TOSHIBA

32-Bit TX System RISC
TX19 Family
TMP19A63F10XBG

Rev1.1

2008-3-28

32-Bit RISC Microprocessor TX19 Family

TMP19A63F10XBG

1. Overview and features

TMP19A63 is equipped with the TX19A processor core that forms a high-performance 32-bit RISC processor series. The core was developed based on the MIPS32ISA that contains a 32-bit instruction set and the MIPS16eISA that contains an instruction set of high code efficiency. TOSHIBA uniquely integrated these two and the MIPS16e-TX ™ASE (Application Specific Extension), which includes an extended instruction set of high code efficiency.

TMP19A63 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 TMP19A63 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 consumption

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

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

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

(2) Internal program memory and data memory

Product name	Built-in ROM	Built-in RAM
TMP19A63CDXBG	512Kbyte	24Kbyte
TMP19A63F10XBG	1Mbyte(Flash)	48Kbyte

- ROM correction function: 8word×12 block

(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

  Added CS recovery function (wait is inserted within RD (WR)↑－CS↑)

  (For 1 clock)

  External wait X+2N-capable （X=2 to 7）
  Changed ALE width

(4) DMA controller : 8 channels

- 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 : 36 channels

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit PPG output
- Input capture function

  2-phase pulse input counter function (2 channels assigned to perform this function):

(6)32-bit timer

- 32-bit input capture register: 4 channels
- 32-bit compare register: 4 channels
- 32-bit time base timer: 2 channels

(7) General-purpose serial interface: 11 channels

- Selectable between the UART mode and the synchronization mode

(8) Serial bus interface: 2 channels

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

(9) 10-bit A/D converter (with S/H): 32 channels

- An optional trigger by the internal timer

- Fixed channel/scan mode
- Single/repeat mode
- Top-priority conversion mode
- Timer monitor function

  1.7usec@27MHz (at 54MHz) 1.15usec@40MHz (at 40MHz)
  (Consists of 2 units. Capable of simultaneous conversion. No definition for error between units)

(11) Watchdog timer: 1 channel
(12) Chip select/ wait controller: 6 channels
(13) Interrupt function

- CPU: 2 factors …software interrupt instruction
- Internal 83 factors…The order of precedence can be set over 7 levels
  (except the watchdog timer interrupt)
  39- independent-interrupt factors are included.
- External: 20 factors…The order of precedence can be set over 7 levels.
  (Except for NMI interrupt)
  8 factors, which are KWUP, are united as an interrupt factor.

(14) Input and output ports: 212 pins
(15) Standby function

- Two stand-by modes (IDLE, STOP)

(16) Clock generator

- Built-in PLL (multiplication by 4)
- Clock gear function: The high-speed clock can be divided into 1/1, 1/2，1/4, 1/8.

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

Big endian

	31 24	23 16	15 8	7 0	Word address
Upper address ↑	8	9	10	11	8
	4	5	6	7	4
Lower address	0	1	2	3	0

- The most significant byte is 0 (bit 31-24).
- The address of the most significant byte specifies the word address.

Little endian

	31 24	23 16	15 8	7 0	Word address
Upper address ↑	11	10	9	8	8
	7	6	5	4	4
Lower address	3	2	1	0	0

- The least significant byte is 0 (bit 7-0).
- The address of the least significant byte specifies the word address.

(18) Operating frequency

- 54MHz (DVCC15 = 1.35V-1.65V)

(19) Operating voltage range

- Core: 1.35 - 1.65V
- I/O: 1.65 - 3.3 V
- ADC: 2.7 - 3.3 V

(20) Temperature range

- -20°C-85°C
- 0°C -70°C（Flash W/E）

(21) Package

- P-TFBGA289 (11mm×11mm, 0.5mm pitch)

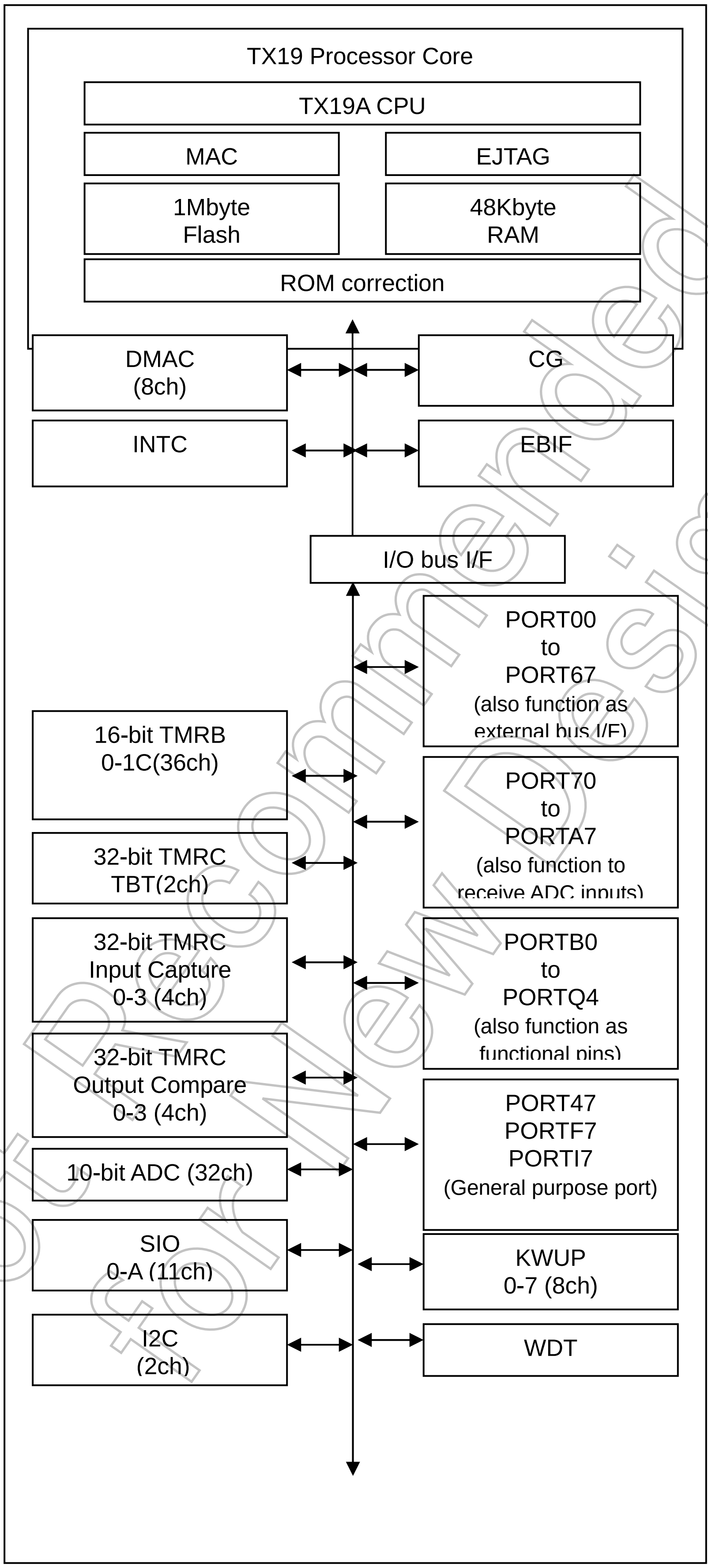


Fig. 1.1 TMP19A63F10XBG Block Diagram

2. Pin Layout and Pin Functions

This section shows the pin layout of TMP19A63 and describes the names and functions of input and output pins.

2.1 Pin Layout (Top view)

Fig. 2.1.1 shows the pin layout of TMP19A63.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	B20
C1	C2																	C19	C20
D1	D2		D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17		D19	D20
E1	E2		E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17		E19	E20
F1	F2		F4	F5											F16	F17		F19	F20
G1	G2		G4	G5		G7	G8	G9	G10	G11	G12	G13	G14		G16	G17		G19	G20
H1	H2		H4	H5		H7	H8	H9	H10	H11	H12	H13	H14		H16	H17		H19	H20
J1	J2		J4	J5		J7	J8	J9				J13	J14		J16	J17		J19	J20
K1	K2		K4	K5		K7	K8					K13	K14		K16	K17		K19	K20
L1	L2		L4	L5		L7	L8					L13	L14		L16	L17		L19	L20
M1	M2		M4	M5		M7	M8					M13	M14		M16	M17		M19	M20
N1	N2		N4	N5		N7	N8	N9	N10	N11	N12	N13	N14		N16	N17		N19	N20
P1	P2		P4	P5		P7	P8	P9	P10	P11	P12	P13	P14		P16	P17		P19	P20
R1	R2		R4	R5											R16	R17		R19	R20
T1	T2		T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17		T19	T20
U1	U2		U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17		U19	U20
V1	V2																	V19	V20
W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18	W19	W20
Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16	Y17	Y18	Y19	Y20

Fig. 2.1.1 Pin Layout Diagram (P-FBGA289)

