFO Operation

② I/O interface mode with HSCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

HSC0TFC <1:0> = 01: Clears the transmit FIFO and sets the condition of interrupt generation.

HSC0TFC <7:2> = 000000: Sets the interrupt to be generated at fill level 0.

HSC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the send buffer, the transmit FIFO interrupt is generated.

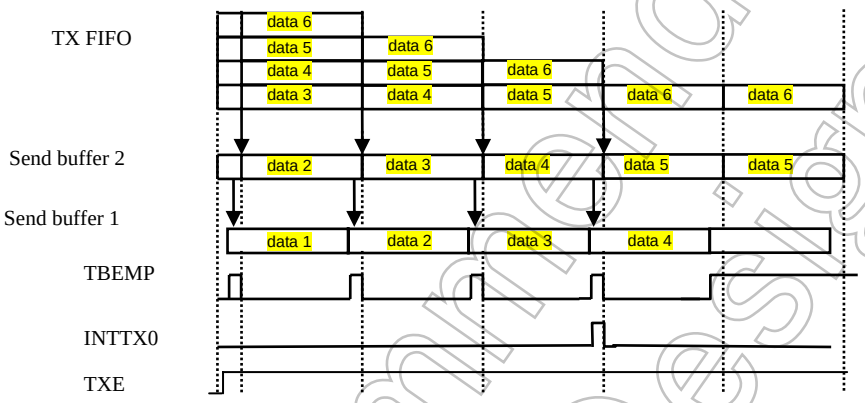


Fig. 14-9 Transmit FIFO Operation

14.1.13 Parity Control Circuit

If the parity addition bit <PE> of the serial control register HSC0CR 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 HSC0CR 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 HSC0MOD 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 HSC0CR register. If there is any difference, a parity error occurs and the <PERR> flag of the HSC0CR register is set.

In the I/O interface mode, the HSC0CR <PERR> flag functions as an under-run error flag, not as a parity flag.

14.1.14 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 HSC0CR

   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 HSC0CR 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 HSC0MOD2 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 HSC0CR 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, HSC0MOD2, 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

14.1.15 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 HSC0MOD2 serial mode control register 2. Don't switch the direction when data is being transferred.

14.1.16 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 HSC0MOD2 register.

14.1.17 Status Flag

If the double buffer function is enabled (HSC0MOD2 <WBUF> = "1"), the bit 6 flag <RBFLL> of the HSC0MOD2 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 (HSC0MOD2 <WBUF> = "1"), the bit 7 flag <TBEMP> of the HSC0MOD2 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.

14.1.18 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

14.1.19 software reset

Software reset is HSC0MOD2 <SWRST1:0>“10” → “01”

HSC0MOD0 **<** RXE **>**、HSC0MOD1<TXE>，HSC0MOD2 **<** TBEMP **>**, **<** RBFLL **>**, **<** TXRUN **>**、HSC0CR **<** OERR **>**、 **<** PERR **>**、 **<** FERR **>** and internal circuit is initialized.

Other states are maintained.

14.1.20 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)	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 HSC0MOD0<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.

14.2 Register Description (Only for Channel 0)

LITTLE / BIG

HSC0MOD0 (0xFFFF_E80E) (0xFFFF_E80D)

	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 TB0OUT 01: Baud rate generator 10: Internal f_{SYS} clock 11: External clock (HSCLK0 input)	

Note) In the I/O interface mode, the serial control register (HSC0CR) is used for clock selection.

→ Wakeup function

	9-bit UART	Other mode
0	Interrupt when received	don't care
1	Interrupt at RB8=1	

→ Handshake function ($\overline{\text{CTS}}$ pin) enable

0	Disable (transmission is always allowed)
1	Enable

Note) With <RXE> set to "0," set each mode register (HSC0MOD0, HSC0MOD1 and HSC0MOD2). Then set <RXE> to "1."

- **The registers must be byte accessed in setting them.**

Fig. 14-10 Serial Mode Control Register 0 (for HSIO0, HSC0MOD0)

HSC0MOD1 (0xFFFF_E805) (0xFFFF_E806)

LITTLE BIG

	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 000: None 100: 8SCLK 001: 1SCLK 101:16SCLK 010: 2SCLK 110: 32SCLK 011: 4SCLK 111: 64SCLK			Write "0."

Fig. 14-11 Serial Mode Control Register 1 (for HSIO0, HSC0MOD1)

<SINT2:0>: Specifies the interval time of continuous transmission when double buffering or/and FIFO is enabled in the I/O interface mode. This parameter is invalid for the UART mode.

<TXE>: This bit enables transmission and is valid for all the transfer modes. If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<FDPX1:0>: Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<I2S0>: Specifies the Idle mode operation.

- **The registers must be byte accessed in setting them.**

LITTLE / BIG

HSC0MOD2 (0xFFFF_E806) (0xFFFF_E805)

	7	6	5	4	3	2	1	0
bit Symbol	TBEMP	RBFLL	TXRUN	SBLEN	DRCHG	WBUF	SWRST1	SWRST0
Read/Write	R/W						W	W
After reset	1	0	0	0	0	0	0	0
Function	Send buffer empty flag 0: full 1: Empty	Receive buffer full flag 0: Empty 1: full	In transmission flag 0: Stop 1: Start	Stop bit 0: 1-bit 1: 2-bit	Setting transfer direction 0: LSB first 1: MSB first	W-buffer 0: Disable 1: Enable	Soft reset Overwrite "01" on "10" to reset	

<SWRST1:0>: Overwriting "01" in place of "10" generates a software reset. When this software reset is executed, the mode register parameters HSC0MOD0 <RXE>, HSC0MOD1<TXE>, HSC0MOD2 <TBEMP>, <RBFLL>, and <TXRUN>, control register parameters HSC0CR <OERR>, <PERR>, and <FERR>, and their internal circuits are initialized.

<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 regardless of the <WBUF> setting.

<DRCHG>: Specifies the direction of data transfer in the I/O interface mode. In the UART mode, it is fixed to LSB first.

<TXRUN>: This is a status flag to show that data transmission is in progress.
When this bit is set to "1," it indicates that data transmission operation is in progress. If it is "0," the bit 7 <TBEMP> is set to "1" to indicate that the transmission has been fully completed and the same <TBEMP> is set to "0" to indicate that the send buffer contains some data waiting for the next transmission.

<RBFLL>: This is a flag to show that the receive double buffers are full. When a receive operation is completed and received data is moved from the receive shift register to the receive double buffers, this bit changes to "1" while reading this bit changes it to "0."
If double buffering is disabled, this flag is insignificant.

<TBEMP>: This flag shows that the send double buffers are empty. When data in the send double buffers is moved to the send shift register and the double buffers are empty, this bit is set to "1." Writing data again to the double buffers sets this bit to "0."
If double buffering is disabled, this flag is insignificant.

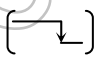
<SBLEN>: This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

(Note) While data transmission is in progress, any software reset operation must be executed twice in succession.

- **The registers must be byte accessed in setting them.**

Fig. 14-12 Serial Mode Control Register

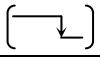
HSC0CR
LITTLE (0xFFFF_E80D)
BIG (0xFFFF_E80E)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to "0" when read)			R/W	
After reset	0	0	0	0	0	0	0	0
Function	Receive data Bit 8	Parity 0: Odd 1: Even	Add parity 0: Disable 1: Enable	1: Error			0: HSCLK0  1: HSCLK0 	0: Baud rate generator 1: HSCLK0 pin input
				Overrun	Parity/ under-run	Framing		

I/O interface input clock selection

0	Baud rate generator
1	HSCLK0 pin input

Edge selection for HSCLK0 input operation

0	Data send/receive at rising edges of HSCLK0	
1	Data send/receive at falling edges of HSCLK0	

Framing error flag
Parity error/under-run error flag
Overrun error flag
} Cleared to "0" when read

Add/check even parity

0	Odd parity
1	Even parity

(Note) Any error flag is cleared when read.

- **The registers must be byte accessed in setting them.**

Fig. 14-13 Serial Control Register (for HSIO0, HSC0CR)

LITTLE / BIG — HBR0CR (0xFFFF_E80F) (0xFFFF_E80C)

	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"					

- **The registers must be byte accessed in setting them.**

LITTLE / BIG — HBR0ADD (0xFFFF_E804) (0xFFFF_E807)

	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			

Setting divide ratio of the baud rate generator

	HBR0CR<HR0DDE> = 1		HBR0CR<HR0DDE> = 0
HBR0CR <HBR0S3:0> / HBR0DD <HR03:0>	000000(N = 64) ~ 0001(N = 1)	000010(N = 2) ~ 111111(N = 63)	000001(N = 1) (ONLY UART) ~ 111111 (N = 63), 000000 (N = 64)
0000	Disable	Disable	
0001(K = 1) ~ 1111(K = 15)	Disable	$N+\frac{(16-K)}{16}$ Division	Divide by N

Fig. 14-14 Baud Rate Generator Control (for HSIO0, HBR0CR, HBR0ADD)

(Note 1) In the UART mode, the division ratio "1" of the baud rate generator can be specified only when the "N + (16 - K)/16" division function is not used. In the I/O interface mode, "divide by 1" must not be specified as a divisor for the baud rate generator.

(Note 2) To use the "N + (16 - K)/16" division function, be sure to set HBR0CR <BR0ADDE> to "1" after setting the K value (K = 1 to 15) to HBR0ADD <BR3K3:0>. However, don't use the "N + (16 - K)/16" division function when HBR0CR <BR0S5:0> is set to either "000000" or "000001" (N = 64 or 1).

(Note 3) The "N + (16 - K)/16" division function can only be used in the UART mode. In the I/O interface mode, the "N + (16 - K)/16" division function must be disabled (prohibited) by setting HBR0CR <BR0ADDE> to "0."

LITTLE / BIG

HSC0BUF
(0xFFFF_E800)
(0xFFFF_E803)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Send BUFFER＋FIFO)
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receive BUFFER＋FIFO)

Note: HSC0BUF works as a send buffer for WR operation and as a receive buffer for RD operation.

Fig. 14-15 FIFO Configuration Register

LITTLE / BIG

HSC0FCNF
(0xFFFF_E80C)
(0xFFFF_E80F)

	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

<CNFG>: If enabled, the HSCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows:

<FDPX1:0> = 01 (Half duplex RX) ---- 4-byte RX FIFO

<FDPX1:0> = 10 (Half duplex TX) ---- 4-byte TX FIFO

<FDPX1:0> = 11 (Full duplex) ---- 2-Byte RX FIFO + 2-Byte TX FIFO

<RXTXCNT>:0 The function to automatically disable RXE/TXE bits is disabled.

1: If enabled, the HSCOMOD1 <FDPX1:0> is used to set as follows:

<FDPX1:0> = 01 (Half duplex RX) ------ When the RX FIFO is filled up to the specified number of valid bytes, RXE is automatically set to "0" to inhibit further reception.

<FDPX1:0> = 10 (Half duplex TX) ------ When the TX FIFO is empty, TXE is automatically set to "0" to inhibit further transmission.

<FDPX1:0> = 11 (Full duplex) ---------- When either of the above two conditions is satisfied, TXE/RXE are automatically set to "0" to inhibit further transmission and reception.

<RFIE>: When RX FIFO is enabled, receive interrupts are enabled or disabled by this parameter.

<TFIE>: When TX FIFO is enabled, transmit interrupts are enabled or disabled by this parameter.

<RFST>: When RX FIFO is enabled, the number of RX FIFO bytes to be used is selected.

0: The maximum number of bytes of the FIFO configured 4 bytes when <FDPX1:0> = 01 (Half duplex RX) and 2 bytes for <FDPX1:0> = 11 (Full duplex)

1: Same as the fill level for receive interrupt generation specified by SC0RFC <RIL1:0>.

(Note 1) Regarding TX FIFO, the maximum number of bytes being configured is always available. The available number of bytes is the bytes already written to the TX FIFO.

- **The registers must be byte accessed in setting them.**

Fig. 14-16 Receive FIFO Control Register

HSC0RFC
LITTLE (0xFFFF_E808)
BIG (0xFFFF_E80B)

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS	–	–	–	–	RIL1	RIL0
Read/Write	W	R						
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 01: 1byte 10: 2byte 11: 3byte Note: RIL1 is ignored when FDPX1:0 = 11 (full duplex)	

0: An interrupt is generated when the specified fill level is reached.
1: An interrupt is generated when the specified fill level is reached or if the specified fill level has been exceeded at the time data is read.

Fig. 14-17 Transmit FIFO Configuration Register

HSC0TFC
LITTLE (0xFFFF_E809)
BIG (0xFFFF_E80A)

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS	–	–	–	–	TIL1	TIL0
Read/Write	W	R						
After reset	0	0	0	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 when FDPX1:0 = 11 (full duplex).	

0: An interrupt is generated when the specified fill level is reached.
1: An interrupt is generated when the specified fill level is reached or if the level is lower than the specified fill level at the time new data is written.

- **The registers must be byte accessed in setting them.**

Fig. 14-18 Receive FIFO Status Register

HSC0RST
(0xFFFF_E80A)
(0xFFFF_E809)

LITTLE
BIG

	7	6	5	4	3	2	1	0
bit Symbol	ROR					RLVL2	RLVL1	RLVL0
Read/Write	R	R						
After reset	0	0	0	0	0	0	0	0
Function	RX FIFO Overrun 1: Generated Cleared when read	Always reads "0."				Status of RX FIFO fill level 000: Empty 001: 1Byte 010: 2Byte 011: 3Byte 100: 4Byte		

Fig. 14-19 Transmit FIFO Status Register

HSC0TST
(0xFFFF_E80B)
(0xFFFF_E808)

LITTLE
BIG

	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	0
Function	TX FIFO Under run 1: Generated Cleared by writing to FIFO	Always reads "0."				Status of TX FIFO fill level 000: Empty 001: 1Byte 010: 2Byte 011: 3Byte 100: 4Byte		

Fig. 14-20 HSIO Enable Register

HSC0EN
(0xFFFF_E807)
(0xFFFF_E804)

LITTLE
BIG

	7	6	5	4	3	2	1	0
bit Symbol	–		–	–	–	–	–	SIOE
Read/Write	R							R/W
After reset	0	0	0	0	0	0	0	0
Function	Always reads "0."							HSIO operation 0: Disable 1: Enable

<SIOE>: It specifies HSIO operation. When HSIO operation is disabled, the clock will not be supplied to the HSIO module except for the register part and thus power dissipation can be reduced (other registers cannot be accessed for read/write 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. If HSIO is enabled once and then disabled, any register setting is maintained.

- **The registers must be byte accessed in setting them.**

14.3 Operation in Each Mode

14.3.1 Mode 0 (I/O Interface Mode)

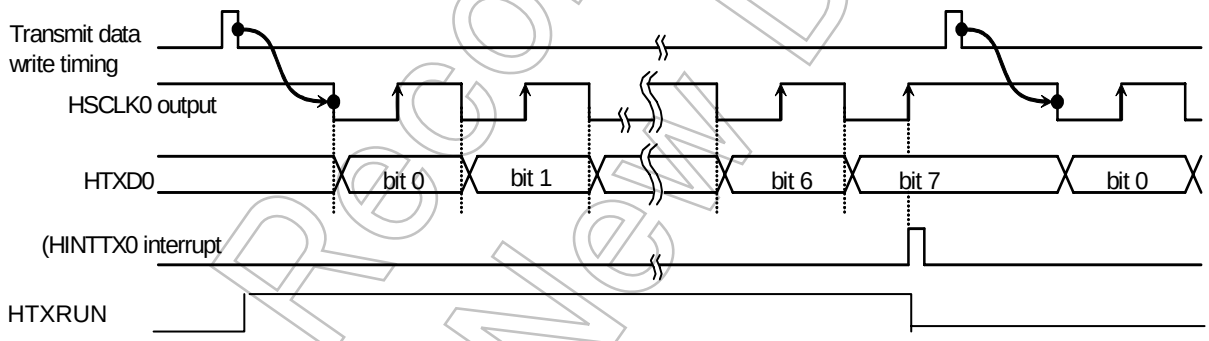
Mode 0 consists of two modes, i.e., the "HSCLK output" mode to output synchronous clock and the "HSCLK input" mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

① Sending data

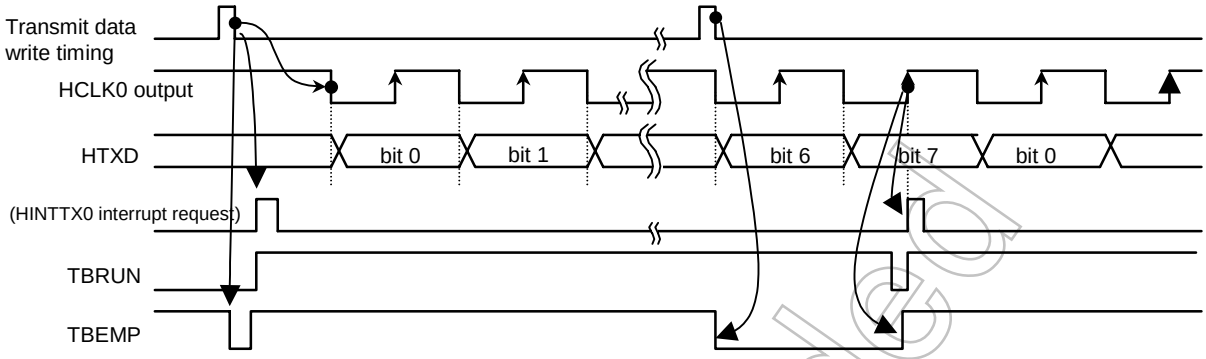
HSCLK output mode

In the HSCLK output mode, if HSC0MOD2<WBUF> is set to "0" and the send double buffers are disabled, 8 bits of data are output from the HXD0 pin and the synchronous clock is output from the HSCLK0 pin each time the CPU writes data to the send buffer. When all data is output, the HINTTX0 interrupt is generated.

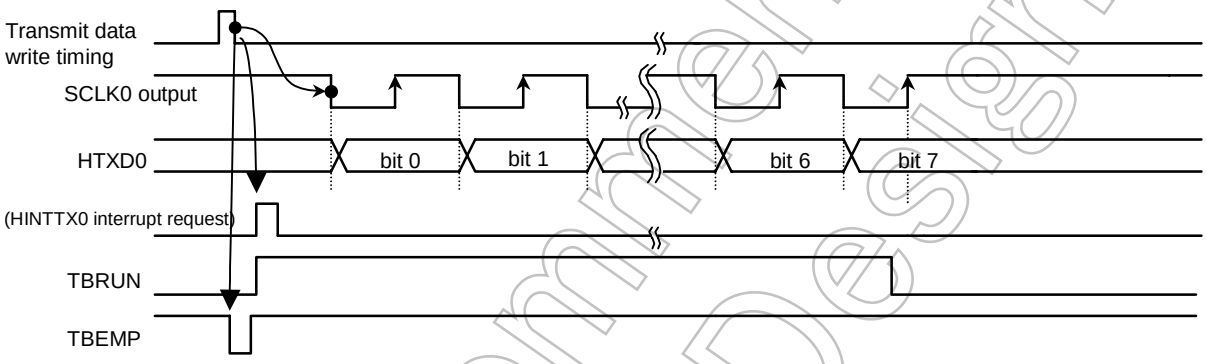
If HSC0MOD2 <WBUF> is set to "1" and the send double buffers are enabled, data is moved from send buffer 2 to send buffer 1 when the CPU writes data to send buffer 2 while data transmission is halted or when data transmission from send buffer 1 (shift register) is completed. When data is moved from send buffer 2 to send buffer 1, the send buffer empty flag HSC0MOD2 <TBEMP> is set to "1," and the HINTTX0 interrupt is generated. If send buffer 2 has no data to be moved to send buffer 1, the HINTTX0 interrupt is not generated and the HSCLK0 output stops.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



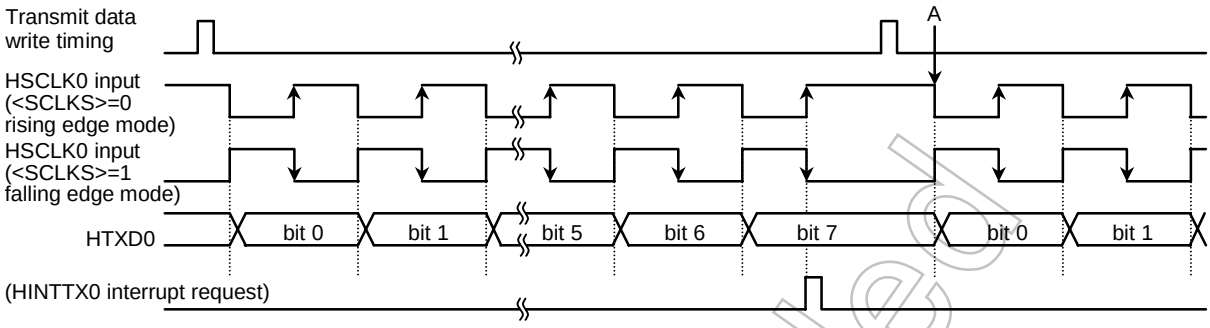
<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 14-21 Send Operation in the I/O Interface Mode (HSCLK0 Output Mode)

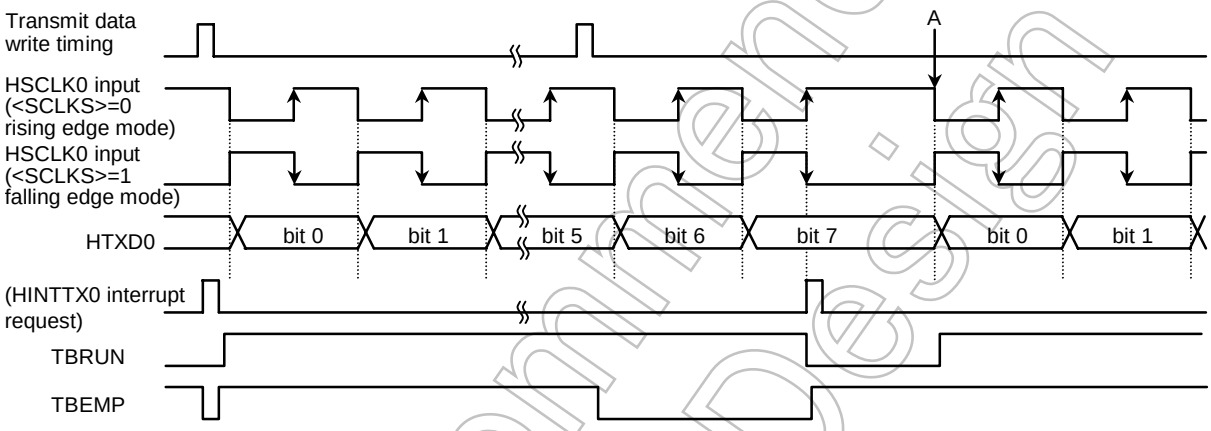
HSCLK input mode

In the HSCLK input mode, if HSC0MOD2 <WBUF> is set to "0" and the send double buffers are disabled, 8-bit data that has been written in the send buffer is output from the HTXD0 pin when the HSCLK0 input becomes active. When all 8 bits are sent, the HINTTX0 interrupt is generated. The next send data must be written before the timing point "A."

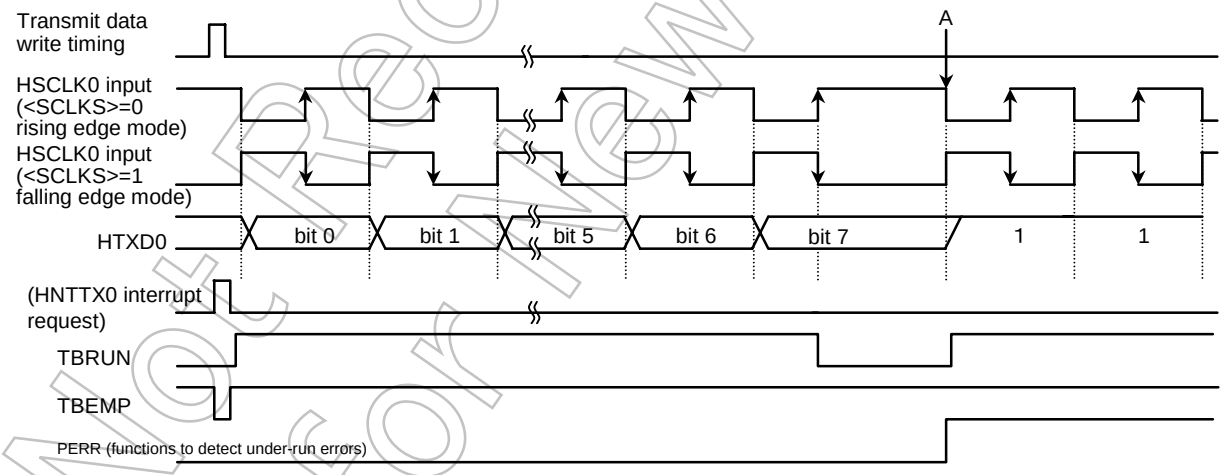
If HSC0MOD2 <WBUF> is set to "1" and the send double buffers are enabled, data is moved from send buffer 2 to send buffer 1 when the CPU writes data to send buffer 2 before the HSCLK0 becomes active or when data transmission from send buffer 1 (shift register) is completed. As data is moved from send buffer 2 to send buffer 1, the send buffer empty flag HSC0MOD2 <TBEMP> is set to "1" and the HINTTX0 interrupt is generated. If the HSCLK0 input becomes active while no data is in send buffer 2, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is sent.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 14-22 Send Operation in the I/O Interface Mode (HSCLK0 Input Mode)

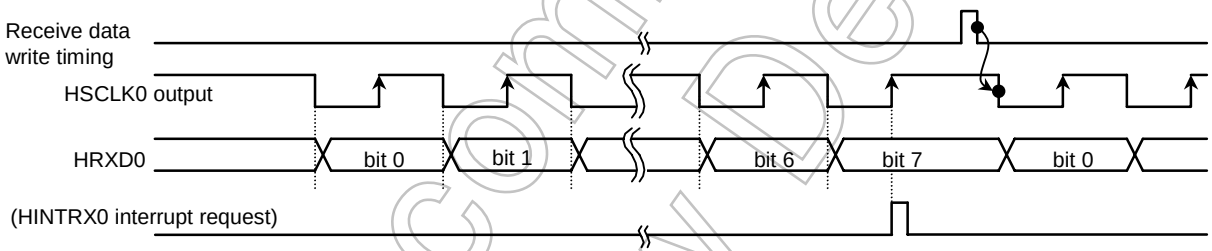
② Receiving data

HSCLK output mode

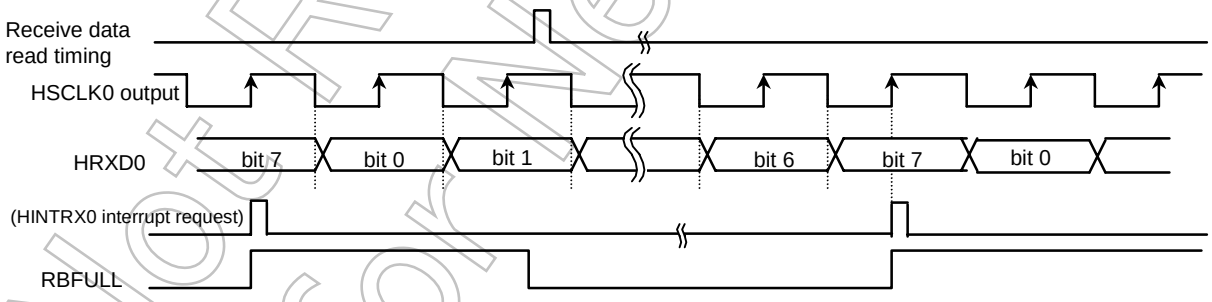
In the HSCLK output mode, if HSC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the HSCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the HINTRX0 interrupt is generated.

The first HSCLK output can be started by setting the receive enable bit HSC0MOD0 <RXE> to "1." If the receive double buffering is enabled with HSC0MOD2 <WBUF> set to "1," the first frame received is moved to receive buffer 2 and receive buffer 1 can receive the next frame successively. As data is moved from receive buffer 1 to receive buffer 2, the receive buffer full flag HSC0MOD2 <RBFULL> is set to "1" and the HINTRX0 interrupt is generated.

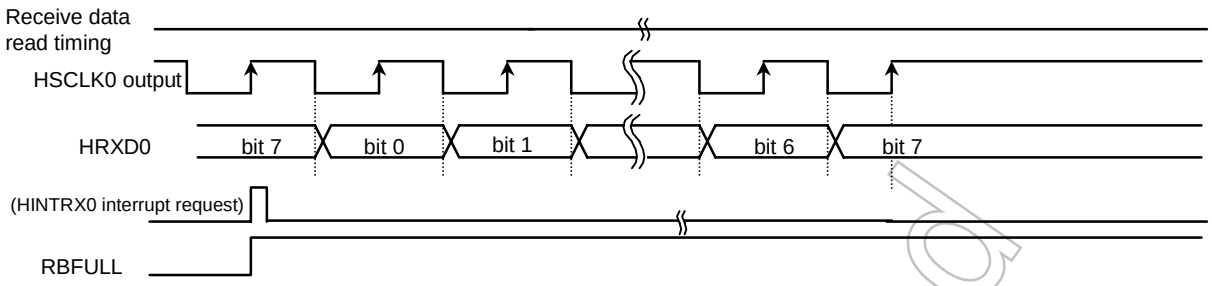
While data is in receive buffer 2, if CPU/DMAC cannot read data from receive buffer 2 in time before completing reception of the next 8 bits, the HINTRX0 interrupt is not generated and the HSCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 2 and thus the HINTRX0 interrupt is generated and data reception resumes.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if data is read from buffer 2)



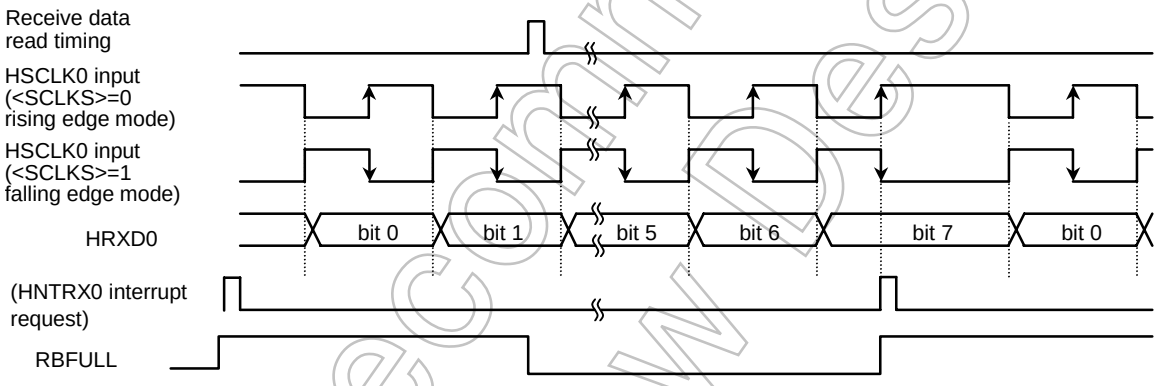
<WBUF> = "1" (if double buffering is enabled) (if data cannot be read from buffer 2)

Fig. 14-23 Receive Operation in the I/O Interface Mode (HSCLK0 Output Mode)

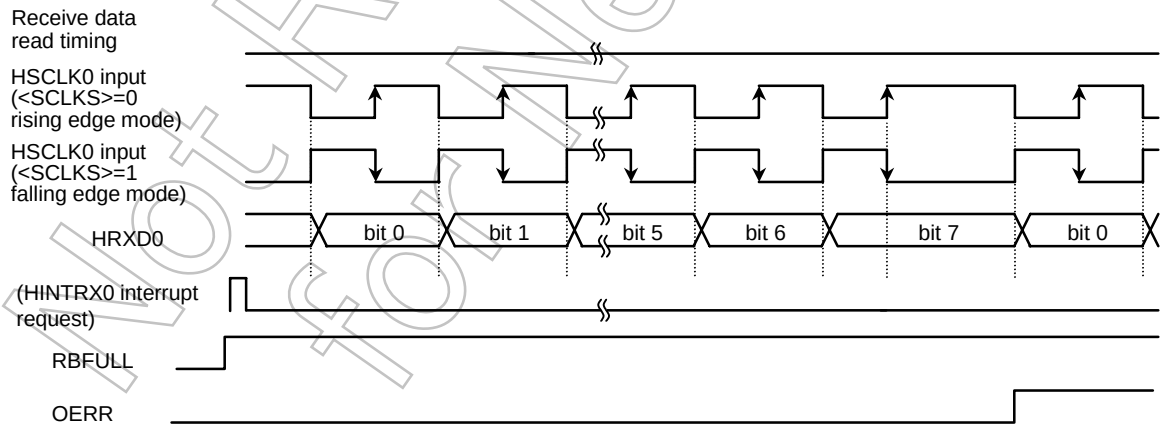
HSCLK input mode

In the HSCLK input mode, since receive double buffering is always enabled, the received frame can be moved to receive buffer 2 and receive buffer 1 can receive the next frame successively.

The HINTRX0 receive interrupt is generated each time received data is moved to received buffer 2.



If data is read from buffer 2



If data cannot be read from buffer 2

Fig. 14-24 Receive Operation in the I/O Interface Mode (HSCLK0 Input Mode)

(Note) To receive data, HSC0MOD <RXE> must always be set to "1" (receive enable) regardless of the HSCLK input or output mode.

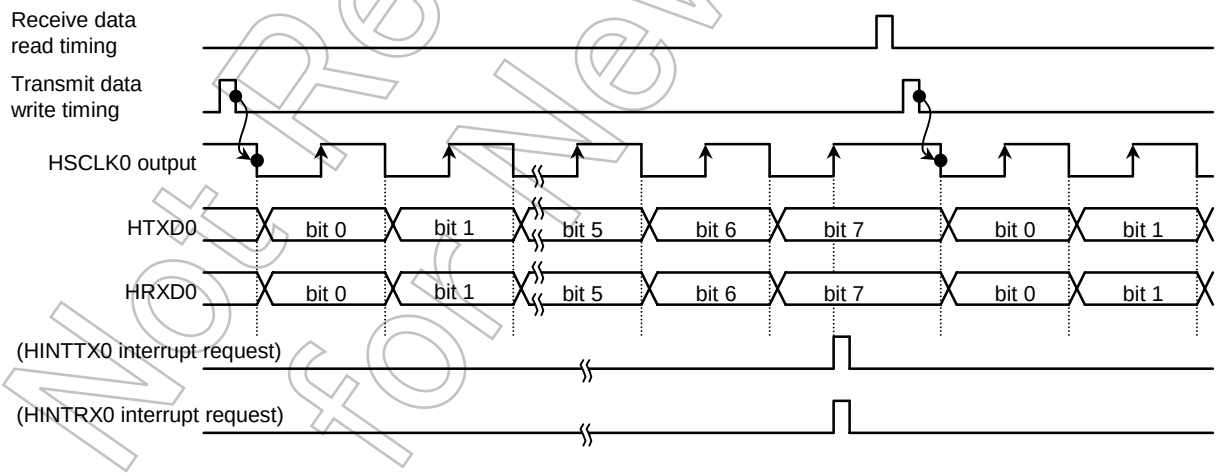
③ Send and receive (full-duplex)

The full-duplex mode is enabled by setting bit 6 <FDPX0> of the serial mode control register 1 (HSC0MOD1) to "1."

HSCLK output mode

In the HSCLK output mode, if HSC0MOD2 <WBUF> is set to "0" and both the send and receive double buffers are disabled, HSCLK is output when the CPU writes data to the send buffer. Subsequently, 8 bits of data are shifted into receive buffer 1 and the HINTRX0 receive interrupt is generated. Concurrently, 8 bits of data written to the send buffer are output from the HTXD0 pin, the HINTTX0 send interrupt is generated when transmission of all data bits has been completed. Then, the HSCLK output stops. In this, the next round of data transmission and reception starts when the data is read from the receive buffer and the next send data is written to the send buffer by the CPU. The order of reading the receive buffer and writing to the send buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

If HSC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, HSCLK is output when the CPU writes data to the send buffer. Subsequently, 8 bits of data are shifted into receive buffer 1, moved to receive buffer 2, and the HINTRX0 interrupt is generated. While 8 bits of data is received, 8 bits of transmit data is output from the HTXD0 pin. When all data bits are sent out, the HINTTX0 interrupt is generated and the next data is moved from the send buffer 2 to send buffer 1. If send buffer 2 has no data to be moved to send buffer 1 (HSC0MOD2 <TBEMP> = 1) or when receive buffer 2 is full (HSC0MOD2 <RBFULL> = 1), the HSCLK clock is stopped. When both conditions are satisfied, i.e., receive data is read and send data is written, the HSCLK output is resumed and the next round of data transmission is started.



<WBUF> = "0" (if double buffering is disabled)

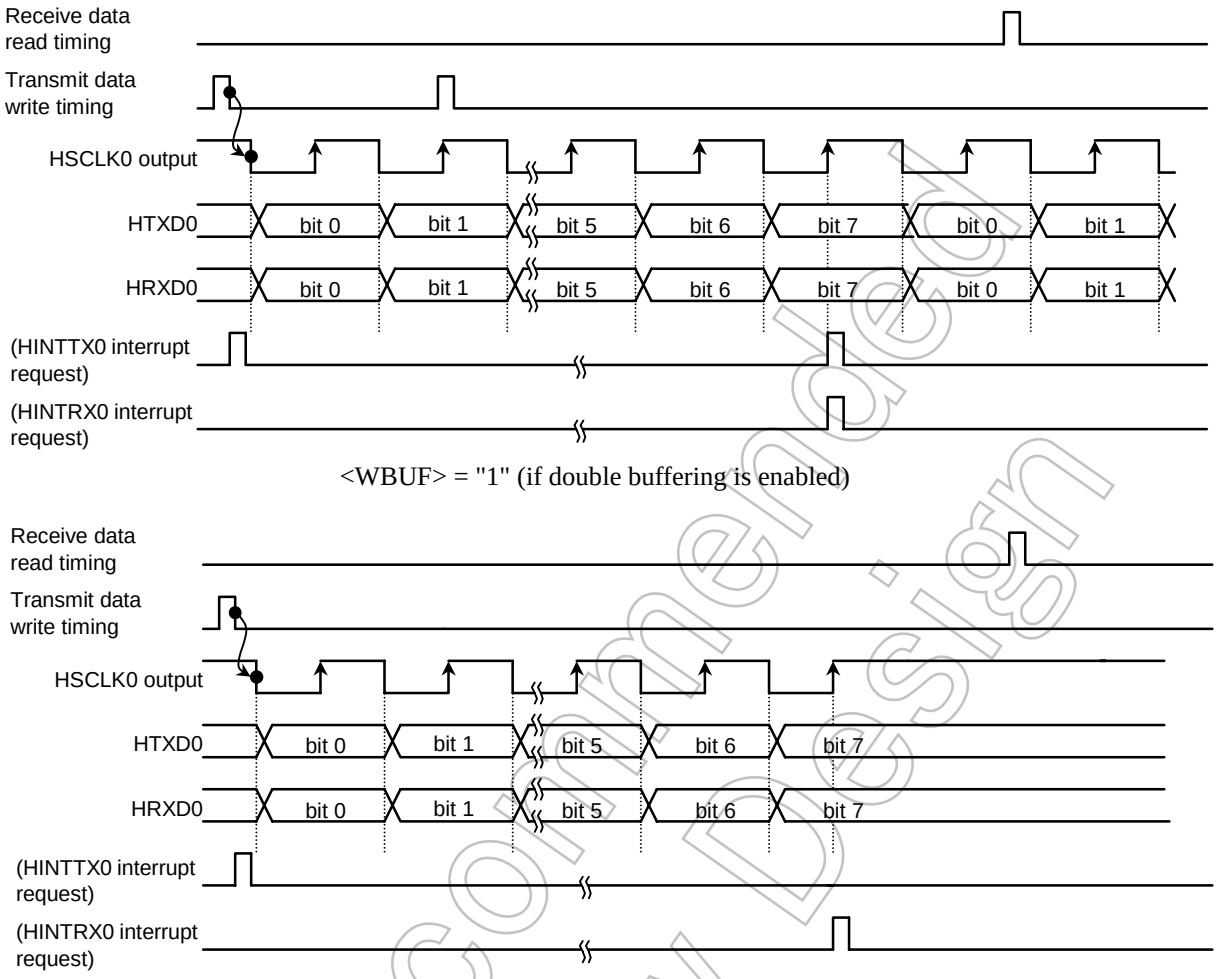
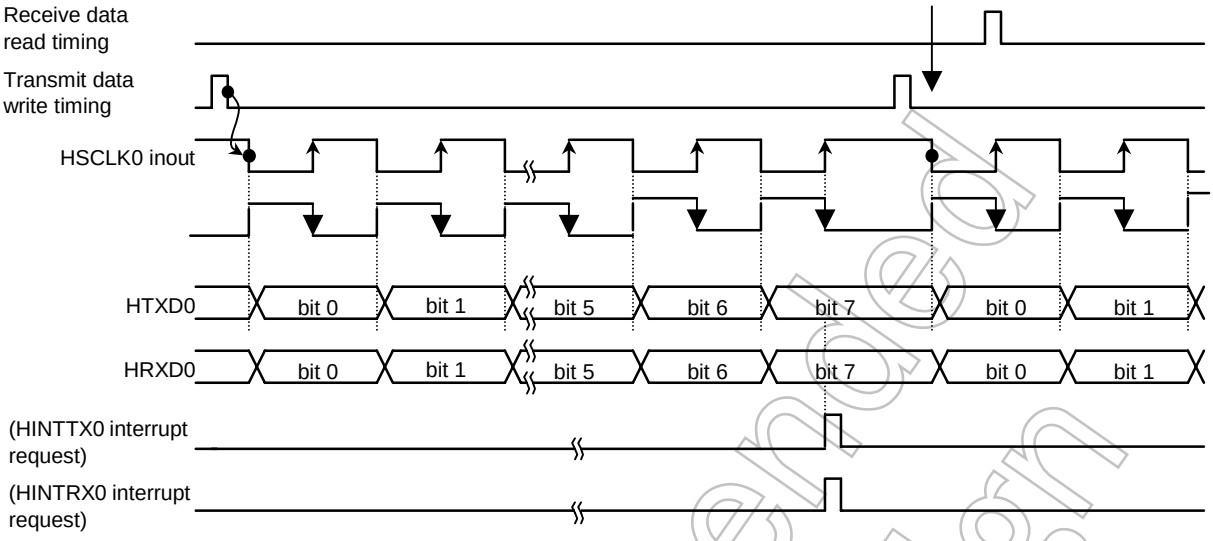


Fig. 14-25 Send/Receive Operation in the I/O Interface Mode (HSCLK0 Output Mode)

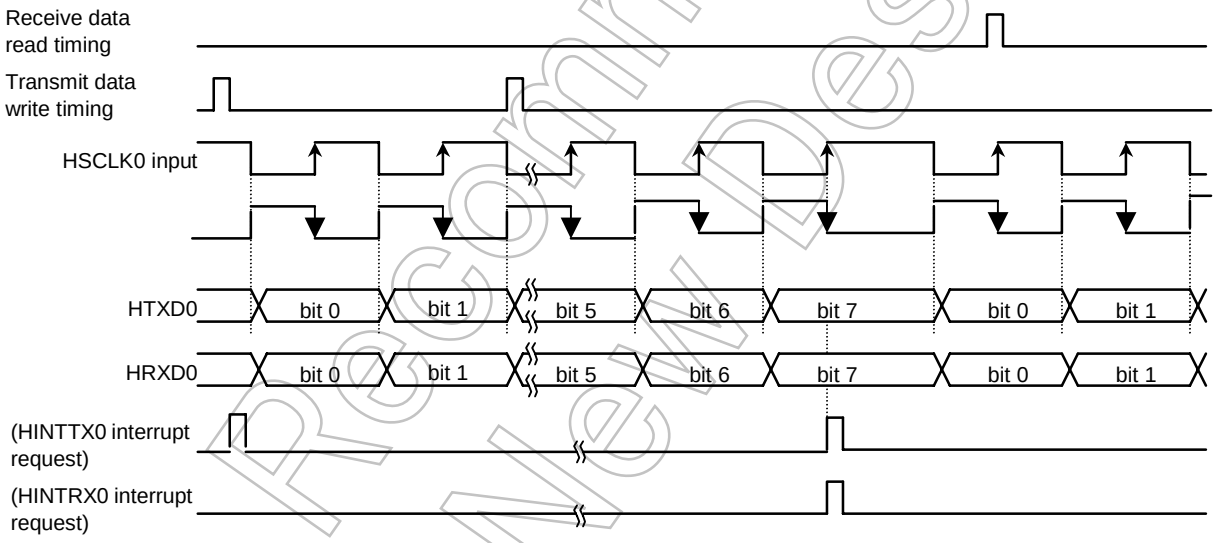
HSCLK input mode

In the HSCLK input mode with HSC0MOD2 <WBUF> set to "0" and the send double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the send buffer is output from the HTXD0 pin and 8 bits of data is shifted into the receive buffer when the HSCLK input becomes active. The HINTTX0 interrupt is generated upon completion of data transmission and the HINTRX0 interrupt is generated at the instant the received data is moved from receive buffer 1 to receive buffer 2. Note that transmit data must be written into the send buffer before the HSCLK input for the next frame (data must be written before the point A). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

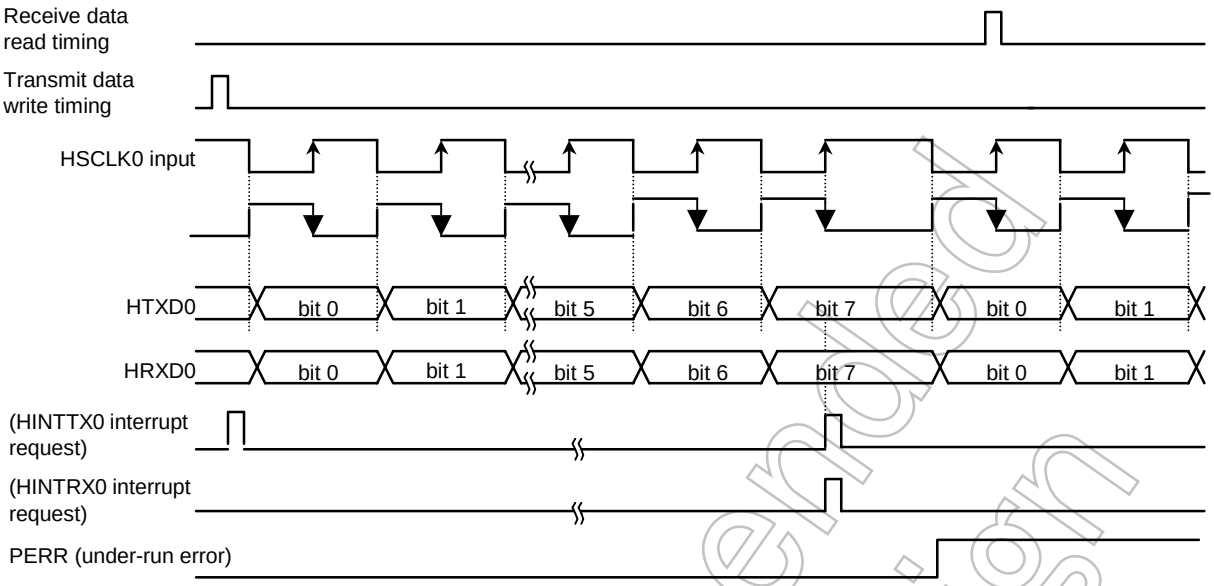
If HSC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt HINTRX0 is generated at the timing send buffer 2 data is moved to send buffer 1 after completing data transmission from send buffer 1. At the same time, the 8 bits of data received is shifted to buffer 1, moved to receive buffer 2, and the HINTRX0 interrupt is generated. Upon the HSCLK input for the next frame, transmission from send buffer 1 (in which data has been moved from send buffer 2) is started while receive data is shifted into receive buffer 1 simultaneously. If data in receive buffer 2 has not been read when the last bit of the frame is received, an overrun error occurs. Similarly, if there is no data written to send buffer 2 when HSCLK for the next frame is input, an under-run error occurs.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (no errors)



<WBUF> = "1" (if double buffering is enabled) (error generation)

Fig. 14-26 Send/Receive Operation in the I/O Interface Mode (HSCLK0 Input Mode)

14.3.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (HSC0MOD <SM1, 0>) to "01."