PIN No.	PIN Name	PIN No.	PIN Name	PIN No.	PIN Name	PIN No.	PIN Name
A1	N.C(GND)	B1	N.C(GND)	C1	PL0/TC4IN	D1	PL2
A2	N.C(GND)	B2	N.C(GND)	C2	PL1/TC5IN	D2	PL3/TCOUTB0
A3	RESET	B3	PCST0				
A4	PCST1	B4	PCST2			D4	DVSS
A5	PCST3	B5	PCST4			D5	PQ0/DREQ2
A6	DCLK	B6	TOVR			D6	TCK
A7	TDO	B7	TDI			D7	DINT
A8	PP6/TPC6/TPD6	B8	PP7/TPC7/TPD7			D8	PO6/TPD6
A9	PP4/TPC4/TPD4	B9	PP5/TPC5/TPD5			D9	PO4/TPD4
A10	PP2/TPC2/TPD2	B10	PP3/TPC3/TPD3			D10	PO2/TPD2
A11	PP0/TPC0/TPD0	B11	PP1/TPC1/TPD1			D11	PO0/TPD0
A12	PJ4/TC1IN	B12	PJ5/SO1/SDA1			D12	PJ6/SI1/SCL1
A13	PJ2/SCLK8/CTS8	B13	PJ3/TC0IN			D13	PM6/TCOUTA0
A14	PJ0/TXD8	B14	PJ1/RXD8			D14	PM4/INT4
A15	PF6/SCLK1/CTS1	B15	PF7			D15	PM2/INT2
A16	PF4/TXD1	B16	PF5/RXD1			D16	PM0/INT0
A17	PF2/SCLK0/CTS0	B17	PF3			D17	PG5/RXD3
A18	PF0/TXD0	B18	PF1/RXD0				
A19	N.C(GND)	B19	N.C(GND)	C19	PG7/TBTIN2	D19	PG4/TXD3
A20	N.C(GND)	B20	N.C(GND)	C20	PG6/SCLK3/CTS3	D20	PG3/TBTIN1

Table 2.1.1 Pin Names and Functions (1/3)

PIN No.	PIN Name	PIN No.	PIN Name	PIN No.	PIN Name	PIN No.	PIN Name
E1	PL4/TXD9	F1	PL6/SCLK9/CTS9	G1	P00/D0/AD0	H1	P02/D2/AD2
E2	PL5/RXD9	F2	PL7/TCOUTB1	G2	P01/D1/AD1	H2	P03/D3/AD3
E4	PQ1/DACK2	F4	PQ2/DREQ3	G4	PK0/KEY0	H4	PK2/KEY2
E5	DVSS	F5	PQ3/DACK3	G5	PK1/KEY1	H5	PK3/KEY3
E6	TRST						
E7	TMS			G7	DVSSC	H7	PK4/KEY4
E8	PO7/TPD7			G8	EJE	H8	DVSSD
E9	PO5/TPD5			G9	DVCC33	H9	FVCC30
E10	PO3/TPD3			G10	DVCC34	H10	FVCC31
E11	PO1/TPD1			G11	DVCC34	H11	FVCC15
E12	PJ7/SCK1			G12	DVCC34	H12	DVCC15
E13	PM7/TCOUTA1			G13	DVCC32	H13	AVSS1
E14	PM5/INT5			G14	AVSS0	H14	P85/ANA13
E15	PM3/INT3						
E16	PM1/INT1	F16	P77/ANA7	G16	P87/ANA15	H16	P84/ANA12
E17	PG2/SCLK2/CTS2	F17	P76/ANA6	G17	P86/ANA14	H17	P83/ANA11
E19	PG1/RXD1	F19	P75/ANA5	G19	P73/ANA3	H19	P71/ANA1
E20	PG0/TXD2	F20	P74/ANA4	G20	P72/ANA2	H20	P70/ANA0
J1	P04/D4/AD4	K1	P06/D6/AD6	L1	P10/D8/AD8/A8	M1	P12/D10/AD10/A10
J2	P05/D5/AD5	K2	P07/D7/AD7	L2	P11/D9/AD9/A9	M2	P13/D11/AD11/A11
J4	P50/A0	K4	P52/A2	L4	P54/A4	M4	P56/A6
J5	P51/A1	K5	P53/A3	L5	P55/A5	M5	P57/A7
J7	PK5/KEY5	K7	PK6/KEY6	L7	PK7/KEY7	M7	BW0
J8	DVCC30	K8	DVCC30	L8	DVCC30	M8	DVCC15
J9	DVSS						
J13	AVCC30	K13	AVREFH0	L13	AVREFH1	M13	AVCC31
J14	P82/ANA10	K14	PA7/ANB15	L14	PA4/ANB12	M14	DVCC15
J16	P81/ANA9	K16	PA6/ANB14	L16	PA3/ANB11	M16	PA1/ANB9
J17	P80/ANA8	K17	PA5/ANB13	L17	PA2/ANB10	M17	PA0/ANB8
J19	P97/ANB7	K19	P95/ANB5	L19	P93/ANB3	M19	P91/ANB1
J20	P96/ANB6	K20	P94/ANB4	L20	P92/ANB2	M20	P90/ANB0
N1	P14/D12/AD12/A12	P1	P16/D14/AD14/A14	R1	P40/*CS0	T1	P42/*CS2
N2	P15/D13/AD13/A13	P2	P17/D15/AD15/A15	R2	P41/*CS1	T2	P43/*CS3
N4	P30/*RD	P4	P32/*HWR	R4	P34/*BUSRQ	T4	P36/R/*W
N5	P31/*WR	P5	P33/*WAIT/*RDY	R5	P35/*BUSAK	T5	P61/A9
						T6	P63/A11
N7	BW1	P7	TEST2			T7	P65/A13
N8	TEST1	P8	TEST3			T8	PN1/INT7
N9	BUSMD	P9	ENDIAN			T9	PN3/ADTRG-A
N10	FVCC15	P10	*NMI			T10	PN5/RXDA
N11	DVCC15	P11	DVCC31			T11	PN7/ADTRG-B
N12	PLLSEL	P12	DVCC31			T12	PH1/RXD4
N13	DVSSF	P13	CVSS			T13	PH3/INT9
N14	CVCC15	P14	DVSS			T14	PH5/RXD5
						T15	PH7/INTA
N16	PC7/TB0FIN1	P16	PC5/TB0EIN1	R16	PC3/TB0DIN1	T16	DVSSG
N17	PC6/TB0FIN0	P17	PC4/TB0EIN0	R17	PC2/TB0DIN0	T17	PC1/TB0CIN1
N19	PB7/TB0BIN1	P19	PB5/TB0AIN1	R19	PB3/TB9IN1	T19	PB1/TB8IN1
N20	PB6/TB0BIN0	P20	PB4/TB0AIN0	R20	PB2/TB9IN0	T20	PB0/TB8IN0

Table 2.1.1 Pin Names and Functions (2/3)

PIN No.	PIN Name	PIN No.	PIN Name	PIN No.	PIN Name	PIN No.	PIN Name
U1	P44/*CS4	V1	P46/SCOUT	W1	N.C(GND)	Y1	N.C(GND)
U2	P45/*CS5	V2	P47	W2	N.C(GND)	Y2	N.C(GND)
				W3	P21/A17/A1/A17	Y3	P20/A16/A0/A16
U4	P37/ALE			W4	P23/A19/A3/A19	Y4	P22/A18/A2/A18
U5	P60/A8			W5	P25/A21/A5/A21	Y5	P24/A20/A4/A20
U6	P62/A10			W6	P27/A23/A7/A23	Y6	P26/A22/A6/A22
U7	P64/A12			W7	P67/A15	Y7	P66/A14
U8	PN0/INT6			W8	PI1/RXD6	Y8	PI0/TXD6
U9	PN2/INT8			W9	PI3/INTB	Y9	PI2/SCLK6/CLS6
U10	PN4/TXDA			W10	PI5/RXD7	Y10	PI4/TXD7
U11	PN6/SCLKA/CTSA			W11	PI7	Y11	PI6/SCLK7/CTS7
U12	PH0/TXD4			W12	PE1/TB17OUT	Y12	PE0/TB16OUT
U13	PH2/SCLK4/CTS4			W13	PE3/TB19OUT	Y13	PE2/TB18OUT
U14	PH4/TXD5			W14	PE5/SO0/SCA0	Y14	PE4/TB1AOUT
U15	PH6/SCLK5/CTS5			W15	PE7/SCK0	Y15	PE6/SI0/SCL0
U16	PD2/TB11IN0			W16	PD1/TB10IN1	Y16	PD0/TB10IN0
U17	DVSSH			W17	PD4/TB12IN0	Y17	PD3/TB11IN1
				W18	PD6/TB14OUT	Y18	PD5/TB12IN1
U19	PC0/TB0CIN0	V19	PD7/TB15OUT	W19	N.C(GND)	Y19	N.C(GND)
U20	X2	V20	X1	W20	N.C(GND)	Y20	N.C(GND)

Table 2.1.1 Pin Names and Functions (3/3)

2.3 Pin Names and Functions

Tables 2.3 show the names and functions of input and output pins.

Table 2.3 Pin Names and Functions (1/8)

Pin name	No. of pins	Input or output	Function
P00~P07 D0~D7 AD0~D7	8	Input/output Input/output Input/output	Port 0: Input/output port that allows input/output to be set in units of bits Data (lower): Data bus 0~7 (separate bus mode) Address data (lower): Address data bus 0~7 (multiplexed bus mode)
P10~P17 D8~D15 AD8~AD15 A8~A15	8	Input/output Input/output Input/output Output	Port 1: Input/output port that allows input/output to be set in units of bits Data (upper): Data bus 8~15: (separate bus mode) Address data (upper): Address data bus 8~15 (multiplexed bus mode) Address: Address bus 8~15 (multiplexed bus mode)
P20~P27 A16~A23 A0~A7 A16~A23	8	Input/output Output Output Output	Port 2: Input/output port that allows input/output to be set in units of bits Address: Address bus 16~23 (separate bus mode) Address: Address bus 0~7 (multiplexed bus mode) Address: Address bus 16~23 (multiplexed bus mode)
P30 *RD	1	Input/output Output	Port 30:Input/output port (with pull-up) Read: Strobe signal for reading external memory
P31 *WR	1	Input/output Output	Port 31:Input/output port (with pull-up) Write: Strobe signal for writing data of D0 to D7 pins
P32 *HWR	1	Input/output Output	Port 32:Input/output port (with pull-up) Write upper-pin data: Strobe signal for writing data of D8 to D15 pins
P33 *WAIT *RDY	1	Input/output Input Input	Port 33:Input/output port (with pull-up) Wait: Pin for requesting CPU to put a bus in a wait state Ready: Pin for notifying CPU that a bus is ready
P34 *BUSRQ	1	Input/output Input	Port 34:Input/output port (with pull-up) Bus request: Signal requesting CPU to allow an external master to take the bus control authority
P35 *BUSAK	1	Input/output Output	Port 35:Input/output port (with pull-up) Bus acknowledge: Signal notifying that CPU has released the bus control authority in response to *BUSREQ
P36 R/*W	1	Input/output Output	Port 36:Input/output port (with pull-up) Read/write: "1" shows a read cycle or a dummy cycle. "0" shows a write cycle.
P37 ALE	1	Input/output Output	Port 37:: Input/output port Address latch enable (address latch is enabled only if access to external memory is taking place, that is multiplex bus mode)
P40 *CS0	1	Input/output Output	Port 40:Input/output port (with pull-up) Chip select 0:"0" is output if the address is in a designated address area.
P41 *CS1	1	Input/output Output	Port 41:Input/output port (with pull-up) Chip select 1:"0" is output if the address is in a designated address area.
P42 *CS2	1	Input/output Output	Port 42:Input/output port (with pull-up) Chip select 2:"0" is output if the address is in a designated address area.
P43 *CS3	1	Input/output Output	Port 43:Input/output port (with pull-up) Chip select 3:"0" is output if the address is in a designated address area.
P44 *CS4	1	Input/output Output	Port 44:Input/output port (with pull-up) Chip select 4:"0" is output if the address is in a designated address area.
P45 *CS5	1	Input/output Output	Port 45:Input/output port (with pull-up) Chip select 5:"0" is output if the address is in a designated address area.

Table 2.3 Pin Names and Functions (2/8)

Pin name	No. of pins	Input or output	Function
P46 SCOUT	1	Input/output Output	Port 46: Input/output port System clock output: Selectable between high- and low-speed clock outputs, as in the case of CPU
P47	1	Input/output	Port 47: Input/output port
P50~P57 A0~A7	8	Input/output Output	Port 5: Input/output port that allows input/output to be set in units of bits Address: Address bus 0~7 (separate bus mode)
P60~P67 A8~A15	1	Input/output Output	Port 60 ~67 :Input/output port Address: Address bus 4 (separate bus mode)
P70~P77 ANA0~ANA7	8	Input Input	Port 7:Port used exclusively for input Analog input: Input from A/D converter
P80~P87 ANA8~ANA15	8	Input Input	Port 8:Port used exclusively for input Analog input: Input from A/D converter
P90~P97 ANB0~ANB7	8	Input Input	Port 9:Port used exclusively for input Analog input: Input from A/D converter
PA0~PA7 ANB8~ANB15	8	Input Input	Port A: Port used exclusively for input Analog input: Input from A/D converter
PB0 TB8IN0	1	Input/output Input	Port B0:Input/output port 16-bit timer 8 input 0:For inputting the capture trigger of a 16-bit timer 8
PB1 TB8IN1	1	Input/output Input	Port B1:Input/output port 16-bit timer 8 input 1:For inputting the capture trigger of a 16-bit timer 8
PB2 TB9IN0	1	Input/output Input	Port B2:Input/output port 16-bit timer 9 input 0:For inputting the capture trigger of a 16-bit timer 9
PB3 TB9IN1	1	Input/output Input	Port B3:Input/output port 16-bit timer 9 input 1:For inputting the capture trigger of a 16-bit timer 9
PB4 TBAIN0	1	Input/output Input	Port B4:Input/output port 16-bit timer A input 0:For inputting the capture trigger of a 16-bit timer A
PB5 TBAIN1	1	Input/output Input	Port B5:Input/output port 16-bit timer A input 1:For inputting the capture trigger of a 16-bit timer A
PB6 TBBIN0	1	Input/output Input	Port B6:Input/output port 16-bit timer B input 0:For inputting the capture trigger of a 16-bit timer B
PB7 TBBIN1	1	Input/output Input	Port B7:Input/output port 16-bit timer B input 1:For inputting the capture trigger of a 16-bit timer B
PC0 TBCIN0	1	Input/output Input	Port C0:Input/output port 16-bit timer C input 0:For inputting the capture trigger of a 16-bit timer C/Two-phase counter input pin
PC1 TBCIN1	1	Input/output Input	Port C1:Input/output port 16-bit timer C input 1:For inputting the capture trigger of a 16-bit timer C/Two-phase counter input pin
PC2 TBDIN0	1	Input/output Input	Port C2:Input/output port 16-bit timer D input 0:For inputting the capture trigger of a 16-bit timer D
PC3 TBDIN1	1	Input/output Input	Port C3:Input/output port 16-bit timer D input 1:For inputting the capture trigger of a 16-bit timer D
PC4 TBEIN0	1	Input/output Input	Port C4:Input/output port 16-bit timer E input 0:For inputting the capture trigger of a 16-bit timer E
PC5 TBEIN1	1	Input/output Input	Port C5:Input/output port 16-bit timer E input 1:For inputting the capture trigger of a 16-bit timer E
PC6 TBFIN0	1	Input/output Input	Port C6:Input/output port 16-bit timer F input 0:For inputting the capture trigger of a 16-bit timer F
PC7 TBFIN1	1	Input/output Input	Port C7:Input/output port 16-bit timer F input 1:For inputting the capture trigger of a 16-bit timer 10
PD0 TB10IN0	1	Input/output Input	Port D0:Input/output port 16-bit timer 10 input 0:For inputting the capture trigger of a 16-bit timer 10

Table 2.3 Pin Names and Functions (3/8)

Pin name	No. of pins	Input or output	Function
PD1 TB10IN1	1	Input/output Input	Port D1:Input/output port 16-bit timer 10 input 1:For inputting the capture trigger of a 16-bit timer 10
PD2 TB11IN0	1	Input/output Input	Port D2:Input/output port 16-bit timer 11 input 0:For inputting the capture trigger of a 16-bit timer 11
PD3 TB11IN1	1	Input/output Input	Port D3:Input/output port 16-bit timer 11 input 1:For inputting the capture trigger of a 16-bit timer 11
PD4 TB12IN0	1	Input/output Input	Port D4:Input/output port 16-bit timer 12 input 0:For inputting the capture trigger of a 16-bit timer 12 /Two-phase counter input pin
PD5 TB12IN1	1	Input/output Input	Port D5:Input/output port 16-bit timer 12 input 1:For inputting the capture trigger of a 16-bit timer 12 /Two-phase counter input pin
PD6 TB14OUT	1	Input/output Output	Port D6:Input/output port 16-bit timer 14 output :16bit timer 14 variable PPG output
PD7 TB15OUT	1	Input/output Output	Port D7:Input/output port 16-bit timer 15 output :16bit timer 15 variable PPG output
PE0 TB16OUT	1	Input/output Output	Port E0:Input/output port 16-bit timer 16 output :16bit timer 16 variable PPG output
PE1 TB17OUT	1	Input/output Output	Port E1:Input/output port 16-bit timer 17 output :16bit timer 17 variable PPG output
PE2 TB18OUT	1	Input/output Output	Port E2:Input/output port 16-bit timer 18 output :16bit timer 18 variable PPG output
PE3 TB19OUT	1	Input/output Output	Port E3:Input/output port 16-bit timer 19 output :16bit timer 19 variable PPG output
PE4 TB1AOUT	1	Input/output Output	Port E4:Input/output port 16-bit timer 1A output :16bit timer 1A variable PPG output
PE5 SO0 SDA0	1	Input/output Output Input/output	Port E5:Input/output port Pin for sending data if the serial bus interface operates in the SIO mode Pin for sending and receiving data if the serial bus interface operates in the I2C mode**(Input with Schmitt trigger) Open drain output pin**
PE6 SI0 SCL0	1	Input/output Input Input/output	Port E6:Input/output port Pin for receiving data if the serial bus interface operates in the SIO mode Pin for inputting and outputting a clock if the serial bus interface operates in the I2C mode**(Input with Schmitt trigger)** **Open drain output pin**
PE7 SCK0	1	Input/output Input/output	Port E7:Input/output port Pin for inputting and outputting a clock if the serial bus interface operates in the I2C mode
PF0 TXD0	1	Input/output Input	Port F0:Input/output port Sending serial data 0: Open drain output pin depending on the program used
PF1 RXD0	1	Input/output Input	Port F1 Input/output port Receiving serial data 0
PF2 *SCLK0 CTS0	1	Input/output Input Input	Port F2:Input/output port Serial clock input/output 0 : Open drain output pin depending on the program used Handshake input pin
PF3	1	Input/output	Port F3:Input/output port
PF4 TXD1	1	Input/output Input	Port F0:Input/output port Sending serial data 1: Open drain output pin depending on the program used
PF5 RXD1	1	Input/output Input	Port F2 Input/output port Receiving serial data 1
PF6 *SCLK1 CTS1	1	Input/output Input Input	Port F3:Input/output port Serial clock input/output 1 : Open drain output pin depending on the program used Handshake input pin
PF7	1	Input/output	Port F7:Input/output port