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (HSC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the HSC0CR <EVEN> bit. The length of the stop bit can be specified using HSC0MOD2<SBLEN>.

14.3.3 Mode 2 (8-bit UART Mode)

The 8-bit UART mode can be selected by setting HSC0MOD0 <SM1:0> to "10." In this mode, parity bits can be added and parity enable/disable is controlled using HSC0CR <PE>. If <PE> = "1" (enabled), either even or odd parity can be selected using HSC0CR <EVEN>.

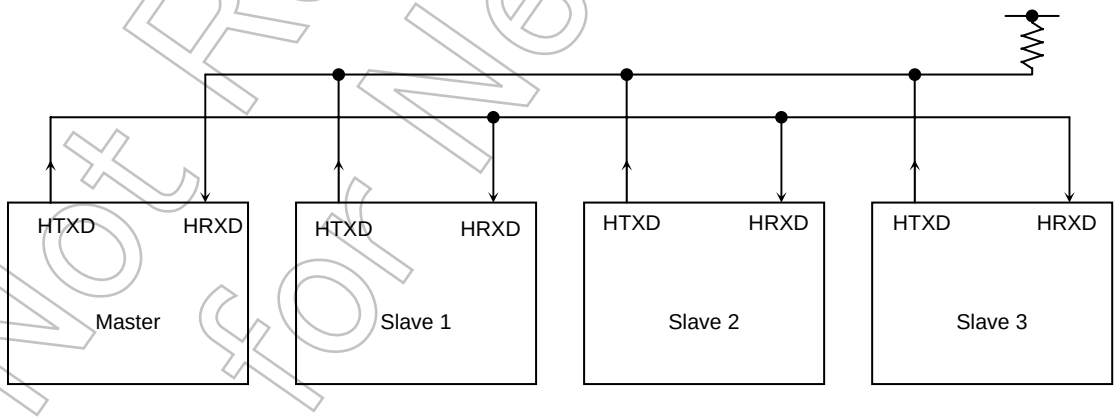
14.3.4 Mode 3 (9-bit UART)

The 9-bit UART mode can be selected by setting HSC0MOD0 <SM1:0> to "11." In this mode, parity bits must be disabled (HSC0CR <PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (HSC0MOD0) for transmit data and it is stored in bit 7 <RB8> of the serial control register HSC0CR upon receiving data. When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from HSC0BUF. The stop bit length can be specified using HSC0MOD2 <SBLEN>.

Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit HSC0MOD0 <WU> to "1." In this case, the interrupt HINTRX0 will be generated only when HSC0CR <RB8> is set to "1."



(Note) The HTXD pin of the slave controller must be set to the open drain output mode using the ODE register.

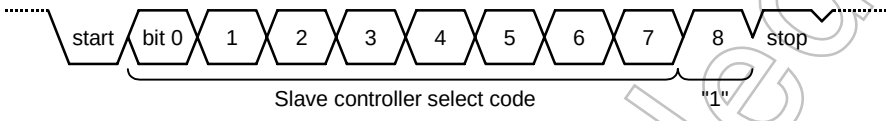
Fig. 14-27 Serial Links to Use Wake-up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

② Set HSC0MOD <WU> to "1" for the slave controllers to make them ready to receive data.

③ The master controller is to send a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1."



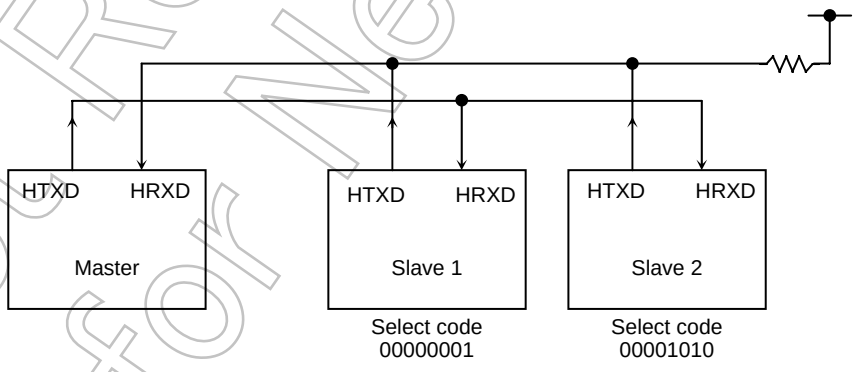
④ Every slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0."

⑤ The master controller transmits data to the designated slave controller (the controller of which HSC0MOD <WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0."

⑥ The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is set to "0" and thus no interrupt (HINTRX0) is generated.

Also, the slave controller with the <WU> bit set to "0" can transmit data to the master controller to inform that the data has been successfully received.

Example setting: Using the internal clock f_{SYS} as the transfer clock, two slave controllers are serially linked as follows:



15. Serial Bus Interface (SBI)

The TMP19A43 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 PC5 (SDA) and PC7 (SCL). In the clock-synchronous 8-bit SIO mode, the SBI is connected to external devices via PC7 (SCK), PC5 (SO) and PC6 (SI).

The following table shows the programming required to put the SBI in each operating mode.

	PCODE <PCODE1:0>	PCCR <PC7C, PC6C, PC5C>	PCFC <PC7F, PC6F, PC5F>
I2C bus mode	11	X11	011
Clock-synchronous 8-bit SIO mode	XX	101 (clock output) 001 (clock input)	111

X: Don't care

15.1 Configuration

The configuration is shown in Fig. 15.1.

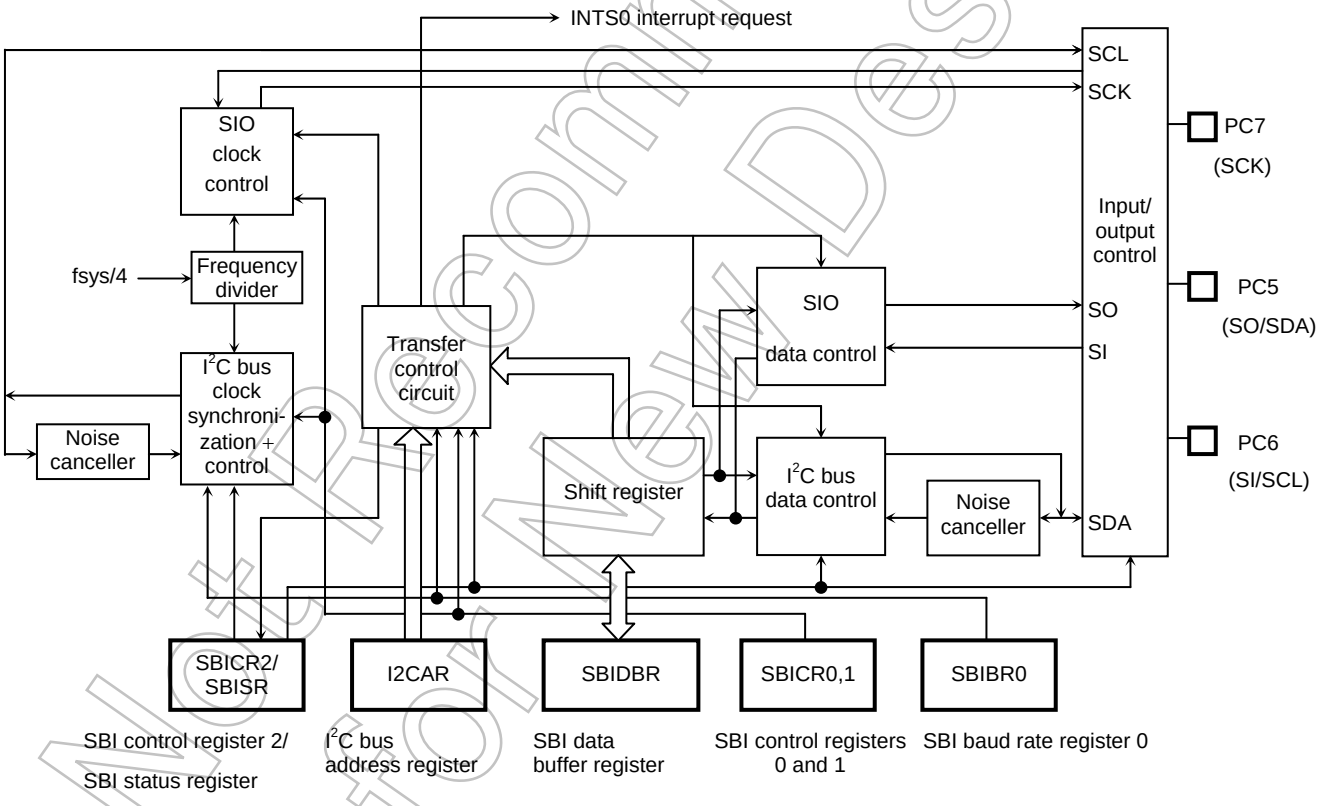


Fig. 15.1 SBI Block Diagram

15.2 Control

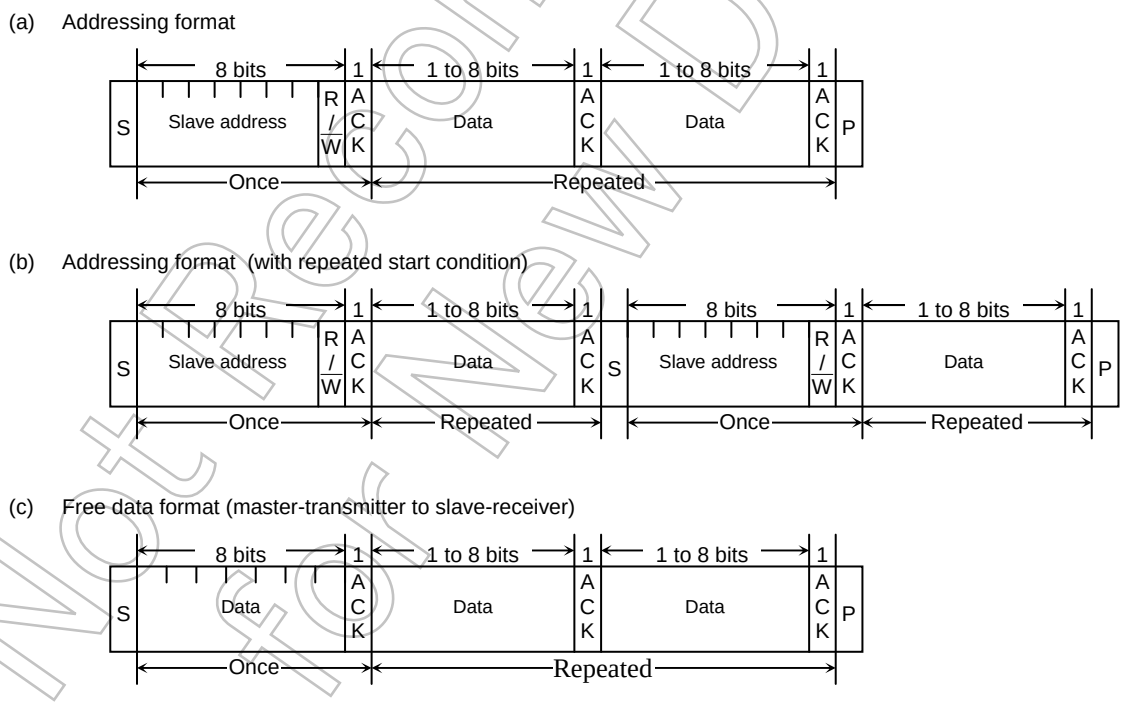
The following registers control the serial bus interface and provide its status information for monitoring.

- Serial bus interface control register 0 (SBICR0)
- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register (SBISR)
- Serial bus interface baud rate register 0 (SBIBR0)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to "3.12.4 Control in the I^2C Bus Mode" and "3.12.7 Control in the Clock-synchronous 8-bit SIO Mode."

15.3 I^2C Bus Mode Data Formats

Fig. 15.1 shows the data formats used in the I^2C bus mode.



Note: S: Start condition
R/$\overline{W}$: Direction bit
ACK: Acknowledge bit
P: Stop condition

Fig. 15.2 I^2C Bus Mode Data Formats

15.4 Control Registers in the I²C Bus Mode

The following registers control the serial bus interface (SBI) in the I²C bus mode and provide its status information for monitoring.

Serial bus interface control register 0

SBICR0 (0xFFFF_F257)

	7	6	5	4	3	2	1	0
Bit symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	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. 15.3 I²C Bus Mode Register

Serial bus interface control register 1

SBICR1 (0xFFFF_F250)

	7	6	5	4	3	2	1	0
Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	R/W			R/W	R	R/W		R/W
After reset	0	0	0	0	1	0	0	1
Function	Select the number of bits per transfer (Note 1)			Acknow-ledgment clock 0: Not generate 1: Generate	This can be read as "1."	Select internal SCL output clock frequency (Note 2) and reset monitor		

On writing <SCK2:0>: Select internal SCL output clock frequency

000	n=5	196 kHz
001	n=6	149 kHz
010	n=7	101 kHz
011	n=8	61 kHz
100	n=9	34 kHz
101	n=10	18 kHz
110	n=11	9 kHz
111		reserved

System clock : fsys (=40 MHz)
Clock gear : fc/1
Frequency = $\frac{fsys/2}{2^n + 70}$ [Hz]

On reading <SWRMON>: Software reset status monitor

0	Software reset operation is in progress.
1	Software reset operation is not in progress.

Select the number of bits per transfer

<BC2:0>	When <ACK> = 0		When <ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

(Note 1) Clear <BC2:0> to "000" before switching the operation mode to the clock-synchronous 8-bit SIO mode.

(Note 2) For details on the SCL line clock frequency, refer to "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. 15.4 I^2C Bus Mode Register

Serial bus interface control register 2

SBICR2 (0xFFFF_F253)

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W (Note 1)		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: I²C bus mode 11: (Reserved)		Software reset generation Write "10" followed by "01" to generate a reset.	

Select serial bus interface operating mode (Note 2)

00	Port mode (Serial bus interface output disabled)
01	Clock-synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

(Note 1) Reading this register causes it to function as the SBISR register.

(Note 2) Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "H" level before switching the operating mode from the port mode to the I^2C bus or clock-synchronous 8-bit SIO mode.

(Note 3) Ensure that serial transfer is completed before switching the mode.

Fig. 15.5 I^2C Bus Mode Register

Table 15.1 Base Clock Resolution

@fsys = 40 MHz

Clock gear value <GEAR1:0>	Base clock resolution
00 (fc)	fsys/2^2 (0.1 μs)
01 (fc/2)	fsys/2^3 (0.2 μs)
10 (fc/4)	fsys/2^4 (0.4 μs)
11 (fc/8)	fsys/2^5 (0.8 μs)

Serial bus interface status register

SBISR (0xFFFF_F253)

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/ slave selection monitor 0: Slave 1: Master	Transmit/ receive selection monitor 0: Receive 1: Transmit	I^2C bus state monitor 0: Free 1: Busy	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	Last received bit monitor 0: "0" 1: "1"

Last received bit monitor

0	The last bit received was "0."
1	The last bit received was "1."

Addressed as slave

0	–
1	Addressed as slave or general call detected

Arbitration lost

0	–
1	Arbitration was lost to another master

(Note) Writing to this register causes it to function as SBICR2.

Fig. 15.6 I^2C Bus Mode Register

Serial bus interface baud rate register 0

SBIBR0 (0xFFFF_F254)

	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 "1."	IDLE 0: Stop 1: Operate	This can be read as "1."					Make sure that you write "0." (Note)

Operation in the IDLE mode

0	Stop
1	Operate

(Note) This is read as "1" in the SIO mode.

Serial bus interface data buffer register

SBIDBR (0xFFFF_F251)

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	0							

(Note) Transmit data must be written to this register, with bit 7 being the most-significant bit (MSB).

I^2C bus address register

I2CAR (0xFFFF_F252)

	7	6	5	4	3	2	1	0
Bit symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set the slave address when the SBI acts as a slave device.							Specify address recognition mode

Specify address recognition mode

0	Recognizes the slave address.
1	Does not recognize slave address.

Fig. 15.7 I^2C Bus Mode Register

(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.

Control in the I²C Bus Mode

15.4.1 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.

15.4.2 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.

15.4.3 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.

$t_{LOW} = 2^{n-1}/(fsys/2) + 58/(fsys/2)$

$t_{HIGH} = 2^{n-1}/(fsys/2) + 12/(fsys/2)$

$fscl = 1/(t_{Low} + t_{HIGH})$

$= \dfrac{fsys/2}{2^n + 70}$

SBI0CR1 <SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Fig. 15.8 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^2C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "L" level overrides other masters producing the "H" level on their clock lines. This must be detected and responded by the masters producing the "H" level.

Clock synchronization assures correct data transfer on a bus that has two or more masters.

For example, the clock synchronization procedure for a bus with two masters is shown below.

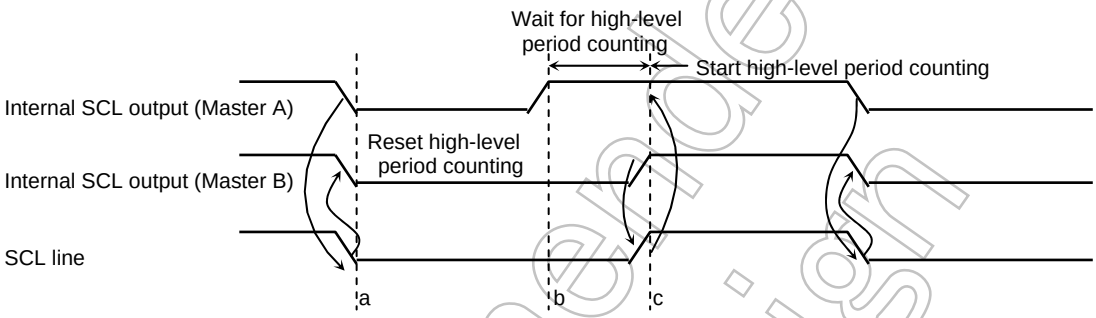


Fig. 15.9 Example of Clock Synchronization

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.

15.4.4 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.

15.4.5 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.

15.4.6 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.

15.4.7 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.

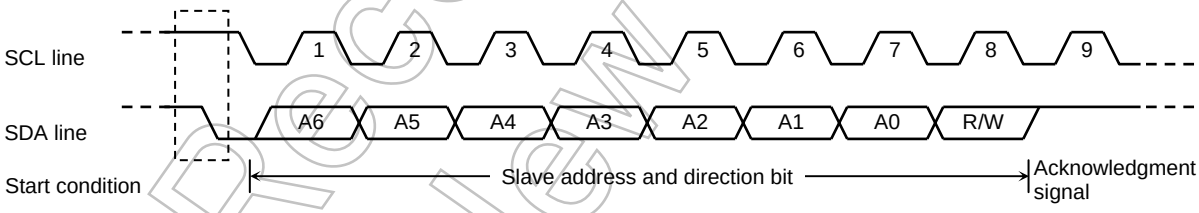


Fig. 15.10 Generating the Start Condition and a Slave Address

When <BB> is "1," writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

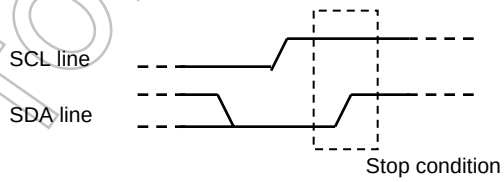


Fig. 15.11 Generating the Stop Condition

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).

15.4.8 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."

15.4.9 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.

15.4.10 Lost-arbitration Detection Monitor

The I^2C bus has the multi-master capability (there are two or more masters on a bus), and requires the bus arbitration procedure to ensure correct data transfer.

A master that attempts to generate the start condition while the bus is busy loses bus arbitration, with no start condition occurring on the SDA and SCL lines. The I^2C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below. Up until point a, Master A and Master B output the same data. At point a, Master A outputs the "L" level and Master B outputs the "H" level. Then Master A pulls the SDA bus line to the "L" level because the line has the wired-AND connection. When the SCL line goes high at point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid. In other words, Master B loses arbitration. Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

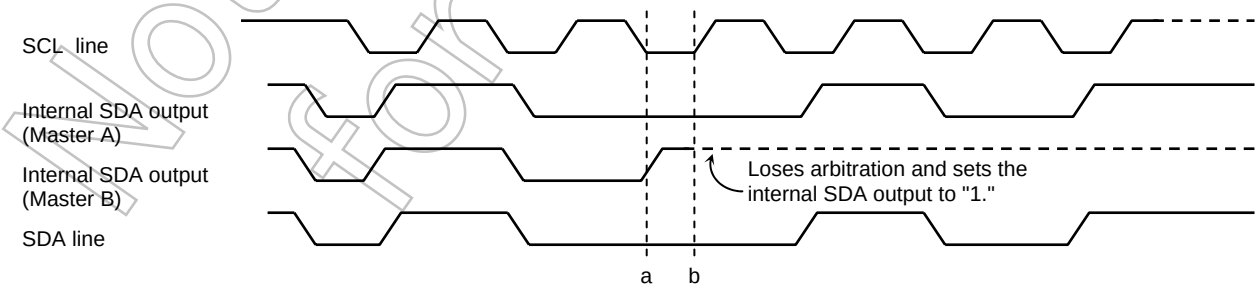


Fig. 15.12 Lost Arbitration

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.

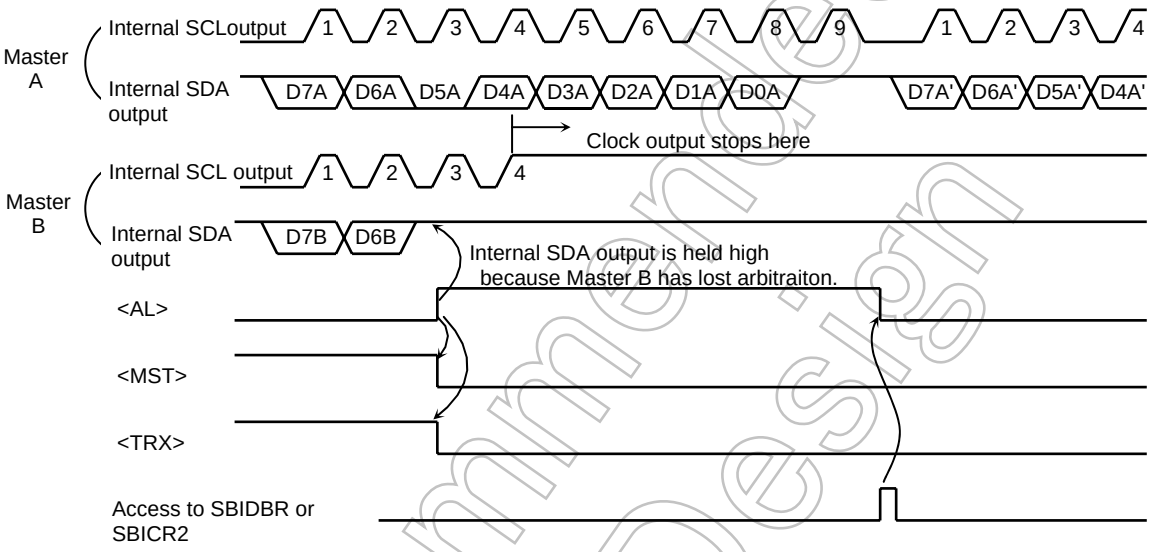


Fig. 15.13 Example of Master B Losing Arbitration (D7A = D7B, D6A = D6B)

15.4.11 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.

15.4.12 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.

15.4.13 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.

15.4.14 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^2C mode to the port mode.

15.4.15 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.

15.4.16 I^2C Bus Address Register (I2CAR)

When the SBI is configured as a slave device, the I2CAR<SA6:0> bit is used to specify a slave address. If I2C0AR <ALS> is set to "0," the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format. If <ALS> is set to "1," the SBI does not recognize a slave address and receives data in the free data format.

15.4.17 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.

15.5 Data Transfer Procedure in the I^2C Bus Mode

15.5.1 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
```

15.5.2 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 ← 0x78                      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.**

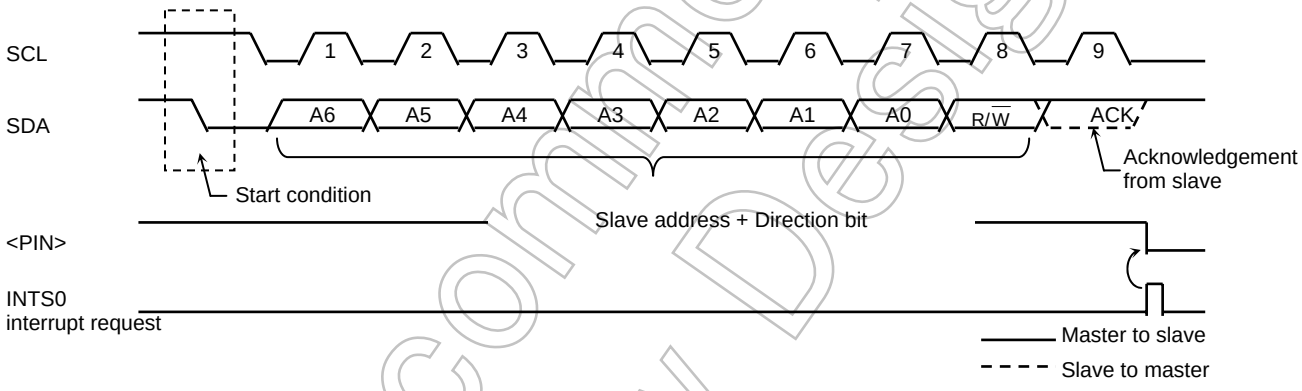


Fig. 15.14 Generation of the Start Condition and a Slave Address

15.5.3 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

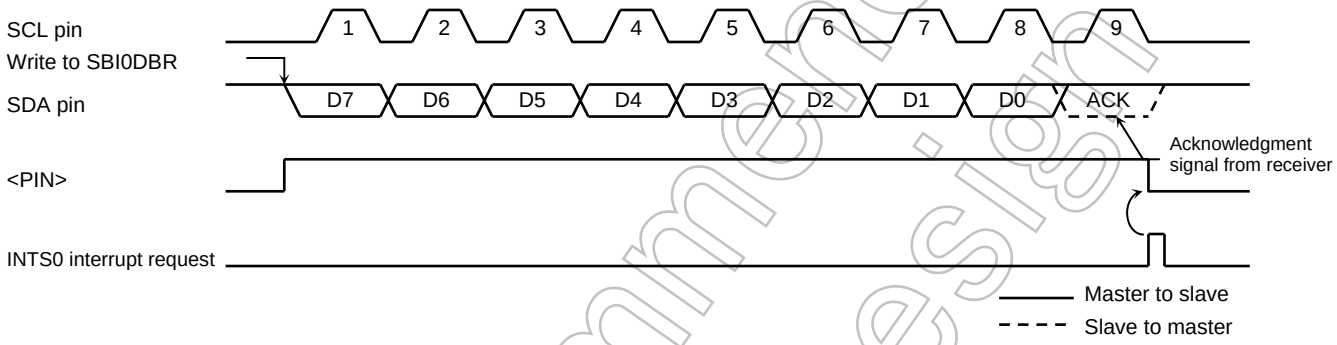


Fig. 15.15 <BC2:0> = "000" and <ACK> = "1" (Transmitter Mode)

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.

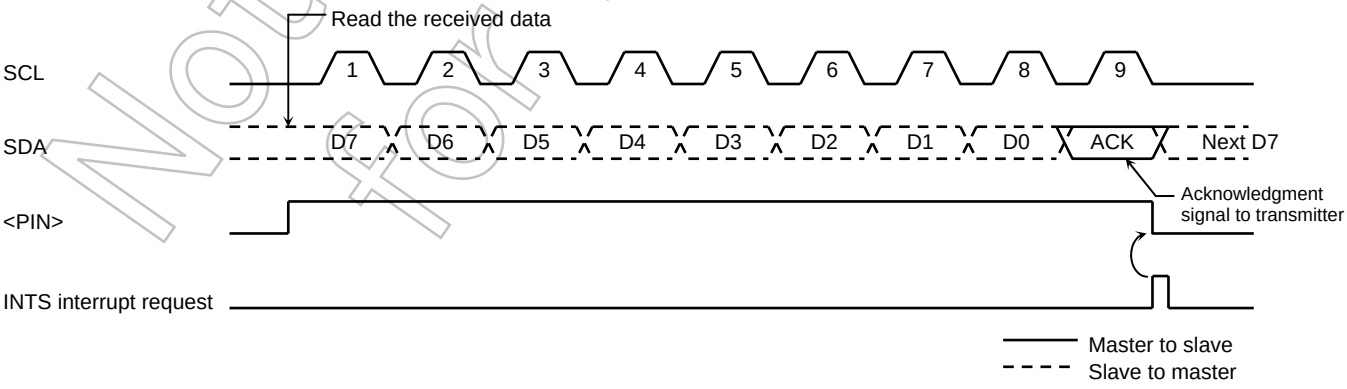


Fig. 15.16 <BC2:0> = "000" and <ACK> = "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be set to "0" immediately before reading the second to last data word. This disables generation of an acknowledgment clock for the last data word. When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC2:0> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer. At this time, the master receiver holds the SDA bus line at the "H" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

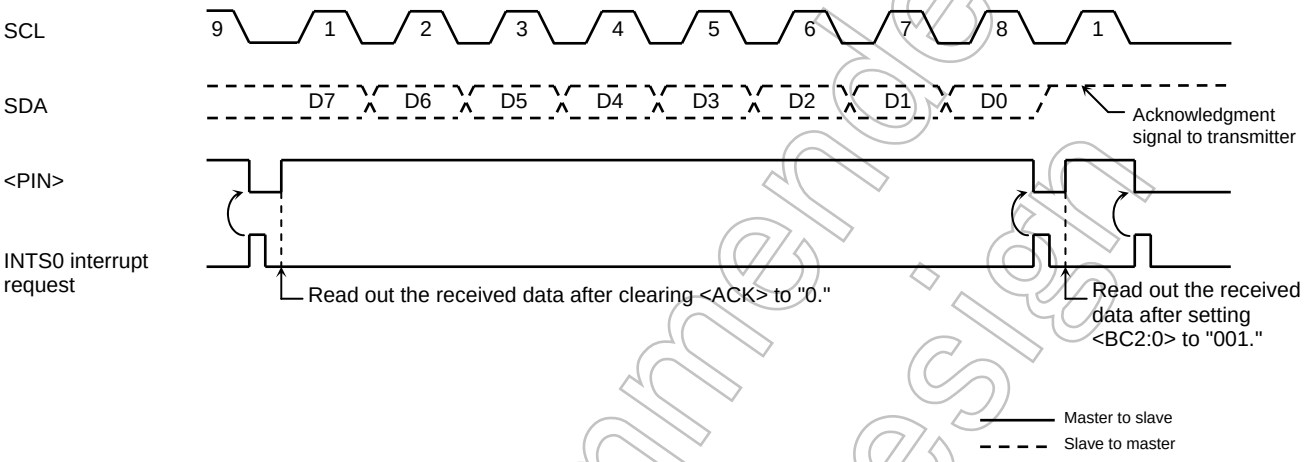


Fig. 15.17 Terminating Data Transmission in the Master Receiver Mode

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 15.1 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 15.1 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration was lost while the slave address was being transmitted, and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC2:0> and write the transmit data into SBI0DBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1," that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0," that means the receiver requires further data. Set the number of bits in the data word to <BC2:0> and write the transmit data to the SBIDBR.
0	1	1	1/0	Arbitration was lost while a slave address was being transmitted, and the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration was lost while a slave address or a data word was being transmitted, and the transfer terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC2:0> and read the received data from SBIDBR.

15.5.4 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.

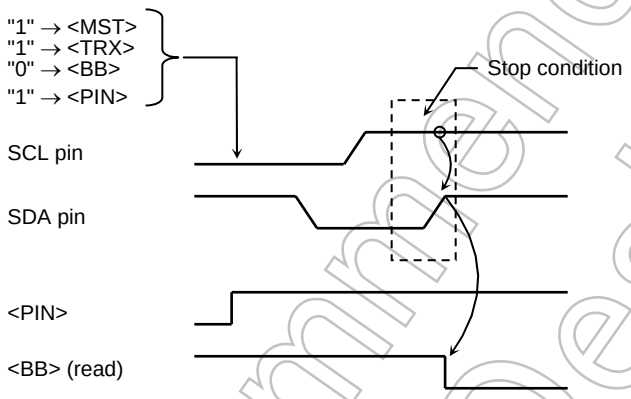


Fig. 15.18 Generating the Stop Condition

15.5.5 Repeated Start Procedure

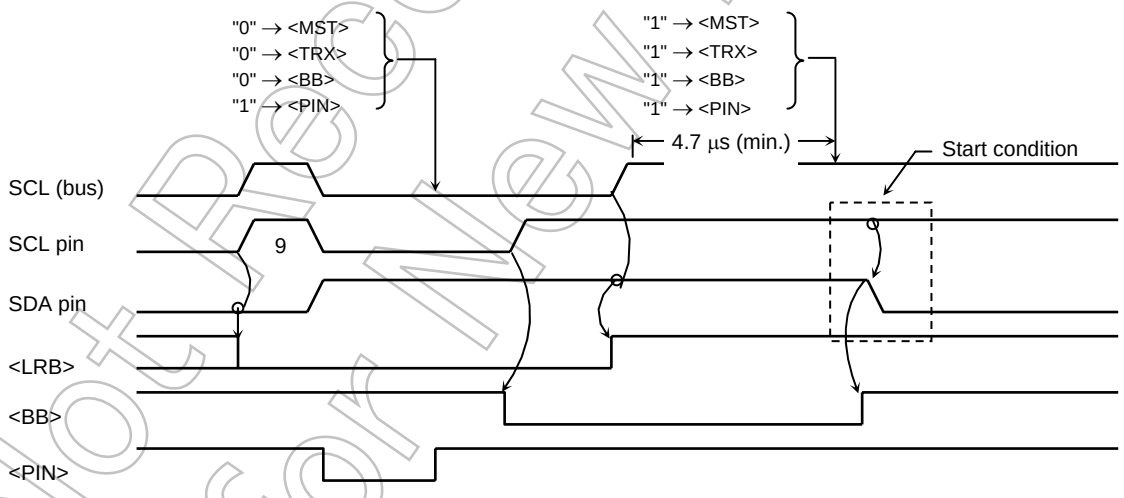
Repeated start is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a repeated start in the master mode is described below.

First, set SBICR2 <MST, TRX, BB> to "0" and write "1" to <PIN> to release the bus. At this time, the SDA pin is held at the "H" level and the SCL pin is released. Because no stop condition is generated on the bus, other devices think that the bus is busy. Then, test SBISR <BB> and wait until it becomes "0" to ensure that the SCL pin is released. Next, test <LRB> and wait until it becomes "1" to ensure that no other device is pulling the SCL bus line to the "L" level. Once the bus is determined to be free this way, use the steps described above in (2) to generate the start condition.

To satisfy the setup time of repeated start, at least 4.7-μs wait period (in the standard mode) must be created by the software after the bus is determined to be free.