Table 2.3 Pin Names and Functions (4/8)

Pin name	No. of pins	Input or output	Function
PG0 TXD2	1	Input/output Output	Port G0:Input/output port Sending serial data 2: Open drain output pin depending on the program used
PG1 RXD2	1	Input/output Input	Port G1 Input/output port Receiving serial data 2
PG2 *SCLK2 CTS2	1	Input/output Input/output Input	Port G2:Input/output port serial clock input/output 2 : Open drain output pin depending on the program used Handshake input pin
PG3 TBTIN1	1	Input/output Input	Port G3:Input/output port 32-bit time base timer input 1:For inputting a 32-bit time base timer
PG4 TXD3	1	Input/output Input	Port G4:Input/output port Sending serial data 3: Open drain output pin depending on the program used
PG5 RXD3	1	Input/output Input	Port G5 Input/output port Receiving serial data 3
PG6 *SCLK3 CTS3	1	Input/output Input/output Input	Port G6:Input/output port Serial clock input/output 3: Open drain output pin depending on the program used Handshake input pin
PG7 TBTIN2	1	Input/output Input	Port G7:Input/output port 32-bit time base timer input 2:For inputting a 32-bit time base timer
PH0 TXD4	1	Input/output Output	Port H0:Input/output port Sending serial data 4: Open drain output pin depending on the program used
PH1 RXD4	1	Input/output Input	Port H1 Input/output port Receiving serial data 4
PH2 *SCLK4 CTS4	1	Input/output Input/output Input	Port H2:Input/output port Serial clock input/output 4 : Open drain output pin depending on the program used Handshake input pin
PH3 INT9	1	Input/output Input	Port H3 Input/output port Interrupt request pin 9: Selectable between “H” level, “L” level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PH4 TXD4	1	Input/output Output	Port H4:Input/output port Sending serial data 5: Open drain output pin depending on the program used
PH5 RXD5	1	Input/output Input	Port H5 Input/output port Receiving serial data 5
PH6 *SCLK5 CTS5	1	Input/output Input/output Input	Port H6:Input/output port Serial clock input/output 5 : Open drain output pin depending on the program used Handshake input pin
PH7 INTA	1	Input/output Input	Port H7 Input/output port Interrupt request pin A: Selectable between “H” level, “L” level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PI0 TXD6	1	Input/output Output	Port I0:Input/output port Sending serial data 6: Open drain output pin depending on the program used
PI1 RXD6	1	Input/output Input	Port I1 Input/output port Receiving serial data 6
PI2 *SCLK6 CTS6	1	Input/output Input/output Input	Port I2:Input/output port Serial clock input/output 6 : Open drain output pin depending on the program used Handshake input pin
PI3 INTB	1	Input/output Input	Port I3 Input/output port Interrupt request pin B: Selectable between “H” level, “L” level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PI4 TXD7	1	Input/output Output	Port I4:Input/output port Sending serial data 7: Open drain output pin depending on the program used
PI5 RXD7	1	Input/output Input	Port I5 Input/output port Receiving serial data 7
PI6 *SCLK7 CTS7	1	Input/output Input/output Input	Port I6:Input/output port Serial clock input/output 7 : Open drain output pin depending on the program used Handshake input pin

Table 2.3 Pin Names and Functions (5/8)

Pin name	No. of pins	Input or output	Function
PI7	1	Input/output	Port I7:Input/output port
PJ0 TXD8	1	Input/output Output	Port J0:Input/output port Sending serial data 8: Open drain output pin depending on the program used
PJ1 RXD8	1	Input/output Input	Port J1 Input/output port Receiving serial data 8
PJ2 *SCLK8 CTS8	1	Input/output Input/output Input	Port J2:Input/output port serial clock input/output 8 : Open drain output pin depending on the program used Handshake input pin
PJ3 TC0IN	1	Input/output Input	Port J3 Input/output port For inputting the capture trigger for 32-bit timer
PJ4 TC1IN	1	Input/output Input	Port J4 Input/output port For inputting the capture trigger for 32-bit timer
PJ5 SO1 SDA1	1	Input/output Output Input/output	Port J5:Input/output port Pin for sending data if the serial bus interface operates in the SIO mode Pin for sending and receiving data if the serial bus interface operates in the I2C mode**(Input with Schmitt trigger)** **Open drain output pin**
PJ6 SI1 SCL1	1	Input/output Input Input/output	Port J6:Input/output port Pin for receiving data if the serial bus interface operates in the SIO mode Pin for inputting and outputting a clock if the serial bus interface operates in the I2C mode**(Input with Schmitt trigger)** **Open drain output pin**
PJ7 SCK1	1	Input/output Input/output	Port J7:Input/output port Pin for inputting and outputting a clock if the serial bus interface operates in the SIO mode
PK0 KEY0	1	Input/output Input	Port K0:Input/output port Key On Wake UP input 0 :(Input with Schmitt trigger with pull-up and Noise filter)
PK1 KYE1	1	Input/output Input	Port K1:Input/output port Key On Wake UP input 1:(Input with Schmitt trigger with pull-up and Noise filter)
PK2 KEY2	1	Input/output Input	Port K2:Input/output port Key On Wake UP input 2:(Input with Schmitt trigger with pull-up and Noise filter)
PK3 KEY3	1	Input/output Input	Port K3:Input/output port Key On Wake UP input 3:(Input with Schmitt trigger with pull-up and Noise filter)
PK4 KEY4	1	Input/output Input	Port K4:Input/output port Key On Wake UP input 4:(Input with Schmitt trigger with pull-up and Noise filter)
PK5 KEY5	1	Input/output Input	Port K5:Input/output port Key On Wake UP input 5 :(Input with Schmitt trigger with pull-up and Noise filter)
PK6 KEY6	1	Input/output Input	Port K6:Input/output port Key On Wake UP input 6 :(Input with Schmitt trigger with pull-up and Noise filter)
PK7 KEY7	1	Input/output Input	Port K7:Input/output port Key On Wake UP input 7 :(Input with Schmitt trigger with pull-up and Noise filter)
PL0 TC4IN	1	Input/output Input	Port L0:Input/output port For inputting the capture trigger for 32-bit timer
PL1 TC5IN	1	Input/output Input	Port L1:Input/output port For inputting the capture trigger for 32-bit timer
PL2	1	Input/output	Port L2:Input/output port
PL3 TCOUTB0	1	Input/output Output	Port L3:Input/output port Outputting 32-bit timer if the result of a comparison is a match
PL4 TXD9	1	Input/output Output	Port L4:Input/output port Sending serial data 9: Open drain output pin depending on the program used
PL5 RXD9	1	Input/output Input	Port L5 Input/output port Receiving serial data 9
PL6 *SCLK9 CTS9	1	Input/output Input/output Input	Port L6:Input/output port serial clock input/output 9 : Open drain output pin depending on the program used Handshake input pin
PL7 TCOUTB1	1	Input/output Output	Port L7:Input/output port Outputting 32-bit timer if the result of a comparison is a match

Table 2.3 Pin Names and Functions (6/8)

Pin name	No. of pins	Input or output	Function
PM0 INT0	1	Input/output Input	Port M0:Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PM1 INT1	1	Input/output Input	Port M0:Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PM2 INT2	1	Input/output Input	Port M0:Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PM3 INT3	1	Input/output Input	Port M0:Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PM4 INT4	1	Input/output Input	Port M0:Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PM5 INT5	1	Input/output Input	Port M0:Input/output port Interrupt request pin 0: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PM6 TCOUTA0	1	Input/output Output	Port M6:Input/output port Outputting 32-bit timer if the result of a comparison is a match
PM7 TCOUTA1	1	Input/output Output	Port M7:Input/output port Outputting 32-bit timer if the result of a comparison is a match
PN0 INT6	1	Input/output Input	Port N0:Input/output port Interrupt request pin 6: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PN1 INT7	1	Input/output Input	Port N1:Input/output port Interrupt request pin 7: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PN2 INT8	1	Input/output Input	Port N2:Input/output port Interrupt request pin 8: Selectable between "H" level, "L" level, rising edge and falling edge **(Input with Schmitt trigger with Noise filter)**
PN3 ADTRG-A	1	Input/output Input	Port N3:Input/output port Pin for starting A/D trigger or A/D converter from an external source
PN4 TXDA	1	Input/output Output	Port N4:Input/output port Sending serial data A: Open drain output pin depending on the program used
PN5 RXDA	1	Input/output Input	Port N5 Input/output port Receiving serial data A
PN6 *SCLK CTSA	1	Input/output Input/output Input	Port N6:Input/output port serial clock input/output A : Open drain output pin depending on the program used Handshake input pin
PN7 ADTRG-B	1	Input/output Input	Port N7:Input/output port Pin for starting A/D trigger or A/D converter from an external source
PO0 TPD0	1	Input/output Output	Port D0:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PO1 TPD1	1	Input/output Output	Port D1:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PO2 TPD2	1	Input/output Output	Port D2:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PO3 TPD3	1	Input/output Output	Port D3:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PO4 TPD4	1	Input/output Output	Port D4:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE

Table 2.3 Pin Names and Functions (7/8)

Pin name	No. of pins	Input or output	Function
PO5 TPD5	1	Input/output Output	Port D5:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PO6 TPD6	1	Input/output Output	Port D5:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PO7 TPD7	1	Input/output Output	Port D5:Input/output port Outputting trace data from the data access address: Signal for DSU-ICE
PP0 TPC0 TPD0	1	Input/output Output Output	Port P0:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP1 TPC1 TPD1	1	Input/output Output Output	Port P1:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP2 TPC2 TPD2	1	Input/output Output Output	Port P2:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP03 TPC3 TPD3	1	Input/output Output Output	Port P3:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP4 TPC4 TPD4	1	Input/output Output Output	Port P4:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP5 TPC5 TPD5	1	Input/output Output Output	Port P5:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP6 TPC6 TPD6	1	Input/output Output Output	Port P6:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PP7 TPC7 TPD7	1	Input/output Output Output	Port P7:Input/output port Outputting trace data from the program counter: Signal for DSU-ICE Outputting trace data from the data access address: Signal for DSU-ICE
PQ0 DREQ2	1	Input/output Input	Port Q0:Input/output port DMA request signal 2: For inputting the request to transfer data by DMA from an external I/O device to DMA2
PQ1 DACK2	1	Input/output Output	Port Q0:Input/output port DMA acknowledge signal 2: Signal showing that DREQ2 have acknowledged a DMA transfer request
PQ2 DREQ3	1	Input/output Input	Port Q2:Input/output port DMA request signal 3: For inputting the request to transfer data by DMA from an external I/O device to DMA3
PQ4 DACK3	1	Input/output Output	Port Q3:Input/output port DMA acknowledge signal 3: Signal showing that DREQ3 have acknowledged a DMA transfer request
DCLK	1	Output	Debug clock: Signal for DSU-ICE
*EJE	1	Input	EJTAG enable: Signal for DSU-ICE**(fixed to pull up)** **(Input with Schmitt trigger with Noise filter)**
*DINT	1	Input	Debug interrupt: Signal for DSU-ICE**(fixed to pull up)** **(Input with Schmitt trigger with Noise filter)**
PCST0	1	Output	PC trace status: Signal for DSU-ICE
PCST1	1	Output	PC trace status: Signal for DSU-ICE
PCST2	1	Output	PC trace status: Signal for DSU-ICE
PCST3	1	Output	PC trace status: Signal for DSU-ICE
PCST4	1	Output	PC trace status: Signal for DSU-ICE
*DINT	1	入力	Debug interrupt: Signal for DSU-ICE
TOVR	1	Output	Outputting the status of PD data overflow status: Signal for DSU-ICE
TCK	1	Input	Test clock input: Signal for DSU-ICE**(fixed to pull up)** **(Input with Schmitt trigger with Noise filter)**
TMS	1	Input	Test mode select input: Signal for DSU-ICE**(fixed to pull up)** **(Input with Schmitt trigger)**

Table 2.3 Pin Names and Functions (8/8)

Pin name	No. of pins	Input or output	Function
TDI	1	Input	Test data input: Signal for DSU-ICE**(fixed to pull up)** **(Input with Schmitt trigger)**
TDO	1	Output	Test data output: Signal for DSU-ICE
*TRST	1	Input	Test reset input: Signal for DSU-ICE**(fixed to pull down)** **(Input with Schmitt trigger with Noise filter)**
*RESET	1	Input	Reset:Intializing LSI **(fixed to pull down)** **(Input with Schmitt trigger with Noise filter)**
X1/X2	2	Input/output	Pin for connecting a high-speed oscillator **(X1:Input with Schmitt trigger)**
*NMI	1	Input	Non-maskable interrupt request pin :**(with Schmitt trigger)**
BUSMD	1	Input	Pin for setting an external bus mode: This pin functions as a multiplexed bus by sampling the "H (DVCC15) level" 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.
ENDIAN	1	Input	Pin for setting endian: This pin is used to set a mode. It performs a big-endian operation by sampling the "H (DVCC15) level" 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. Input with Schmitt trigger.
PLLSEL	1	Input	Pin for setting PLL operation with MASK **(Input with Schmitt trigger)** High(DVCC15) :11~13.5MHz(=X1) Low:8~11MHz(=X1) When performing a reset operation, pull it up or down according to the type of oscillator to be used.
BW0	1	Input	TEST pin: To be fixed to DVCC15:**(Input with Schmitt trigger)**
BW1	1	Input	TEST pin: To be fixed to DVCC15:**(Input with Schmitt trigger)**
TEST1	1	Input	TEST pin: Set to OPEN
TEST2	1	Input	TEST pin: Set to OPEN
TEST3	1	Input	TEST pin: Set to OPEN
AVREFH0	1	Input	Reference power supply pin for the A/D converter (H) If the A/D converter is not used, connect (fix) this pin to AVCC3x.
AVREFH1	1	Input	Reference power supply pin for the D/A converter (H) If the A/D converter is not used, connect (fix) this pin to AVCC3x.
AVSS0	1	-	GND pin (0 V) for the D/A converter (0V)。Connect this pin to GND even if the D/A converter is not used.
AVCC30	1	-	Power supply pin for the A/D converter. Connect this pin to power supply even if the A/D converter is not used.
AVCC31	1		Power supply pin for the A/D converter. Connect this pin to power supply even if the A/D converter is not used.
AVSS1	1	-	GND pin (0 V) for the D/A converter (0V)。Connect this pin to GND even if the D/A converter is not used.
CVCC15	1	-	Power supply pin for a high-frequency oscillator: 1.5 V power supply
CVSS	1	-	GND pin (0V) for a high-frequency oscillator
DVCC15	4	-	Power supply pin: 1.5 V power supply
DVCC30	4	-	Power supply pin: 3 V power supply
DVCC31	2	-	Power supply pin: 3 V power supply
DVCC32	1	-	Power supply pin: 3 V power supply
DVCC33	1	-	Power supply pin: 3 V power supply
DVCC34	3	-	Power supply pin: 3 V power supply
DVSS	9	-	Power supply pin: GND pin (0V)
FVCC3	2		Power supply pin: 3 V power supply(for FLASH)
FVCC15	2		Power supply pin: 1.5 V power supply(for FLASH)

2.4 Pin Names and Power Supply Pins

Table 2.4 Pin Names and Power Supplies

Power supply	Pin number	Voltage range	Power supply
P0	DVCC30	PL	DVCC33
P1	DVCC30	PM	DVCC32
P2	DVCC30	PN	DVCC31
P3	DVCC30	PO	DVCC34
P4	DVCC30	PP	DVCC34
P5	DVCC30	PQ	DVCC34
P6	DVCC30	*NMI	DVCC15
P7	AVCC30	PCST4~0	DVCC34
P8	AVCC30	DCLK	DVCC34
P9	AVCC31	*EJE	DVCC34
PA	AVCC31	*TRST	DVCC34
PB	DVCC31	TDI	DVCC34
PC	DVCC31	TDO	DVCC34
PD	DVCC31	TMS	DVCC34
PE	DVCC31	TCK	DVCC34
PF	DVCC32	*DINT	DVCC34
PG	DVCC32	*RESET	DVCC15
PH	DVCC31	PLLSEL	DVCC15
PI	DVCC31	X1、X2	CVCC15
PJ	DVCC32	BUSMD	DVCC15
PK	DVCC33	BW0,BW1	DVCC15

2.5 Pin Numbers and Power Supply Pins

Table 2.5 Pin Numbers and Power Supplies

Power supply	Pin number	Voltage range
DVCC15	M8,M14,N11,H12	1.35V~1.65V
DVCC3	G9,G10,G11,G12, G13,J8,K8,L8,P11, P12,	2.7V~3.3V
AVCC	J13,M13	2.7V~3.3V
CVCC15	N14	1.35V~1.65V
FVCC15	H11,N10	1.35V~1.65V
FVCC3	H9,H10	2.7V~3.3V

3. Processor Core

The TMP19A63 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 TMP19A63 that are not explained in that document.

3.1 Reset Operation

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

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

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

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

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

(Note 3) 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 4) In the FLASH program, the reset period of 0.5 μs or longer is required independently of the system clock.

(Note 5) Please be sure to turn the 1.5V power supply for core on first and then turn the 3V power supply for I/O on.

4. Memory Map

Fig. 4.1 & Fig 4.2 shows the memory map of the TMP19A63.