```
                    7 6 5 4 3 2 1 0
SBICR2      ← 0 0 0 1 1 0 0 0      Releases the bus.
if SBISR<BB> ≠ 0                   Checks that the SCL pin is released.
Then
if SBISR<LRB> ≠ 1                  Checks that no other device is pulling the SCL pin to the
                                     "L" level.
Then
4.7 μs Wait
SBICR1      ← X X X 1 0 X X X      Selects the acknowledgment mode.
SBIDBR      ← X X X X X X X X      Sets the desired slave address and direction.
SBICR2      ← 1 1 1 1 1 0 0 0      Generates the start condition.
```

(Note) X: Don't care



(Note) Do not write <MST> to "0" when it is "0." (Repeated start cannot be done.)

Fig. 15.19 Timing Chart of Generating a Repeated Start

15.6 Control in the Clock-synchronous 8-bit SIO Mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

Serial bus interface control register 0

SBICR0 (0xFFFF_F257)

	7	6	5	4	3	2	1	0
Bit symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0						
Function	SBI operation 0: Disable 1: Enable	This can be read as "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 (0xFFFF_F250)

	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		

On writing <SCK2:0>: Select serial clock frequency

000	n = 3	1.25 MHz
001	n = 4	625 kHz
010	n = 5	313 kHz
011	n = 6	156 kHz
100	n = 7	78 kHz
101	n = 8	39 kHz
110	n = 9	20 kHz
111	—	External clock

System clock : fsys (=40 MHz)

Clock gear : fc/1

Frequency = $\frac{fsys/4}{2^n}$ [Hz]

(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							

Fig. 15.20 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^2C bus mode 11: (Reserved)		This can be read as "1."	

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."	

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."

Fig. 15.11 SIO Mode Registers

15.6.1 Serial Clock

① Clock source

Internal or external clocks can be selected by programming SBICR1 <SCK2:0>.

Internal clocks

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCK pin. At the beginning of a transfer, the SCK pin output becomes the "H" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

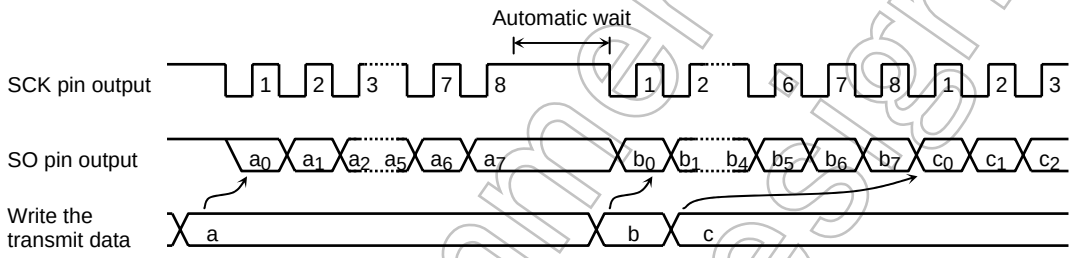


Fig. 15.22 Automatic Wait

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.

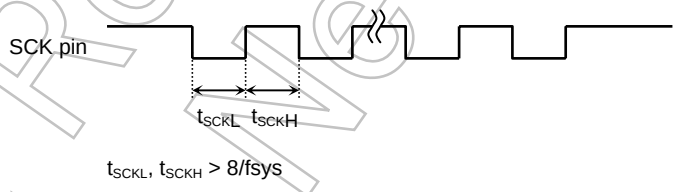


Fig. 15.23 Maximum Transfer Frequency of External Clock Input

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).

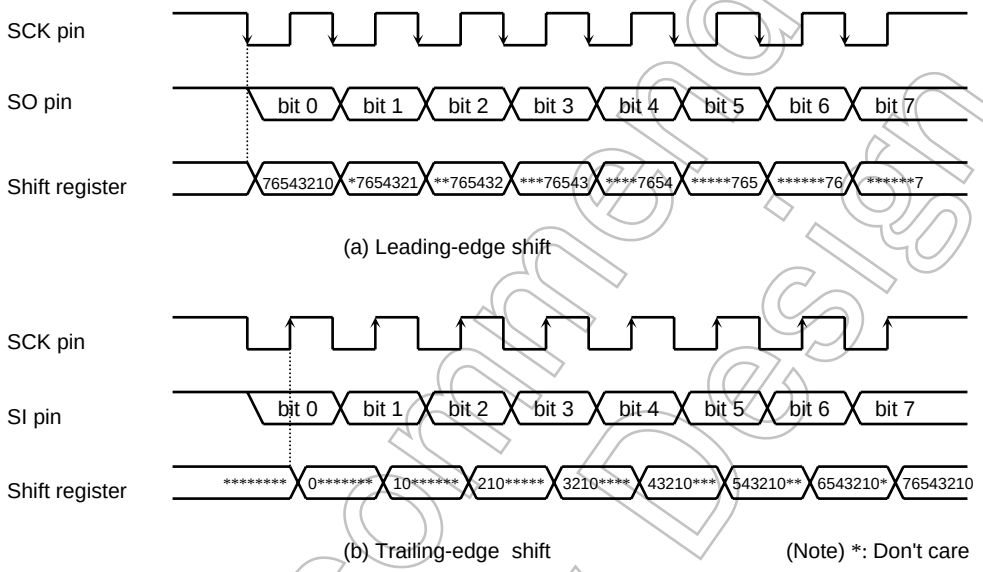


Fig. 15.24 Shift Edge

15.6.2 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.
```

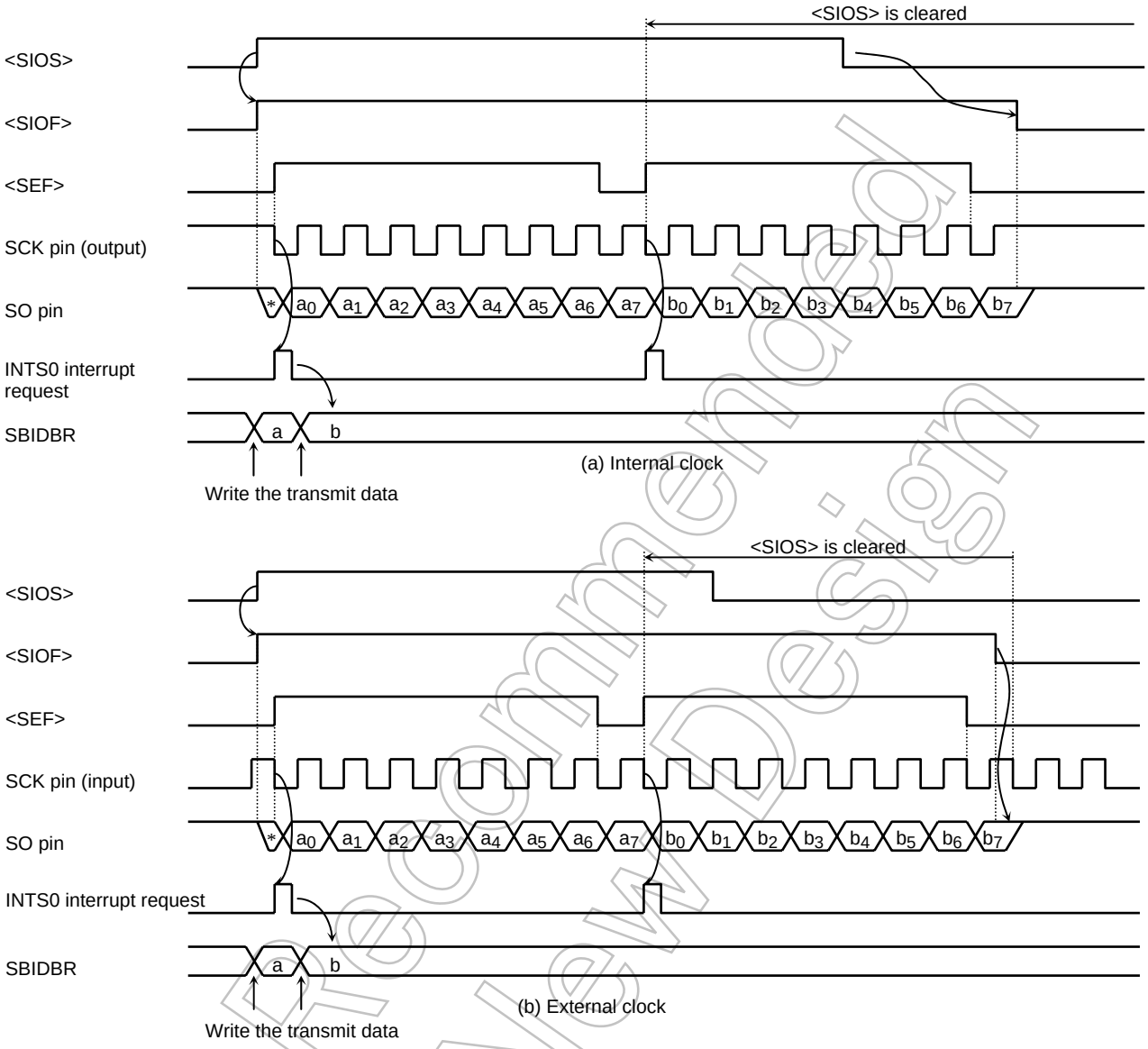


Fig. 15.25 Transmit Mode

Example: Example of programming (MIPS16) to terminate transmission by <SIO> (external clock)

```
        ADDIU   r3,  r0,  0x04
STEST1  :  LB   r2, (SBISR)          ; If SBISR<SEF> = 1 then loop
        AND     r2,  r3
        BNEZ    r2,  STEST1
        ADDIU   r3,  r0,  0x20
STEST2  :  LB   r2, (PA)             ; If SCK = 0 then loop
        AND     r2,  r3
        BEQZ    r2,  STEST2
        ADDIU   r3,  r0,  0y00000111
        STB     r3, (SBICR1)         ; <SIOS> ← 0
```

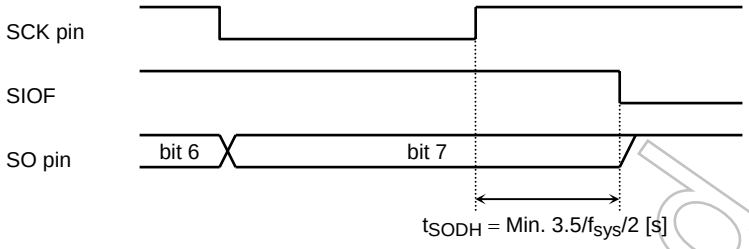


Fig. 15.26 Transmit Data Retention Time at the End of Transmission

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.
```

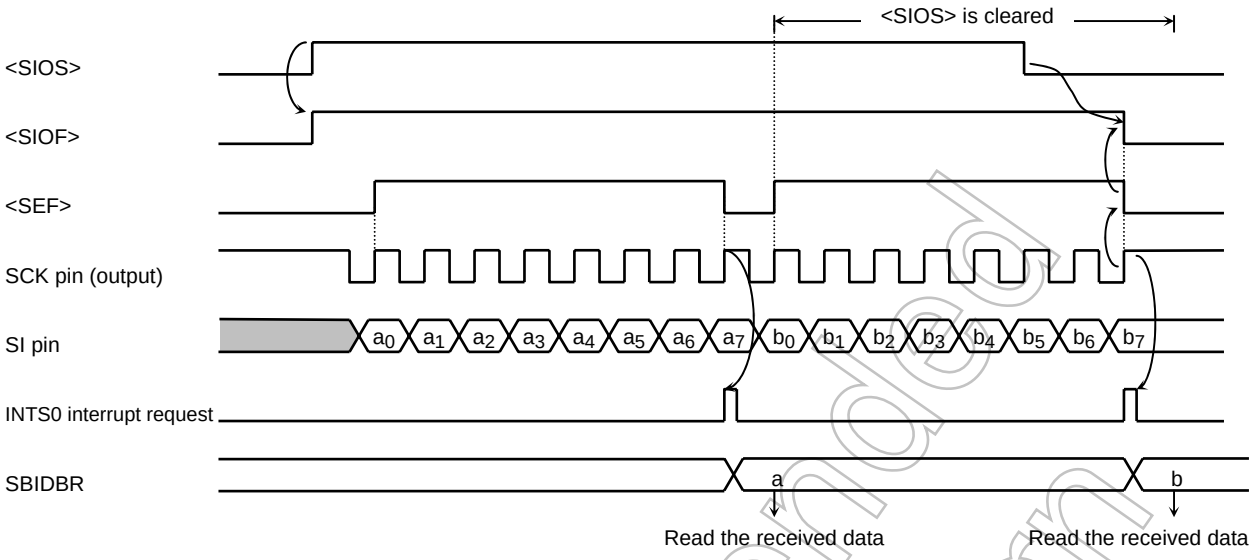


Fig. 15.27 Receive Mode (Example: Internal Clock)

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.

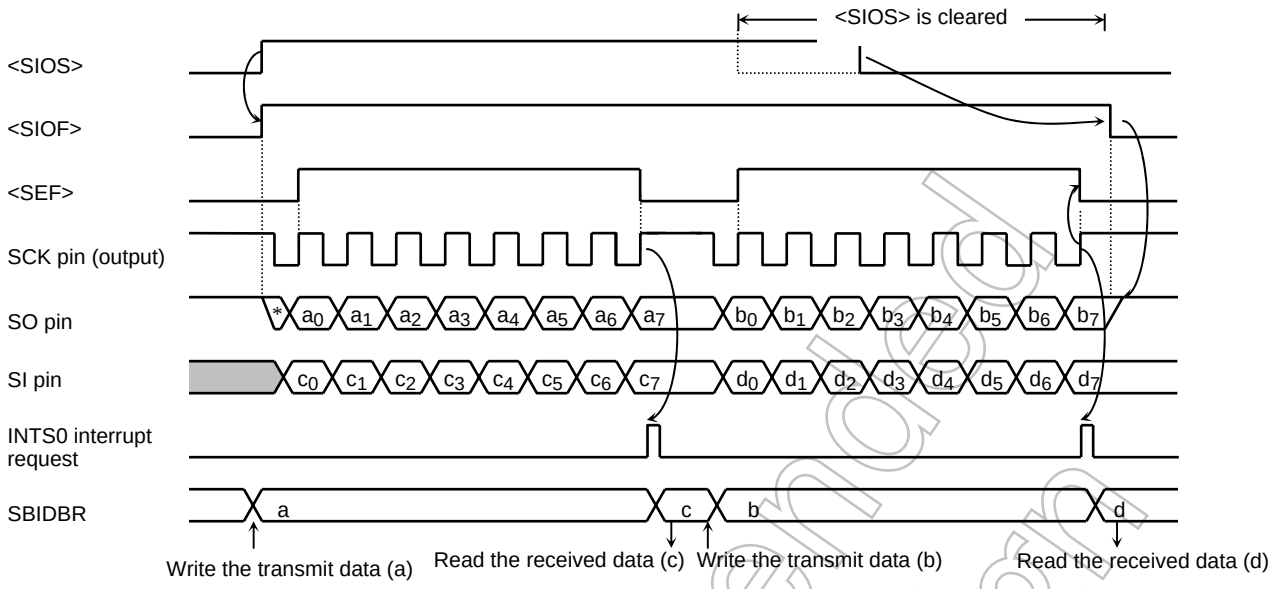


Fig. 15.28 Transmit/Receive Mode (Example: Internal Clock)

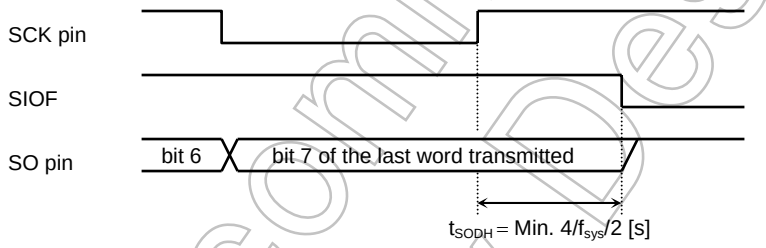


Fig. 15.6.2 Transmit Data Retention Time at the End of Transmission/Reception (In the Transmit/Receive Mode)

```
                    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.
```

16. Analog/Digital Converter

A 10-bit, sequential-conversion analog/digital converter (A/D converter) is built into the TMP19A43. This A/D converter is equipped with 16 analog input channels.

Fig. 16-1 shows the block diagram of this A/D converter.

These 16 analog input channels (pins AN0 through AN15) are also used as input ports.

(Note) If it is necessary to reduce a power current by operating the TMP19A43 in IDLE, SLEEP, SLOW or STOP mode and if either case shown below is applicable, you must first stop the A/D converter and then execute the instruction to put the TMP19A43 into standby mode:

1) The TMP19A43 must be put into IDLE mode when ADMOD1<I2AD> is "0."
2) The TMP19A43 must be put into SLEEP, SLOW or STOP mode.

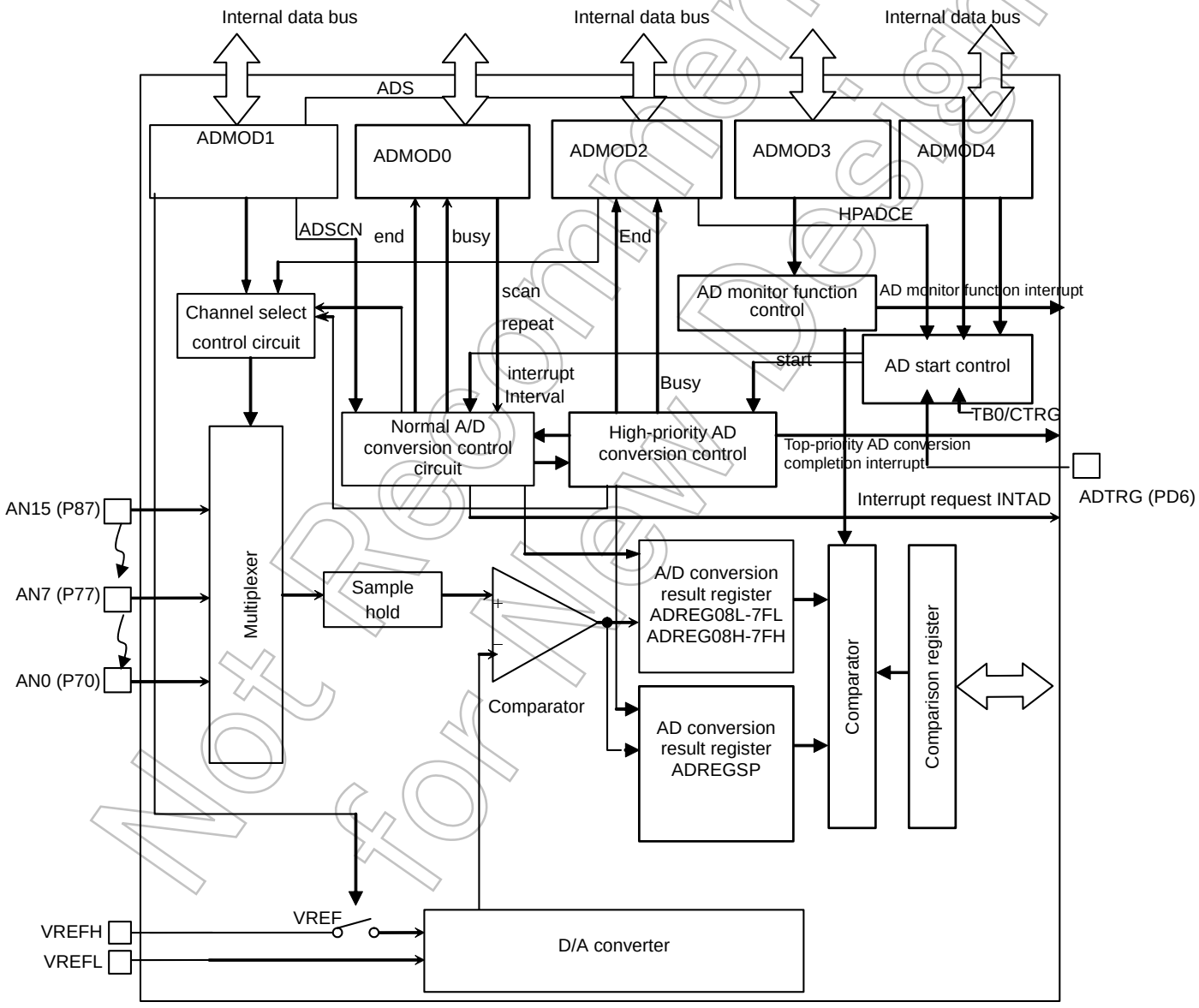


Fig. 16-1 A/D Converter Block Diagram

Note) Please set the following before the analog to digital conversion begins to guarantee the conversion accuracy.

0xFFFF_F319 = 0x58

ADCBAS (0xFFFF_F319)

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

16.1 Control Register

The A/D converter is controlled by A/D mode control registers (ADMOD0, ADMOD1, ADMOD2, ADMOD3 and ADMOD4). Results of A/D conversion are stored in 16 upper and lower A/D conversion result registers ADREG08H/L through ADREG7FH/L. Results of top-priority conversion are stored in ADREGSPH/L.

Fig. 16-2 shows the registers related to the A/D converter.

A/D Mode Control Register 0

ADMOD0 (0xFFFF_F314)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFN	ADBFN		ITM1	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R	R/W				
After reset	0	0	0	0	0	0	0	0
Function	Normal A/D conversion completion flag 0: Before or during conversion 1: Completion	Normal A/D conversion BUSY flag 0: Conversion stop 1: During conversion	"0" is read.	Specify interrupt in fixed channel repeat conversion mode	Specify interrupt in fixed channel repeat conversion mode	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode	Specify scan mode 0: Fixed channel mode 1: Channel scan mode	Start A/D conversion 0: Don't care 1: Start conversion "0" is always read.

Specify A/D conversion interrupt in fixed channel repeat conversion mode

	Fixed channel repeat conversion mode <SCAN> = "0," <REPEAT> = "1"
00	Generate interrupt once every single conversion
01	Generate interrupt once every 4 conversions
10	Generate interrupt once every 8 conversions
11	Setting prohibited

Fig. 16-2 Registers related to the A/D Converter

A/D Mode Control Register 1

ADMOD1 (0xFFFF_ F315)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON	I2AD	ADSCN	–	ADCH3	ADCH2	ADCH1	ADCH0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	VREF application control 0: OFF 1: ON	IDLE 0: Stop 1: Activate	Specify operation mode for channel scanning 0: 4ch scan 1: 8ch scan	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	AN0	AN0	AN0
0001	AN1	AN0 to AN1	AN0 to AN1
0010	AN2	AN0 to AN2	AN0 to AN2
0011	AN3	AN0 to AN3	AN0 to AN3
0100	AN4	AN4	AN0 to AN4
0101	AN5	AN4 to AN5	AN0 to AN5
0110	AN6	AN4 to AN6	AN0 to AN6
0111	AN7	AN4 to AN7	AN0 to AN7
1000	AN8	AN8	AN8
1001	AN9	AN8 to AN9	AN8 to AN9
1010	AN10	AN8 to AN10	AN8 to AN10
1011	AN11	AN8 to AN11	AN8 to AN11
1100	AN12	AN12	AN8 to AN12
1101	AN13	AN12 to AN13	AN8 to AN13
1110	AN14	AN12 to AN14	AN8 to AN14
1111	AN15	AN12 to AN15	AN8 to AN15

(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. 16-3 Registers related to the A/D Converter

A/D Mode Control Register 2

ADMOD2 (0xFFFF_ F316)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFHP	ADBFHP	HPADCE	–	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	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	AN5
0110	AN6
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	AN12
1101	AN13
1110	AN14
1111	AN15

A/D Mode Control Register 3

ADMOD3 (0xFFFF _F317)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC	REGS3	REGS2	REGS1	REGS0	ADOBSV
Read/Write	R/W	R	R/W					
After reset	0	0	0	0	0	0	0	0
Function	Write "0."	"0" is read.	Make AD monitor function interrupt setting 0: Smaller than comparison Regi 1: Larger than comparison Regi	BIT for selecting the AD conversion result storage Regi that is to be compared with the comparison Regi if the AD monitor function is enabled				AD monitor function 0: Disable 1: Enable

<REGS.2, 1, 0>	AD conversion result storage Regi to be compared
0000	ADREG08
0001	ADREG19
0010	ADREG2A
0011	ADREG3B
0100	ADREG4C
0101	ADREG5D
0110	ADREG6E
0111	ADREG7F
1XXX	ADREGSP

A/D Mode Control Register 4

ADMOD4 (0xFFFF_F318)

	7	6	5	4	3	2	1	0
bit Symbol	HADHS	HADHTG	ADHS	ADHTG			ADRST1	ADRST0
Read/Write	R/W				R		W	W
After reset	0	0	0	0	0		–	–
Function	HW source for activating top-priority A/D conversion 0: External TRG 1: TB9TRG	HW for activating top-priority A/D conversion 0: Disable 1: Enable	HW source for activating normal A/D conversion 0: External TRG 1: TB1TRG	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.

Lower A/D Conversion Result Register 08

ADREG08L (0xFFFF_F300)

	7	6	5	4	3	2	1	0
bit Symbol	ADR01	ADR00					OVR0	ADR0RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 08

ADREG08H (0xFFFF_F301)

	7	6	5	4	3	2	1	0
bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 19

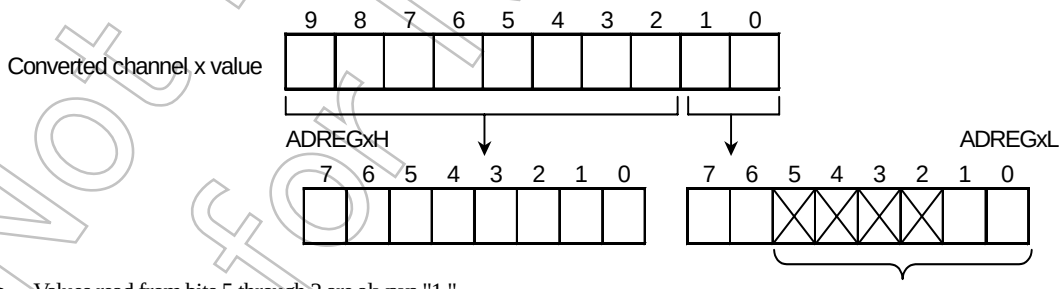
ADREG19L (0xFFFF_F302)

	7	6	5	4	3	2	1	0
bit Symbol	ADR11	ADR10					OVR1	ADR1RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUNflag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 19

ADREG19H (0xFFFF_F303)

	7	6	5	4	3	2	1	0
bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 are always "1."
- Bit 0 is the A/D conversion result storage flag <ADRxRF>. This bit is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) will set this bit to "0."
- Bit 1 is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers, first read upper registers and then lower registers.

Fig. 16-4 Registers related to the A/D Converter

Lower A/D Conversion Result Register 2A

ADREG2AL (0xFFFF_F304)

	7	6	5	4	3	2	1	0
bit Symbol	ADR21	ADR20					OVR2	ADR2RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 2A

ADREG2AH (0xFFFF_F305)

	7	6	5	4	3	2	1	0
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 3B

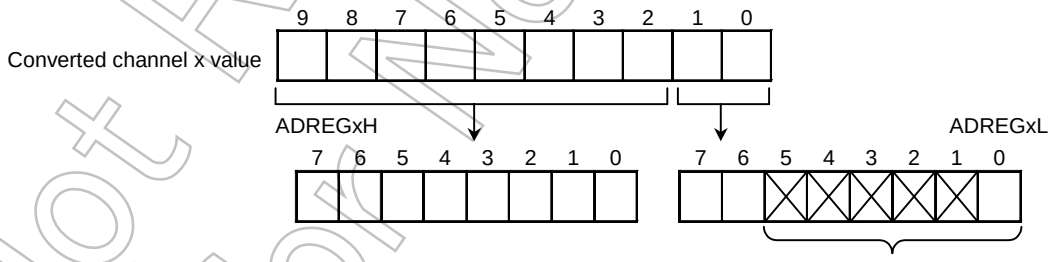
ADREG3BL (0xFFFF_F306)

	7	6	5	4	3	2	1	0
bit Symbol	ADR31	ADR30					OVR3	ADR3RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 3B

ADREG3BH (0xFFFF_F307)

	7	6	5	4	3	2	1	0
bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 are always "1."
- Bit 0 is the A/D conversion result storage flag <ADRxRF>. It is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) will set this bit to "0."
- Bit 1 is the over RUN flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH,ADREGxL) are read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers, first read upper registers and then read lower registers.

Fig. 16-5 Registers related to the A/D Converter (1 of 2)

Lower A/D Conversion Result Register 4C

ADREG4CL (0xFFFF_F308)

	7	6	5	4	3	2	1	0
bit Symbol	ADR41	ADR40					OVR4	ADR4RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 4C

ADREG4CH (0xFFFF_F309)

	7	6	5	4	3	2	1	0
bit Symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 5D

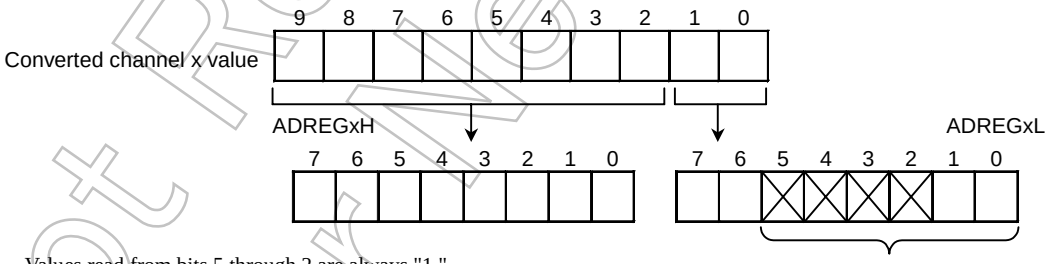
ADREG5DL (0xFFFF_F30A)

	7	6	5	4	3	2	1	0
bit Symbol	ADR51	ADR50					OVR5	ADR5RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 5D

ADREG5DH (0xFFFF_F30B)

	7	6	5	4	3	2	1	0
bit Symbol	ADR59	ADR58	ADR57	ADR56	ADR55	ADR54	ADR53	ADR52
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 are always "1."
- Bit 0 is the A/D conversion result storage flag <ADRxRF>. It is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) will set this bit to "0."
- Bit 1 is the over Run flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers, first read upper registers and then read lower registers.

Lower A/D Conversion Result Register 6E

ADREG6EL (0xFFFF_F30C)

	7	6	5	4	3	2	1	0
bit Symbol	ADR61	ADR60					OVR6	ADR6RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 6E

ADREG6EH (0xFFFF_F30D)

	7	6	5	4	3	2	1	0
bit Symbol	ADR69	ADR68	ADR67	ADR66	ADR65	ADR64	ADR63	ADR62
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							

Lower A/D Conversion Result Register 7F

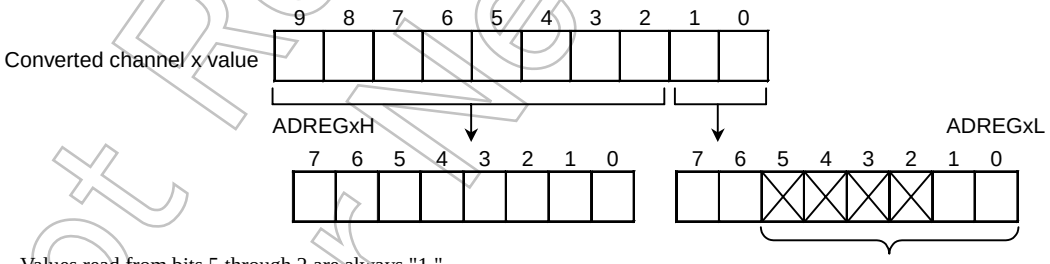
ADREG7FL (0xFFFF_F30E)

	7	6	5	4	3	2	1	0
bit Symbol	ADR71	ADR70					OVR7	ADR7RF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUNflag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 7F

ADREG7FH (0xFFFF_F30F)

	7	6	5	4	3	2	1	0
bit Symbol	ADR79	ADR78	ADR77	ADR76	ADR75	ADR74	ADR73	ADR72
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 are always "1."
- Bit 0 is the A/D conversion result storage flag <ADRxRF>. It is set to "1" if an A/D converted value is stored. A read of a lower register (ADREGxL) will set this bit to "0."
- Bit 1 is the over Run flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers, first read upper registers and then read lower registers.

Lower A/D Conversion Result Register SP

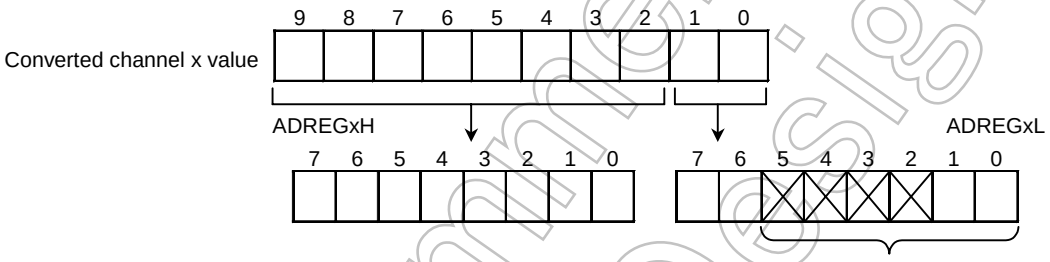
ADREGSPL (0xFFFF_F310)

	7	6	5	4	3	2	1	0
bit Symbol	ADRSP1	ADRSP0					OVRSP	ADRSPRF
Read/Write	R		R				R	R
After reset	0		1				0	0
Function	Store lower 2 bits of A/D conversion result		"1" is read.				Over RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register SP

ADREGSPH (0xFFFF_F311)

	7	6	5	4	3	2	1	0
bit Symbol	ADRSP9	ADRSP8	ADRSP7	ADRSP6	ADRSP5	ADRSP4	ADRSP3	ADRSP2
Read/Write	R							
After reset	0							
Function	Store upper 8 bits of A/D conversion result							



- Values read from bits 5 through 2 are always "1."
- Bit 0 is the A/D conversion result storage flag <ADRxRF>. It is set to "1" after an A/D converted value is stored. A read of a lower register (ADREGxL) will set this bit to "0."
- Bit 1 is the over RUN flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADREGxH and ADREGxL) are read. A read of a flag will clear this bit to "0."
- When reading conversion result storage registers, first read upper registers and then read lower registers.

Lower A/D Conversion Result Comparison Register

ADCOMREGL (0xFFFF_F312)

	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 comparison		"0" is read.					

Upper A/D Conversion Result Comparison Register

ADCOMREGH (0xFFFF_F313)

	7	6	5	4	3	2	1	0
bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R/W							
After reset	0							
Function	Store upper 8 bits of A/D conversion result comparison							

(Note) To set or change a value in this register, the AD monitor function must be disabled (ADMOD3<ADOBSV>="0").

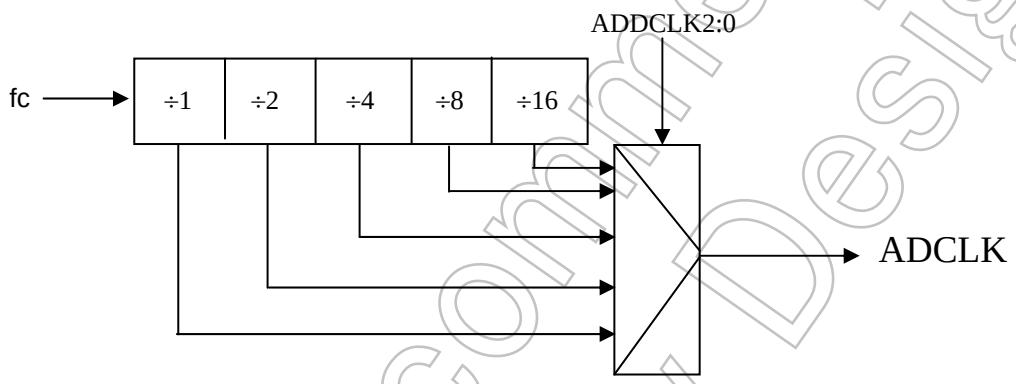
16.2 Conversion Clock

- The conversion time is calculated by the 46 conversion clock.

A/D Conversion Clock Setting Register

ADCLK (0xFFFF_F31C)

	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: 8×2 conversion clock 1010: 8×3 conversion clock 1011: 8×4 conversion clock 0011: 8×8 conversion clock 1100: 8×16 conversion clock 1101: 8×64 conversion clock				"0" is read.	Select the A/D prescaler output 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 111: Reserved		



Example: If fsys = fc = 40 MHz

fc	prescalar	tconv. (conversion time)
40 MHz	1	1.15 us
	1/2	2.3 us
	1/4	4.6 us

Variable S/H time

Conversion clock	S/H time	tconv. (conversion time)
40 MHz	Conversion clk*8*1 (0.2 us)	1.15 us
	Conversion clk*8*2 (0.4 us)	1.35 us
	Conversion clk*8*3 (0.6 us)	1.55 us
	Conversion clk*8*4 (0.8 us)	1.75 us
	Conversion clk*8*8 (1.6 us)	2.55 us
	Conversion clk*8*16 (3.2 us)	4.15 us
	Conversion clk*8*64 (12.8 us)	13.75 us

"Please do not change the analog to digital conversion clock setting in the analog to digital translation. "

Description of Operations

16.2.1 Analog Reference Voltage

The "H" level of the analog reference voltage shall be applied to the VREFH pin, and the "L" level shall be applied to the VREFL pin. By writing "0" to the ADMOD1<VREFON> bit, a switched-on state of VREFH-VREFL can be turned into a switched-off state. To start AD conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

16.2.2 Selecting the Analog Input Channel

How the analog input channel is selected is different depending on A/D converter operation mode used.

(1) Normal AD conversion mode

If the analog input channel is used in a fixed state (ADMOD0<SCAN>="0"):

One channel is selected from analog input pins AIN0 through AIN15 by setting ADMOD1<ADCH3 to 0> to an appropriate setting.

If the analog input channel is used in a scan state (ADMOD0<SCAN>="1"):

One scan mode is selected from 16 scan modes by setting ADMOD1<ADCH3 to 0> and ADSCN to appropriate settings.

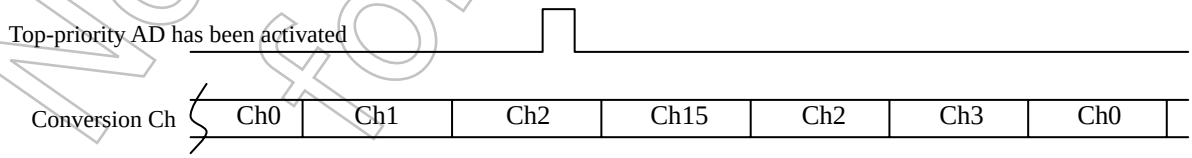
(2) Top-priority AD conversion mode

One channel is selected from analog input pins AIN0 through AIN15 by setting ADMOD2<HPADCH3 to 0> to an appropriate setting.

After a reset, ADMOD0<SCAN> is initialized to "0" and ADMOD1<ADCH3:0> is initialized to "0000." This initialization works as a trigger to select a fixed channel input through the AN0 pin. The pins that are not used as analog input channels can be used as ordinary input ports.

If 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 AIN0 through AIN3 with ADMOD0<REPEAT:SCAN> set to "11" and ADMOD1<ADCH3:0> set to 0011, and top-priority AD conversion has been activated at AIN15 with ADMOD2<HPADCH3:0>=1111:



16.2.3 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 ADMOD0<ADS> to "1." Top-priority AD conversion is software activated by setting ADMOD2<HPADCE> to "1." 4 operation modes are made available to normal AD conversion. In performing normal AD conversion, one of these operation modes must be selected by setting ADMOD0<2:1> to an appropriate setting. For 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 ADMOD4<ADHS>, and top-priority AD conversion can be activated using the HW activation source selected by ADMOD4<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 $\overline{\text{ADTRG}}$ 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.

When normal A/D conversion starts, the A/D conversion Busy flag (ADMOD0<ADBF>) showing that A/D conversion is under way is set to "1." When top-priority A/D conversion starts, the A/D conversion Busy flag (ADMOD2<ADBFHP>) showing that A/D conversion is under way is set to "1." If normal A/D conversion is interrupted by 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 (ADMOD4<ADRST1:0>) must be performed before starting A/D conversion. The HW activation method must not be used to reactivate normal A/D conversion.

If ADMOD2<HPADCE> is set to "1" during normal A/D conversion, ongoing A/D conversion is discontinued and top-priority A/D conversion starts; specifically, A/D conversion (fixed channel single conversion) is executed for a channel designated by ADMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADREGSP, 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 ADMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADREGSP, normal A/D conversion is resumed.

16.2.4 A/D Conversion Modes and A/D Conversion Completion Interrupts

For A/D conversion, the following four operation modes are supported. For normal A/D conversion, an operation mode can be selected by setting ADMOD0<2:1> to an appropriate setting. For top-priority A/D conversion, the fixed channel single conversion mode is automatically selected, irrespective of the ADMOD0<2:1> setting.

Fixed channel single conversion mode

Channel scan single conversion mode

Fixed channel repeat conversion mode

Channel scan repeat conversion mode

(1) Normal A/D conversion

An operation mode is selected with ADMOD0<REPEAT, SCAN>. As A/D conversion starts, ADMOD0<ADBFN> is set to "1." When specified A/D conversion is completed, the A/D conversion completion interrupt (INTAD) is generated, and ADMOD0<EOCF> showing the completion of A/D conversion is set to "1." If <REPEAT>="0," <ADBFN> returns to "0" concurrently with the setting of EOCF. If <REPEAT> is set to "1," <ADBFN> remains at "1" and A/D conversion continues.

① Fixed channel single conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "00," A/D conversion is performed in the fixed channel single conversion mode.

In this mode, A/D conversion is performed once for one channel selected. After A/D conversion is completed, ADMOD0<EOCF> is set to "1," ADMOD0<ADBF> is cleared to "0," and the interrupt request INTAD is generated. <EOCF> is cleared to "0" upon read.

② Channel scan single conversion mode

If ADMOD0 <REPET,SCAN> is set to "01," A/D conversion is performed in the channel scan single conversion mode.

In this mode, A/D conversion is performed once for each scan channel selected. After A/D scan conversion is completed, ADMOD0<EOCF> is set to "1," ADMOD0<ADBF> is cleared to "0," and the interrupt request INTAD is generated. <EOCF> is cleared to "0" upon read.

③ Fixed channel repeat conversion mode

If ADMOD0<REPEAT,SCAN> is set to "10," A/D conversion is performed in fixed channel repeat conversion mode.

In this mode, A/D conversion is performed repeatedly for one channel selected. After A/D conversion is completed, ADMOD <EOCF> is set to "1." ADMOD0 <ADBF> is not cleared to "0." It remains at "1." The timing with which the interrupt request INTAD is generated can be selected by setting ADMOD0 <ITM1:0> to an appropriate setting. <EOCF> is set with the same timing as this interrupt INTAD is generated.

<EOCF> is cleared to "0" upon read.

With <ITM1:0> set to "00," an interrupt request is generated each time one A/D conversion is completed. In this case, the conversion results are always stored in the storage register ADREG08. After the conversion result is stored, EOCF changes to "1."

With <ITM1:0> set to "01," an interrupt request is generated each time four A/D conversion are completed. In this case, the conversion results are sequentially stored in storage registers ADREG08 through ADREG3B. After the conversion results are stored in ADREG3B, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADREG08. <EOCF> is

cleared to "0" upon read.

With <ITM1:0> set to "10," an interrupt request is generated each time eight A/D conversions are completed. In this case, the conversion results are sequentially stored in storage registers ADREG08 through ADREG7F. After the conversion results are stored in ADREG7F, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADREG08.

<EOCF> is cleared to "0" upon read.

④ Channel scan repeat conversion mode

If ADMOD0 <REPEAT, SCAN> is set to "11," A/D conversion is performed in the channel scan repeat conversion mode.

In this mode, A/D conversion is performed repeatedly for a scan channel selected. Each time one A/D scan conversion is completed, ADMOD0 <EOCF> is set to "1," and the interrupt request INTAD is generated. ADMOD0 <ADBF> is not cleared to "0." It remains at "1." <EOCF> is cleared to "0" upon read.

To stop the A/D conversion operation in the repeat conversion mode (modes described in ③ and ④ above), write "0" to ADMOD0 <REPEAT>. When ongoing A/D conversion is completed, the repeat conversion mode terminates, and ADMOD0 <ADBF> is set to "0."

Before switching from one mode to standby mode (such standby modes as IDLE, STOP, 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 ADMOD0<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 ADMOD2<HPADCH3:0>. After the A/D conversion is completed, the top-priority A/D conversion completion interrupt is generated, ADMOD2<EOCFHP> is set to "1," and <ADBFHP> returns to "0." The EOCFHP Flag is cleared upon read.

Relationships between A/D Conversion Modes, Interrupt Generation Timings and Flag Operations

Conversion mode	Interrupt generation timing	EOCF setting timing (see Note)	ADBF (after the interrupt is generated)	ADMOD0		
				ITM1:0	REPEAT	SCAN
Fixed channel single conversion	After conversion is completed	After conversion is completed	0	—	0	0
Fixed channel repeat conversion	Each time one conversion is completed	After one conversion is completed	1	00	1	0
	Each time four conversions are completed	After four conversions are completed	1	01		
	Each time eight conversions are completed	After eight conversions are completed	1	10		
Channel scan single conversion	After scan conversion is completed	After scan conversion is completed	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.

16.2.5 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 ADMOD2<HPADCE> to "1" or it can be activated using the HW resource by setting ADMOD4<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 ADMOD2<3:0>. The result of single conversion is stored in ADREGSP, 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 AN0 through AN8 and if <HPADCE> is set to "1" during AN3 conversion, AN3 conversion is suspended, and conversion is performed for a channel designated by <HPADC3:0>. After the result of conversion is stored in ADREGSP, channel repeat conversion is resumed, starting from AN3.

16.2.6 A/D Monitor Function

If ADMOD3<ADOBSV> is set to "1," the A/D monitor function is enabled. If the value of the conversion result storage register specified by REGS<3:0> becomes larger or smaller ("larger" or "smaller" to be designated by ADOBIC) than the value of a comparison register, the A/D monitor function interrupt is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result storage register, and the interrupt is generated if the conditions are met. Because storage registers assigned to perform the A/D monitor function are usually not read by software, overrun flag <OVRn> is always set and the conversion result storage flag <ADRnRF> is also set. To use the A/D monitor function, therefore, a flag of a corresponding conversion result storage register must not be used.

16.2.7 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 (ADREG08H/L through ADRG7FH/L).

In fixed channel repeat conversion mode, A/D conversion results are sequentially stored in ADREG08H/L through ADREG7FH/L. If <ITM1:0> is so set as to generate the interrupt each time one A/D conversion is completed, conversion results are stored only in ADREG08H/L. If <ITM1:0> is so set as to generate the interrupt each time four A/D conversions are completed, conversion results are sequentially stored in ADREG08H/L through ADREG3BH/L.

Table 16. 1 shows analog input channels and related A/D conversion result registers.

Table 16-1 Analog Input Channels and Related A/D Conversion Result Registers

Analog input channel (port A)	A/D conversion result register			
	Conversion modes other than shown to the right	Fixed channel repeat conversion mode (every one conversion)	Fixed channel repeat conversion mode (every four conversions)	Fixed channel repeat conversion mode (every eight conversions)
AN0	ADREG08H/L	ADREG08H/L fixed	ADREG08H/L ↔ ADREG3BH/L	ADREG08H/L ↔ ADREG7FH/L
AN1	ADREG19H/L			
AN2	ADREG2AH/L			
AN3	ADREG3BH/L			
AN4	ADREG4CH/L			
AN5	ADREG5DH/L			
AN6	ADREG6EH/L			
AN7	ADREG7FH/L			
AN8	ADREG08H/L			
AN9	ADREG19H/L			
AN10	ADREG2AH/L			
AN11	ADREG3BH/L			
AN12	ADREG4CH/L			
AN13	ADREG5DH/L			
AN14	ADREG6EH/L			
AN15	ADREG7FH/L			

16.2.8 Data Polling

To process A/D conversion results without using interrupts, ADMOD0<EOCF> must be polled. If this flag is set, conversion results are stored in a specified A/D conversion result register. After confirming that this flag is set, read that conversion result storage register. In reading the register, make sure that you first read upper bits and then lower bits to detect an overrun. If OVRn is "0" and ADRnRF is "1" in lower bits, a correct conversion result has been obtained.

17. Digital/Analog Converter

This section describes the D/A converter that is built into the TMP19A43.

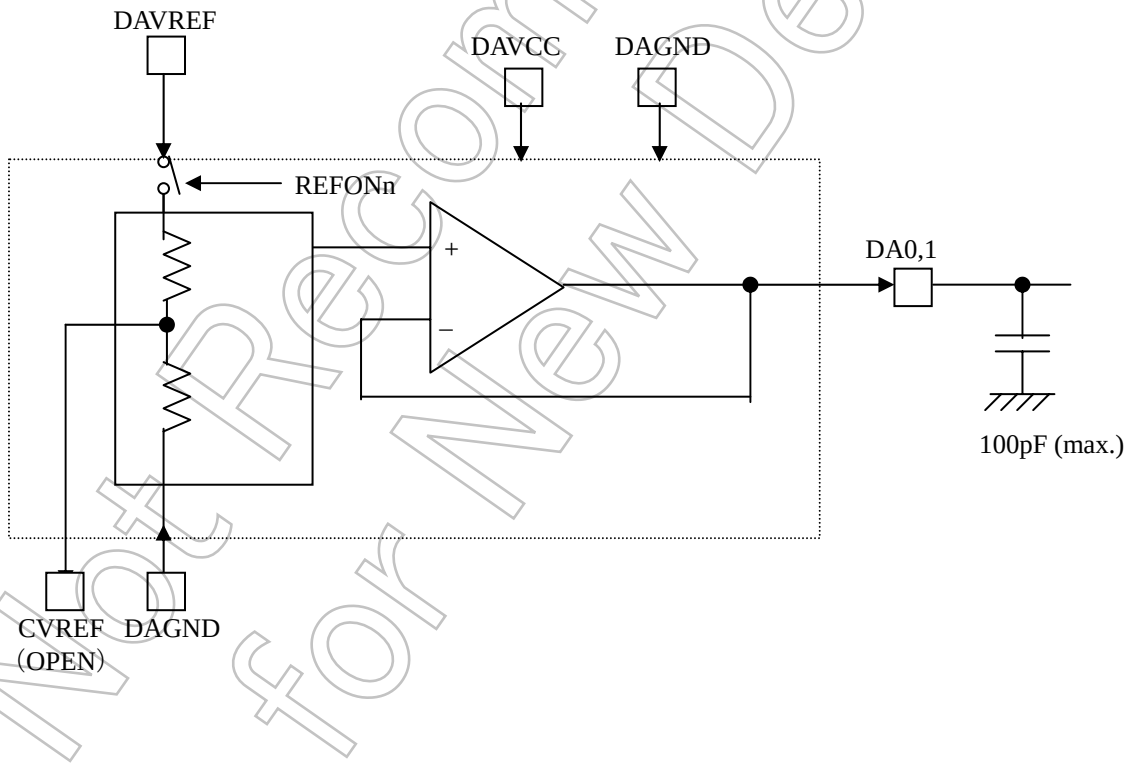
17.1 Features

- A high-resolution, 8-bit D/A converter is built into each of two channels of the TM19A43.
- Each channel is provided with a buffer amplifier.
- Each channel can be individually put into standby mode by making an appropriate control register setting.

17.2 Operation (about the Operation of the D/A Converter)

Basic settings:

- Set the control register DACCNTn<OPn><REFON> to <1:1>.
- Assign an output CODE to the output register DAREGn so that an output voltage specified for that CODE appears at the output pin DAn.
- Set DACCNTn<OPn> to "0," and the output pin DAn goes into Power Down mode. (The electric potential of the pin becomes equal to that of the DAVREF power supply.)
- By setting DACCNTn<REFON> to "0," Iref can be cut and a consumption current can be reduced.



(note)Please open the terminal CVREF.

Fig. 17.2.1 D/A Converter Block Diagram

DACCNT0 Register

DACCNT0 (0xFFFF_F330)

	7	6	5	4	3	2	1	0
Bit Symbol							REFON0	OP0
Read/Write							R/W	R/W
After reset	0						0	0
Function	"0" is read.						0: Ref off 1: Ref on	0: Power Down 1: Output

Output Register DAREG0

DAREG0 (0xFFFF_F331)

	7	6	5	4	3	2	1	0
Bit Symbol	DAC7	DAC6	DAC5	DAC4	DAC3	DAC2	DAC1	DAC0
Read/Write	R/W							
After reset	0							
Function								

DACCNT1 Register

DACCNT1 (0xFFFF_F338)

	7	6	5	4	3	2	1	0
Bit Symbol							REFON1	OP1
Read/Write							R/W	R/W
After reset	0						0	0
Function	"0" is read.						0: Ref off 1: Ref on	0: Power Down 1: Output

Output Register DAREG1

DAREG1 (0xFFFF_F339)

	7	6	5	4	3	2	1	0
Bit Symbol	DAC7	DAC6	DAC5	DAC4	DAC3	DAC2	DAC1	DAC0
Read/Write	R/W							
After reset	0							
Function								

- **When setting REFONn to "0," OPn must also be set to "0."**
- **After "0" of REFONn is changed "1," wait for 10 μ sec(open the terminal CVREF) msec during which time circuitry stabilizes.**
- **All voltages in the range from DAGND through DAVCC (DAVREF) cannot be output. A correct range of output voltages must be verified by checking the electrical characteristics which are later described.**

• (note 3)100μs(typ.) is necessary by the time the voltage output to DAn after output CODE is set is steady.

18. Watchdog Timer (Runaway Detection Timer)

The TMP19A43 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.

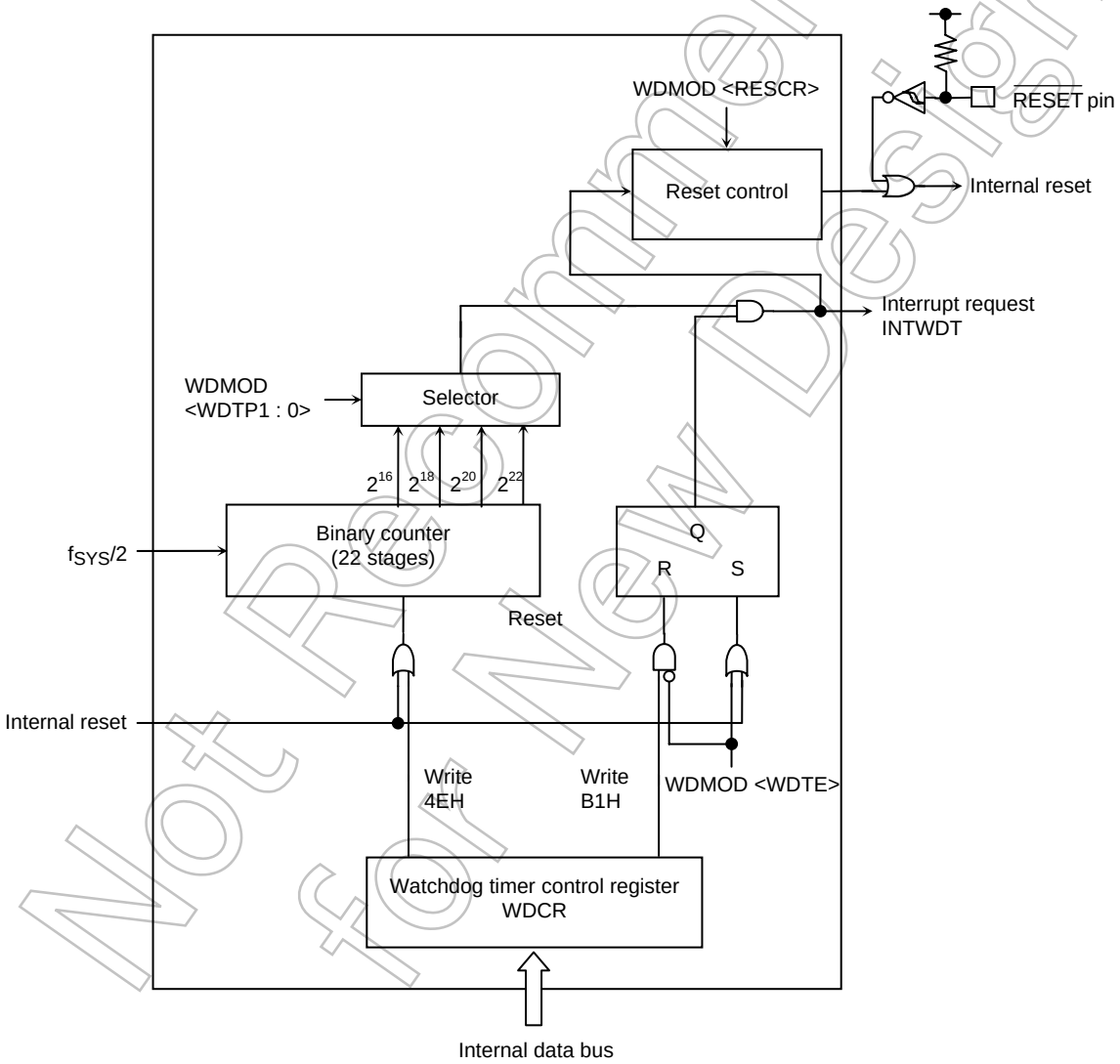


Fig. 18.1 Block Diagram of the Watchdog Timer

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} and 2^{22}. 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..

Because the watchdog timer interrupt is a non-maskable interrupt factor, NMIFLG <WDT> at the INTC performs a task of identifying it.

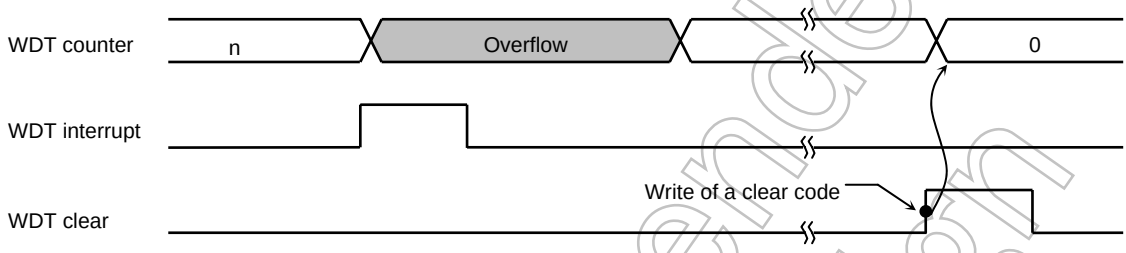


Fig. 18.2 Normal Mode

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is effected for a 32-state time, as shown in Fig. 18.. If this reset is effected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 8 is used as an input clock $f_{SYS/2}$.

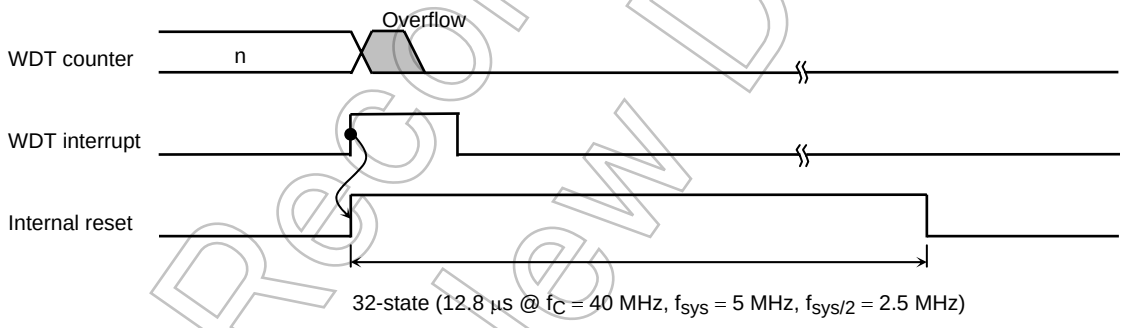


Fig. 18.3 Reset Mode

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. 18.1 shows the detection time of the watchdog timer.

② Enabling/disabling the watchdog timer <WDTE>

When reset, WDMOD <WDTE> is initialized to "1" and the watchdog timer is enabled.

To disable the watchdog timer, this bit must be set to "0" and, at the same time, the disable code (B1H) must be written to the WDCR register. This dual setting is intended to minimize the probability that the watchdog timer may inadvertently be disabled if a runaway occurs.

To change the status of the watchdog timer from "disable" to "enable," set the <WDTE> bit to "1."

③ Watchdog timer out reset connection <RESCR>

This 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 (0xFFFF_F090)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	—
Read/Write	R/W	R/W		R		R/W		R/W
After reset	1	0	0	0		0	0	0
Function	WDT control 1: Enable	Selects WDT detection time 00: $2^{16}/f_{SYS}$ 01: $2^{18}/f_{SYS}$ 10: $2^{20}/f_{SYS}$ 11: $2^{22}/f_{SYS}$		"0" is read.		IDLE 0: Stop 1: Start	1: Internally connect WDT output to reset pin	Write "0."

Watchdog timer out control

0	Generates NMI interrupt
1	Connects WDT out to reset

Detection time of watchdog timer @ fc = 40 MHz

SYSCR1 clock gear value <GEAR2:0>	Detection Time of Watchdog Timer WDMOD<WDTP1, 0>			
	00	01	10	11
000 (fc)	1.6 ms	6.5 ms	26.2 ms	105 ms
100 (fc/2)	3.3 ms	13.1 ms	52.4 ms	210 ms
110 (fc/4)	6.5 ms	26.2 ms	105 ms	419 ms
111 (fc/8)	13.1 ms	52.4 ms	209 ms	839 ms

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Fig. 18.4 Watchdog Timer Mode Register

WDCR (0xFFFF_F091)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	—							
Function	B1H : WDT disable code 4EH : WDT clear code							

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 <WDTP1, 0> register. Before generating the INTWD interrupt, the binary counter for the watchdog timer must be cleared to "0" using software (instruction). If the CPU malfunctions (runs away) due to noise or other disturbances and cannot execute the instruction to clear the binary counter, the binary counter overflows and the INTWD interrupt is generated. The CPU is able to recognize the occurrence of a malfunction (runaway) by identifying the INTWD interrupt and to restore the faulty condition to normal by using a malfunction (runaway) countermeasure program. Additionally, it is possible to resolve the problem of a malfunction (runaway) of the CPU by connecting the watchdog timer out pin to reset pins of peripheral devices.

The watchdog timer begins operation immediately after a reset is cleared.

In STOP mode, the watchdog timer is reset and in an idle state. When the bus is open ($\overline{\text{BUSAK}}$ = "L"), it continues counting. In IDLE mode, its operation depends on the WDMOD <I2WDT> setting. Before putting it in IDLE mode, WDMOD <I2WDT> must be set to an appropriate setting, as required.

Examples:

① To clear the binary counter

		7	6	5	4	3	2	1	0	
WDCR	←	0	1	0	0	1	1	1	0	Writes the clear code (4EH)

② To set the detection time of the watchdog timer to $2^{18}/f_{SYS}$

		7	6	5	4	3	2	1	0
WDMOD	←	1	0	1	–	–	–	–	–

③ To disable the watchdog timer

		7	6	5	4	3	2	1	0	
WDMOD	←	0	–	–	–	–	–	–	–	Clears WDTE to "0"
WDCR	←	1	0	1	1	0	0	0	1	Writes the disable code (B1H)

Note: If the watchdog timer is operated when the high-frequency oscillator is idle, the system reset operation initiated by the watchdog timer becomes erratic due to the unstable oscillation of the high-frequency oscillator. Therefore, do not operate the watchdog timer when the high-frequency oscillator is idle.

19. Clock Timer

19.1 Features

TMP19A43 can be used in these operation modes.

This clock timer using a low clock frequency of 32.768 kHz can generate interrupts at time intervals of 0.125s, 0.250s, 0.500s and 1.000s so that the TMP19A43 is able to use the clock function when operating in low-power-dissipation operation modes.

This clock timer can be operated in all operation modes of low-frequency oscillation. The interrupt generated by this clock allows the TMP19A43 to recover from standby mode (except STOP mode) and return to normal operation mode. To use the clock timer interrupt (INTRTC), the IMCGD register in the CG must be set to an appropriate setting.

Fig. 19-1 shows the block diagram of the clock timer.

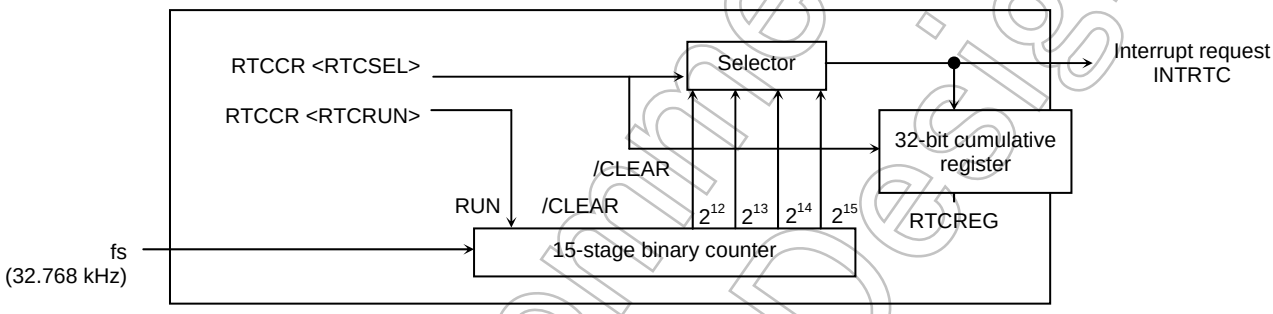


Fig. 19-1 Block Diagram of the Clock Timer

Note:A built-in register is initialized by reset with the terminal RESET.
On the other hand, When resetting it by WDT and DSU, it is not initialized.

Register

The clock timer is controlled by the clock timer control register (RTCCR).

Fig. 19-2 shows the clock timer control register.

(fs = 32.768 kHz)

LITTLE / BIG

RTCCR (0xFFFF_E704) (0xFFFF_E707)

	7	6	5	4	3	2	1	0
Bit Symbol					RTCRCLR	RTCSEL1	RTCSEL0	RTCRUN
Read/Write	R/W	R/W	R		W	R/W		R/W
After reset	0	0	0	0	0	0	0	0
Function	Write "0."	Write "0."	"0" is read.		Clear cumulative register 0: Clear 1: Don't Care	Interrupt generation cycle 00: 2^{15}/fs (1.000 s) 01: 2^{14}/fs (0.500 s) 10: 2^{13}/fs (0.250 s) 11: 2^{12}/fs (0.125 s)		Binary counter 0: Stop & clear 1: Count

	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function								

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

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

Fig. 19-2 Clock Timer Control Register

(Note 1) To access this register, 32-bit access is required.

(Note 2) Values read from RTCCR<RTCRCLR> are always "1."

(Note 3) Before changing the RTCCR<RTCSSEL1:0> setting, make sure that RTCCR<RTCRUN> is "0" and that the RTC interrupt is disabled.

(Note 4) A built-in register is initialized by reset with the terminal RESET.

When resetting it by WDT and DSU, it is not initialized.

The clock timer is provided with a clock count cumulative register for counting the number of times interrupts are generated.

Clock Count Cumulative Register

RTCREG (0xFFFF_E708)

	7	6	5	4	3	2	1	0
Bit Symbol	RUI7	RUI6	RUI5	RUI4	RUI3	RUI2	RUI1	RUI0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							
	15	14	13	12	11	10	9	8
Bit Symbol	RUI15	RUI14	RUI13	RUI12	RUI11	RUI10	RUI9	RUI8
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							
	23	22	21	20	19	18	17	16
Bit Symbol	RUI23	RUI22	RUI21	RUI20	RUI19	RUI18	RUI17	RUI16
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							
	31	30	29	28	27	26	25	24
Bit Symbol	RUI31	RUI30	RUI29	RUI28	RUI27	RUI26	RUI25	RUI24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Accumulate count value							

Fig. 19-3 Clock Count Cumulative Register

(Note 1) A write to this cumulative register clears the prescaler.

(Note 2) Interrupts must be disabled during a read.

(Note 3) To access this register, 32-bit access is required.

Example of the clock timer interrupt setting:

Initialization