1) For 1024KB ROM/ 48KB Type (TMP19A63F10XBG)

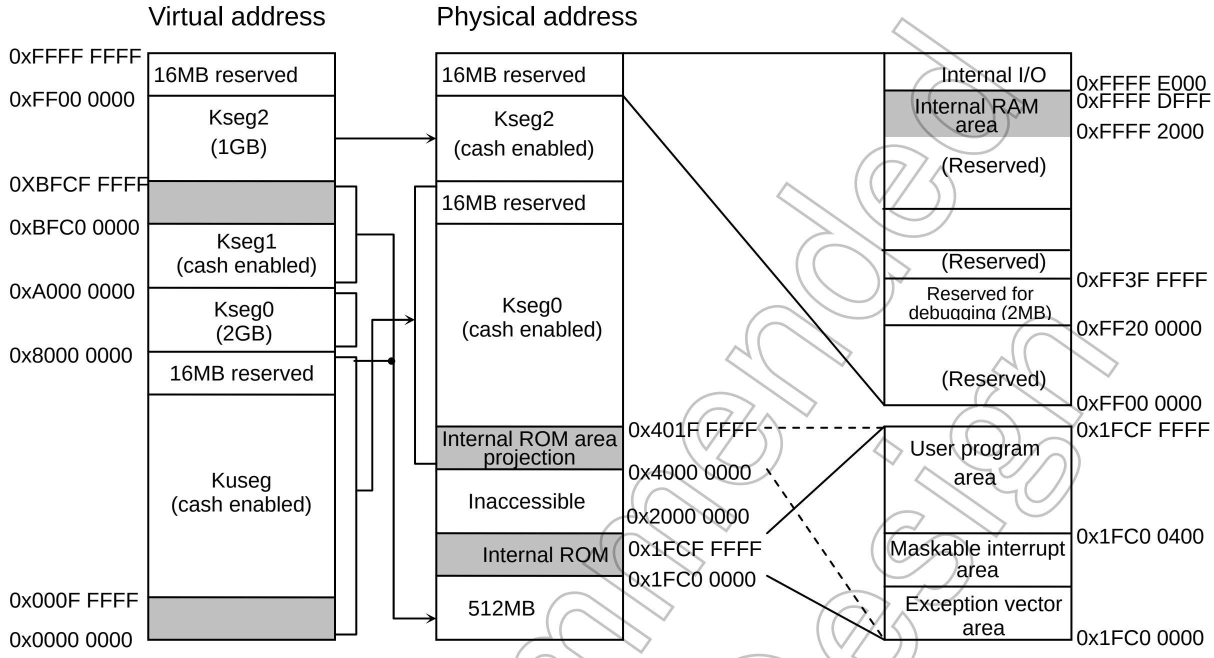


Fig. 4.1 Memory Map

2) For 512KB ROM/ 24KB Type (TMP19A63CDXBG)

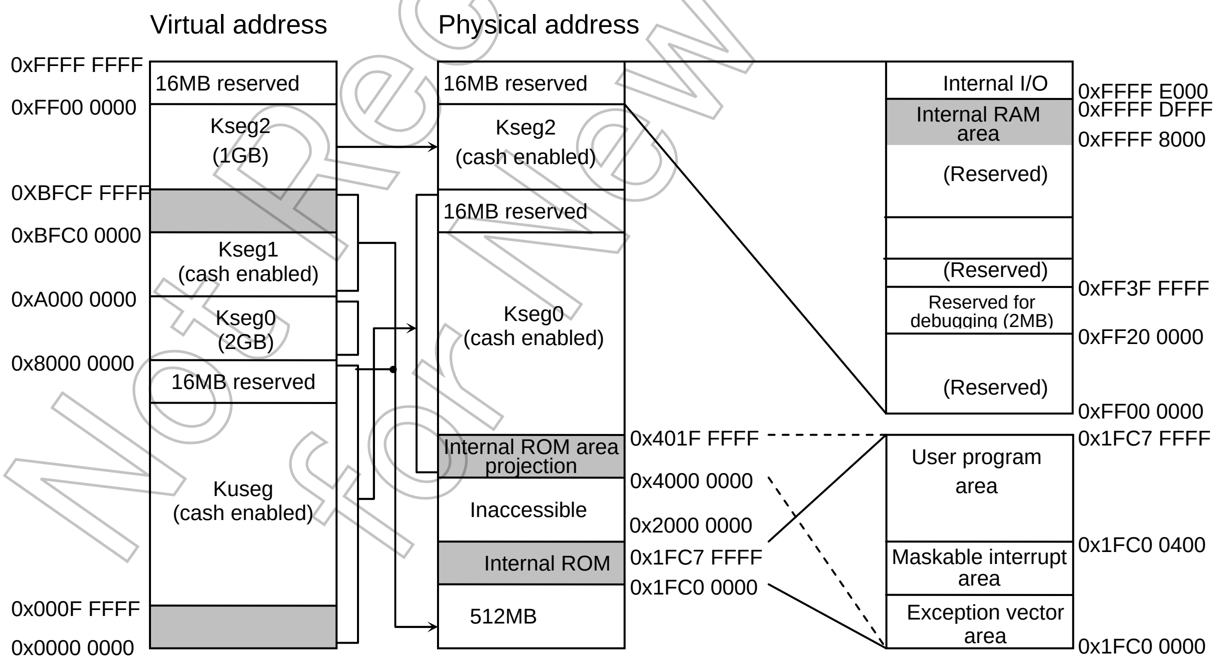


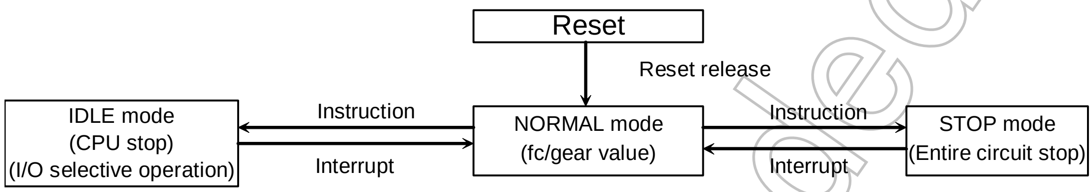
Fig. 4.2 Memory Map

(Note 1)	**The internal ROM is mapped to:** **0x1FC0_0000~0x1FCF_FFFF (1024KB)** **0x1FC0_0000~0x1FC7_FFFF (512KB)** **The internal RAM is mapped to:** **0xFFFF_2000~0xFFFF_DFFF (48KB)** **0xFFFF_8000~0xFFFF_DFFF (24KB)**
(Note 2)	**For the TMP19A63, 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. However, it is not possible to set internal memory, internal I/O space and reserved areas.**
(Note 3)	**Do not place an instruction in the last four words of a physical area.** **Internal ROM: 0x1FCF_FFF0 ~ 0x1FCF_FFFF (1024KB)** **Internal ROM: 0x1FC7_FFF0 ~ 0x1FC7_FFFF (512KB)** **The last four words of an area where memory is mounted for external ROM extension (this varies depending on the system of the user).**

5. Clock/Standby Control

5.1 Operation Mode

The system operation modes contain the standby modes in which the processor core operations are stopped to reduce power consumption. Fig. 5.1 State Transition Diagram of Each Operation Mode is shown below



Clock mode without power supply to backup module

Fig. 5.1 State Transition Diagram of Each Operation Mode

5.2 Default Setting for System Clock

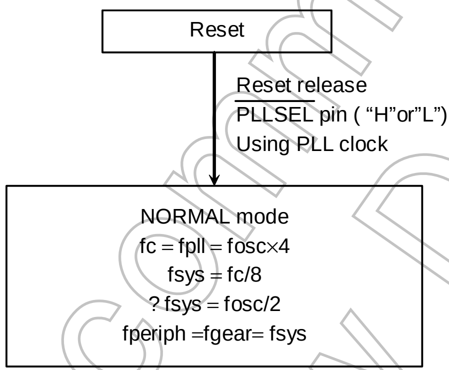


Fig. 5.2 Default Setting for System Clock

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 (clock frequency multiplied (quadrupled) by the PLL)

fgear : Clock frequency selected by the system control register SYSCR1<GEAR2:0> in the clock generator

fsys : System clock frequency
The CPU, ROM, RAM, DMAC and INTC all operate according to this clock.
The internal peripheral I/O operates according to the fsys/2 clock.

fperiph : Clock frequency selected by SYSCR1<FPSEL> (Clock to be input to the peripheral I/O prescaler)

PLLSEL pin:to select frequency that adjusts to PLL depending on the clock frequency connected to X1,X2 pins

	X1	PLL output	Fc
PLLSEL 1:	11-13.5MHz =>	44-54MHz =====>	44-54MHz
0:	8-11 MHz =>	64-88MHz =1/2=>	32-44MHz

5.3 Clock System Block Diagram

5.3.1 Main System Clock

- Allows for oscillator connection or external clock input.
- Sets $\overline{\text{PLLON}}$ (quadrupled) at reset
- Selects PLL setting that corresponds to X1 input frequency by PLLSEL pin
- Clock gear: 1,1/2, 1/4 ,1/8 (Default is 1/8)
- Input frequency (high frequency)

	Input frequency range	Maximum operating frequency	Lowest operating frequency
PLLSEL="H" or "L" (both oscillator and external input)	8~13.5 (MHz)	54 MHz	4 MHz

5.3.2 Clock Gear

- Divides high speed clock into 1/1, 1/2, 1/4 and 1/8.
- The internal I/O prescaler clock ϕT0: fperiph/2, fperiph/4, fperiph/8 and fperiph/16

Fig. 5.3.2 shows a system clock transition diagram.