```
                    7 6 5 4 3 2 1 0
IMCD        ← 0 0 0 1 0 0 0 0   Disables the interrupt INTRTC
                                Sets the bit <15:8> of a 32-bit register
RTCCR       ← 0 0 0 0 X X X 0   Stops the RTC timer count
                                Sets the bit <7:0> of a 32-bit register
IMCGD       ← 0 0 1 0 0 0 0 1   Sets the bit <15:8> of a 32-bit register
EICRCG      ← 0 0 0 0 1 1 0 1   Clears the interrupt request for the CG block
                                Set the bit <7:0> of a 32-bit register
INTCLR      ← 1 1 1 1 0 0 0 0   Clears the interrupt request for the INTC block
                                Sets the bit <8:0> of a 32-bit register
RTCCR       ← 0 0 0 0 1 X X 1   Starts the timer count
                                Sets the bit <7:0> of a 32-bit register
IMCD        ← 0 0 0 1 0 X X X   Sets the interrupt level
                                Set the bit <15:8> of a 32-bit register
```

INTRTC interrupt

```
                    7 6 5 4 3 2 1 0
EICRCG      ← 0 0 0 0 1 1 0 1   Clears the interrupt request for the CG block
                                Sets the bit <7:0> of a 32-bit register
INTCLR      ← 1 1 1 1 0 0 0 0   Clears the interrupt request for the INTC block
                                Sets the bit <8:0> of a 32-bit register

Processing
Interruption finished
```

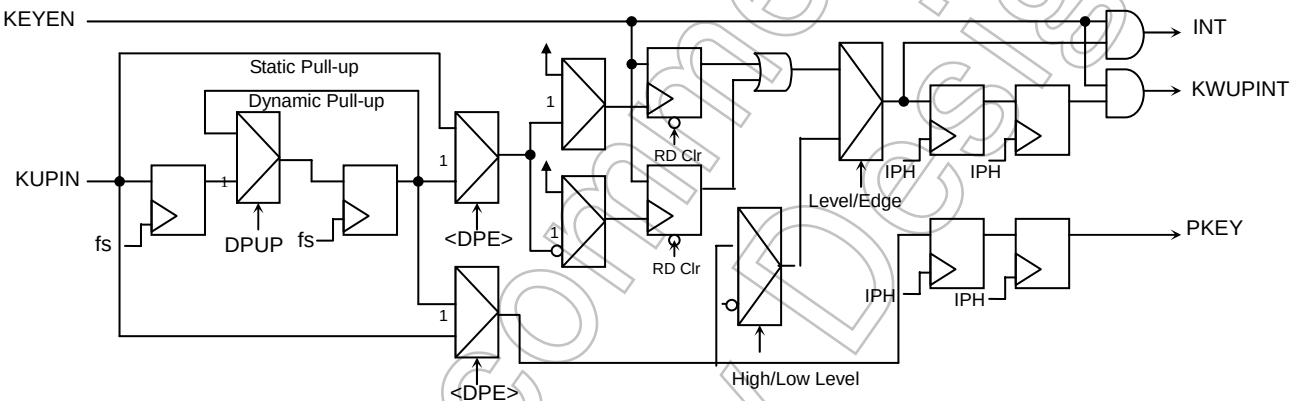
(Note 1) X means "don't care."

(Note 2) To disable interrupts, IMCE must be first set and then IMCGD.

20. Key-on Wakeup Circuit

20.1 Outline

- The TMP19A43 has 32 key inputs, KEY00 to KEY31, which can be used for releasing the STOP 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 TMP19A43 has 32 key input pins, KEY00 to KEY31. Program the IMCGD3<KWUPEN> register in the CG to determine whether to use the key inputs for releasing the STOP 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 IMCGD3 register in the CG as the active high level. Therefore, program IMCGD3<EMCGD0:1> to "01" to determine the detection level to the high level. The results of detection in the CG are also notified to the interrupt controller INTC as the active high level. Therefore, program the INTC to "01" to define the corresponding interrupt as the high level. Setting IMCGD3<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, can it not depend on KWUPSTn<KEYnEN > and the pull-up be used.

20.3.1 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.

   PEFC<PE0F> = "1" The function is set to the key.

   PEPE<PEE0> = "1" Pull-up ON control

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<EMCGC1:0> to "01" 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) Set KWUPST08<KEY08EN> to "0" for the key input to be used.
2) Disable key interrupts by setting IMC4<IL112:110> to "000" at the INTC.
3) Set KWUPST08<KEY08EN> to "1" for the key input to be used.
4) Change the active state by setting KWUPST08<KEY082:KEY080> to "000" for the key input to be changed. (Example: Lo level interrupt)
5) Clear interrupt requests by setting KWUPCLR to "1010."
6) Enable the key interrupt at the INTC. Set IMC4<IL112:110> to a desired level "xxx."

C) To enable a key input during operation

1) Disable key interrupts by setting IMC4<IL112:110> to "000" at the INTC.
2) Set KWUPSTn<KEYnEN> to "0" for the key input to be used.
3) Define the active state of the key input to be used at the corresponding KWUPSTn.
4) Set KWUPSTn<KEYnEN> 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 IMC4<IL112:110> to a desired level.)

20.3.2 Cautions on Use of Key Inputs With Pull-up Disabled

A) When you make the first setting after turning the power ON

1) Set PEPE<PEE0> to "0" to select the pull-up OFF control.
2) Set KWUPSTn<KEYnEN> 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 KWUPSTn<KEYnEN> to "1" for the key input to be used.
5) Set KWUPCLR to "1010" to clear interrupt requests.
6) 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 IMC4<IL112:110> to "000" at the INTC.
2) Set KWUPSTn<KEYnEN> 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 KWUPSTn<KEYnEN> to "1" for the key input to be used.
5) Clear interrupt requests by setting KWUPCLR.
6) Enable key interrupts at the INTC. (Set IMC4<IL112:110> to a desired level.)

C) To enable a key input during operation

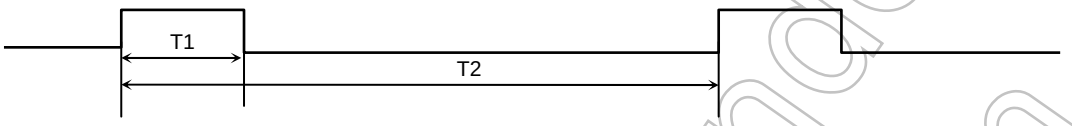
1) Disable key interrupts by setting IMC4<IL112:110> to "000" at the INTC.
2) Set KWUPSTn<KEYnEN> to "0" for the key input to be used.
3) Define the active state by setting KWUPSTn for the key input to be used.
4) Set KWUPSTn<KEYnEN> to "1" for the key input to be used.
5) Clear interrupt requests by setting KWUPCLR.
6) Enable key interrupts at the INTC. (Set IMC4<IL112:110> to a desired level.)

Key-on Wakeup Control

KWUPCNT (0xFFFF_F384)

	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 cycle 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."	

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.

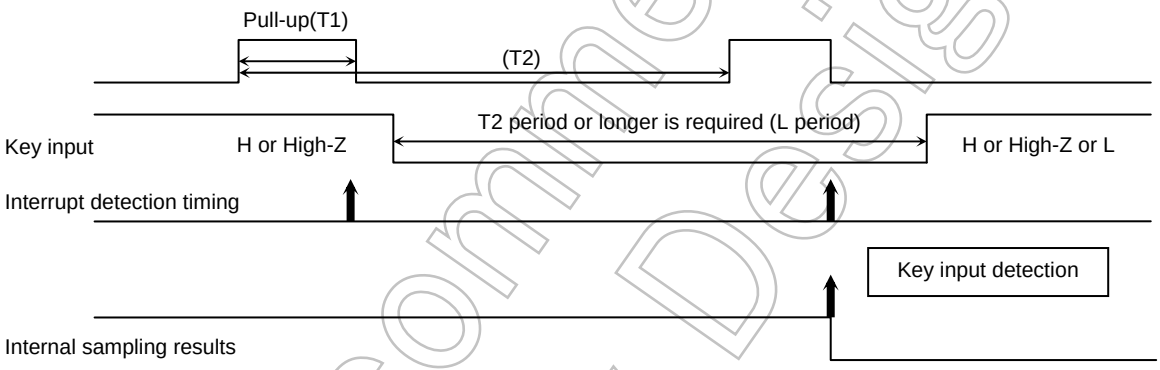
20.4 Key Input Detection Timing

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.



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 (0xFFFF_F380)

	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"

PKEY1 (0xFFFF_F381)

	7	6	5	4	3	2	1	0
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"

PKEY2 (0xFFFF_F382)

	7	6	5	4	3	2	1	0
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"

PKEY3 (0xFFFF_F383)

	7	6	5	4	3	2	1	0
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 (0xFFFF_F360)

	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

KWUPST01 (0xFFFF_F361)

	7	6	5	4	3	2	1	0
bit Symbol	DPE01	KEY012	KEY011	KEY010				KEY01EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY01 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY01 interrupt input 0: Disable 1: Enable

KWUPST02 (0xFFFF_F362)

	7	6	5	4	3	2	1	0
bit Symbol	DPE02	KEY022	KEY021	KEY020				KEY02EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY02 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY02 interrupt input 0: Disable 1: Enable

KWUPST03 (0xFFFF_F363)

	7	6	5	4	3	2	1	0
bit Symbol	DPE03	KEY032	KEY031	KEY030				KEY03EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY03 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY03 interrupt input 0: Disable 1: Enable

KWUPST04 (0xFFFF_F364)

	7	6	5	4	3	2	1	0
bit Symbol	DPE04	KEY042	KEY041	KEY040				KEY04EN
Read/Write	R/W				R			R/W
After reset	0		1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY04 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY04 interrupt input 0: Disable 1: Enable

KWUPST05 (0xFFFF_F365)

	7	6	5	4	3	2	1	0
bit Symbol	DPE05	KEY052	KEY051	KEY050				KEY05EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY05 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY05 interrupt input 0: Disable 1: Enable

KWUPST06 (0xFFFF_F366)

	7	6	5	4	3	2	1	0
bit Symbol	DPE06	KEY062	KEY061	KEY060				KEY06EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY06 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY06 interrupt input 0: Disable 1: Enable

KWUPST07 (0xFFFF_F367)

	7	6	5	4	3	2	1	0
bit Symbol	DPE07	KEY072	KEY071	KEY070				KEY07EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY07 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY07 interrupt input 0: Disable 1: Enable

KWUPST08 (0xFFFF_F368)

	7	6	5	4	3	2	1	0
bit Symbol	DPE08	KEY082	KEY081	KEY080				KEY08EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY08 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY08 interrupt input 0: Disable 1: Enable

KWUPST09 (0xFFFF_F369)

	7	6	5	4	3	2	1	0
bit Symbol	DPE09	KEY092	KEY091	KEY090				KEY09EN
Read/Write	R/W				R			R/W
After reset	0		1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY09 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY09 interrupt input 0: Disable 1: Enable

KWUPST10 (0xFFFF_F36A)

	7	6	5	4	3	2	1	0
bit Symbol	DPE10	KEY102	KEY101	KEY100				KEY10EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY10 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY10 interrupt input 0: Disable 1: Enable

KWUPST11 (0xFFFF_F36B)

	7	6	5	4	3	2	1	0
bit Symbol	DPE11	KEY112	KEY111	KEY110				KEY11EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY11 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY11 interrupt input 0: Disable 1: Enable

KWUPST12 (0xFFFF_F36C)

	7	6	5	4	3	2	1	0
bit Symbol	DPE12	KEY122	KEY121	KEY120				KEY12EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY12 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY12 interrupt input 0: Disable 1: Enable

KWUPST13 (0xFFFF_F36D)

	7	6	5	4	3	2	1	0
bit Symbol	DPE13	KEY132	KEY131	KEY130				KEY13EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY13 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY13 interrupt input 0: Disable 1: Enable

KWUPST14 (0xFFFF_F36E)

	7	6	5	4	3	2	1	0
bit Symbol	DPE14	KEY142	KEY141	KEY140				KEY14EN
Read/Write	R/W				R			R/W
After reset	0		1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY14 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY14 interrupt input 0: Disable 1: Enable

KWUPST15 (0xFFFF_F36F)

	7	6	5	4	3	2	1	0
bit Symbol	DPE15	KEY152	KEY151	KEY150				KEY15EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY15 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY15 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST16	bit Symbol	DPE16	KEY162	KEY161	KEY160				KEY16EN
(0xFFFF_F370)	Read/Write	R/W				R			R/W
	After reset	0	0	1	0	0			0
	Function	Pull-up 0:Static 1:Dynamic	Define the KEY16 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY16 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST17	bit Symbol	DPE17	KEY172	KEY171	KEY170				KEY17EN
(0xFFFF_F371)	Read/Write	R/W				R			R/W
	After reset	0	0	1	0	0			0
	Function	Pull-up 0:Static 1:Dynamic	Define the KEY17 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY17 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST18	bit Symbol	DPE18	KEY182	KEY181	KEY180				KEY18EN
(0xFFFF_F372)	Read/Write	R/W				R			R/W
	After reset	0	0	1	0	0			0
	Function	Pull-up 0:Static 1:Dynamic	Define the KEY18 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY18 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST19	bit Symbol	DPE19	KEY192	KEY191	KEY190				KEY19EN
(0xFFFF_F373)	Read/Write	R/W				R			R/W
	After reset	0		1	0	0			0
	Function	Pull-up 0:Static 1:Dynamic	Define the KEY19 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY19 interrupt input 0: Disable 1: Enable

KWUPST20 (0xFFFF_F374)

	7	6	5	4	3	2	1	0
bit Symbol	DPE20	KEY202	KEY201	KEY200				KEY20EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY20 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY20 interrupt input 0: Disable 1: Enable

KWUPST21 (0xFFFF_F375)

	7	6	5	4	3	2	1	0
bit Symbol	DPE21	KEY212	KEY211	KEY210				KEY21EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY21 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY21 interrupt input 0: Disable 1: Enable

KWUPST22 (0xFFFF_F376)

	7	6	5	4	3	2	1	0
bit Symbol	DPE22	KEY221	KEY221	KEY220				KEY22EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY22 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY22 interrupt input 0: Disable 1: Enable

KWUPST23 (0xFFFF_F377)

	7	6	5	4	3	2	1	0
bit Symbol	DPE23	KEY232	KEY231	KEY230				KEY23EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY23 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY23 interrupt input 0: Disable 1: Enable

KWUPST24 (0xFFFF_F378)

	7	6	5	4	3	2	1	0
bit Symbol	DPE24	KEY242	KEY241	KEY240				KEY24EN
Read/Write	R/W				R			R/W
After reset	0		1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY24 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY24 interrupt input 0: Disable 1: Enable

KWUPST25 (0xFFFF_F379)

	7	6	5	4	3	2	1	0
bit Symbol	DPE25	KEY252	KEY251	KEY250				KEY25EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY25 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY25 interrupt input 0: Disable 1: Enable

KWUPST26 (0xFFFF_F37A)

	7	6	5	4	3	2	1	0
bit Symbol	DPE26	KEY262	KEY261	KEY260				KEY26EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY26 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY26 interrupt input 0: Disable 1: Enable

KWUPST27 (0xFFFF_F37B)

	7	6	5	4	3	2	1	0
bit Symbol	DPE27	KEY272	KEY271	KEY270				KEY27EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY27 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY27 interrupt input 0: Disable 1: Enable

KWUPST28 (0xFFFF_F37C)

	7	6	5	4	3	2	1	0
bit Symbol	DPE28	KEY282	KEY281	KEY280				KEY28EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY28 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY28 interrupt input 0: Disable 1: Enable

KWUPST29 (0xFFFF_F37D)

	7	6	5	4	3	2	1	0
bit Symbol	DPE29	KEY292	KEY291	KEY290				KEY29EN
Read/Write	R/W				R			R/W
After reset	0		1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY29 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY29 interrupt input 0: Disable 1: Enable

KWUPST30 (0xFFFF_F37E)

	7	6	5	4	3	2	1	0
bit Symbol	DPE30	KEY302	KEY301	KEY300				KEY30EN
Read/Write	R/W				R			R/W
After reset	0	0	1	0	0			0
Function	Pull-up 0:Static 1:Dynamic	Define the KEY30 active state 000: "L" level 001: "H" level 010: Falling edge 011: Rising edge 100: Both edges			This can be read as "0."			KEY30 interrupt input 0: Disable 1: Enable

KWUPST31 (0xFFFF_F37F)

	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

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.

KWUPINT0 (0xFFFF_F388)

	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

KWUPINT1 (0xFFFF_F389)

	7	6	5	4	3	2	1	0
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

KWUPINT2 (0xFFFF_F38A)

	7	6	5	4	3	2	1	0
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

KWUPINT3 (0xFFFF_F38B)

	7	6	5	4	3	2	1	0
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 (0xFFFF_F385)

	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."			

21. ROM Correction Function

This chapter describes the ROM correction function built into the TMP19A43.

21.1 Features

- Using this function, eight pieces of one-word data or four pieces of eight-word data can be replaced.
- If an address (lower 5 or 2 bits are "don't care" bits) written to the address register matches an address generated by the PC or DMAC, ROM data is replaced by data generated by the ROM correction data register which is established in a RAM area assigned to the above address register.
- ROM correction is automatically authorized by writing an address to each address register.
- 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<20:5> 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	0xFFFF_E540	0xFFFF_DF60 - 0xFFFF_DF7C	8
ADDREG1	0xFFFF_E544	0xFFFF_DF80 - 0xFFFF_DF9C	8
ADDREG2	0xFFFF_E548	0xFFFF_DFA0 - 0xFFFF_DFBC	8
ADDREG3	0xFFFF_E54C	0xFFFF_DFC0 - 0xFFFF_DFDC	8
ADDREG4	0xFFFF_E550	0xFFFF_DFE0	1
ADDREG5	0xFFFF_E554	0xFFFF_DFE4	1
ADDREG6	0xFFFF_E558	0xFFFF_DFE8	1
ADDREG7	0xFFFF_E55C	0xFFFF_DFEC	1
ADDREG8	0xFFFF_E560	0xFFFF_DFF0	1
ADDREG9	0xFFFF_E564	0xFFFF_DFF4	1
ADDREGA	0xFFFF_E568	0xFFFF_DFF8	1
ADDREGB	0xFFFF_E56C	0xFFFF_DFFC	1

Note: To use the ROM correction function, the ROM must be unprotected.

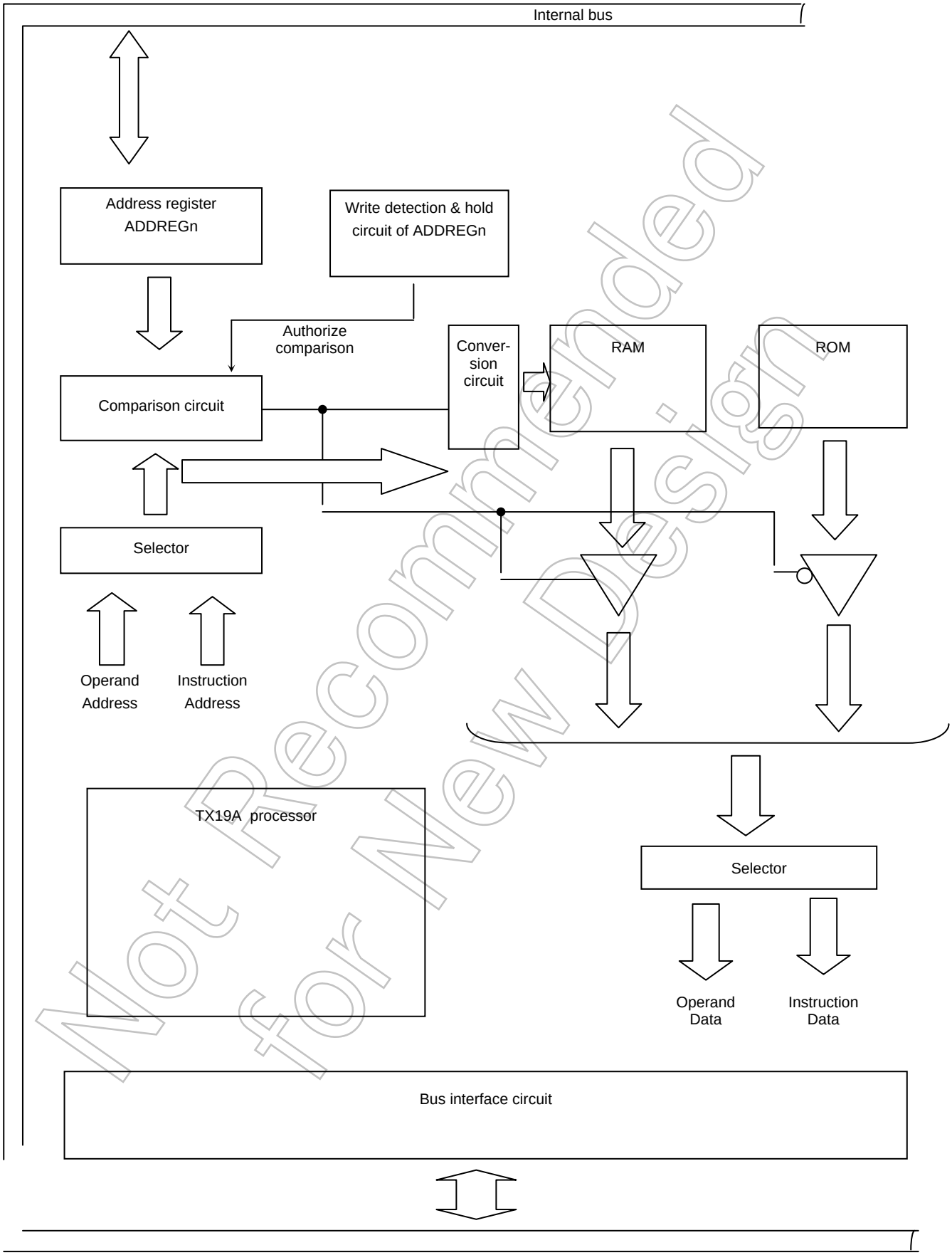


Fig. 21-1 ROM Correction System Diagram

21.3 Registers

(1) Address registers

ADDREG0 (0xFFFF_E540)

	7	6	5	4	3	2	1	0
bit Symbol	ADD07	ADD06	ADD05					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD015	ADD014	ADD013	ADD012	ADD011	ADD010	ADD09	ADD08
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD023	ADD022	ADD021	ADD020	ADD019	ADD018	ADD017	ADD016
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD031	ADD030	ADD029	ADD028	ADD027	ADD026	ADD025	ADD024
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG1 (0xFFFF_E544)

	7	6	5	4	3	2	1	0
bit Symbol	ADD17	ADD16	ADD15					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD115	ADD114	ADD113	ADD112	ADD111	ADD110	ADD19	ADD18
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD123	ADD122	ADD121	ADD120	ADD119	ADD118	ADD117	ADD116
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD131	ADD130	ADD129	ADD128	ADD127	ADD126	ADD125	ADD124
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG2 (0xFFFF_E548)

	7	6	5	4	3	2	1	0
bit Symbol	ADD27	ADD26	ADD25					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD215	ADD214	ADD213	ADD212	ADD211	ADD210	ADD29	ADD28
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD223	ADD222	ADD221	ADD220	ADD219	ADD218	ADD217	ADD216
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD231	ADD230	ADD229	ADD228	ADD227	ADD226	ADD225	ADD224
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG3 (0xFFFF_E54C)

	7	6	5	4	3	2	1	0
bit Symbol	ADD37	ADD36	ADD35					
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD315	ADD314	ADD313	ADD312	ADD311	ADD310	ADD39	ADD38
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD323	ADD322	ADD321	ADD320	ADD319	ADD318	ADD317	ADD316
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD331	ADD330	ADD329	ADD328	ADD327	ADD326	ADD325	ADD324
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG4 (0xFFFF_E550)

	7	6	5	4	3	2	1	0
bit Symbol	ADD47	ADD46	ADD45	ADD44	ADD43	ADD42		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD415	ADD414	ADD413	ADD412	ADD411	ADD410	ADD49	ADD48
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD423	ADD422	ADD421	ADD420	ADD419	ADD418	ADD417	ADD416
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD431	ADD430	ADD429	ADD428	ADD427	ADD426	ADD425	ADD424
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG5 (0xFFFF_E554)

	7	6	5	4	3	2	1	0
bit Symbol	ADD57	ADD56	ADD55	ADD54	ADD53	ADD52		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD515	ADD514	ADD513	ADD512	ADD511	ADD510	ADD59	ADD58
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD523	ADD522	ADD521	ADD520	ADD519	ADD518	ADD517	ADD516
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD531	ADD530	ADD529	ADD528	ADD527	ADD526	ADD525	ADD524
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG6 (0xFFFF_E558)

	7	6	5	4	3	2	1	0
bit Symbol	ADD67	ADD66	ADD65	ADD64	ADD63	ADD62		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD615	ADD614	ADD613	ADD612	ADD611	ADD610	ADD69	ADD68
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD623	ADD622	ADD621	ADD620	ADD619	ADD618	ADD617	ADD616
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD631	ADD630	ADD629	ADD628	ADD627	ADD626	ADD625	ADD624
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG7 (0xFFFF_E55C)

	7	6	5	4	3	2	1	0
bit Symbol	ADD77	ADD76	ADD75	ADD74	ADD73	ADD72		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD715	ADD714	ADD713	ADD712	ADD711	ADD710	ADD79	ADD78
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD723	ADD722	ADD721	ADD720	ADD719	ADD718	ADD717	ADD716
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD731	ADD730	ADD729	ADD728	ADD727	ADD726	ADD725	ADD724
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG8 (0xFFFF_E560)

	7	6	5	4	3	2	1	0
bit Symbol	ADD87	ADD86	ADD85	ADD84	ADD83	ADD82		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD815	ADD814	ADD813	ADD812	ADD811	ADD810	ADD89	ADD88
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD823	ADD822	ADD821	ADD820	ADD819	ADD818	ADD817	ADD816
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD831	ADD830	ADD829	ADD828	ADD827	ADD826	ADD825	ADD824
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREG9 (0xFFFF_E564)

	7	6	5	4	3	2	1	0
bit Symbol	ADD97	ADD96	ADD95	ADD94	ADD93	ADD92		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADD915	ADD914	ADD913	ADD912	ADD911	ADD910	ADD99	ADD98
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADD923	ADD922	ADD921	ADD920	ADD919	ADD918	ADD917	ADD916
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADD931	ADD930	ADD929	ADD928	ADD927	ADD926	ADD925	ADD924
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREGA (0xFFFF_E568)

	7	6	5	4	3	2	1	0
bit Symbol	ADDA7	ADDA6	ADDA5	ADDA4	ADDA3	ADDA2		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADDA15	ADDA14	ADDA13	ADDA12	ADDA11	ADDA10	ADDA9	ADDA8
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADDA23	ADDA22	ADDA21	ADDA20	ADDA19	ADDA18	ADDA17	ADDA16
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADDA31	ADDA30	ADDA29	ADDA28	ADDA27	ADDA26	ADDA25	ADDA24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

ADDREGB (0xFFFF_E56C)

	7	6	5	4	3	2	1	0
bit Symbol	ADDB7	ADDB6	ADDB5	ADDB4	ADDB3	ADDB2		
Read/Write	R/W						R	
After reset	0	0	0	0	0	0	1	1
	15	14	13	12	11	10	9	8
bit Symbol	ADDB15	ADDB14	ADDB13	ADDB12	ADDB11	ADDB10	ADDB9	ADDB8
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
bit Symbol	ADDB23	ADDB22	ADDB21	ADDB20	ADDB19	ADDB18	ADDB17	ADDB16
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
bit Symbol	ADDB31	ADDB30	ADDB29	ADDB28	ADDB27	ADDB26	ADDB25	ADDB24
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

(Note 1) Data cannot be transferred by DMA to the address register. However, data can be transferred by DMA to the RAM area where data for replacement is placed. The ROM correction function supports data replacement for both CPU and DMA access.

(Note 2) Writing back the initial value "0x00" allows data at the reset address to be replaced.

22. Table of Special Function Registers

Special function registers are allocated to an 8K-byte address space from FFFFE000H to FFFFFFFFH.

[1] Port registers
[2] Watchdog timer
[3] 16-bit timer
[4] I^2CBUS/serial channel
[5] UART/serial channel
[6] 10-bit A/D converter
[7] 8-bit D/A converter
[8] Key-on wake-up
[9] 32-bit input capture
[10] 32-bit compare
[11] Interrupt controller
[12] DMA controller
[13] Chip select/wait controller
[14] FLASH control
[15] ROM correction
[16] Clock timer
[17] UART/high-speed serial channel
[18] Clock generator

(Note) 0xFFFF_F000 to 0xFFFF_FFFF are a little-endian area.
0xFFFF_E000 to 0xFFFF_EFFF are a bi-endian area.

(Note) For continuous 8-bit long registers, 16- or 32-bit access is possible. The use of 16- or 32-bit access requires that an even-number address be accessed and that an even-number address does not contain undefined areas.

Big-endian

[1] PORT registers

ADR	Register name
FFFFF000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	
7H	
8H	
9H	
AH	
BH	
CH	P0PE
DH	P1PE
EH	
FH	

ADR	Register name
FFFFF010H	
1H	
2H	P2
3H	
4H	P2CR
5H	P2FC
6H	P2FC2
7H	
8H	P3
9H	
AH	P3CR
BH	P3FC
CH	P2PE
DH	P3PE
EH	P4
FH	

ADR	Register name
FFFFF020H	P4CR
1H	P4FC
2H	
3H	
4H	
5H	P4PE
6H	P5PE
7H	P6PE
8H	P5
9H	P6
AH	
BH	
CH	P5CR
DH	P5FC
EH	P6CR
FH	P6FC

ADR	Register name
FFFFF030H	P6ODE
1H	P9ODE
2H	
3H	
4H	PBODE
5H	PCODE
6H	PDODE
7H	
8H	
9H	
AH	
BH	
CH	P5FC2
DH	P6FC2
EH	reserved
FH	reserved

ADR	Register name
FFFFF040H	P7
1H	P8
2H	P9
3H	PA
4H	–
5H	–
6H	P9CR
7H	PACR
8H	P7FC
9H	P8FC
AH	P9FC
BH	PAFC
CH	P7PE
DH	P8PE
EH	P9PE
FH	PAPE

ADR	Register name
FFFFF050H	PB
1H	PC
2H	PD
3H	PE
4H	PBCR
5H	PCCR
6H	PDCR
7H	PECR
8H	PBFC
9H	PCFC
AH	PDFC
BH	PEFC
CH	PBPE
DH	PCPE
EH	PDPE
FH	PEPE

ADR	Register name
FFFFF060H	PF
1H	PG
2H	PH
3H	
4H	PFCR
5H	PGCR
6H	PHCR
7H	
8H	PFFC
9H	PGFC
AH	–
BH	–
CH	PFPE
DH	PGPE
EH	PHPE
FH	

ADR	Register name
FFFFF070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0C0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-endian

[2] WDT

ADR	Register name
FFFFF080H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF090H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] 16-bit timer

ADR	Register name
FFFFF1 40H	TB0RUN
1H	TB0CR
2H	TB0MOD
3H	TB0FFCR
4H	TB0ST
5H	
6H	TB0UCL
7H	TB0UCH
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

ADR	Register name
FFFFF150H	TB1RUN
1H	TB1CR
2H	TB1MOD
3H	TB1FFCR
4H	TB1ST
5H	
6H	TB1UCL
7H	TB1UCH
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

ADR	Register name
FFFFF160H	TB2RUN
1H	TB2CR
2H	TB2MOD
3H	TB2FFCR
4H	TB2ST
5H	
6H	TB2UCL
7H	TB2UCH
8H	TB2RG0L
9H	TB2RG0H
AH	TB2RG1L
BH	TB2RG1H
CH	TB2CP0L
DH	TB2CP0H
EH	TB2CP1L
FH	TB2CP1H

ADR	Register name
FFFFF170H	TB3RUN
1H	TB3CR
2H	TB3MOD
3H	TB3FFCR
4H	TB3ST