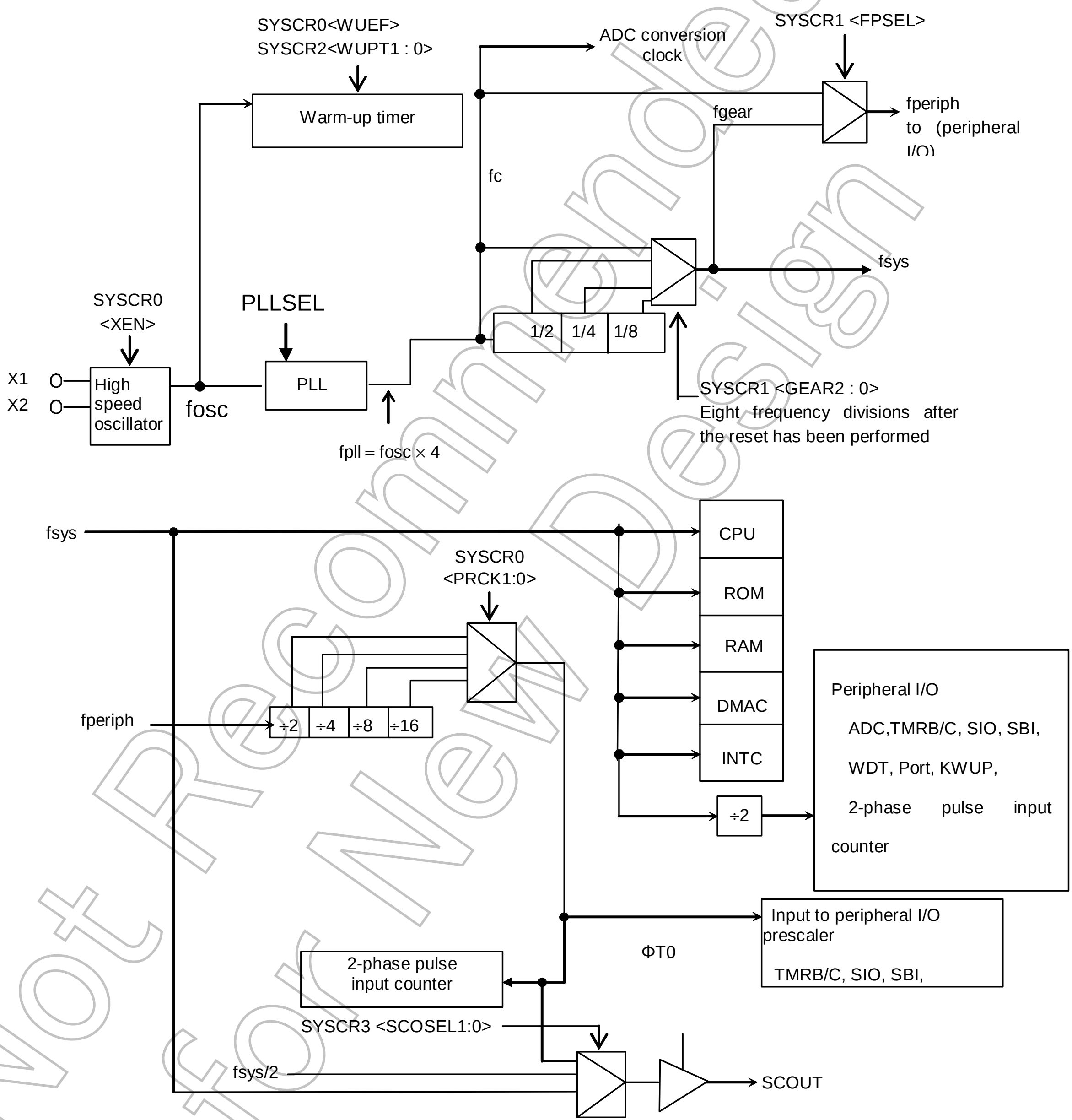


Fig. 5.3.2 System Clock Transition Diagram

5.4 CG Registers

5.4.1 System Control Registers

		7	6	5	4	3	2	1	0
SYSCR0	bit Symbol	XEN		RXEN			WUEF	PRCK1	PRCK0
(0xFFFF_EE00)	Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
	After reset	1	1	1	1	0	0	0	0
	Function	High-speed oscillator 0: Stop 1: Oscilla-tion	Write “1”.	Write “1”.	Write “1”.	This can be read as “0.”	Write “0”.	Select prescaler clock 00:fperiph/16 01:fperiph/8 10:fperiph/4 11:fperiph/2	

		15	14	13	12	11	10	9	8
SYSCR1	Bit symbol		SYSCKFLG	SYSCK	FPSEL	SGEAR	GEAR2	GEAR1	GEAR0
(0xFFFF_EE01)	Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	1	1	1
	Function	This can be read as “0.”	This can be read as “0.”	Write “0”.	Select fperiph 0:fgear 1:fc	Write “0”.	Select gear of high-speed clock (fc) 000: fc 100: fc1/2 001: reserved 101: reserved 010: reserved 110: fc1/4 011: reserved 111: fc1/8		

		23	22	21	20	19	18	17	16
SYSCR2	Bit symbol	DRVOSCH		WUPT1	WUPT0	STBY1	STBY0		DRVE
(0xFFFF_EE02)	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
	After reset	0	0	1	0	1	1	0	0
	Function	High-speed oscillator current control 0: High capability 1: Low capability	Write “0”.	Select oscillator warm-up time 00:no WUP 01:2^8/ oscillating frequency 10:2^{14}/ oscillating frequency 11:2^{16}/ oscillating frequency		Select stand-by mode 00:reserved 01:STOP 10:reserved 11:IDLE		This can be read as “0.”	1: Drive the pin even at the STOP mode.

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

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

(Note) Restriction to use clock gear

To activate peripheral I/O, use fc, fc1/2, fc1/4 or fc1/8 for SYSCR1<GEAR2:0>. Otherwise, it cannot operate properly.

5.5 System Clock Controller

By resetting the system clock controller, the controller status switches to single clock mode and is initialized to <XEN>="1 and <GEAR2:0>="111" and the system clock fsys changes to fc/8. (fc=fosc (original oscillation frequency)×4, because the original oscillation is quadrupled by PLL.) For example, when a 13-MHz oscillator is connected to the X1 or X2 pin, fsys becomes 6.25 MHz (=13.5×4×1/8) after the reset.

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

(Note)Set the system clock frequency to be 4MHz or more as the default.

5.5.1 Oscillation Stabilization Time (Switching between the NORMAL and STOP 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.

Table 5.5.1 shows warm-up time at switching.

(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 5.5.1 Warm-up Time

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

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

<Example 1> Transition from STOP mode to NORMAL mode
SYSCR2<WUPT1:0>="xx": Select the warm-up time
SYSCR0<XEN>="1" :Enable the high speed oscillation (fosc)

SYSCR1<SYSCK>="0" :Switch the system clock to high speed (fgear)
SYSCR1<SYSCKFLG>Read :"0"(confirm the current system is fgear)

5.5.2 System Clock Pin Output Function

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

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

Table 5.5.2 SCOUT Output State in Each Standby Mode

<table>
<tr><th rowspan="2">Mode
SCOUT selection</th><th rowspan="2" colspan="2">NORMAL</th><th colspan="2">Stand-by mode</th></tr>
<tr><th>IDLE</th><th>STOP</th></tr>
<tr><td><SCOSEL1:0> ="00"</td><td colspan="4">Reserved. No other setting is allowed.</td></tr>
<tr><td><SCOSEL1:0> ="01"</td><td colspan="3">Output the fperiph clock.</td><td rowspan="2">Fixed to "0" or "1".</td></tr>
<tr><td><SCOSEL1:0> ="10"</td><td colspan="3">Output the fsys clock.</td></tr>
<tr><td><SCOSEL1:0> ="11"</td><td>Output the ΦT0 clock.</td><td>Fix to "0".</td><td>Output the ΦT0 clock.</td><td>Fixed to "0".</td></tr>
</table>

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

5.5.3 Reducing the Oscillator Driving Capability

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

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

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

- Reducing the driving capability of the high-speed oscillator

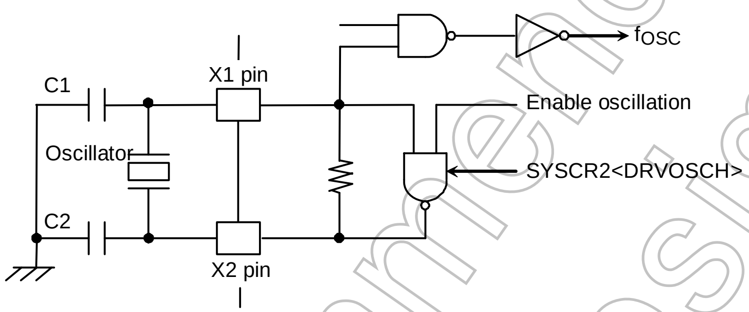


Fig. 5.5.3 Oscillator Driving Capability

5.6 Prescaler Clock Controller

Each internal I/O (TMRB0 to 23, TMRCA to B, SIO0 to A and SBI0 to 1) has a prescaler for dividing a clock. The clock ϕT0 to be input to each prescaler is obtained by selecting the "fperiph" clock, which is divided according to the setting of SYSCR0<PRCK1:0>, from the SYSCR1<FPSEL>. After the controller is reset, fperiph/16 is selected as ϕT0. For details, please refer to Fig. 5.3 System Clock Transition Diagram.

5.7 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.

5.8 Standby Controller

The TX19A core has several low-consumption modes. To shift to the STOP 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 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 at the IDLE mode in the register of each module. This enables operation settings at 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-5.8 shows a list of IDLE setting registers.

Table 5.8 Internal I/O setting registers for the IDLE mode

Internal I/O	IDLE mode setting register
TMRB0~23	Textron<I2TBx>
TBTA~B	TBTxRUN<I2TBT>
SIO0~A	SCxMOD1<I2Sx>
SBI	SBIxBR0<I2SBI>
A/DC A~B	ADxMOD1<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.