5H	
6H	TB3UCL
7H	TB3UCH
8H	TB3RG0L
9H	TB3RG0H
AH	TB3RG1L
BH	TB3RG1H
CH	TB3CP0L
DH	TB3CP0H
EH	TB3CP1L
FH	TB3CP1H

ADR	Register name
FFFFF180H	TB4RUN
1H	TB4CR
2H	TB4MOD
3H	TB4FFCR
4H	TB4ST
5H	
6H	TB4UCL
7H	TB4UCH
8H	TB4RG0L
9H	TB4RG0H
AH	TB4RG1L
BH	TB4RG1H
CH	TB4CP0L
DH	TB4CP0H
EH	TB4CP1L
FH	TB4CP1H

ADR	Register name
FFFFF190H	TB5RUN
1H	TB5CR
2H	TB5MOD
3H	TB5FFCR
4H	TB5ST
5H	
6H	TB5UCL
7H	TB5UCH
8H	TB5RG0L
9H	TB5RG0H
AH	TB5RG1L
BH	TB5RG1H
CH	TB5CP0L
DH	TB5CP0H
EH	TB5CP1L
FH	TB5CP1H

ADR	Register name
FFFFF1A0H	TB6RUN
1H	TB6CR
2H	TB6MOD
3H	TB6FFCR
4H	TB6ST
5H	
6H	TB6UCL
7H	TB6UCH
8H	TB6RG0L
9H	TB6RG0H
AH	TB6RG1L
BH	TB6RG1H
CH	TB6CP0L
DH	TB6CP0H
EH	TB6CP1L
FH	TB6CP1H

ADR	Register name
FFFFF1B0H	TB7RUN
1H	TB7CR
2H	TB7MOD
3H	TB7FFCR
4H	TB7ST
5H	
6H	TB7UCL
7H	TB7UCH
8H	TB7RG0L
9H	TB7RG0H
AH	TB7RG1L
BH	TB7RG1H
CH	TB7CP0L
DH	TB7CP0H
EH	TB7CP1L
FH	TB7CP1H

Big-endian

ADR	Register name
FFFFF1C0H	TB8RUN
1H	TB8CR
2H	TB8MOD
3H	TB8FFCR
4H	TB8ST
5H	
6H	TB8UCL
7H	TB8UCH
8H	TB8RG0L
9H	TB8RG0H
AH	TB8RG1L
BH	TB8RG1H
CH	TB8CP0L
DH	TB8CP0H
EH	TB8CP1L
FH	TB8CP1H

ADR	Register name
FFFFF1D0H	TB9RUN
1H	TB9CR
2H	TB9MOD
3H	TB9FFCR
4H	TB9ST
5H	
6H	TB9UCL
7H	TB9UCH
8H	TB9RG0L
9H	TB9RG0H
AH	TB9RG1L
BH	TB9RG1H
CH	TB9CP0L
DH	TB9CP0H
EH	TB9CP1L
FH	TB9CP1H

ADR	Register name
FFFFF1E0H	TBARUN
1H	TBACR
2H	TBAMOD
3H	TBAFFCR
4H	TBAST
5H	
6H	TBAUCL
7H	TBAUCH
8H	TBARG0L
9H	TBARG0H
AH	TBARG1L
BH	TBARG1H
CH	TBACP0L
DH	TBACP0H
EH	TBACP1L
FH	TBACP1H

ADR	Register name
FFFFF1F0H	TBBRUN
1H	TBBCR
2H	TBBMOD
3H	TBBFFCR
4H	TBBST
5H	
6H	TBBUCL
7H	TBBUCH
8H	TBBRG0L
9H	TBBRG0H
AH	TBBRG1L
BH	TBBRG1H
CH	TBBCP0L
DH	TBBCP0H
EH	TBBCP1L
FH	TBBCP1H

ADR	Register name
FFFFF200H	TBCRUN
1H	TBCCR
2H	TBCMOD
3H	TBCFFCR
4H	TBCST
5H	
6H	TBCUCL
7H	TBCUCH
8H	TBCRG0L
9H	TBCRG0H
AH	TBCRG1L
BH	TBCRG1H
CH	TBCCP0L
DH	TBCCP0H
EH	TBCCP1L
FH	TBCCP1H

ADR	Register name
FFFFF210H	TBDRUN
1H	TBDCR
2H	TBDMOD
3H	TBDFFCR
4H	TBDST
5H	
6H	TBDUCL
7H	TBDUCH
8H	TBDRG0L
9H	TBDRG0H
AH	TBDRG1L
BH	TBDRG1H
CH	TBDCP0L
DH	TBDCP0H
EH	TBDCP1L
FH	TBDCP1H

ADR	Register name
FFFFF220H	TBERUN
1H	TBECR
2H	TBEMOD
3H	TBEFFCR
4H	TBEST
5H	
6H	TBEUCL
7H	TBEUCH
8H	TBERG0L
9H	TBERG0H
AH	TBERG1L
BH	TBERG1H
CH	TBECP0L
DH	TBECP0H
EH	TBECP1L
FH	TBECP1H

ADR	Register name
FFFFF230H	TBFRUN
1H	TBFCR
2H	TBFMOD
3H	TBFFFCR
4H	TBFST
5H	
6H	TBFUCL
7H	TBFUCH
8H	TBFRG0L
9H	TBFRG0H
AH	TBFRG1L
BH	TBFRG1H
CH	TBFCP0L
DH	TBFCP0H
EH	TBFCP1L
FH	TBFCP1H

Big-endian

[4] I2C/SIO

ADR	Register name
FFFFF250H	SBICR1
1H	SBIDBR
2H	I2CAR
3H	SBICR2/SR
4H	SBIBR0
5H	
6H	
7H	SBICR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] UART/SIO

ADR	Register name
FFFFF260H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	SC0MOD2
7H	SC0EN
8H	SC0RFC
9H	SC0TFC
AH	SC0RST
BH	SC0TST
CH	SC0FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF270H	SC1BUF
1H	SC1CR
2H	SC1MOD0
3H	BR1CR
4H	BR1ADD
5H	SC1MOD1
6H	SC1MOD2
7H	SC1EN
8H	SC1RFC
9H	SC1TFC
AH	SC1RST
BH	SC1TST
CH	SC1FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF280H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	SC2MOD2
7H	SC2EN
8H	SC2RFC
9H	SC2TFC
AH	SC2RST
BH	SC2TST
CH	SC2FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF290H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF2A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF2B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF2C0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[6] 10-bit ADC

ADR	Register name
FFFFF300H	ADREG08L
1H	ADREG08H
2H	ADREG19L
3H	ADREG19H
4H	ADREG2AL
5H	ADREG2AH
6H	ADREG3BL
7H	ADREG3BH
8H	ADREG4CL
9H	ADREG4CH
AH	ADREG5DL
BH	ADREG5DH
CH	ADREG6EL
DH	ADREG6EH
EH	ADREG7FL
FH	ADREG7FH

ADR	Register name
FFFFF310H	ADREGSPL
1H	ADREGSPH
2H	ADCOMREGL
3H	ADCOMREGH
4H	ADMOD0
5H	ADMOD1
6H	ADMOD2
7H	ADMOD3
8H	ADMOD4
9H	ADCBAS
AH	Reserved
BH	Reserved
CH	ADCLK
DH	
EH	
FH	

[7] 8-bit DAC

ADR	Register name
FFFFF330H	DACCNT0
1H	DAREG0
2H	
3H	
4H	
5H	
6H	
7H	Reserved
8H	DACCNT1
9H	DAREG1
AH	
BH	
CH	
DH	
EH	
FH	Reserved

ADR	Register name
FFFFF340H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-endian

[8] KWUP

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FFFFF360H	KWUPST00	FFFFF370H	KWUPST16	FFFFF380H	PKEY0	FFFFF390H	
1H	KWUPST01	1H	KWUPST17	1H	PKEY1	1H	
2H	KWUPST02	2H	KWUPST18	2H	PKEY2	2H	
3H	KWUPST03	3H	KWUPST19	3H	PKEY3	3H	
4H	KWUPST04	4H	KWUPST20	4H	KWUPCNT	4H	
5H	KWUPST05	5H	KWUPST21	5H	KWUPCLR	5H	
6H	KWUPST06	6H	KWUPST22	6H		6H	
7H	KWUPST07	7H	KWUPST23	7H		7H	
8H	KWUPST08	8H	KWUPST24	8H	KWUPINT0	8H	
9H	KWUPST09	9H	KWUPST25	9H	KWUPINT1	9H	
AH	KWUPST10	AH	KWUPST26	AH	KWUPINT2	AH	
BH	KWUPST11	BH	KWUPST27	BH	KWUPINT3	BH	
CH	KWUPST12	CH	KWUPST28	CH		CH	
DH	KWUPST13	DH	KWUPST29	DH		DH	
EH	KWUPST14	EH	KWUPST30	EH		EH	
FH	KWUPST15	FH	KWUPST31	FH		FH	

[9] 32-bit input capture

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FFFFF 400H	TCCR	FFFFF410H	CAP0CR	FFFFF420H	CAP2CR	FFFFF430H	
1H	TBTRUN	1H		1H		1H	
2H	TBTCR	2H		2H		2H	
3H		3H		3H		3H	
4H	TBTCAP0	4H	TCCAP0LL	4H	TCCAP2LL	4H	
5H	TBTCAP1	5H	TCCAP0LH	5H	TCCAP2LH	5H	
6H	TBTCAP2	6H	TCCAP0HL	6H	TCCAP2HL	6H	
7H	TBTCAP3	7H	TCCAP0HH	7H	TCCAP2HH	7H	
8H	TBTRDCAPLL	8H	CAP1CR	8H	CAP3CR	8H	
9H	TBTRDCAPLH	9H		9H		9H	
AH	TBTRDCAPHL	AH		AH		AH	
BH	TBTRDCAPHH	BH		BH		BH	
CH	TCG0IM	CH	TCCAP1LL	CH	TCCAP3LL	CH	
DH	TCG0ST	DH	TCCAP1LH	DH	TCCAP3LH	DH	
EH	Reserved	EH	TCCAP1HL	EH	TCCAP3HL	EH	
FH	Reserved	FH	TCCAP1HH	FH	TCCAP3HH	FH	

[10] 32-bit output compare

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FFFFF 440H	TCCMP0LL	FFFFF450H	TCCMP4LL	FFFFF460H		FFFFF470H	CMPCTL0
1H	TCCMP0LH	1H	TCCMP4LH	1H		1H	CMPCTL1
2H	TCCMP0HL	2H	TCCMP4HL	2H		2H	CMPCTL2
3H	TCCMP0HH	3H	TCCMP4HH	3H		3H	CMPCTL3
4H	TCCMP1LL	4H	TCCMP5LL	4H		4H	CMPCTL4
5H	TCCMP1LH	5H	TCCMP5LH	5H		5H	CMPCTL5
6H	TCCMP1HL	6H	TCCMP5HL	6H		6H	CMPCTL6
7H	TCCMP1HH	7H	TCCMP5HH	7H		7H	CMPCTL7
8H	TCCMP2LL	8H	TCCMP6LL	8H		8H	
9H	TCCMP2LH	9H	TCCMP6LH	9H		9H	
AH	TCCMP2HL	AH	TCCMP6HL	AH		AH	
BH	TCCMP2HH	BH	TCCMP6HH	BH		BH	
CH	TCCMP3LL	CH	TCCMP7LL	CH		CH	
DH	TCCMP3LH	DH	TCCMP7LH	DH		DH	
EH	TCCMP3HL	EH	TCCMP7HL	EH		EH	
FH	TCCMP3HH	FH	TCCMP7HH	FH		FH	

Big-endian

[11] INTC

ADR	Register name
FFFFE000H	IMC0
1H	ditto
2H	ditto
3H	ditto
4H	IMC1
5H	ditto
6H	ditto
7H	ditto
8H	IMC2
9H	ditto
AH	ditto
BH	ditto
CH	IMC3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE010H	IMC4
1H	ditto
2H	ditto
3H	ditto
4H	IMC5
5H	ditto
6H	ditto
7H	ditto
8H	IMC6
9H	ditto
AH	ditto
BH	ditto
CH	IMC7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE020H	IMC8
1H	ditto
2H	ditto
3H	ditto
4H	IMC9
5H	ditto
6H	ditto
7H	ditto
8H	IMCA
9H	ditto
AH	ditto
BH	ditto
CH	IMCB
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE030H	IMCC
1H	ditto
2H	ditto
3H	ditto
4H	IMCD
5H	ditto
6H	ditto
7H	ditto
8H	IMCE
9H	ditto
AH	ditto
BH	ditto
CH	IMCF
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE040H	IVR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE050H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE060H	INTCLR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE100H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	ILEV
DH	ditto
EH	ditto
FH	ditto

Big-endian

[12] DMAC

ADR	Register name
FFFFE200H	CCR0
1H	ditto
2H	ditto
3H	ditto
4H	CSR0
5H	ditto
6H	ditto
7H	ditto
8H	SAR0
9H	ditto
AH	ditto
BH	ditto
CH	DAR0
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE210H	BCR0
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR0
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE220H	CCR1
1H	ditto
2H	ditto
3H	ditto
4H	CSR1
5H	ditto
6H	ditto
7H	ditto
8H	SAR1
9H	ditto
AH	ditto
BH	ditto
CH	DAR1
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE230H	BCR1
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR1
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE240H	CCR2
1H	ditto
2H	ditto
3H	ditto
4H	CSR2
5H	ditto
6H	ditto
7H	ditto
8H	SAR2
9H	ditto
AH	ditto
BH	ditto
CH	DAR2
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE250H	BCR2
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR2
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE260H	CCR3
1H	ditto
2H	ditto
3H	ditto
4H	CSR3
5H	ditto
6H	ditto
7H	ditto
8H	SAR3
9H	ditto
AH	ditto
BH	ditto
CH	DAR3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE270H	BCR3
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR3
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE280H	CCR4
1H	ditto
2H	ditto
3H	ditto
4H	CSR4
5H	ditto
6H	ditto
7H	ditto
8H	SAR4
9H	ditto
AH	ditto
BH	ditto
CH	DAR4
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE290H	BCR4
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR4
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2A0H	CCR5
1H	ditto
2H	ditto
3H	ditto
4H	CSR5
5H	ditto
6H	ditto
7H	ditto
8H	SAR5
9H	ditto
AH	ditto
BH	ditto
CH	DAR5
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2B0H	BCR5
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR5
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

Big-endian

ADR	Register name
FFFFE2C0H	CCR6
1H	ditto
2H	ditto
3H	ditto
4H	CSR6
5H	ditto
6H	ditto
7H	ditto
8H	SAR6
9H	ditto
AH	ditto
BH	ditto
CH	DAR6
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2D0H	BCR6
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR6
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2E0H	CCR7
1H	ditto
2H	ditto
3H	ditto
4H	CSR7
5H	ditto
6H	ditto
7H	ditto
8H	SAR7
9H	ditto
AH	ditto
BH	ditto
CH	DAR7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2F0H	BCR7
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR7
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE300H	DCR
1H	ditto
2H	ditto
3H	ditto
4H	RSR
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	DHR
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE310H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE320H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE330H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE340H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE350H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE360H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE370H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-endian

[13] CS/WAIT controller

ADR	Register name
FFFFE400H	BMA0
1H	ditto
2H	ditto
3H	ditto
4H	BMA1
5H	ditto
6H	ditto
7H	ditto
8H	BMA2
9H	ditto
AH	ditto
BH	ditto
CH	BMA3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE410H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE480H	B01CS
1H	ditto
2H	ditto
3H	ditto
4H	B23CS
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	
DH	
EH	BEXCS
FH	ditto

ADR	Register name
FFFFE490H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] FLASH control

ADR	Register name
FFFFE510H	SEQMOD
1H	ditto
2H	ditto
3H	ditto
4H	SEQCNT
5H	ditto
6H	ditto
7H	ditto
8H	ROMSEC1
9H	ditto
AH	ditto
BH	ditto
CH	ROMSEC2
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE520H	FLCS
1H	ditto
2H	ditto
3H	ditto
4H	Reserved
5H	Reserved
6H	Reserved
7H	Reserved
8H	Reserved
9H	Reserved
AH	Reserved
BH	Reserved
CH	
DH	
EH	
FH	

ADR	Register na
FFFFE620H	
1H	
2H	
3H	
4H	Reserved
5H	Reserved
6H	Reserved
7H	Reserved
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Attention

[15] ROM correction

ADR	Register name
FFFFE540H	ADDREG0
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG1
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG2
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE550H	ADDREG4
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG5
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG6
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE560H	ADDREG8
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG9
5H	ditto
6H	ditto
7H	ditto
8H	ADDREGA
9H	ditto
AH	ditto
BH	ditto
CH	ADDREGB
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE570H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Big-endian

[16] Clock timer

ADR	Register name
FFFFE700H	
1H	
2H	
3H	
4H	RTCCR
5H	ditto
6H	ditto
7H	ditto
8H	RTCREG
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE710H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] UART/HSIO

ADR	Register name
FFFFE800H	
1H	
2H	
3H	HSC0BUF
4H	HSC0EN
5H	HSC0MOD2
6H	HSC0MOD1
7H	HBR0ADD
8H	HSC0TST
9H	HSC0RST
AH	HSC0TFC
BH	HSC0RFC
CH	HBR0CR
DH	HSC0MOD0
EH	HSC0CR
FH	HSC0FCNF

ADR	Register name
FFFFE810H	
1H	
2H	
3H	HSC1BUF
4H	HSC1EN
5H	HSC1MOD2
6H	HSC1MOD1
7H	HBR1ADD
8H	HSC1TST
9H	HSC1RST
AH	HSC1TFC
BH	HSC1RFC
CH	HBR1CR
DH	HSC1MOD0
EH	HSC1CR
FH	HSC1FCNF

ADR	Register name
FFFFE820H	
1H	
2H	
3H	HSC2BUF
4H	HSC2EN
5H	HSC2MOD2
6H	HSC2MOD1
7H	HBR2ADD
8H	HSC2TST
9H	HSC2RST
AH	HSC2TFC
BH	HSC2RFC
CH	HBR2CR
DH	HSC2MOD0
EH	HSC2CR
FH	HSC2FCNF

ADR	Register name
FFFFE840H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[18] CG

ADR	Register name
FFFFEE00H	SYSCR3
1H	SYSCR2
2H	SYSCR1
3H	SYSCR0
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFEE10H	IMCGA
1H	ditto
2H	ditto
3H	ditto
4H	IMCGB
5H	ditto
6H	ditto
7H	ditto
8H	IMCGC
9H	ditto
AH	ditto
BH	ditto
CH	IMCGD
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFEE20H	EICRCG
1H	ditto
2H	ditto
3H	ditto
4H	NMIFLG
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFEE40H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-endian

[1] PORT registers

ADR	Register name
FFFFF000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	
7H	
8H	
9H	
AH	
BH	
CH	P0PE
DH	P1PE
EH	
FH	

ADR	Register name
FFFFF010H	
1H	
2H	P2
3H	
4H	P2CR
5H	P2FC
6H	P2FC2
7H	
8H	P3
9H	
AH	P3CR
BH	P3FC
CH	P2PE
DH	P3PE
EH	P4
FH	

ADR	Register name
FFFFF020H	P4CR
1H	P4FC
2H	
3H	
4H	
5H	P4PE
6H	P5PE
7H	P6PE
8H	P5
9H	P6
AH	
BH	
CH	P5CR
DH	P5FC
EH	P6CR
FH	P6FC

ADR	Register name
FFFFF030H	P6ODE
1H	P9ODE
2H	
3H	
4H	PBODE
5H	PCODE
6H	PDODE
7H	
8H	
9H	
AH	
BH	
CH	P5FC2
DH	P6FC2
EH	reserved
FH	reserved

ADR	Register name
FFFFF040H	P7
1H	P8
2H	P9
3H	PA
4H	–
5H	–
6H	P9CR
7H	PACR
8H	P7FC
9H	P8FC
AH	P9FC
BH	PAFC
CH	P7PE
DH	P8PE
EH	P9PE
FH	PAPE

ADR	Register name
FFFFF050H	PB
1H	PC
2H	PD
3H	PE
4H	PBCR
5H	PCCR
6H	PDCR
7H	PECR
8H	PBFC
9H	PCFC
AH	PDFC
BH	PEFC
CH	PBPE
DH	PCPE
EH	PDPE
FH	PEPE

ADR	Register name
FFFFF060H	PF
1H	PG
2H	PH
3H	
4H	PFCR
5H	PGCR
6H	PHCR
7H	
8H	PFFC
9H	PGFC
AH	–
BH	–
CH	PFPE
DH	PGPE
EH	PHPE
FH	

ADR	Register name
FFFFF070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0C0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0D0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0E0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-endian

[2] WDT

ADR	Register name
FFFFF080H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF090H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF0B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] 16-bit timer

ADR	Register name
FFFFF1 40H	TB0RUN
1H	TB0CR
2H	TB0MOD
3H	TB0FFCR
4H	TB0ST
5H	
6H	TB0UCL
7H	TB0UCH
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

ADR	Register name
FFFFF150H	TB1RUN
1H	TB1CR
2H	TB1MOD
3H	TB1FFCR
4H	TB1ST
5H	
6H	TB1UCL
7H	TB1UCH
8H	TB1RG0L
9H	TB1RG0H
AH	TB1RG1L
BH	TB1RG1H
CH	TB1CP0L
DH	TB1CP0H
EH	TB1CP1L
FH	TB1CP1H

ADR	Register name
FFFFF160H	TB2RUN
1H	TB2CR
2H	TB2MOD
3H	TB2FFCR
4H	TB2ST
5H	
6H	TB2UCL
7H	TB2UCH
8H	TB2RG0L
9H	TB2RG0H
AH	TB2RG1L
BH	TB2RG1H
CH	TB2CP0L
DH	TB2CP0H
EH	TB2CP1L
FH	TB2CP1H

ADR	Register name
FFFFF170H	TB3RUN
1H	TB3CR
2H	TB3MOD
3H	TB3FFCR
4H	TB3ST
5H	
6H	TB3UCL
7H	TB3UCH
8H	TB3RG0L
9H	TB3RG0H
AH	TB3RG1L
BH	TB3RG1H
CH	TB3CP0L
DH	TB3CP0H
EH	TB3CP1L
FH	TB3CP1H

ADR	Register name
FFFFF180H	TB4RUN
1H	TB4CR
2H	TB4MOD
3H	TB4FFCR
4H	TB4ST
5H	
6H	TB4UCL
7H	TB4UCH
8H	TB4RG0L
9H	TB4RG0H
AH	TB4RG1L
BH	TB4RG1H
CH	TB4CP0L
DH	TB4CP0H
EH	TB4CP1L
FH	TB4CP1H

ADR	Register name
FFFFF190H	TB5RUN
1H	TB5CR
2H	TB5MOD
3H	TB5FFCR
4H	TB5ST
5H	
6H	TB5UCL
7H	TB5UCH
8H	TB5RG0L
9H	TB5RG0H
AH	TB5RG1L
BH	TB5RG1H
CH	TB5CP0L
DH	TB5CP0H
EH	TB5CP1L
FH	TB5CP1H

ADR	Register name
FFFFF1A0H	TB6RUN
1H	TB6CR
2H	TB6MOD
3H	TB6FFCR
4H	TB6ST
5H	
6H	TB6UCL
7H	TB6UCH
8H	TB6RG0L
9H	TB6RG0H
AH	TB6RG1L
BH	TB6RG1H
CH	TB6CP0L
DH	TB6CP0H
EH	TB6CP1L
FH	TB6CP1H

ADR	Register name
FFFFF1B0H	TB7RUN
1H	TB7CR
2H	TB7MOD
3H	TB7FFCR
4H	TB7ST
5H	
6H	TB7UCL
7H	TB7UCH
8H	TB7RG0L
9H	TB7RG0H
AH	TB7RG1L
BH	TB7RG1H
CH	TB7CP0L
DH	TB7CP0H
EH	TB7CP1L
FH	TB7CP1H

Little-endian

ADR	Register name
FFFFF1C0H	TB8RUN
1H	TB8CR
2H	TB8MOD
3H	TB8FFCR
4H	TB8ST
5H	
6H	TB8UCL
7H	TB8UCH
8H	TB8RG0L
9H	TB8RG0H
AH	TB8RG1L
BH	TB8RG1H
CH	TB8CP0L
DH	TB8CP0H
EH	TB8CP1L
FH	TB8CP1H

ADR	Register name
FFFFF1D0H	TB9RUN
1H	TB9CR
2H	TB9MOD
3H	TB9FFCR
4H	TB9ST
5H	
6H	TB9UCL
7H	TB9UCH
8H	TB9RG0L
9H	TB9RG0H
AH	TB9RG1L
BH	TB9RG1H
CH	TB9CP0L
DH	TB9CP0H
EH	TB9CP1L
FH	TB9CP1H

ADR	Register name
FFFFF1E0H	TBARUN
1H	TBACR
2H	TBAMOD
3H	TBAFFCR
4H	TBAST
5H	
6H	TBAUCL
7H	TBAUCH
8H	TBARG0L
9H	TBARG0H
AH	TBARG1L
BH	TBARG1H
CH	TBACP0L
DH	TBACP0H
EH	TBACP1L
FH	TBACP1H

ADR	Register name
FFFFF1F0H	TBBRUN
1H	TBBCR
2H	TBBMOD
3H	TBBFFCR
4H	TBBST
5H	
6H	TBBUCL
7H	TBBUCH
8H	TBBRG0L
9H	TBBRG0H
AH	TBBRG1L
BH	TBBRG1H
CH	TBBCP0L
DH	TBBCP0H
EH	TBBCP1L
FH	TBBCP1H

ADR	Register name
FFFFF200H	TBCRUN
1H	TBCCR
2H	TBCMOD
3H	TBCFFCR
4H	TBCST
5H	
6H	TBCUCL
7H	TBCUCH
8H	TBCRG0L
9H	TBCRG0H
AH	TBCRG1L
BH	TBCRG1H
CH	TBCCP0L
DH	TBCCP0H
EH	TBCCP1L
FH	TBCCP1H

ADR	Register name
FFFFF210H	TBDRUN
1H	TBDCR
2H	TBDMOD
3H	TBDFFCR
4H	TBDST
5H	
6H	TBDUCL
7H	TBDUCH
8H	TBDRG0L
9H	TBDRG0H
AH	TBDRG1L
BH	TBDRG1H
CH	TBDCP0L
DH	TBDCP0H
EH	TBDCP1L
FH	TBDCP1H

ADR	Register name
FFFFF220H	TBERUN
1H	TBECR
2H	TBEMOD
3H	TBEFFCR
4H	TBEST
5H	
6H	TBEUCL
7H	TBEUCH
8H	TBERG0L
9H	TBERG0H
AH	TBERG1L
BH	TBERG1H
CH	TBECP0L
DH	TBECP0H
EH	TBECP1L
FH	TBECP1H

ADR	Register name
FFFFF230H	TBFRUN
1H	TBFCR
2H	TBFMOD
3H	TBFFFCR
4H	TBFST
5H	
6H	TBFUCL
7H	TBFUCH
8H	TBFRG0L
9H	TBFRG0H
AH	TBFRG1L
BH	TBFRG1H
CH	TBFCP0L
DH	TBFCP0H
EH	TBFCP1L
FH	TBFCP1H

Little-endian

[4] I2C/SIO

ADR	Register name
FFFFF250H	SBICR1
1H	SBIDBR
2H	I2CAR
3H	SBICR2/SR
4H	SBIBR0
5H	
6H	
7H	SBICR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] UART/SIO

ADR	Register name
FFFFF260H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	SC0MOD2
7H	SC0EN
8H	SC0RFC
9H	SC0TFC
AH	SC0RST
BH	SC0TST
CH	SC0FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF270H	SC1BUF
1H	SC1CR
2H	SC1MOD0
3H	BR1CR
4H	BR1ADD
5H	SC1MOD1
6H	SC1MOD2
7H	SC1EN
8H	SC1RFC
9H	SC1TFC
AH	SC1RST
BH	SC1TST
CH	SC1FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF280H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	SC2MOD2
7H	SC2EN
8H	SC2RFC
9H	SC2TFC
AH	SC2RST
BH	SC2TST
CH	SC2FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF290H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF2A0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF2B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF2C0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[6] 10-bit ADC

ADR	Register name
FFFFF300H	ADREG08L
1H	ADREG08H
2H	ADREG19L
3H	ADREG19H
4H	ADREG2AL
5H	ADREG2AH
6H	ADREG3BL
7H	ADREG3BH
8H	ADREG4CL
9H	ADREG4CH
AH	ADREG5DL
BH	ADREG5DH
CH	ADREG6EL
DH	ADREG6EH
EH	ADREG7FL
FH	ADREG7FH

ADR	Register name
FFFFF310H	ADREGSPL
1H	ADREGSPH
2H	ADCOMREGL
3H	ADCOMREGH
4H	ADMOD0
5H	ADMOD1
6H	ADMOD2
7H	ADMOD3
8H	ADMOD4
9H	ADCBAS
AH	Reserved
BH	Reserved
CH	ADCLK
DH	
EH	
FH	

[7] 8-bit ADC

ADR	Register name
FFFFF330H	DACCNT0
1H	DAREG0
2H	
3H	
4H	
5H	
6H	
7H	Reserved
8H	DACCNT1
9H	DAREG1
AH	
BH	
CH	
DH	
EH	
FH	Reserved

ADR	Register name
FFFFF340H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-endian

[8] KWUP

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FFFFF360H	KWUPST00	FFFFF370H	KWUPST16	FFFFF380H	PKEY0	FFFFF390H	
1H	KWUPST01	1H	KWUPST17	1H	PKEY1	1H	
2H	KWUPST02	2H	KWUPST18	2H	PKEY2	2H	
3H	KWUPST03	3H	KWUPST19	3H	PKEY3	3H	
4H	KWUPST04	4H	KWUPST20	4H	KWUPCNT	4H	
5H	KWUPST05	5H	KWUPST21	5H	KWUPCLR	5H	
6H	KWUPST06	6H	KWUPST22	6H		6H	
7H	KWUPST07	7H	KWUPST23	7H		7H	
8H	KWUPST08	8H	KWUPST24	8H	KWUPINT0	8H	
9H	KWUPST09	9H	KWUPST25	9H	KWUPINT1	9H	
AH	KWUPST10	AH	KWUPST26	AH	KWUPINT2	AH	
BH	KWUPST11	BH	KWUPST27	BH	KWUPINT3	BH	
CH	KWUPST12	CH	KWUPST28	CH		CH	
DH	KWUPST13	DH	KWUPST29	DH		DH	
EH	KWUPST14	EH	KWUPST30	EH		EH	
FH	KWUPST15	FH	KWUPST31	FH		FH	

[9] 32-bit input capture

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FFFFF 400H	TCCR	FFFFF410H	CAP0CR	FFFFF420H	CAP2CR	FFFFF430H	
1H	TBTRUN	1H		1H		1H	
2H	TBTCR	2H		2H		2H	
3H		3H		3H		3H	
4H	TBTCAP0	4H	TCCAP0LL	4H	TCCAP2LL	4H	
5H	TBTCAP1	5H	TCCAP0LH	5H	TCCAP2LH	5H	
6H	TBTCAP2	6H	TCCAP0HL	6H	TCCAP2HL	6H	
7H	TBTCAP3	7H	TCCAP0HH	7H	TCCAP2HH	7H	
8H	TBTRDCAPLL	8H	CAP1CR	8H	CAP3CR	8H	
9H	TBTRDCAPLH	9H		9H		9H	
AH	TBTRDCAPHL	AH		AH		AH	
BH	TBTRDCAPHH	BH		BH		BH	
CH	TCG0IM	CH	TCCAP1LL	CH	TCCAP3LL	CH	
DH	TCG0ST	DH	TCCAP1LH	DH	TCCAP3LH	DH	
EH	Reserved	EH	TCCAP1HL	EH	TCCAP3HL	EH	
FH	Reserved	FH	TCCAP1HH	FH	TCCAP3HH	FH	

[10] 32-bit output compare

ADR	Register name	ADR	Register name	ADR	Register name	ADR	Register name
FFFFF 440H	TCCMP0LL	FFFFF450H	TCCMP4LL	FFFFF460H		FFFFF470H	CMPCTL0
1H	TCCMP0LH	1H	TCCMP4LH	1H		1H	CMPCTL1
2H	TCCMP0HL	2H	TCCMP4HL	2H		2H	CMPCTL2
3H	TCCMP0HH	3H	TCCMP4HH	3H		3H	CMPCTL3
4H	TCCMP1LL	4H	TCCMP5LL	4H		4H	CMPCTL4
5H	TCCMP1LH	5H	TCCMP5LH	5H		5H	CMPCTL5
6H	TCCMP1HL	6H	TCCMP5HL	6H		6H	CMPCTL6
7H	TCCMP1HH	7H	TCCMP5HH	7H		7H	CMPCTL7
8H	TCCMP2LL	8H	TCCMP6LL	8H		8H	
9H	TCCMP2LH	9H	TCCMP6LH	9H		9H	
AH	TCCMP2HL	AH	TCCMP6HL	AH		AH	
BH	TCCMP2HH	BH	TCCMP6HH	BH		BH	
CH	TCCMP3LL	CH	TCCMP7LL	CH		CH	
DH	TCCMP3LH	DH	TCCMP7LH	DH		DH	
EH	TCCMP3HL	EH	TCCMP7HL	EH		EH	
FH	TCCMP3HH	FH	TCCMP7HH	FH		FH	

Little-endian

[11] INTC

ADR	Register name
FFFFE000H	IMC0
1H	ditto
2H	ditto
3H	ditto
4H	IMC1
5H	ditto
6H	ditto
7H	ditto
8H	IMC2
9H	ditto
AH	ditto
BH	ditto
CH	IMC3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE010H	IMC4
1H	ditto
2H	ditto
3H	ditto
4H	IMC5
5H	ditto
6H	ditto
7H	ditto
8H	IMC6
9H	ditto
AH	ditto
BH	ditto
CH	IMC7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE020H	IMC8
1H	ditto
2H	ditto
3H	ditto
4H	IMC9
5H	ditto
6H	ditto
7H	ditto
8H	IMCA
9H	ditto
AH	ditto
BH	ditto
CH	IMCB
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE030H	IMCC
1H	ditto
2H	ditto
3H	ditto
4H	IMCD
5H	ditto
6H	ditto
7H	ditto
8H	IMCE
9H	ditto
AH	ditto
BH	ditto
CH	IMCF
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE040H	IVR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE050H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE060H	INTCLR
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE100H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	ILEV
DH	ditto
EH	ditto
FH	ditto

Little-endian

[12] DMAC

ADR	Register name
FFFFE200H	CCR0
1H	ditto
2H	ditto
3H	ditto
4H	CSR0
5H	ditto
6H	ditto
7H	ditto
8H	SAR0
9H	ditto
AH	ditto
BH	ditto
CH	DAR0
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE210H	BCR0
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR0
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE220H	CCR1
1H	ditto
2H	ditto
3H	ditto
4H	CSR1
5H	ditto
6H	ditto
7H	ditto
8H	SAR1
9H	ditto
AH	ditto
BH	ditto
CH	DAR1
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE230H	BCR1
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR1
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE240H	CCR2
1H	ditto
2H	ditto
3H	ditto
4H	CSR2
5H	ditto
6H	ditto
7H	ditto
8H	SAR2
9H	ditto
AH	ditto
BH	ditto
CH	DAR2
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE250H	BCR2
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR2
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE260H	CCR3
1H	ditto
2H	ditto
3H	ditto
4H	CSR3
5H	ditto
6H	ditto
7H	ditto
8H	SAR3
9H	ditto
AH	ditto
BH	ditto
CH	DAR3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE270H	BCR3
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR3
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE280H	CCR4
1H	ditto
2H	ditto
3H	ditto
4H	CSR4
5H	ditto
6H	ditto
7H	ditto
8H	SAR4
9H	ditto
AH	ditto
BH	ditto
CH	DAR4
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE290H	BCR4
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR4
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2A0H	CCR5
1H	ditto
2H	ditto
3H	ditto
4H	CSR5
5H	ditto
6H	ditto
7H	ditto
8H	SAR5
9H	ditto
AH	ditto
BH	ditto
CH	DAR5
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2B0H	BCR5
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR5
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

Little-endian

ADR	Register name
FFFFE2C0H	CCR6
1H	ditto
2H	ditto
3H	ditto
4H	CSR6
5H	ditto
6H	ditto
7H	ditto
8H	SAR6
9H	ditto
AH	ditto
BH	ditto
CH	DAR6
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2D0H	BCR6
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR6
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2E0H	CCR7
1H	ditto
2H	ditto
3H	ditto
4H	CSR7
5H	ditto
6H	ditto
7H	ditto
8H	SAR7
9H	ditto
AH	ditto
BH	ditto
CH	DAR7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE2F0H	BCR7
1H	ditto
2H	ditto
3H	ditto
4H	
5H	
6H	
7H	
8H	DTCR7
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE300H	DCR
1H	ditto
2H	ditto
3H	ditto
4H	RSR
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	DHR
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE310H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE320H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE330H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE340H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE350H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE360H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE370H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-endian

[13] CS/WAIT controller

ADR	Register name
FFFFE400H	BMA0
1H	ditto
2H	ditto
3H	ditto
4H	BMA1
5H	ditto
6H	ditto
7H	ditto
8H	BMA2
9H	ditto
AH	ditto
BH	ditto
CH	BMA3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE410H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE480H	B01CS
1H	ditto
2H	ditto
3H	ditto
4H	B23CS
5H	ditto
6H	ditto
7H	ditto
8H	
9H	
AH	
BH	
CH	BEXCS
DH	ditto
EH	
FH	

ADR	Register name
FFFFE490H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] FLASH control

ADR	Register name
FFFFE510H	SEQMOD
1H	ditto
2H	ditto
3H	ditto
4H	SEQCNT
5H	ditto
6H	ditto
7H	ditto
8H	ROMSEC1
9H	
AH	
BH	
CH	ROMSEC2
DH	
EH	
FH	

ADR	Register name
FFFFE520H	FLCS
1H	
2H	
3H	
4H	Reserved
5H	Reserved
6H	Reserved
7H	Reserved
8H	Reserved
9H	Reserved
AH	Reserved
BH	Reserved
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE620H	
1H	
2H	
3H	
4H	Reserved
5H	Reserved
6H	Reserved
7H	Reserved
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Attention

[15] ROM correction

ADR	Register name
FFFFE540H	ADDREG0
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG1
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG2
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG3
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE550H	ADDREG4
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG5
5H	ditto
6H	ditto
7H	ditto
8H	ADDREG6
9H	ditto
AH	ditto
BH	ditto
CH	ADDREG7
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE560H	ADDREG8
1H	ditto
2H	ditto
3H	ditto
4H	ADDREG9
5H	ditto
6H	ditto
7H	ditto
8H	ADDREGA
9H	ditto
AH	ditto
BH	ditto
CH	ADDREGB
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFE570H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Little-endian

[16] Clock timer

ADR	Register name
FFFFE700H	
1H	
2H	
3H	
4H	RTCCR
5H	
6H	
7H	
8H	RTCREG
9H	ditto
AH	ditto
BH	ditto
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE710H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] UART/HSIO

ADR	Register name
FFFFE800H	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
FFFFE810H	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
FFFFE820H	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
FFFFE840H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[18] CG

ADR	Register name
FFFFEE00H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	SYSCR3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFEE10H	IMCGA
1H	ditto
2H	ditto
3H	ditto
4H	IMCGB
5H	ditto
6H	ditto
7H	ditto
8H	IMCGC
9H	ditto
AH	ditto
BH	ditto
CH	IMCGD
DH	ditto
EH	ditto
FH	ditto

ADR	Register name
FFFFEE20H	EICRCG
1H	ditto
2H	ditto
3H	ditto
4H	NMIFLG
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFEE40H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

23. JTAG Interface

The TMP19A43 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.

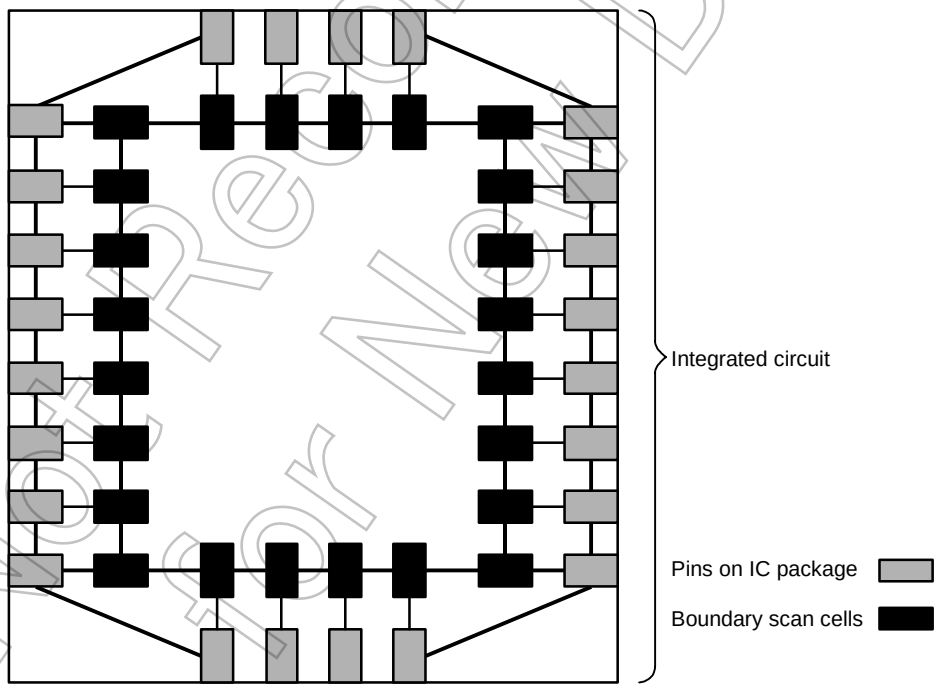


Fig. 23-1 JTAG Boundary Scan Cells

Note) The optional instructions IDCODE, USERCODE, INTEST and RUNBIST are not implemented in the TMP19A43.

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

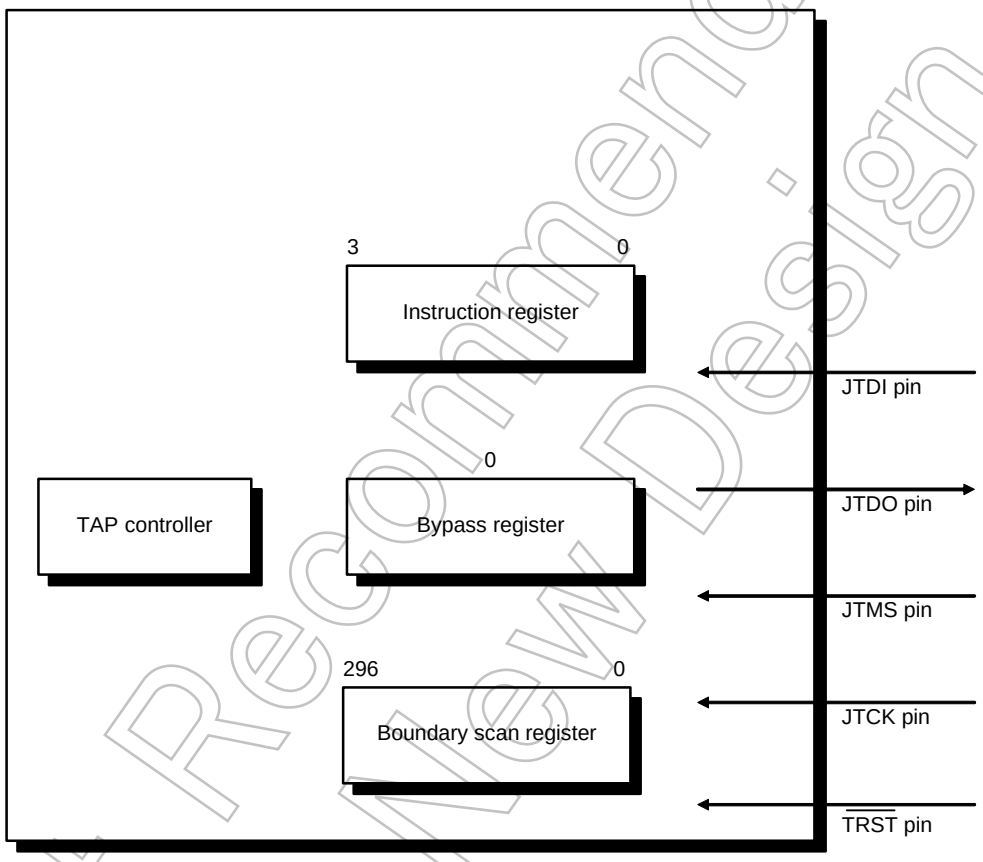


Fig. 23-2 JTAG Interface Signals and Registers

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.

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.

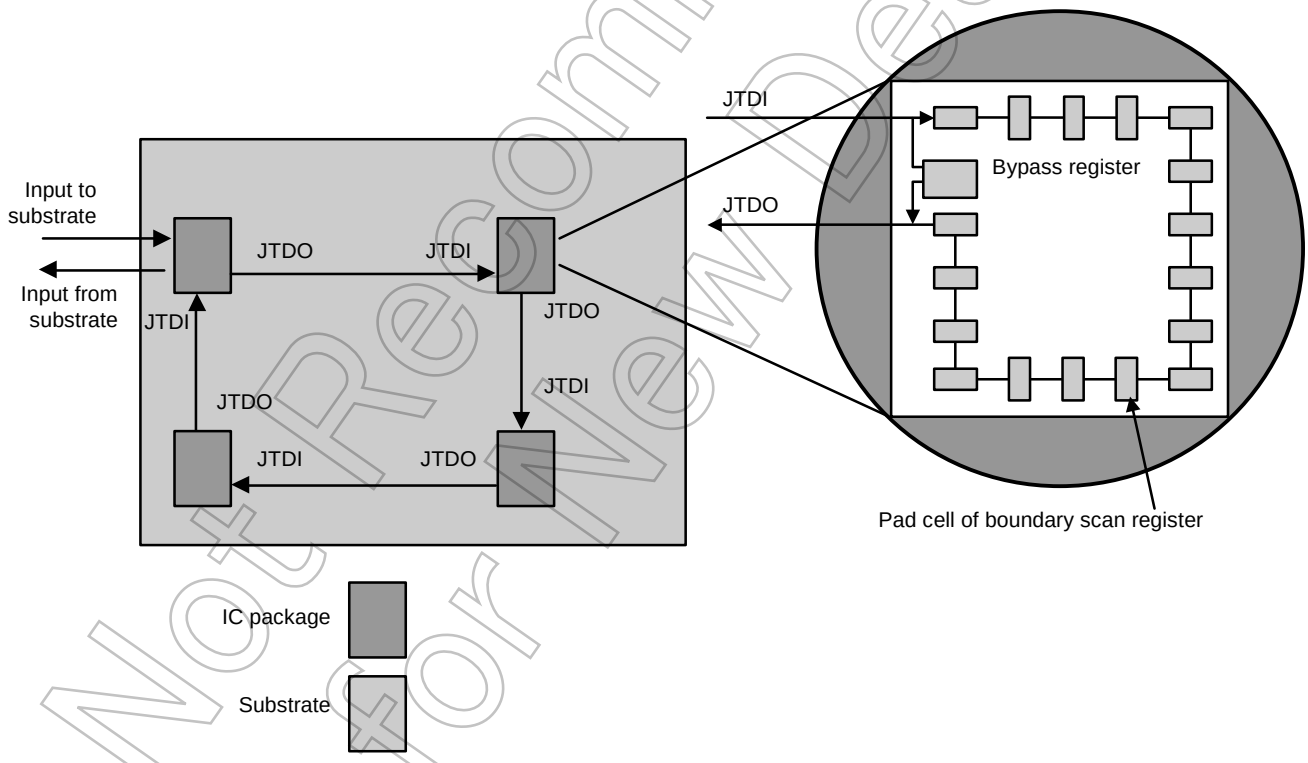


Fig. 23-5 Function of the Bypass Register

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 TMP19A43 except control signals. Pins of the TMP19A43 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 297-bit width. Cells in this data path are connected to all input and output pads of the TMP19A43.

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.

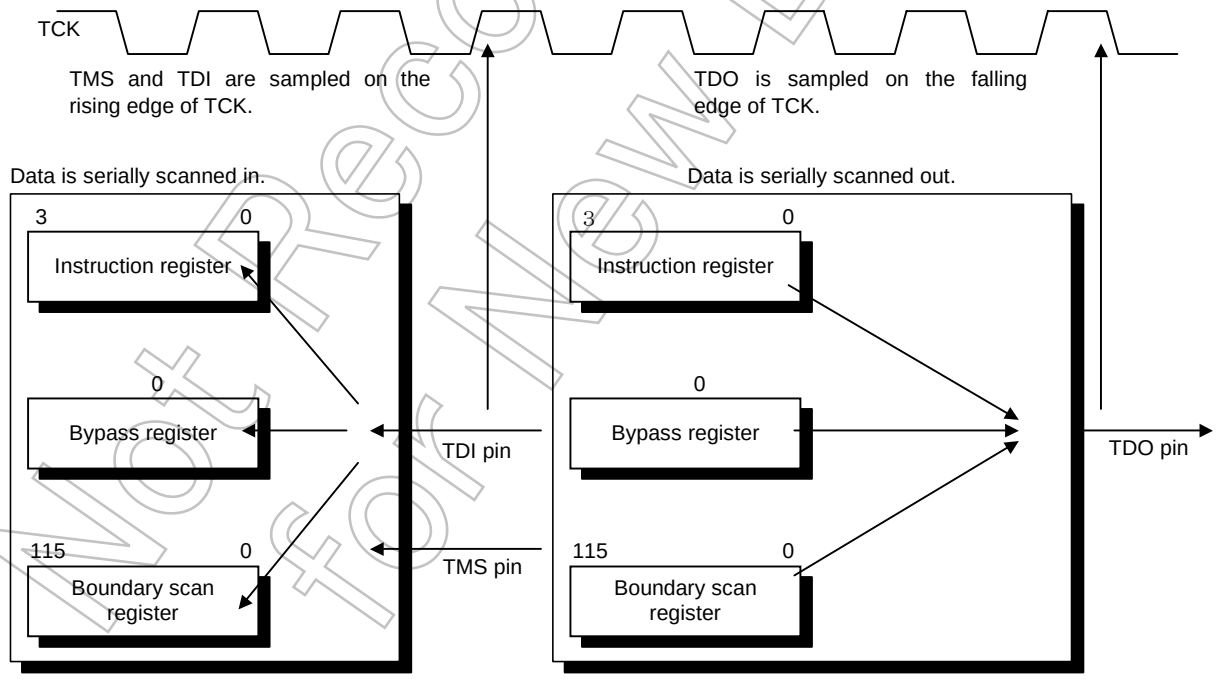


Fig. 23-6 JTAG Test Access Port

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.

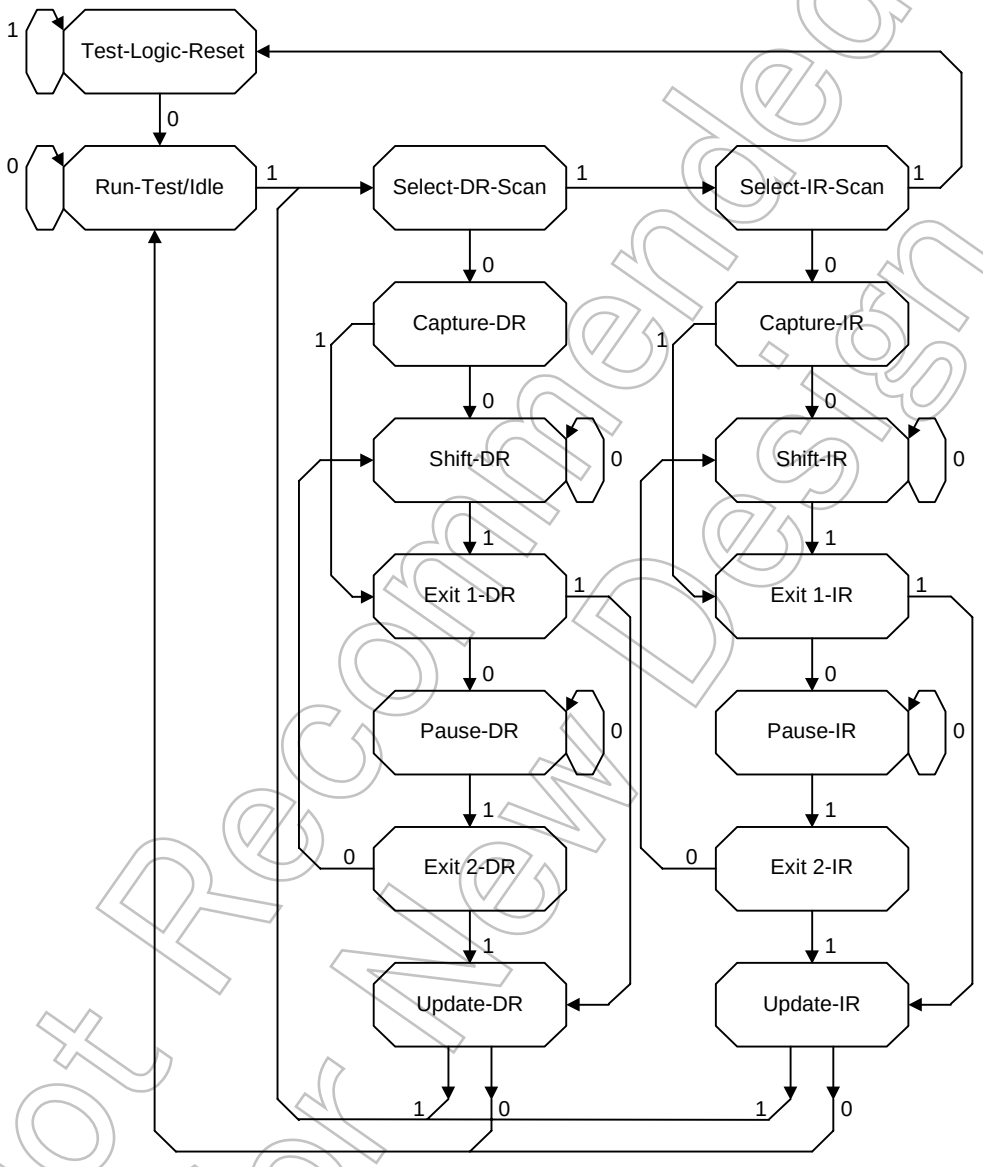


Fig. 23-7 State Transition Diagram of the TAP Controller

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 TMP19A43 Processor Pins

[TDI]	1:P90	2: PE5	3:P91	4: P93	5:PE6	6:P92
7: P94	8: PE7	9: P95	10: P97	11: P96	12: PA0	13: PA1
14: PA2	15: PA3	16: PA4	17: PA7	18:PA6	19: PA5	20: PB2
21: PB1	22: PB5	23:PB4	24: PB0	25: PB7	26: PB6	27: PB3
28: BOOT	29: P32	30: P36	31: P00	32: P37	33: P33	34: P04
35: P01	36: P34	37: P35	38: P30	39: P05	40:P10	41: P31
42: P02	43: P11	44: P03	45: P06	46: P14	47: P07	48: P15
49: P12	50: P13	51:P16	52:P17	53:P20	54:P21	55:P22
56: P25	57: P24	58: P23	59: P27	60: P26	61: P53	62: P52
63: P57	64: P51	65: P56	66: P50	67: P63	68: P62	69: P55
70: P67	71: P66	72:P54	73: P61	74: P41	75: P40	76:P43
77:P42	78:P65	79: P60	80:P44	81:P45	82: P46	83: P47
84: P64	85: PG3	86: PG6	87: PG7	88:PG4	89:PG5	90:PG0
91:PG1	92:PG2	93:TOV	94:PH3	95:DINT	96:PH2	97:PH1
98:PH7	99:DCLK	100:PCST4	101:PCST3	102: PCST2	103: PCST1	104:PCST0
105:PH6	106:PH0	107:PC3	108: PC4	109: PH5	110: PH4	111: PC1
112:PC7	113:PC6	114:PC5	115:PC2	116:PC0	117:PF6	118:PF3
119:PF7	120:PF4	121:PF5	122:PF1	123:PF2	124:PF0	125:PD0
126:PD1	127:PD2	128:PD3	129:PD4	130:PD5	131:PD6	132:P77
133:P76	134:P75	135:P87	136:P74	137:P85	138:P86	139:P84
140:P83	141:P73	142:P82	143:P81	144:P80	145:P71	146:P72
147:P70	148:PE0	149:PE1	150:PE2	151:PE3	152:PE4	
		[TDO]				

Terminal list to which JTAG can be scanned.

Instructions Supported by the JTAG Controller Cells

This section describes the instructions supported by the JTAG controller cells of the TMP19A43.

23.3.8 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.

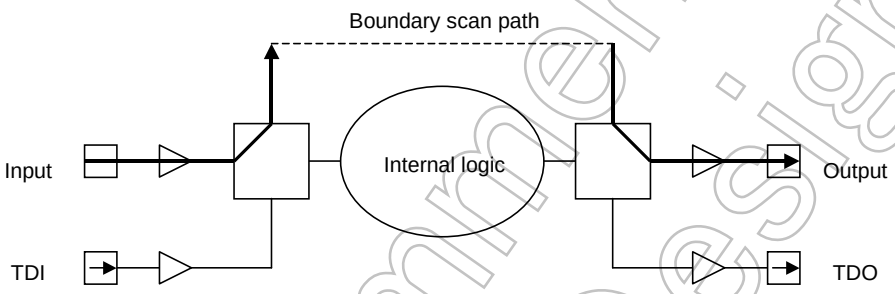


Fig. 23-8 Flow of Data While the EXTEST Instruction Is Selected

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.」

23.3.9 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.

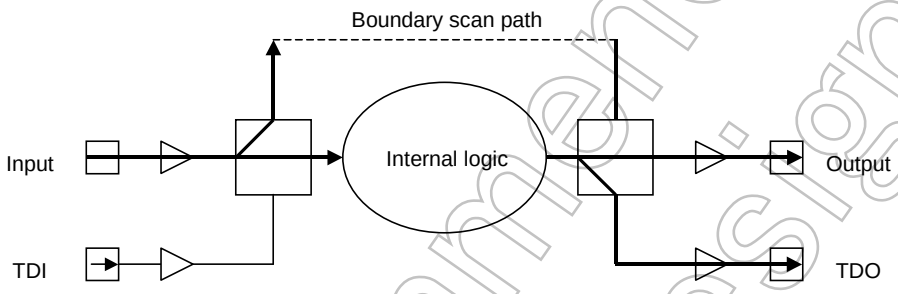


Fig. 23-9 Flow of Data While SAMPLE Is Selected

- 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.

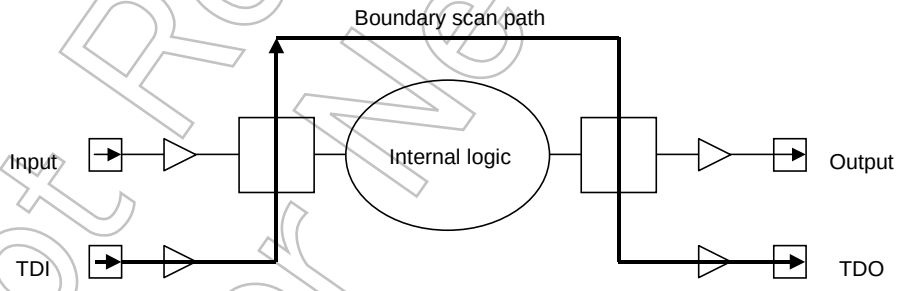


Fig. 23-10 Flow of Test Data While PRELOAD Is Selected

23.3.10 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.

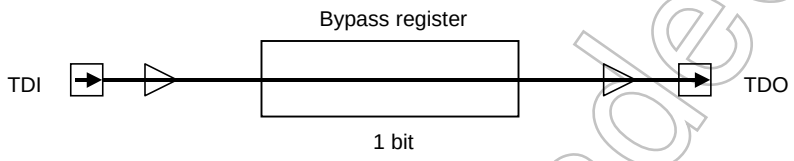


Fig. 23-11 Flow of Data While the Bypass Register Is Selected

23.4 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. Various protecting functions

24.1 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 TMP19A43. The read protect functions specifically include the following:

- Flash protect function
- ROM data protect function
- DSU protect function

24.2 Features

24.2.1 Flash Protect Function

<FLASH>

A built-in flash can prohibit the operation of writing and the deletion at every the block of every 128 Kbyte. This function is called the block protecting.

To make the block protecting function effective, it protects it corresponding to the block where it wants to put protecting.

The bit is made "1". The block protecting can be released by making the protecting bit "0". (Please see the chapter of the Flash operation explanation about the program method.)The protecting bit can be monitored by FLCS register < BLPRO3:0 > bit.

The state to put protecting on all blocks is called the FLASH protecting. It is necessary to note it because all the protecting bits become "0" after automatically deleting all data of the flash when the protecting release operates after it puts it into the state of the LASH protection of 1°F(operation that makes the protecting bit "0").

<Mask>

FLASH is always being protected in the mask version, and the FLASH protecting cannot be released. This function doesn't influence usual operation in the mask version.

<FLASH/MASK>

It is necessary to be protecting FLASH to make "ROM data protecting" and "DSU protecting" that will explain in the future effective.

24.2.2 ROM data Protect

As for ROM data protecting, the execution of the command to the flash is prohibited in the function it, and the flash version that limits reading data to building FLASH/ROM into. When ROM protecting register ROMSEC1<RSECON > bit is "1", ROM data protecting becomes effective with FLASH protected.

If instructions in the ROM area have been replaced with instructions in the RAM area in a PC by using the ROM correction function, a PC shows the instructions as residing in the flash ROM area. Because they actually reside in the RAM area, data cannot be read in a ROM protected state. To read data by using instructions held in the overwritten RAM area, it is necessary to write data to RAM by using a program available in the ROM area or to use other means.

If the ROM area is put in a protected state, the following operations cannot be performed:

- Using instructions placed in areas other than the ROM area to load or store the data taken from the ROM area
- Store to DMAC register (NMI by the bus error is generated.)
- Loading or storing the data taken from the ROM area in accordance with EJTAG
- Using BOOT-ROM to load or store the data taken from the ROM area (FLASH only)
- Executing flash writer to load or store the data taken from the ROM area(FLASH only)
- Using instructions placed in areas other than the ROM area to access the registers (ROMSEC1, ROMSEC2) that concern the protection of the ROM area
- Executing the command to unprotect automatic blocking in writer mode, performing the flash command sequences other than the automatic blocking unprotect command sequence, and performing the flash command sequence in single or boot mode by specifying an address in the ROM area(FLASH only)

The following operations can be performed even if the ROM area is in a protected state:

- Using instructions placed in the ROM area to load the data taken from the ROM area
- Using instructions placed in all areas to load the data taken from areas other than the ROM area
- Using instructions placed in all areas to make instructions branch off to the ROM area
- Performing PC trace (there are restrictions) or break on the ROM area in accordance with EJTAG
- Data transfer of ROM area by DMAC

24.2.3 DSU Protect

The DSU protecting function is a function for invalidating the connection of DSU-probe to enable third parties other than the user to read the data of a built-in flash easily.

When SEQMOD register < DSUOFF > bit is "1", the DSU protecting becomes effective with FLASH protected.

In the DSUOFF bit, the flash version, the mask version, and the state of the first stage are "1. "It enters the state of the DSU protecting as long as the FLASH protecting is always effective in the mask version, and the DSUOFF bit is not set to "0" by the user program.

It doesn't enter the state of the DSU protecting if protecting is not put on all blocks of FLASH in the flash version. An initial state enters the state of the DSU protecting as well as the mask version when FLASH is being protected putting protecting on all blocks of FLASH.

(note)

The DSUOFF bit can be accessed only with the instruction put on built-in ROM in the state of ROM data protecting. It is necessary to note it because it is necessary to put the program of the DSU protecting release on built-in ROM.

24.3 Protect Configuration and Protect Statuses

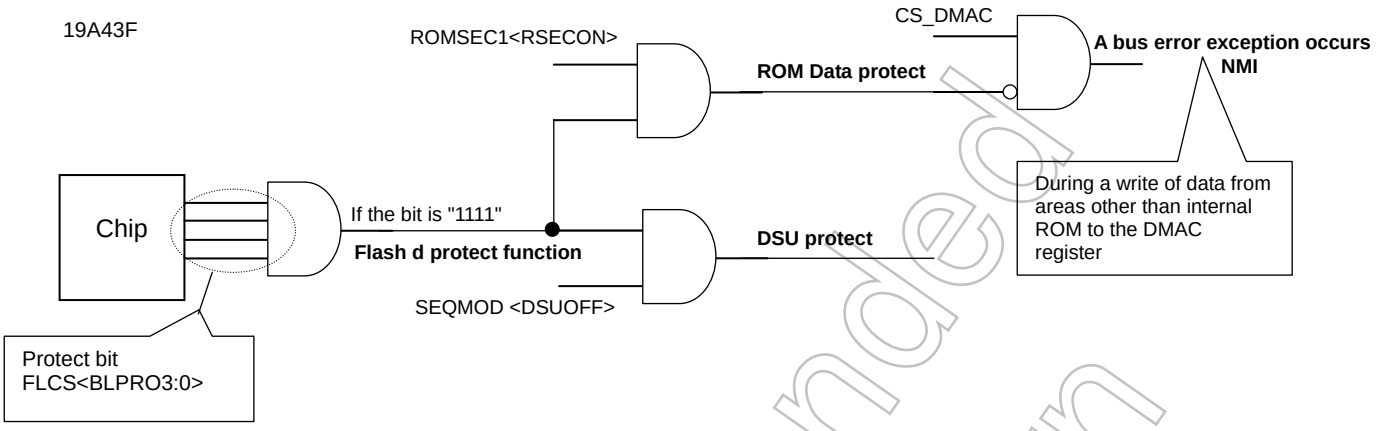


Fig. 24-1 Various Protect Statuses

Table 24-1 Protect Statuses in Each Mode

	Protect bit setting FLCS<BLPRO 3:0>	1111				≠ 1111
	ROM protect enable bit ROMSEC1<RSECON>	1		0		Don't Care
	DSU protect enable bit SEQMOD <DSUOFF>	1	0	1	0	Don't Care
	Flash read protect status	ON				OFF
	ROM protect status	ON		OFF		OFF
	DSU protect status	ON	OFF	ON	OFF	OFF
Single /single boot mode	Read of flash from internal ROM	○	○	○	○	○
	Read of flash from areas other than internal ROM	× *1	× *1	○	○	○
	Clearing of ROM protect enable status (from ROM)	○	○			○
	Clearing of ROM protect enable status (from areas other than ROM)	× *2	× *2			○
	Clearing of DSU protect enable status (from ROM)	○		○		○
	Clearing of DSU protect enable status (from areas other than ROM)	× *3		○		○
	Issuing of the command to erase protect bits	× *4	× *4	○ *8	○ *8	○
	Issuing of commands other than the command to erase protect bits	× *5	× *5	× *7	× *7	Δ *9
	Writing of data to the DMAC setting register (from ROM)	○	○	○	○	○
	Writing of data to the DMAC setting register (from areas other than ROM)	× *6	× *6	○	○	○

*1 : The data of address "0xBFC0_0000" or "0xBFC0_0002" can be read.
*2 : Stored data is masked. A write to registers cannot be executed (data in registers cannot be cleared).
*3 : Stored data is masked. A write to registers cannot be executed (data in registers cannot be cleared).
*4 : A command address is masked, and flash memory does not recognize commands.
*5 : A command address is masked, and flash memory does not recognize commands.
*6 : A bus error exception occurs (when making the DMAC register setting).
*7 : Because a read of flash memory is prohibited, commands are not recognized.
*8 : Because a read of flash memory is prohibited, issued commands are converted to the command for erasing the whole flash memory area and the command for erasing all protect bits.

24.4 Register

Flash control/status register

This register shows the status of flash memory being monitored and the block protect status of flash memory.

Table 24-2 Flash Control Register

FLCS (0xFFFF_E520)

	7	6	5	4	3	2	1	0
Bit Symbol	BLPRO3	BLPRO2	BLPRO1	BLPRO0		ROMTYPE		RDY/BSY
Read/Write	R				R	R	R	R
After reset by power-on	0 (1)	0 (1)	0 (1)	0 (1)	0	0 (1)	0	1
Function	Protect area setting (in units of 128 KB) 0000: All blocks unprotected xxx1: Block 0 protected xx1x: Block 1 protected x1xx: Block 2 protected 1xxx: Block 3 protected (MASK : "1111")				"0" is read.	ROM identification bit 0: Flash 1: MROM	"0" is read.	Ready/Busy 0: In auto operation 1: Auto operation completed (MASK:"1")

	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset by power-on	0	0	0	0	0	0	0	0
Function								

	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset by power-on	0	0	0	0	0	0	0	0
Function								

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset by power-on	0	0	0	0	0	0	0	0
Function								

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided to identify the status of auto operation. This bit is a functional bit for monitoring this function by communicating with the CPU. If flash memory is in auto operation, "0" is output to show that flash memory is busy. As flash memory completes auto operation and goes into a ready state, "1" is output and the next command will be accepted. If the result of auto operation is faulty, this bit continues to output "0." It returns to "1" upon a hardware reset.

(Note) Before issuing a command, make sure that flash memory is in a ready state. If a command is issued when flash memory is busy, a right command cannot be generated and there is the possibility that subsequent commands may not be able to be input. In this case, you must return to a normal functional state by executing a system reset or issuing a reset command.

Bit 2: ROM type identification bit

This bit is used to identify the type of flash ROM or the type of mask ROM based on the value after a reset.

Flash ROM: "0"

Mask ROM: "1"

Bit [7:4]: Protect bit (x: A combination setting can be made for each block)

The protect bit (4-bit) value corresponds to the protect status of each block. If this bit is "1," the corresponding block is in a protected state. A protected block cannot be overwritten.

Table 24-3 ROM Protect Register

ROMSEC1 (0xFFFF_E518)

	7	6	5	4	3	2	1	0
Bit Symbol								RSECON
Read/Write	R							R/W
After reset by power-on	0							1
Function	"0" is always read.							ROM protect 1: ON 0: OFF (see note)

	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset by power-on	0							
Function	"0" is always read.							

	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset by power-on	0							
Function								

	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset by power-on	0							
Function	"0" is always read.							

(Note) This register is initialized only by power-on reset in the FLASH version.
The mask version is usually initialized at each reset.

(Note) To access this register, 32-bit access is required.

Table 24-4 ROM Protect Lock Register

ROMSEC2 (0xFFFF_E51C)

	7	6	5	4	3	2	1	0
Bit Symbol								
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) If this register is set to "0x0000_003D" after ROMSEC1<RSECON> is set, appropriate bit values are automatically set in ROMSEC1<RSECON>.

(Note) If the ROM area is protected, the registers ROMSEC1 and ROMSEC2 can be accessed only by using the instructions residing in the ROM area.

(Note) To access this register, 32-bit access is required.

(Note) This register is a write-only register. If it is read, values will be undefined.

Table 24-5 DSU Protect Mode Register

SEQMOD (0xFFFF_E510)

	7	6	5	4	3	2	1	0
Bit Symbol								DSUOFF
Read/Write	R							R/W
After reset	0							1
Function	"0" is always read.							1: DSU disabled 0: DSU enabled
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is always read.							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	R							
After reset	0							
Function								
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	R							
After reset	0							
Function	"0" is always read.							

(Note) This register is initialized only by power-on reset in the FLASH version.

(Note) The mask version is usually initialized at each reset. To access this register, 32-bit access is required.

Table 24-6 DSU Protect Control Register

SEQCNT (0xFFFF_E514)

	7	6	5	4	3	2	1	0
Bit Symbol	DSECODE07	DSECODE06	DSECODE05	DSECODE04	DSECODE03	DSECODE02	DSECODE01	DSECODE00
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							
	15	14	13	12	11	10	9	8
Bit Symbol	DSECODE15	DSECODE14	DSECODE13	DSECODE12	DSECODE11	DSECODE10	DSECODE09	DSECODE08
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							
	23	22	21	20	19	18	17	16
Bit Symbol	DSECODE23	DSECODE22	DSECODE21	DSECODE20	DSECODE19	DSECODE18	DSECODE17	DSECODE16
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							
	31	30	29	28	27	26	25	24
Bit Symbol	DSECODE31	DSECODE30	DSECODE29	DSECODE28	DSECODE27	DSECODE26	DSECODE25	DSECODE24
Read/Write	W							
After reset	0							
Function	Write "0x0000_00C5."							

(Note) To access this register, 32-bit access is required.

(Note) This register is a write-only register. If it is read, values will be undefined.

24.5 Proted-related / Release Settings

If it is necessary to overwrite flash memory or protect bits in a protected state, "automatic protect bit deletion" must be executed or the ROM protect function must be disabled. DSU cannot be used if it is in a protected state.

Flash memory may go into a read-protected state after the automatic protect bit program is executed. In this case, it is necessary to set DSU-PROBE to "enable" before the automatic protect bit program is executed.

（The mask version is possible only the release of ROM security, and the protecting bit cannot be rewritten.）

If "automatic protect bit deletion" is executed when flash memory is in a read-protected state, flash memory is automatically initialized inside this device. Therefore, extra caution must be used when switching from one state to a read-protected state. (FLASH only)

24.5.1 Flash Protect Function

The flash protecting function cannot be released always effectively in the mask version.

It becomes effective by putting the block protecting on all of the four blocks in the flash version.

The flash memory command sequence and protect bit program commands are used to enable or disable the flash read protect function. For further information, refer to the command sequence explained in the chapter describing the operations of flash memory.

(notes concerning FLASH version)

The protecting bit is cleared after all the data of the flash is deleted when the protecting bit release command is executed with the flash protected, and the flash protecting is released.

In the state of ROM data protecting, explains as follows, the command execution to the flash is disregarded. It is necessary to release ROM data protecting first clearing the RSECON bit of ROM protecting register when the flash protecting is released with ROM protected.

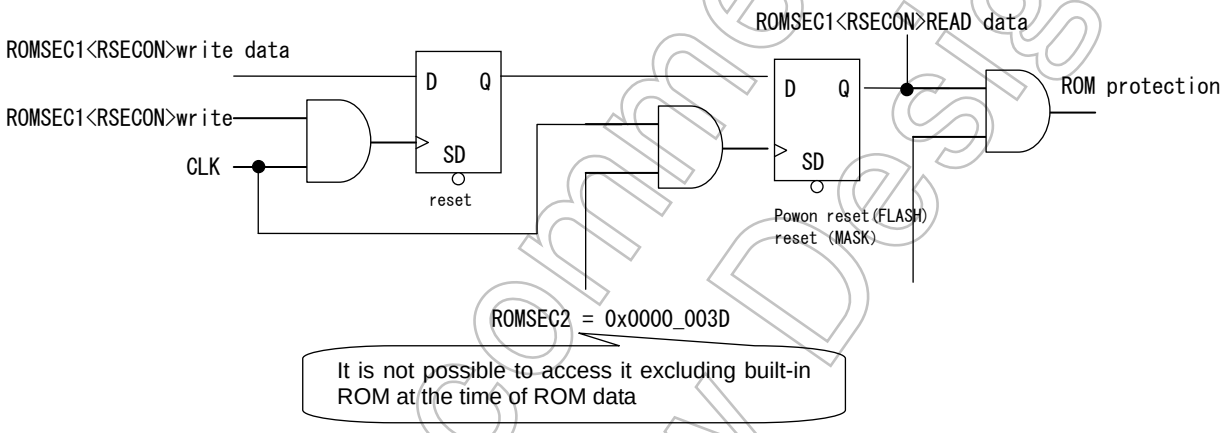
24.5.2 ROM data Protect

ROM data protecting is effective the flash protecting and becomes effective at ROM protecting register ROMSEC1<RSECON>=“1".

After releasing reset, the RSECON bit is initialized by "1". The flash protecting is sure to enter the state of ROM data protecting in the mask version after releasing reset because it is always effective.

It decides whether to enter the state of ROM data protecting by the state of the flash protecting in the flash version.

When ROM protecting register is rewritten with ROM data protected, rewriting can be executed only from the program put on built-in ROM. Therefore, it is necessary to prepare the release program of ROM data protecting on built-in ROM.



ROM data protecting is released by setting ROM protecting register ROMSEC1<RSECON>“0" when protecting is released, and writing protecting code “0x0000_003D" in ROM protecting lock register ROMSEC2. Moreover, ROM data protecting function can be set again by similarly setting ROM protecting register ROMSEC1<RSECON>“1" when ROM protecting is set, and writing protecting code “0x0000_003D" in ROM protecting lock register ROMSEC2.

It is necessary to note the ROMSEC2 register because the reading data is different from original write data because of the register only for writing.

The initialization of ROM protecting register is different in the flash version and the mask version.

It provides with the power-on reset circuit in the flash version, ROM protecting register is initialized by power-on reset, and the value doesn't usually change in reset.

It is usually initialized by reset in the mask version because power-on reset is not provided.

It is necessary to note it in the mask version because it is usually initialized at each reset.

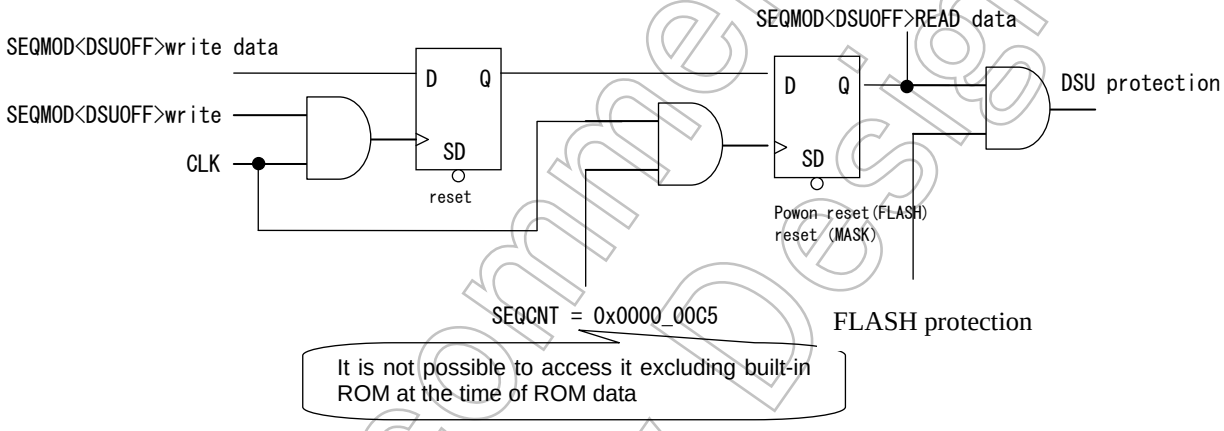
24.5.3 DSU Protect

DSU data protecting is effective the flash protecting and becomes effective at DSU protecting register SEQMOD<RSECON>="1".

After releasing reset, the DSUOFF bit is initialized by "1". The flash protecting is sure to enter the state of DSU data protecting in the mask version after releasing reset because it is always effective.

It decides whether to enter the state of ROM data protecting by the state of the flash protecting in the flash version.

When DSU protecting register is rewritten with ROM data protected, rewriting can be executed only from the program put on built-in ROM. Therefore, it is necessary to prepare the release program of DSU data protecting on built-in ROM.



DSU protecting is released by setting DSU protecting register SEQMOD<DSUOFF>"0" when protecting is released, and writing protecting code "0x0000_00C5" in ROM protecting lock register SEQCNT. Moreover, DSU protecting function can be set again by similarly setting ROM protecting register SEQMOD<DSUOFF>"1" when DSU protecting is set, and writing protecting code "0x0000_00C5" in DSU protecting lock register SEQCNT.

It is necessary to note the SEQCNT register because the reading data is different from original write data because of the register only for writing.

The initialization of DSU protecting register is different in the flash version and the mask version.

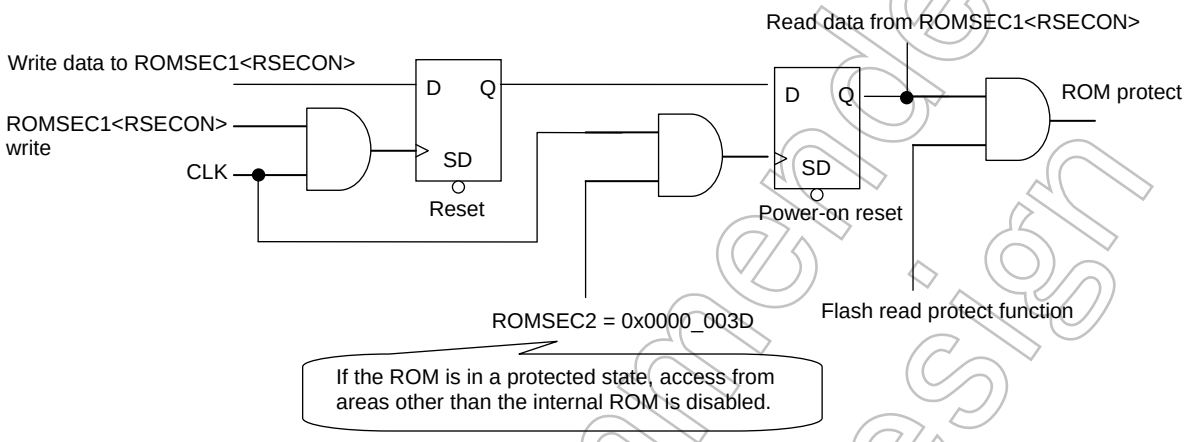
It provides with the power-on reset circuit in the flash version, DSU protecting register is initialized by power-on reset, and the value doesn't usually change in reset.

It is usually initialized by reset in the mask version because power-on reset is not provided.

It is necessary to note it in the mask version because it is usually initialized at each reset.

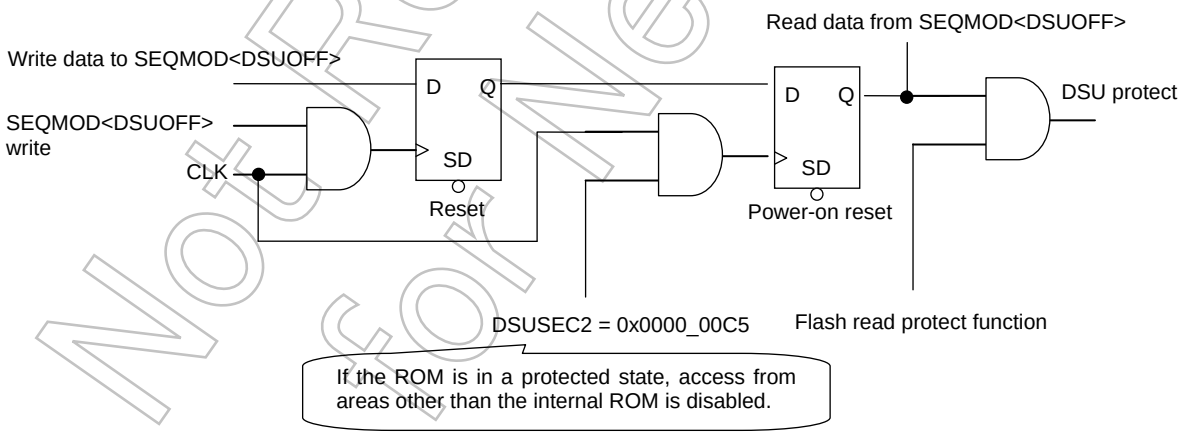
24.5.4 ROM Protect Register: ROMSEC1<RSECON>

The ROM protect register is equipped with a power-on reset circuit. Caution must be exercised as data read from the ROMSEC1<RSECON> bit is different from the actually written data. How data is processed is shown below. The mask version is usually initialized by reset though FLASH goods are initialized by power-on reset.



24.5.5 DSU Protect Mode Register: SEQMOD <DSUOFF>

The DSU protect mode register is equipped with a power-on reset circuit. Caution must be exercised as data read from the SEQMOD <DSUOFF> bit is different from the actually written data. How data is processed is shown below. The mask version is usually initialized by reset though FLASH goods are initialized by power-on reset.



25. Electrical Characteristics

The letter x in equations presented in this chapter represents the cycle period of the fsys clock selected through the programming of the SYSCR1.SYSCK bit. The fsys clock may be derived from either the high-speed or low-speed crystal oscillator. The programming of the clock gear function also affects the fsys frequency. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.SYSCK = 0) and a clock gear factor of 1/fc (SYSCR1.GEAR[2:0] = 000).

25.1 Absolute Maximum Ratings

Parameter		Symbol	Rating	Unit
Supply voltage		V_{CC15} (Core)	− 0.3to3.0	V
		Vcc3 (I/O)	− 0.3to3.9	
		AVCC3 (A/D)	− 0.3to3.9	
		DAVCC (D/A)	− 0.3to3.5	
		DVCC3	− 0.3to3.9	
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 dissipation (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

V_{CC15}＝DVCC15＝CVCC15＝CVCCH、V_{CC}3＝DVCC3=CVCCL、
V_{SS}＝DVSS＝AVSS＝CVSS=DAGND

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 dissipation, 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.

25.2 DC ELECTRICAL CHARACTERISTICS (1/3)

Ta＝－20 to 85℃

Parameter		Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
Supply voltage	AVCC3 = 3.3V CVCCH＝DVCC15 DVCC3=CVCCL CVSS＝DVSS＝ AVSS=DAGND=0V	DVCC15 CVCCH	fosc = 8to10MHz fs = 30kHzto34kHz fsys = 15KHzto34KHz 4MHzto40MHz	1.35		1.65	V
		DAVCC		2.3		2.7	
		DVCC3 CVCCL		2.7		3.6	
Low-level input voltage	P7 to P8 (Used as a port)	V_{IL1}	2.7V≦AVCC3≦3.6V	－0.3		0.3 AVCC3	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	
	X1	V_{IL4}	1.35V≦CVCCH≦1.65V			0.1 CVCCH	
	XT1	V_{IL5}	2.7V≦CVCCL≦3.6V			0.1 CVCCL	

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3= AVCC3=3.3V,DVCC=2.5V, unless otherwise noted

Ta＝－20 to 85℃

Parameter		Symbol	Rating		Min.	Typ. (Note 1)	Max.	Unit
High-level input voltage	P7 to P8 (Used as a port) Normal port	V_{IH1}	2.7V≦AVCC3≦3.6V		0.7 AVCC3		DVCC3 + 0.3 DVCC15+ 0.2 CVCCH+ 0.2 CVCCL+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			
	X1	V_{IH4}	1.35V≦CVCCH≦1.65V		0.9 CVCCH			
	XT2	V_{IH4}	2.7V≦CVCCL≦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			

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3= AVCC3=3.3V,DVCC=2.5V, unless otherwise noted

25.3 DC ELECTRICAL CHARACTERISTICS (2/3)

Ta＝－20 to 85℃

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
Input leakage current	I_{LI}	$0.0 \leqq V_{IN} \leqq$DVCC15 $0.0 \leqq V_{IN} \leqq$DVCC3 $0.0 \leqq V_{IN} \leqq$AVCC3 $0.0 \leqq V_{IN} \leqq$DAVCC		0.02	±5	μA
Output leakage current	I_{LO}	$0.2 \leqq V_{IN} \leqq$ DVCC15 – 0.2 $0.2 \leqq V_{IN} \leqq$DVCC3 – 0.2 $0.2 \leqq V_{IN} \leqq$AVCC3– 0.2 $0.2 \leqq V_{IN} \leqq$DAVCC– 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, DVCC15=1.5V,DVCC3= AVCC3=3.3V,DVCC=2.5V, unless otherwise noted

25.4 DC ELECTRICAL CHARACTERISTICS (3/3)

DVCC15＝CVCCH＝1.35Vto1.65V, CVCCL= DVCC3＝AVCC3＝2.7Vto3.6V,
DAVCC=2.3Vto2.7V

Ta＝－20 to 85℃

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL(Note 2): Gear = 1/1	I_{CC}	f_{sys} =40 MHz (f_{osc} = 10 MHz)		50	74	mA
IDLE(Doze) (Note 3)				20	29	
IDLE(Halt) (Note 3)				18	28	
SLOW (Note 4)		fs = 32.768kHz		140	995	μA
SLEEP (Note 4)		fs = 32.768kHz		30	985	μA
STOP				27	980	μA

Note 1: Ta = 25°C, DVCC15=1.5V,DVCC3= AVCC3=3.3V,DVCC=2.5V, unless otherwise noted

(note1) I_{CC} NORMAL:

Measured with the CPU dhrystone operating (DSU is excluded.), RAM, FLASH.
All functions operating. D/A and A/D excluded.

(note2) I_{CC} IDLE :

Measured with all functions stoping.

(note3) I_{CC} SLOW, SLEEP :

Measured with RTC on low-speed

I_{CC} ：The current where flows is included.

（DVCC15、DVCC3、CVCCH、CVCCL、AVCC3、DAVCC）

25.5 10-bit ADC Electrical Characteristics

DVCC15＝CVCCH＝1.35Vto1.65V, CVCCL= DVCC3＝AVCC3＝VREFH＝2.7Vto3.6V,
DAVCC=2.3Vto2.7V,AVSS = DVSS ,Ta＝－20 to 85℃
AVCC3 load capacitanc= 3.3μF, VREFH load capacitanc= 3.3μF

Parameter		Symbol	Rating	Min	Typ	Max	Unit
Analog reference voltage (＋)		VREFH		2.7	3.3	3.6	V
Analog reference voltage (－)		VREFL		AVSS	AVSS	AVSS	V
Analog input voltage		VAIN		VREFL		VREFH	V
Analog supply current	A/D conversion	IREF	DVSS = AVSS = VREFL		4.5	5.5	mA
	Non-A/D conversion		DVSS = AVSS = VREFL		±0.02	±5	μA
supply current	A/D conversion	—	Non-IREF			3	mA
INL error		—	AIN resistance ≦600Ω AIN load capacitance≦30pF Conversion time≧1.15μs		±2	±3	LSB
DNL error					±1	±2	
Offset error					±2	±4	
Fullscale error					±2	±4	
INL error		—	AIN resistance ≦600Ω AIN load capacitance≧0.1μF Conversion time≧1.15μs		±2	±3	
DNL error					±1	±2	
Offset error					±2	±4	
Fullscale error					±2	±4	
INL error		—	AIN resistance ≦1.3kΩ AIN load capacitance≧0.1μF Conversion time≧1.15μs		±2	±3	
DNL error					±1	±2	
Offset error					±2	±4	
Fullscale error					±2	±4	
INL error		—	AIN resistance ≦10kΩ AIN load capacitance≧0.1μF Conversion time≧2.30μs		±2	±3	
DNL error					±1	±2	
Offset error					±2	±4	
Fullscale error					±2	±4	

(Note 1) 1LSB = (VREFH－VREFL) / 1024[V]

25.6 8bit D/A Electrical Characteristics

DVCC15＝CVCCH＝1.35V to 1.65V, CVCCL= DVCC3＝AVCC3＝2.7V to 3.6V,
DAVCC=2.3V to 2.7V

Parameter		Symbol	Rating	Min	Typ	Max	Unit
Analog reference voltage (＋)		DAVREF		2.3	2.5	2.7	V
Analog supply current	<REFONn> = 1 n=0,1	IDREF			1	2	mA
	<REFONn> = 0 n=0,1				±0.02	±5	μA
supply current		Icc				5	mA
Output current		IDA0, IDA1		±1			mA
Range of output voltage		DA0, DA1		DAGND＋0.3		DAVCC－0.3	V
Fullscale error		—			±2	±3	LSB

(Note 1) 1LSB = (DAVREF − DAGND) / 256[V]

(Note 2) IDREF current valu is in the Dual channel operation .

(Note 3) No guarantee about Relative accuracy in the Dual channel operation

(Note 4) Load Maximum capacitance of each DAx pin is 100pF.

25.7 AC Electrical Characteristics

25.7.1 Multiplex Bus mode

(1) DVCC15＝CVCC15＝1.35Vto1.65V, AVCC3= 2.7Vto3.6V
DVCC3＝ 2.7Vto3.6V, DAVCC = 2.3Vto2.7V,Ta =－20to85°C

① ALE width = 1 clock cycle, 2 programmed wait state

No.	Parameter	Symbol	Equation		40 MHz (fsys)(Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	25				ns
2	A0-A15 VALID TO ALE LOW	t_{AL}	x –11		14.0		ns
3	A0-A15 HOLD AFTER ALE LOW	t_{LA}	x – 8		17.0		ns
4	ALE pulse width high	t_{LL}	x – 6		19.0		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x – 8		17.0		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 8		17.0		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x – 11		39.0		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x – 11		39.0		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 11		14.0		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (2 + TW+ALE) – 43		82.0	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (2 + TW+ALE) – 43		82.0	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + TW) –40		35.0	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + TW) –6		69		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– 6		19.0		ns
16	$\overline{WR}$/$\overline{HWR}$ width low	t_{WW}	x (1 + TW) – 6		69.0		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + TW) – 11		64.0		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 11		14.0		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x+ x (ALE)＋ x (TW-1)－32		43.0	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x+ x (ALE)＋ x (TW-1)－32		43.0	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (TW - 3)-16	x (TW – 1) – 29	9.0	46.0	ns

Note 1: No. 1 to 21:

Internal 2 wait insertion，ALE“1”Clock, @40MHz

TW = W + 2N ,

W : Number of Auto wait insertion , 2N : Number of external wait insertion

TW = 2 + 2*1 = 4

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

② ALE width = 1 clock cycles, 2 programmed wait state

No.	Parameter	Symbol	Equation		40 MHz (fsys)(Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	25				ns
2	A0-A15 VALID TO ALE LOW	t_{AL}	x –11		14.0		ns
3	A0-A15 HOLD AFTER ALE LOW	t_{LA}	x – 8		17.0		ns
4	ALE pulse width high	t_{LL}	x – 6		19.0		ns
5	ALE low to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x – 8		17.0		ns
6	$\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x – 8		17.0		ns
7	A0-A15 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x – 11		39.0		ns
8	A16-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x – 11		39.0		ns
9	A16-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 11		14.0		ns
10	A0-A15 valid to D0-D15 Data in	t_{ADL}		x (2 + TW+ALE) – 43		82.0	ns
11	A16-A23 valid to D0-D15 Data in	t_{ADH}		x (2 + TW+ALE) – 43		82.0	ns
12	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + TW) –40		35.0	ns
13	$\overline{RD}$ width low	t_{RR}	x (1 + TW) –6		69		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– 6		19.0		ns
16	$\overline{WR}$/$\overline{HWR}$ width low	t_{WW}	x (1 + TW) – 6		69.0		ns
17	D0-D15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + TW) – 11		64.0		ns
18	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 11		14.0		ns
19	A16-A23 valid to $\overline{WAIT}$ input	t_{AWH}		x+ x (ALE)＋ x (TW-1)－32		43.0	ns
20	A0-A15 valid to $\overline{WAIT}$ input	t_{AWL}		x+ x (ALE)＋ x (TW-1)－32		43.0	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (TW - 3)-16	x (TW – 1) – 29	9.0	46.0	ns

Note 1: No. 1 to 21:

Internal 2 wait insertion，ALE“1”Clock，@40MHz

TW = W + 2N ，

W : Number of Auto wait insertion ，2N : Number of external wait insertion

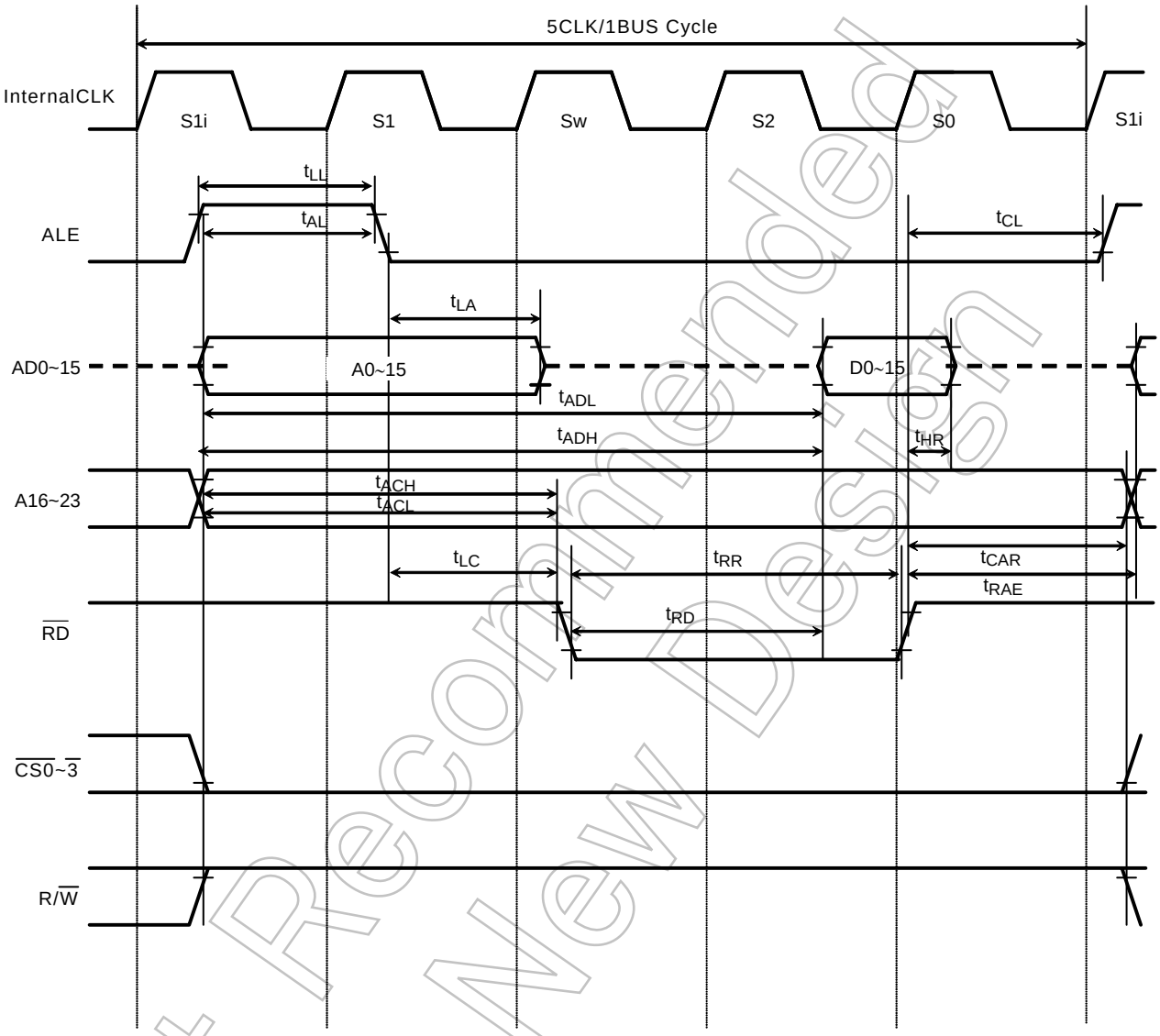
TW = 2 + 2*1 = 4

AC measurement conditions:

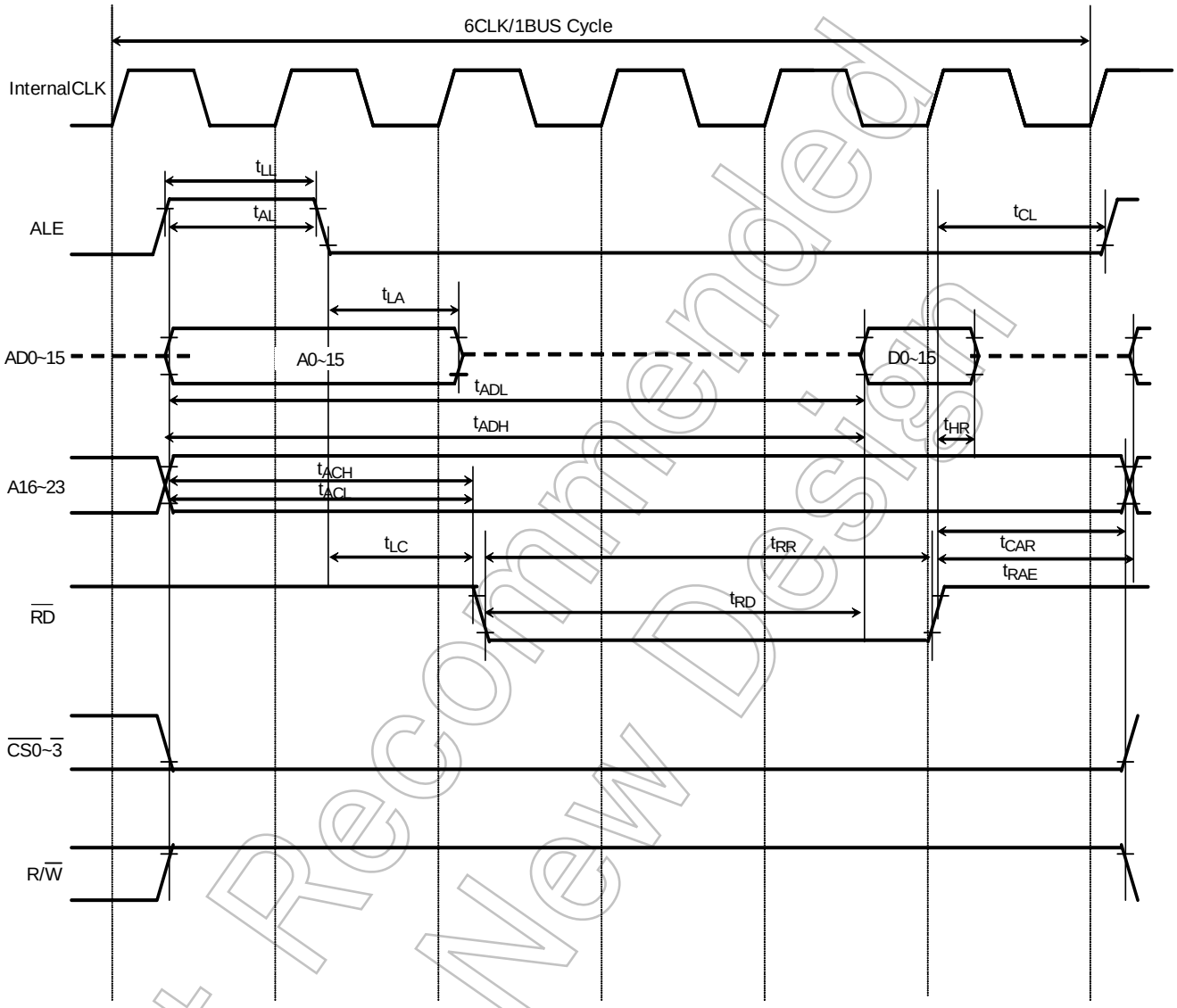
Output levels: High = 0.8DVCC3 V/Low 0.2DVCC3 V, CL = 30 pF

Input levels: High = 0.7DVCC3 V/Low 0.3DVCC3 V

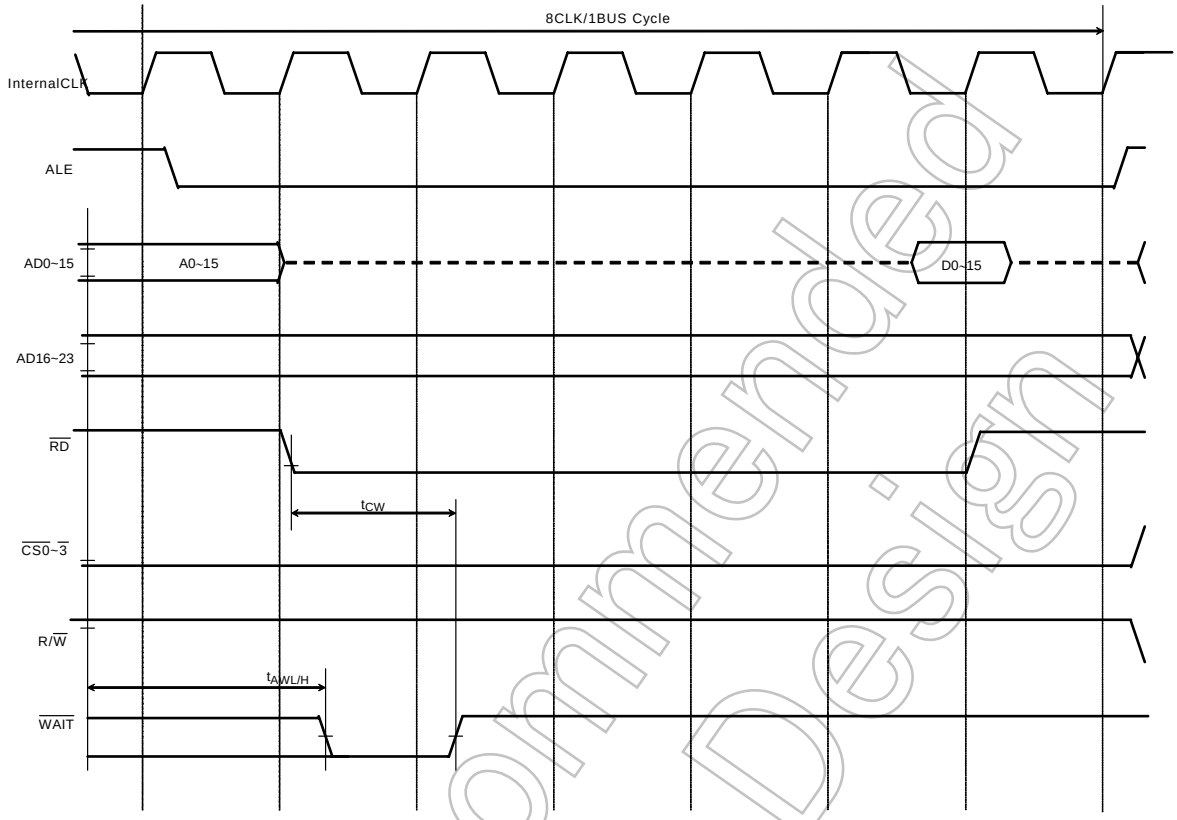
(1) Read cycle timing, ALE width = 1 clock cycle, 1 programmed wait state



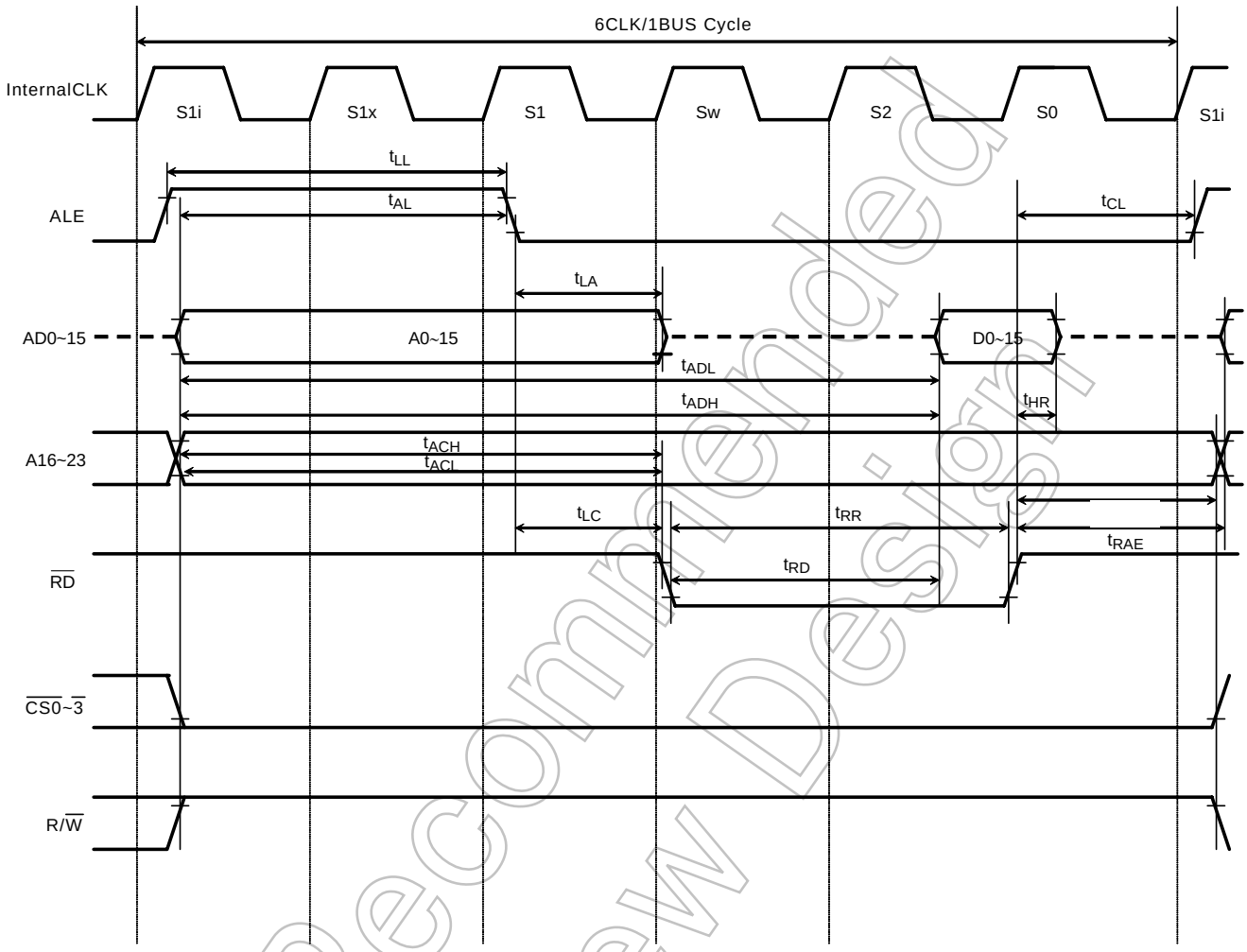
(2) Read cycle timing, ALE width = 1 clock cycle, 2 programmed wait state



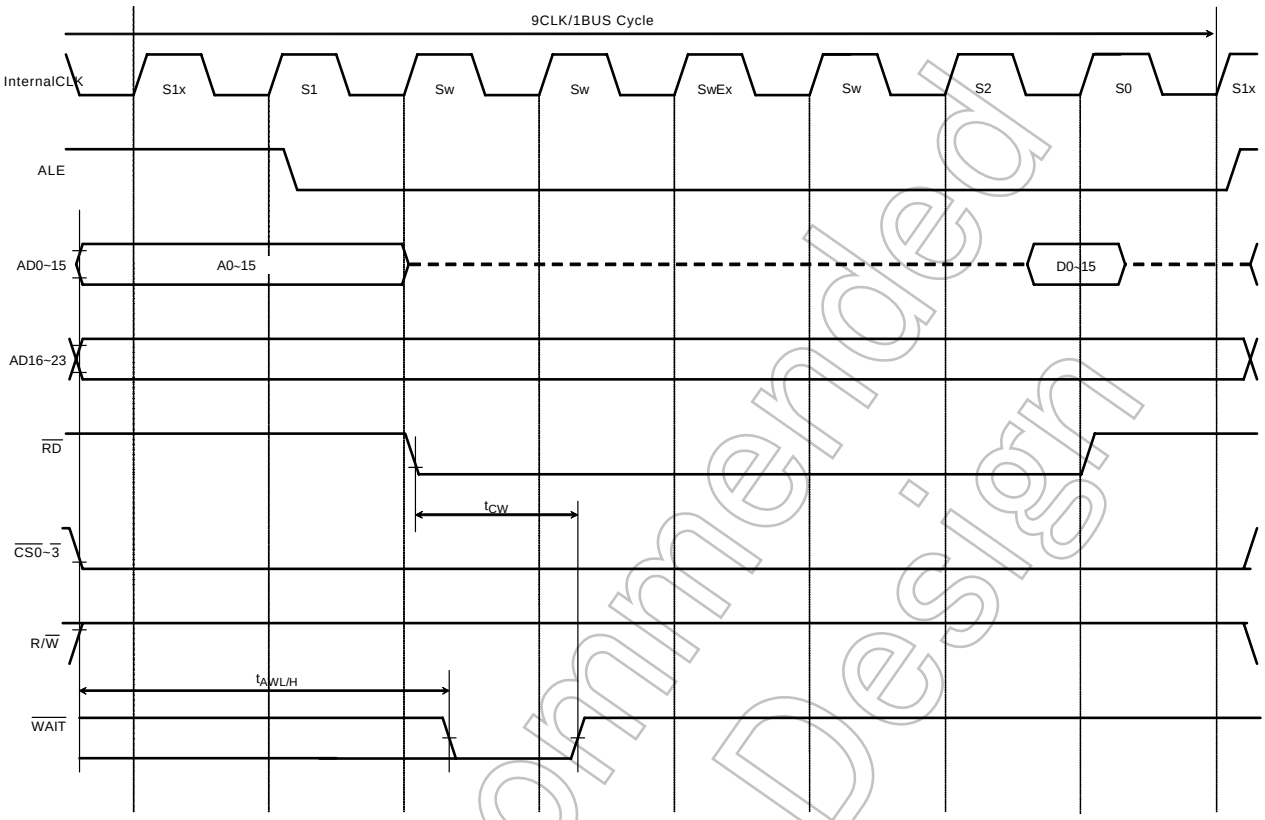
(3) Read cycle timing, ALE width = 1 clock cycle, 4 programmed wait state



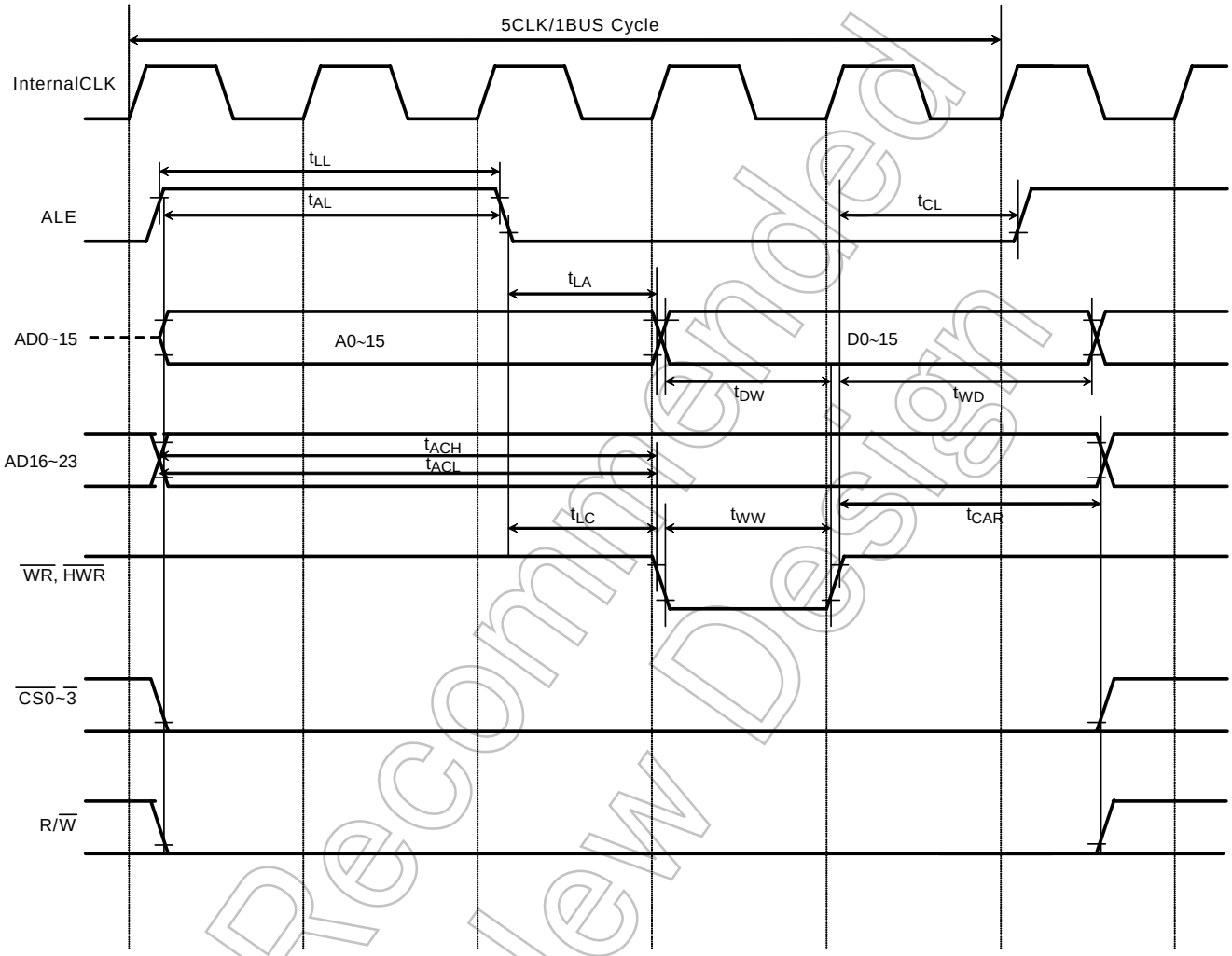
(4) Read cycle timing, ALE width = 2 clock cycle, 1 programmed wait state



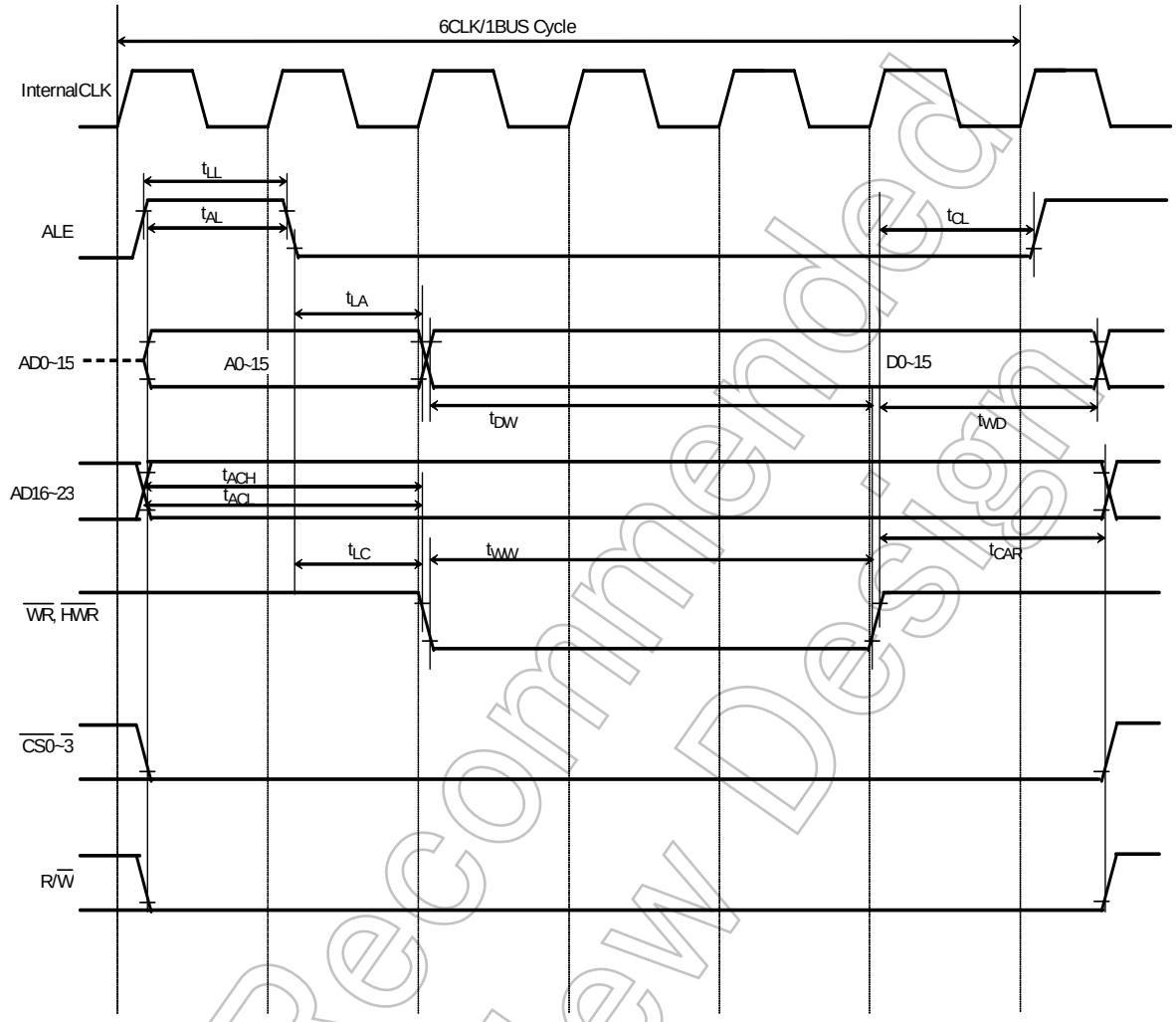
(5) Read cycle timing, ALE width = 2 clock cycle, 4 programmed wait state



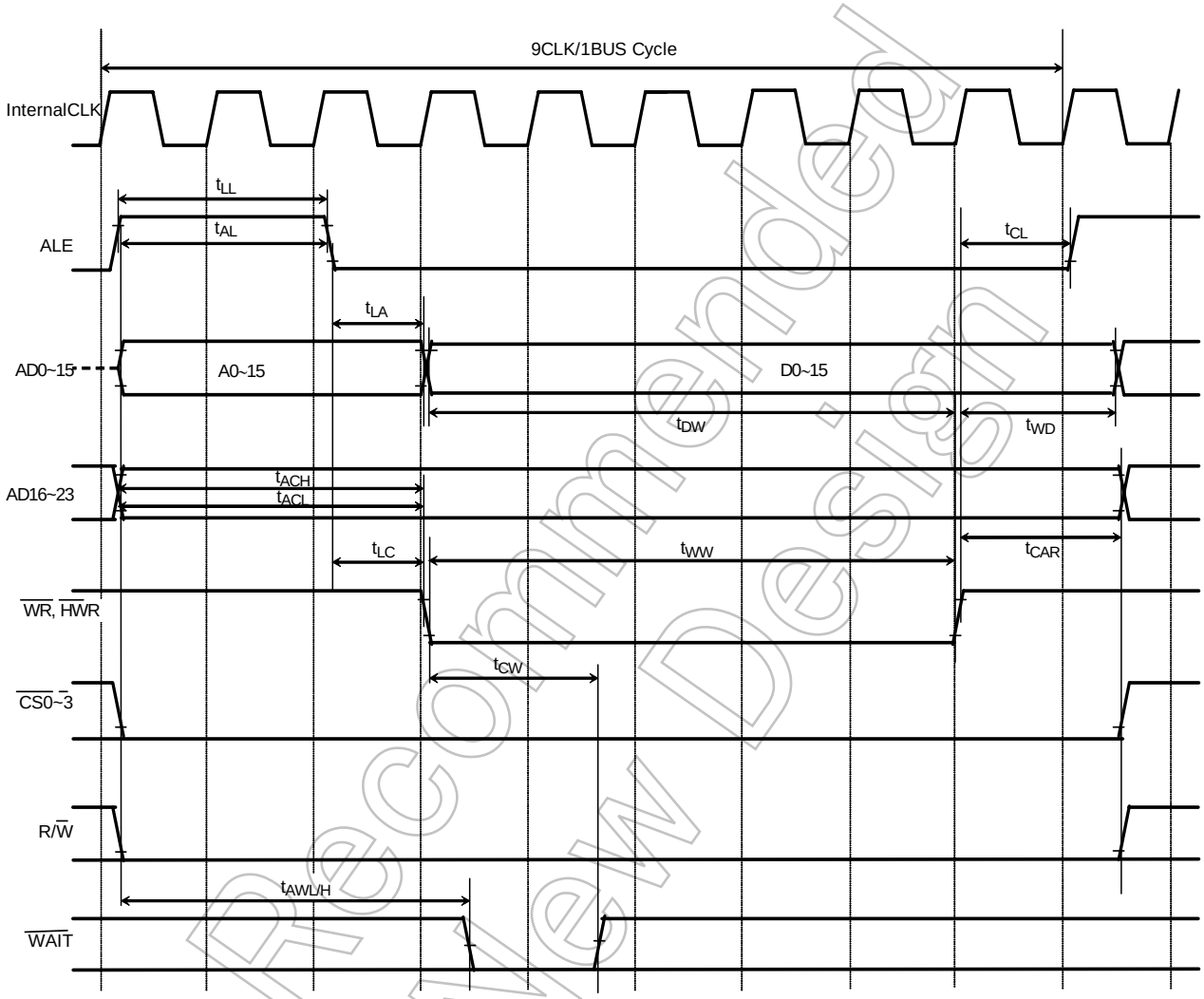
(6) Write cycle timing, ALE width = 2 clock cycles, zero wait state



(7) Write cycle timing, ALE width = 1 clock cycles, 2 wait state



(8) Write cycle timing, ALE width = 2 clock cycles, 4 wait state



25.7.2 Separate Bus mode

(1) DVCC15＝CVCCH＝1.35Vto1.65V, DVCC3＝AVCC3=2.7Vto3.6V,
DAVCC =2.3 Vto2.7V, Ta = －20 to 85°C

① SYSCR3<ALESEL> = “0”, 2 programmed wait state

No.	Parameter	Symbol	Equation		40 MHz (fsys)(Note)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	25				ns
2	A0-A23 valid to $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{AC}	X(1＋ALE) –11		39.0		ns
3	A0-A23 hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x – 11		14.0		ns
4	A0-A23 valid to D0-D15 Data in	t_{AD}		x (2 + TW+ALE) – 43		82.0	ns
5	$\overline{RD}$ asserted to D0-D15 data in	t_{RD}		x (1 + TW) – 40		35.0	ns
6	$\overline{RD}$ width low	t_{RR}	x (1 +TW) –6		69.0		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 – 6		19.0		ns
9	$\overline{WR}$ /$\overline{HWR}$ width low	t_{WW}	x (1 + TW) –6		69.0		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-D15 valid	t_{DO}		9.7		9.7	ns
11	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x (1 + TW) – 11		64.0		ns
12	D0-D15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x – 11		14.0		ns
13	A0-A23 valid to $\overline{WAIT}$ input	t_{AW}		x+ x (ALE)＋ x (TW-1)－32		43.0	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$, $\overline{WR}$ or $\overline{HWR}$ asserted	t_{CW}	x (TW - 3)-16	x (TW – 1) – 29	9.0	46.0	ns

Note 1: No. 1 to 14:

Internal 2 wait insertion，ALE “1” Clock, @40MHz

TW = W + 2N ,

W : Number of Auto wait insertion , 2N : Number of external wait insertion

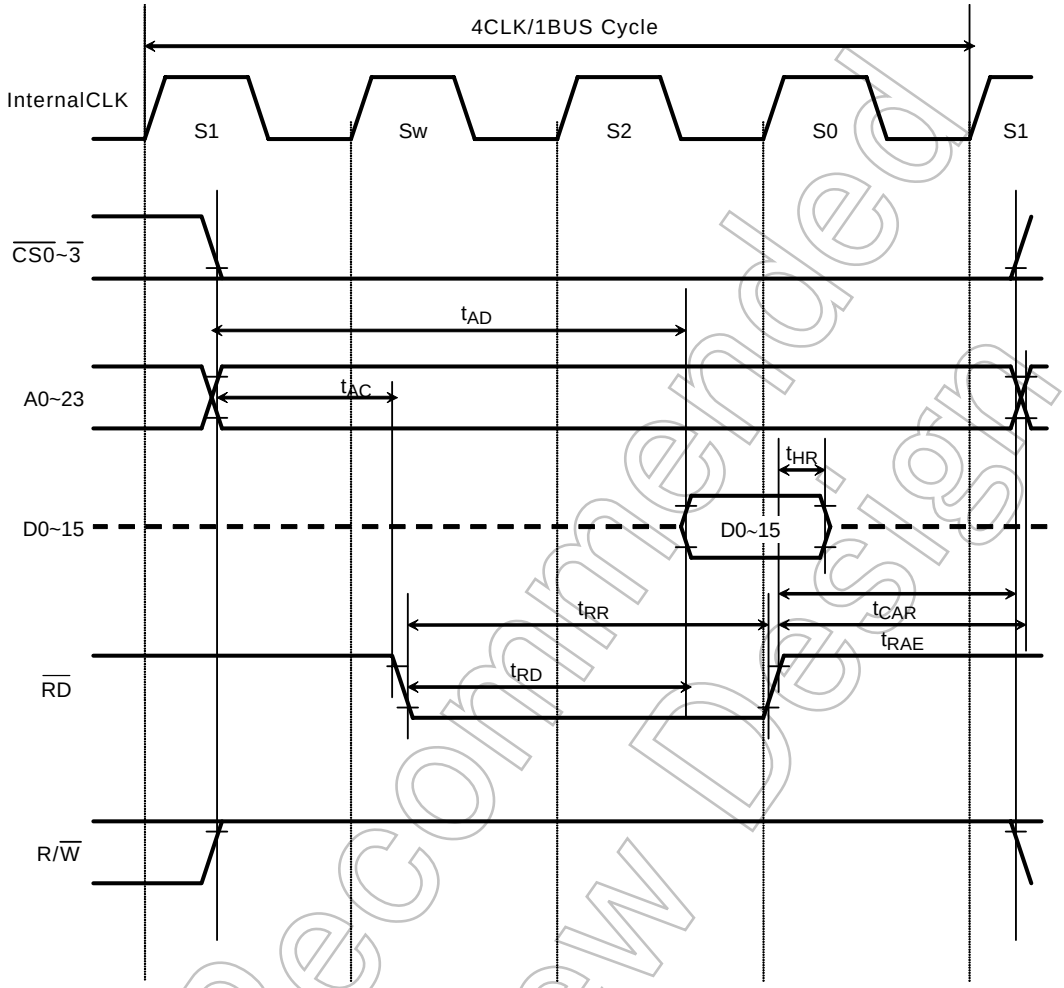
TW = 2 + 2*1 = 4

AC measurement conditions:

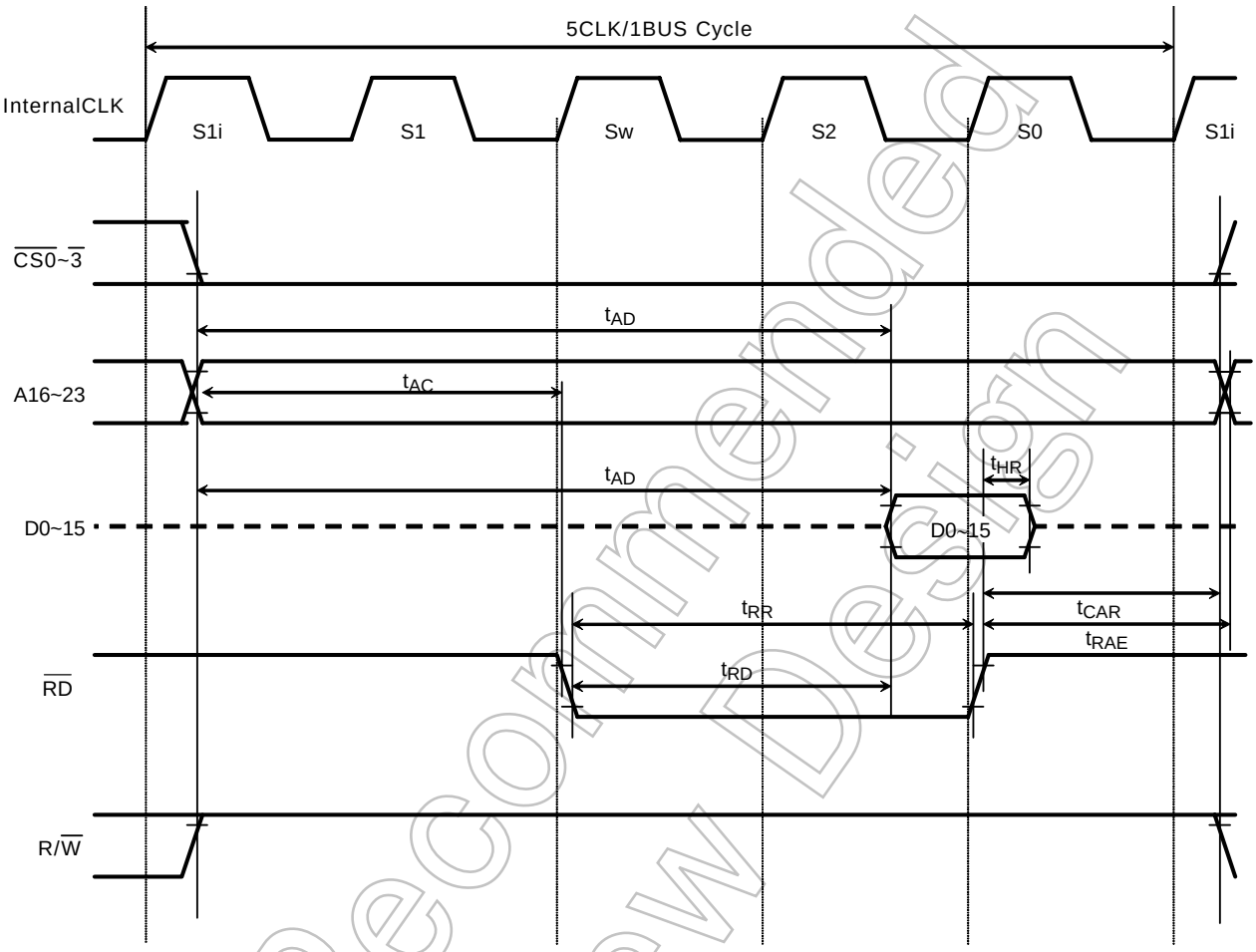
Output levels: High = 0.8DVCC3 V/Low 0.2DVCC3 V, CL = 30 pF

Input levels: High = 0.7DVCC3 V/Low 0.3DVCC3 V

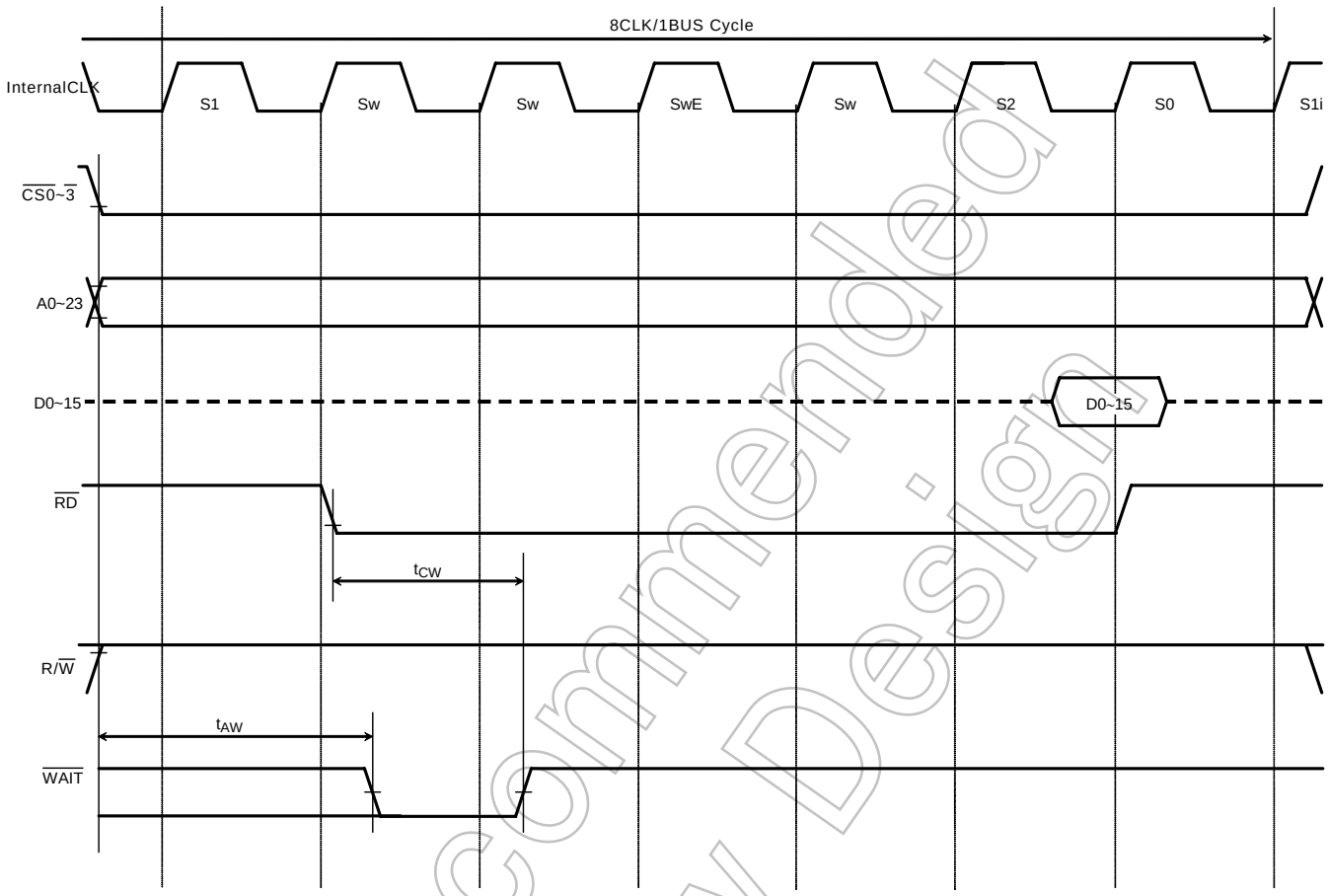
(1) Read cycle timing (SYSCR3<ALESEL> = 0, 1 programmed wait state)



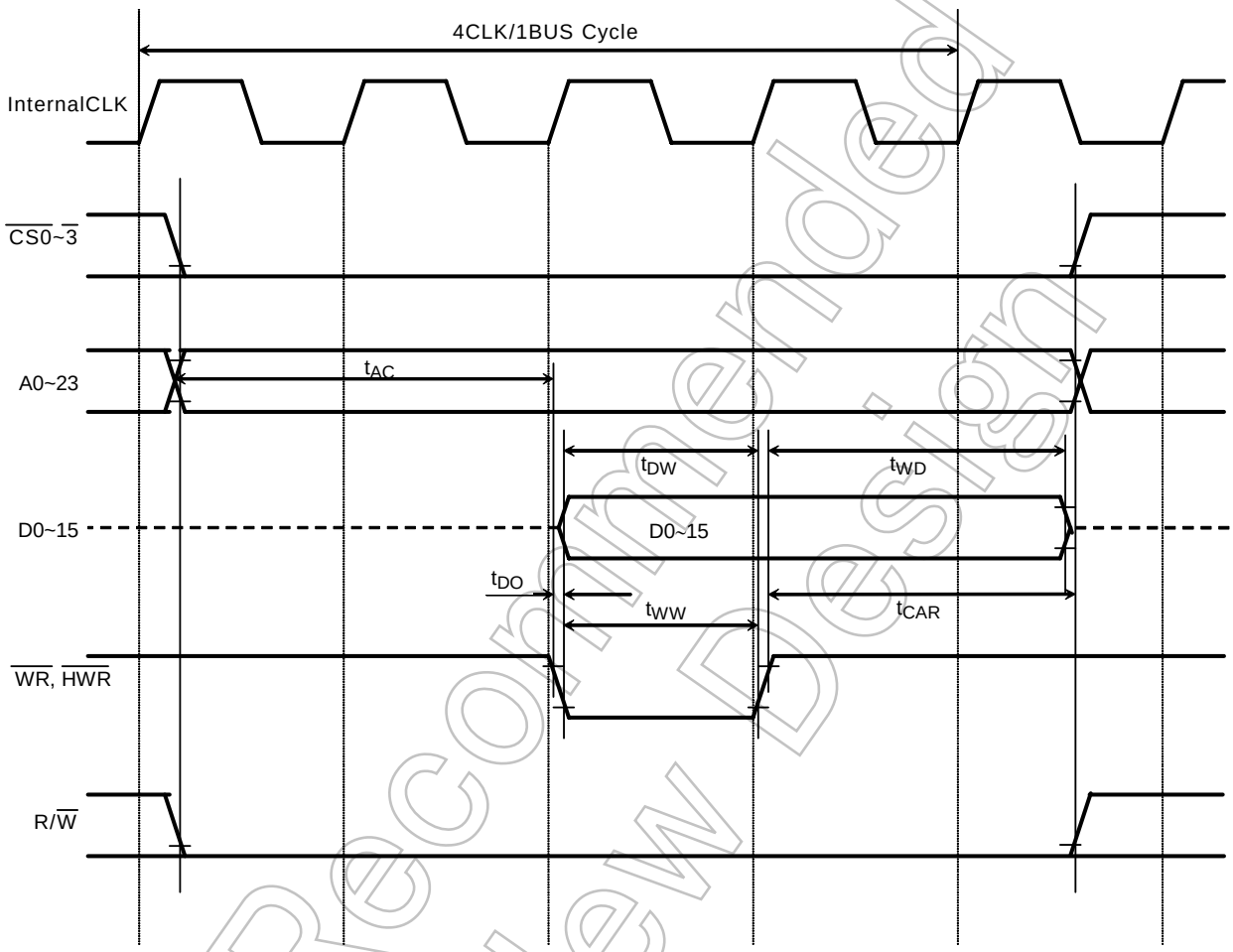
(2) Read cycle timing (SYSCR3<ALESEL> = 1, 1 programmed wait state)



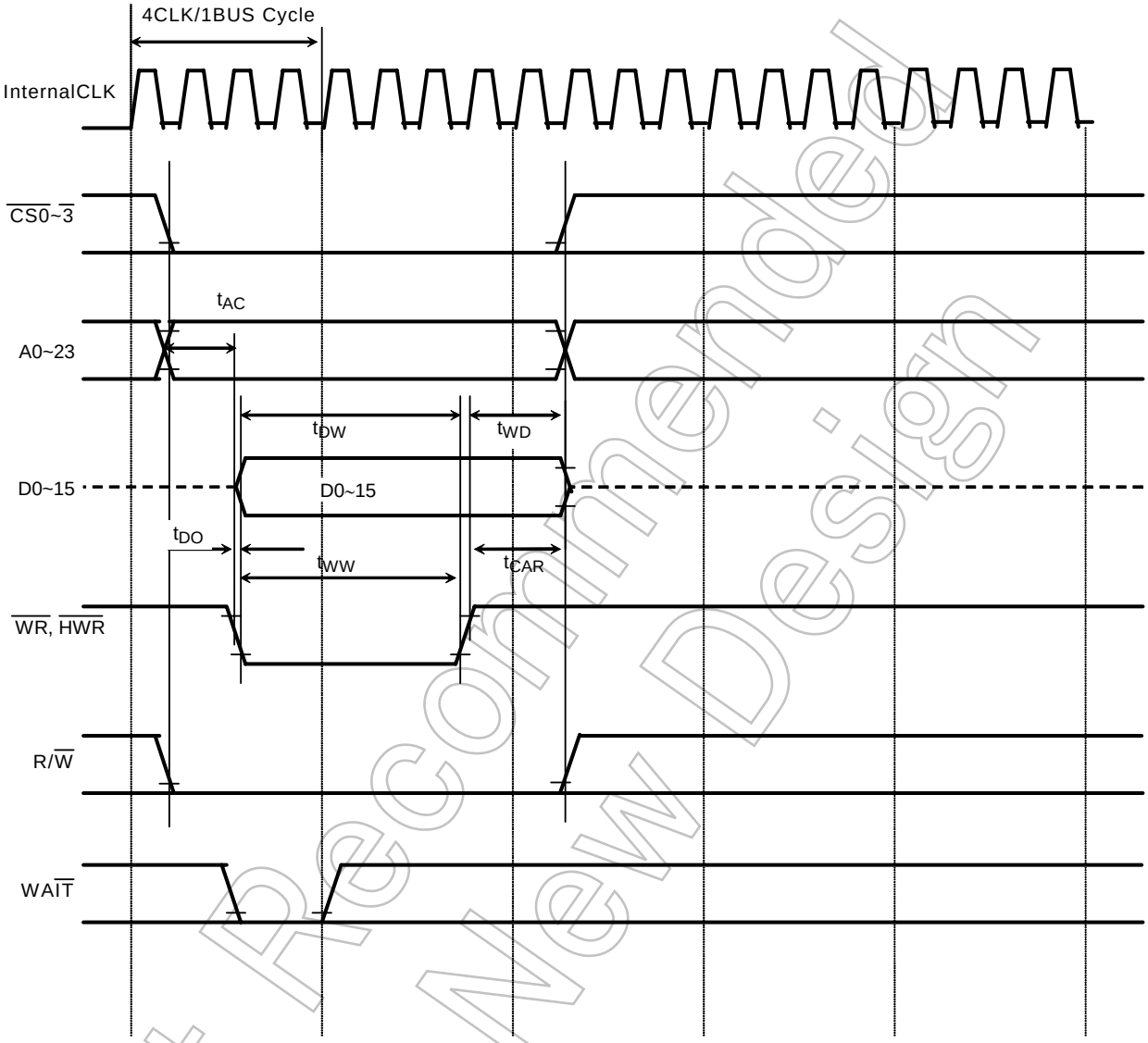
(3)Read cycle timing SYSCR3<ALESEL> = 1, 4 externally generated wait states with N = 1)



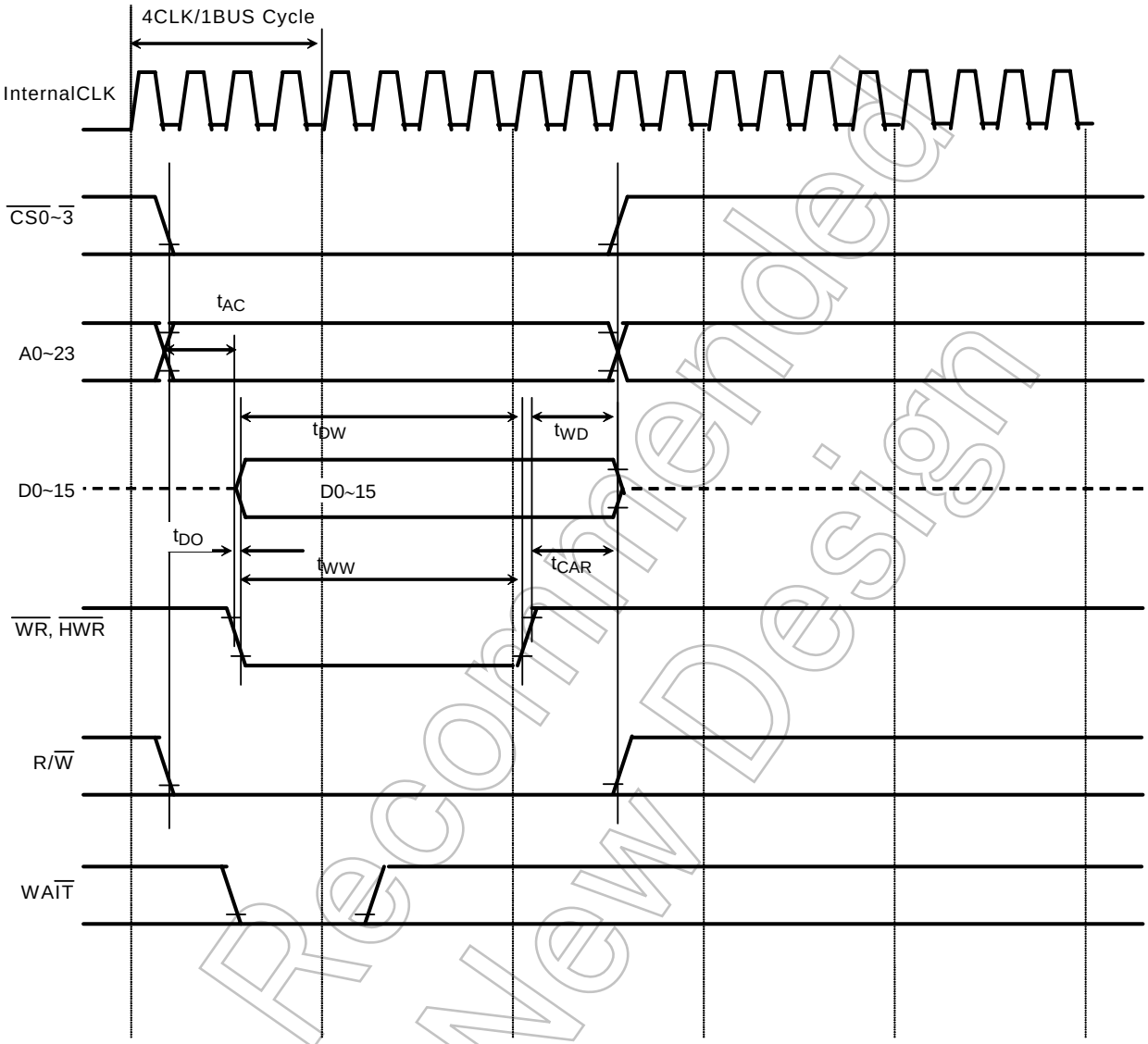
(4) Write cycle timing (SYSCR3<ALESEL> = 1, zero wait sate)



(5) Write cycle timing (SYSCR3<ALESEL> =1, 4 wait state)



(6) Write cycle timing (SYSCR3<ALESEL> = 1, 5 wait state)

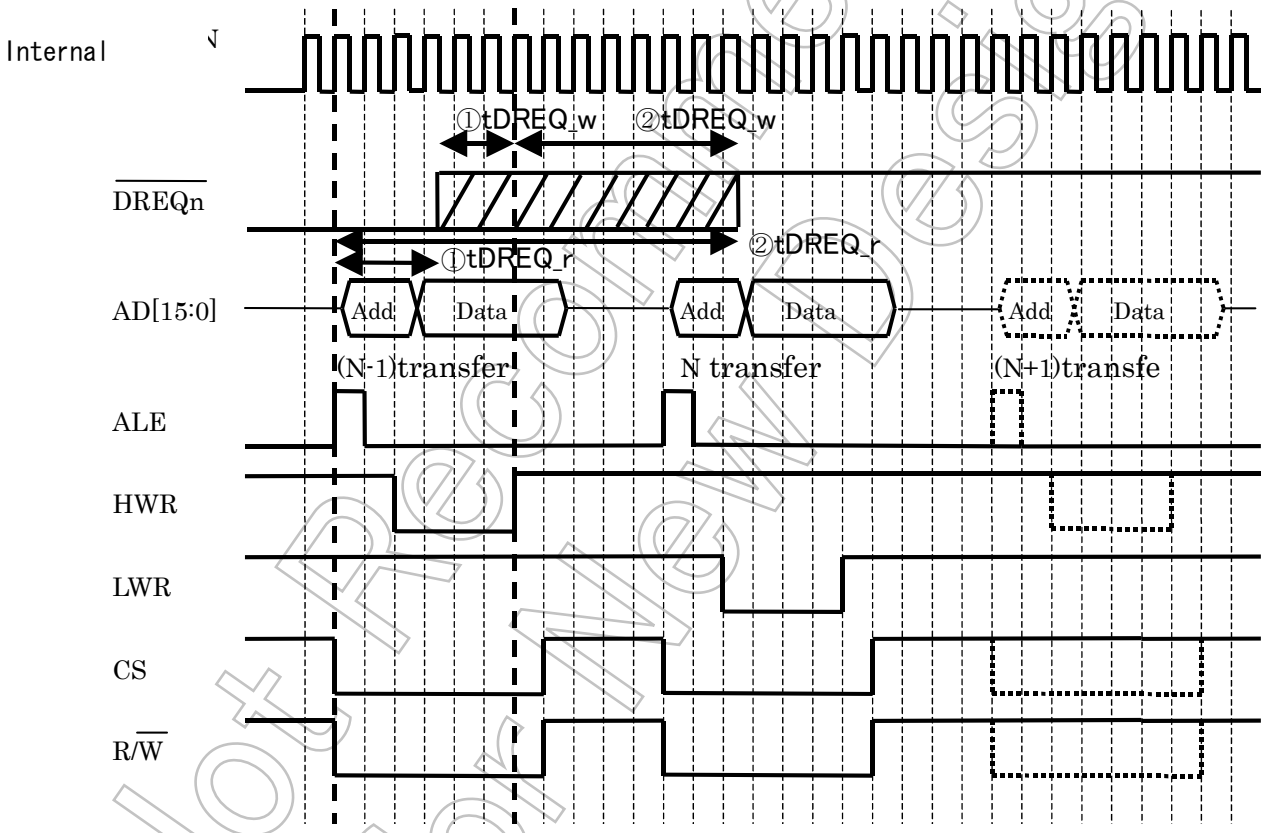


25.8 Transfer with DMA Request

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.

DVCC15＝CVCCH＝1.35Vto1.65V, DVCC3＝AVCC3=2.7Vto3.6V,

DAVCC =2.3 Vto2.7V, Ta = －20to85°C

Parameter	Symbol	Equation		40 MHz (fsys)		Unit
		①Min	②Max	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	50	224	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	−75	98.2	ns

25.9 Serial Channel Timing

(1) I/O Interface mode (DVCC3＝2.7Vto3.6V)

In the table below, the letter x represents the fsys cycle period, which varies depending on the programming of the clock gear function.

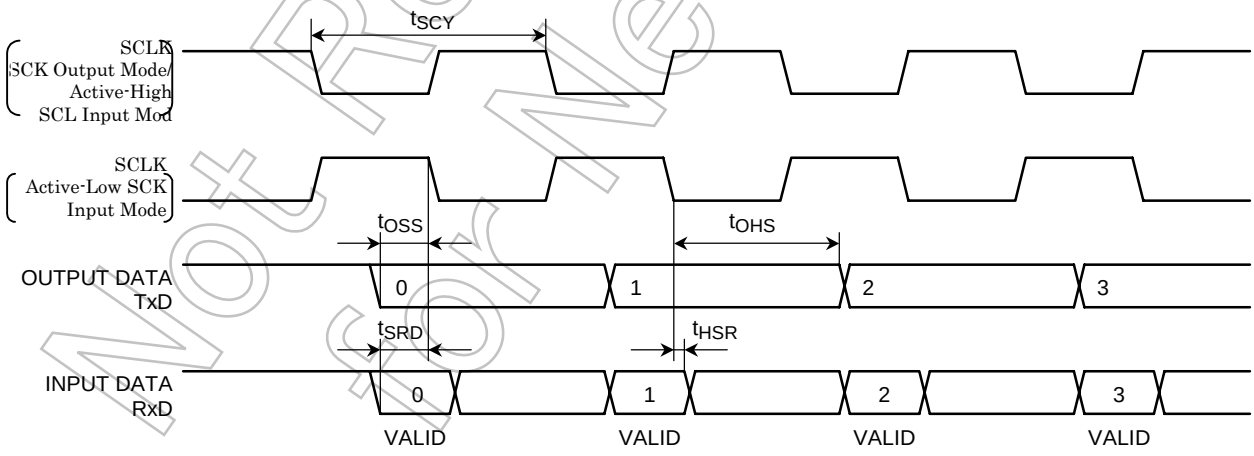
① SCLK input mode (SIO0 to SIO2)

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK period	tSCY	12x		300		ns
SCLK Clock High width(input)	TscH	6x		150		ns
SCLK Clock Low width (input)	TscL	6x		150		ns
TxD data to SCLK rise or fall*	tOSS	2x-30		20		ns
TxD data hold after SCLK rise or fall*	tOHS	8x-15		185		ns
RxD data valid to SCLK rise or fall*	tSRD	30		30		ns
RxD data hold after SCLK rise or fall*	tHSR	2x+30		80		ns

* SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

② SCLK output mode (SIO0 to SIO2)

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCLK period	tSCY	8x		200		ns
TxD data to SCLK rise or fall*	tOSS	4x-10		90		ns
TxD data hold after SCLK rise or fall*	tOHS	4x-10		90		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



25.10 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.

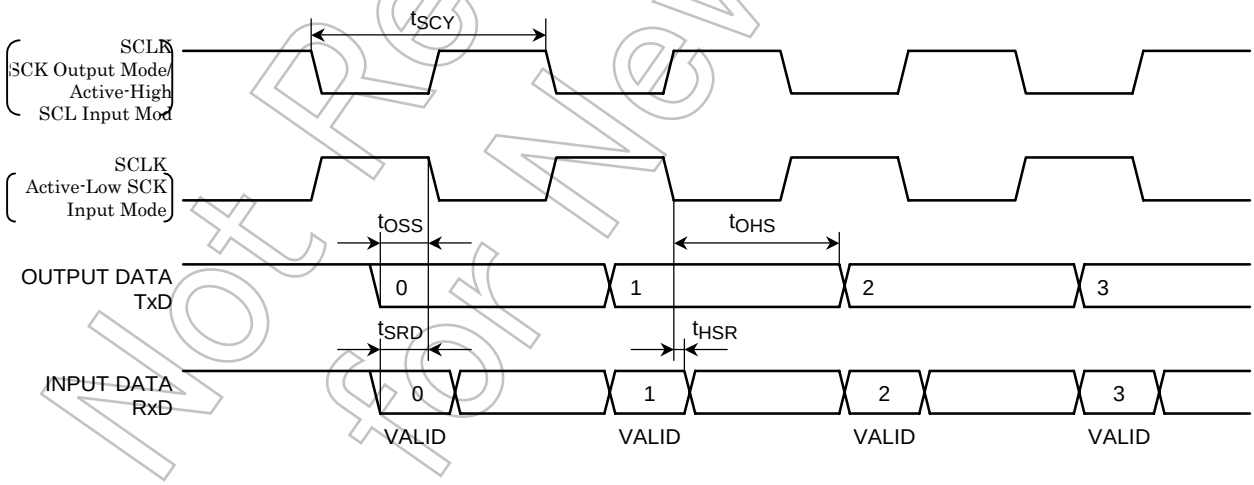
① HSCLK input mode (HSIO0 to HSIO2)

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
HSCLK period	tSCY	12(x/2)		150		ns
HSCLK Clock High width(input)	TscH	3x		75		ns
HSCLK Clock Low width (input)	TscL	3x		75		ns
TxD data to HSCLK rise or fall*	tOSS	2(x/2)-30		-5		ns
TxD data hold after HSCLK rise or fall*	tOHS	8(x/2)-15		85		ns
RxD data valid to HSCLK rise or fall*	tSRD	30		30		ns
RxD data hold after HSCLK rise or fall*	tHSR	2(x/2)+30		55		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	8(x/2)		100		ns
TxD data to HSCLK rise or fall*	tOSS	4(x/2)-10		40		ns
TxD data hold after HSCLK rise or fall*	tOHS	4(x/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



25.11 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: $(2^{n-1}+58)/(fsys/2)$

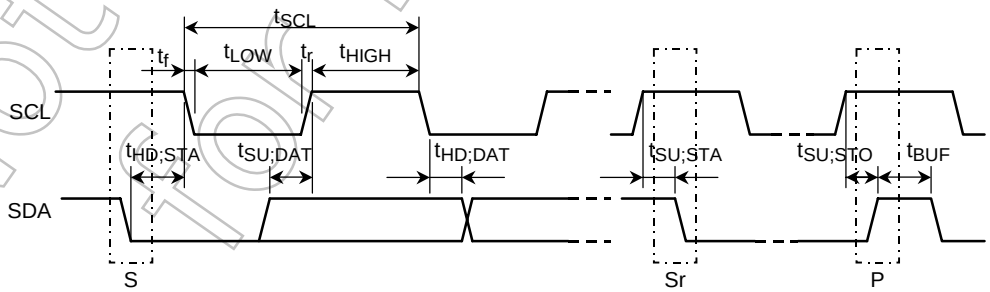
Note 2: SCL clock high width (output) is calculated with $(2^{n-1}+12)/(fsys/2)$

Notice: On I^2C-bus specification, Maximum Speed of Standard Mode is 100KHz ,Fast mode is 400Khz. Internal SCL clock Frequency setting should be shown above Note1 & Note2.

Note 3: The output data hold time is equal to 12x

Note 4: The Philips I^2C-bus specification states that a device must internally provide a hold time of at least 300 ns for the SDA signal to bridge the undefined region of the fall edge of SCL. However, 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.

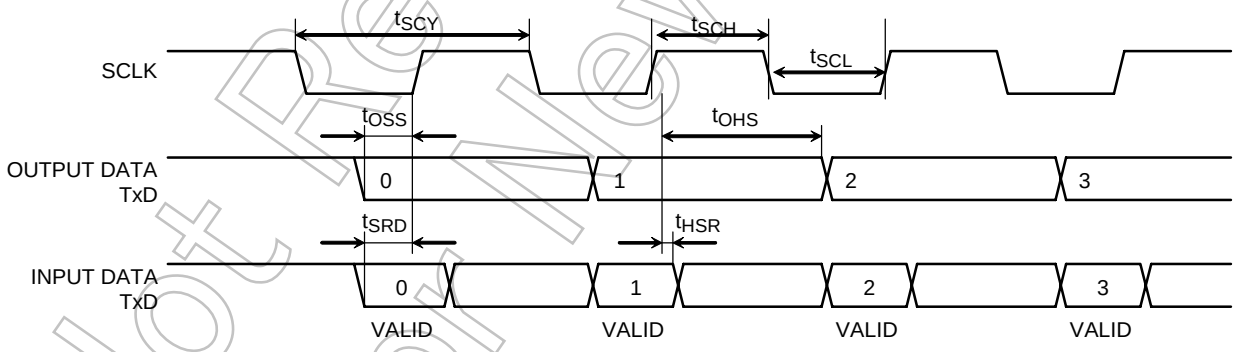
The electrical specifications below are for an SCK signal with a 50% duty cycle.

③ SCK Input mode

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
SCK period	tSCY	16x		400		ns
SCK Clock High width(input)	TscH	8x		200		ns
SCKClock Low width(input)	TscH	8x		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 MHz		Unit
		Min	Max	Min	Max	
SCK period (programmable)	t_{SCY}	$2^n \cdot T$		800		ns
SO data to SCK rise	t_{OSS}	$(t_{SCY}/2) - 20$		380		ns
SO data hold after SCK rise	t_{OHS}	$(t_{SCY}/2) - 20$		T380		ns
SI data valid to SCK rise	t_{SRD}	2x + 30		55		ns
SI data hold after SCK rise	t_{HSR}	0		0		ns



25.12 Event Counter

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2X + 100		150		ns
Clock high pulse width	t_{VCKH}	2X + 100		150		ns

25.13 Timer Capture

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2X + 100		150		ns
High pulse width	t_{CPH}	2X + 100		150		ns

25.14 General Interrupts

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTA	t_{INTAL}	X + 100		125		ns
High pulse width for INT0-INTA	t_{INTAH}	X + 100		125		ns

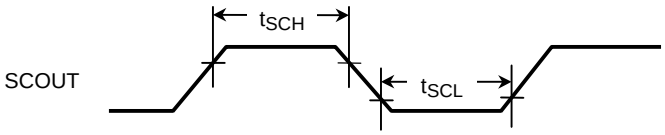
25.15 STOP /SLEEP/SLOW Wake-up Interrupts

Parameter	Symbol	Equation		40 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

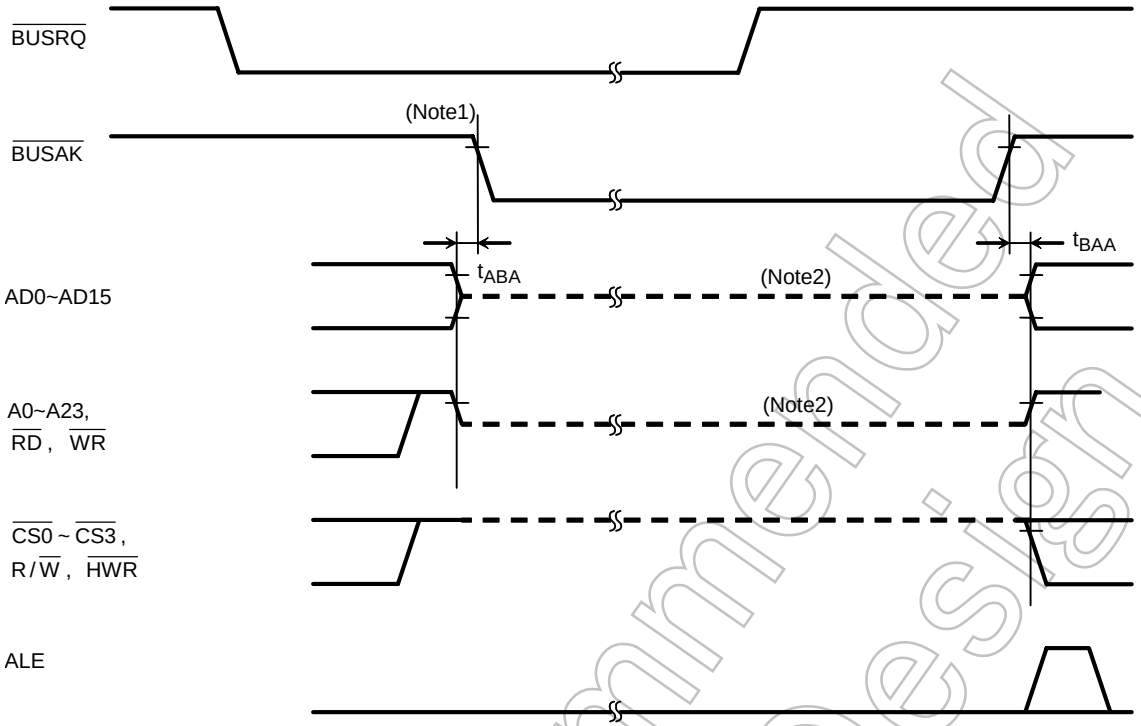
25.16 SCOUT Pin

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Clock high pulse width	t_{SCH}	0.5T – 5		7.5		ns
Clock low pulse width	t_{SCL}	0.5T – 5		7.5		ns

Note: In the above table, the letter T represents the cycle period of the SCOUT output clock.



25.17 Bus Request and Bus Acknowledge Signals



Parameter	Symbol	Equation		40 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 TMP19A43FDXBG 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.

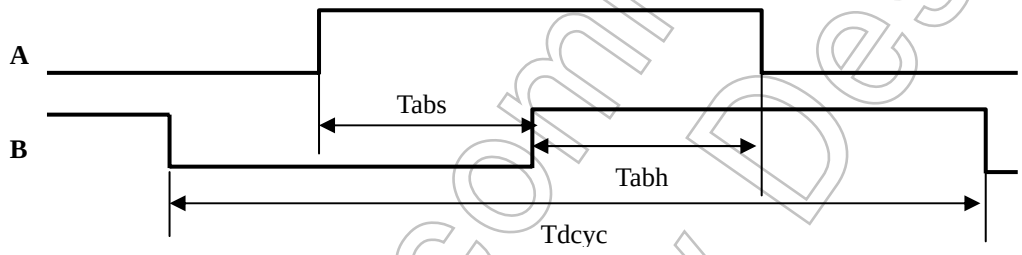
25.18 KWUP Input

Parameter	Symbol	Equation		40 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

25.19 Dual Pulse Input

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Dual input pulse period	Tdcyc	8Y		400		ns
Dual input pulse setup	Tabs	Y+20		70		ns
Dual input pulse hold	Tabh	Y+20		70		ns

Y : fsys/2

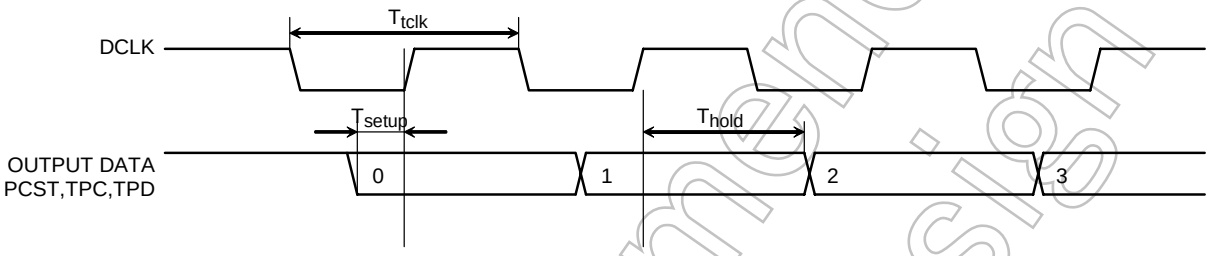


25.20 ADTRG input

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for ADTRG	tad_L	fsys/2+20		32.5		ns
High pulse width for ADTRG	Tadh	fsys/2+20		32.5		ns

25.21 DSU

Parameter	Symbol	Equation		40 MHz		Unit
		Min	Max	Min	Max	
PCST valid to DCLK negated	Tsetup	11		11		ns
PCST hold after DCLK negated	Thold	0.5		0.5		ns
TPC valid to DCLK negated	Tsetup	11		11		ns
TPC hold after DCLK negated	Thold	0.5		0.5		ns
TPD valid to DCLK negated	Tsetup	11		11		ns
TPD hold after DCLK negated	Thold	0.5		0.5		ns



25.22 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