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

5.8.1 CG Operations in Each Mode

Table 5.8.1 Status of CG in Each Operation Mode

Clock source	Mode	Oscillation circuit	PLL	Clock supply to peripheral I/O	Clock supply to CPU
Oscillator	Normal	○	○	○	○
	Idle (Halt)	○	○	Selectable	×
	Idle (Doze)	○	○	Selectable	×
	Stop	×	×	×	×

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

5.8.2 Block Operations in Each Mode

Table 5.8.2 Block Operating Status in Each Operation Mode

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

○: ON ×: OFF

5.8.3 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 5.8.3 Standby Release Sources and Standby Release Operations.

- Release by an interrupt request

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

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

- Release by the reset

Any standby state can be released by the reset. It initializes the setting (the precedent status of the stand-by is maintained in the case of release by the interrupt).
Note that releasing of the STOP mode requires sufficient reset time to allow the oscillator operation to become stable (it must be longer than “the time required for stable oscillation + 500μs”).

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

Table 5.8.3 Standby Release Sources and Standby Release Operations

(Interrupt level)>(Interrupt mask)

Interrupt accepting state			Interrupt enable EI="1"		Interrupt disable EI= "0"	
Stand-by mode			IDLE (programmable)	STOP	IDLE (programmable)	STOP
Stand-by release source	Interrupt	INTNMI	◎	◎	◎	◎
		INTWDT	◎	×	◎	-
		INT0~B	◎	◎	○	○
		KWUP00~7	◎	◎	○	○
		INTTB0~23	◎	×	○	×
		INTTBT/CAPG/CMPG	◎	×	○	×
		INTRX0~A,INTTX0~A	◎	×	○	×
		INTADA/INTADHPA/	◎	×	○	×
		INTADM	◎	×	○	×
		INTADB/INTADHPB	◎	×	○	×
		INTDMAx	◎	×	○	×
	RESET		◎	◎	◎	◎

◎: Starts the interrupt handling after the standby mode is released. (The LSI is initialized by the reset.)
○: Starts the processing at the address next to the standby instruction (without executing the interrupt handling) after the standby mode is released.
×: Cannot be used for releasing the standby mode.
-: Cannot execute masking with an interruption mask when a non-maskable interrupt is selected.

5.8.4 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 5.8.4 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), 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>.

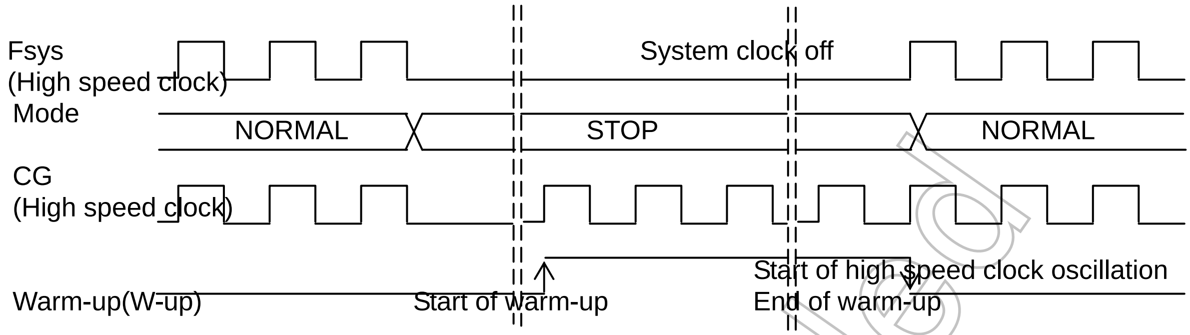
(Note)To shift from the NORMAL mode to the STOP mode on the TMP19A63, do not set the SYSCR2<WUPT1:0> to "00" or "01" for the warm-up time setting. The internal system recovery time cannot be satisfied when the system recovers from the STOP mode.

Table 5.8.4 Warm-up Settings for Transitions of Operation Modes

Transition of operating mode	Warm-up setting
NORMAL→IDLE	Not required
NORMAL→STOP	Not required
IDLE→NORMAL	Not required
STOP→NORMAL	Required

5.8.5 Recovery from the STOP Mode

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



@fosc=13.5MHz

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

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

Table 5.8.5 Pin States in the STOP Mode in Each State of SYSCR2<DRVE> (1/3)

Pin name	Input/Output	<DRVE>=0	<DRVE>=1
P00~P07	Input mode Output mode AD0~AD7, D0~D7	- - -	- Output -
P10~P17	Input mode Output mode, A8~A15 AD8~AD15, D8~D15	- - -	- Output -
P20~P27	Input mode Output mode, A0~A7/A16~A23	- -	Input Output
P30 (*RD), P31 (*WR)	Output pin	-	Output
P32,P35,P36	Input mode Output mode,*HWR,*BUSAK,R/W_	PU* PU*	Input Output
P33	Input mode,*WAIT,*RDY Output mode	PU* PU*	Input Output
P34	Input mode Output mode *BUSRQ	PU* PU* PU*	Input Output Input
P37 (ALE)	Input mode Output mode ALE(Output mode)	- - -	Input Output “-
P40~P45	Input mode Output mode,CS0~CS5	PU* PU*	Input Output
P46 (SCOUT)	Input mode Output mode	- -	Input Output
P47	Input mode Output mode	- -	Input Output
P50~P57	Input mode Output mode, A0~A7	- -	Input Output
P60~P67	Input mode Output mode, A8~A15	- -	Input Output
P7, P8, P9,PA	Input pin,ANx0~ANx15	Input	Input
PB0~PB7	Input mode Output mode TB8IN0~TBBIN1(Input mode)	- - -	Input Output Input
PC0~PC7	Input mode Output mode TBCIN0~TBFIN1(Input mode)	- - -	Input Output Input
PD0,PD1,PD2,PD3, PD4,PD5	Input mode Output mode TB10IN0~TB12IN1(Input mode)	- - -	Input Output Input
PD6,PD7	Input mode Output mode,TB14OUT,TB15OUT	- -	Input Output
PE0,PE1,PE2,PE3, PE4	Input mode, Output mode,TB16OUT,TB17OUT, TB18OUT,TB19OUT,TB1AOUT	- -	Input Output
PE5,PE6,PE7	Input mode,SI0/SCL0/SCK0 Output mode,SO0/SCA0/SCK0	- -	Input Output
PF0~PF2 PF4~PF6	Input mode,SCLK0,RXD0,*CTS0 SCLK1,RXD1,*CTS1 Output mode,SCLK0,TXD0 SCLK1,TXD1	- - 	Input Output
PF3 ,PF7	Input mode Output mode	- -	Input Output
PG0~PG2 PG4~PG6	Input mode,SCLK2,RXD2,*CTS2 SCLK3,RXD3,*CTS3 Output mode,SCLK2,TXD2 SCLK3,TXD3	- - 	Input Output

Table 5.8.5 Pin States in the STOP Mode in Each State of SYSCR2<DRVE> (2/3)

Pin name	Input/Output	<DRVE>=0	<DRVE>=1
PG3,PG7	Input mode,TBTIN1,TBTIN2 Output mode	- -	Input Output
PH0~PH2 PH4~PG6	Input mode,SCLK4,RXD4,*CTS4 SCLK5,RXD5,*CTS5 Output mode,SCLK4,TXD4,SCLK5,TXD5	- -	Input Output
PH3,PH7,PI3	Input mode Output mode INT9,INTA,INTB(Input mode)	- - Input	Input Output Input
PI0~PI2 PI4~PI6	Input mode,SCLK6,RXD6,*CTS6 SCLK7,RXD7,*CTS7 Output mode,SCLK6,TXD6,SCLK7,TXD7	- -	Input Output
PI7	Input mode Output mode	- -	Input Output
PJ0~PJ2	Input mode,SCLK8,RXD8,*CTS8 Output mode,SCLK8,TXD8	- -	Input Output
PJ3,PJ4	Input mode Output mode TC0IN,TC1IN(Input mode)	- - -	Input Output Input
PJ5~PJ7	Input mode,SI1/SCL1/SCK1 Output mode,SO1/SCA1/SCK1	- -	Input Output
PK0~PK7	Input mode Output mode KEY0~KEY7(Input mode)	- - Input	Input Output Input
PL0,PL1	Input mode Output mode TC4IN,TC5IN(Input mode)	- - -	Input Output Input
PL2	Input mode Output mode	- -	Input Output
PL3,PL7	Input mode Output mode,TCOUT6,TCOUT7	- -	Input Output
PL4~PL6	Input mode,SCLK9,RXD9,*CTS9 Output mode,SCLK9,TXD9	- -	Input Output
PM0~PM5	Input mode Output mode INT0~INT5(Input mode)	- - Input	Input Output Input
PM6,PM7	Input mode Output mode,TCOUT2,TCOUT3	- -	Input Output
PN0~PN2			
PN3,PN7	Input mode Output mode ADTRG-1,ADTRG-2(Input mode)	- - -	Input Output Input
PN4~PN6	Input mode,SCLKA,RXDA,*CTSA Output mode,SCLKA,TXDA	- -	Input Output
PO0~PO7	Input mode Output mode,TPD0~TPD7	- -	Input Output
PP0~PP7	Input mode Output mode,TPD0~TPD7 TPC0~TPC7	- - -	Input Output -
PQ0~PQ3	Input mode Output mode DREQ2,DACK2,DREQ3,DACK3	- -	Input Output
EJE,	Input pin	Input	Input
DINT,TMS,TCK	Input pin	Input	Input
TRST	Input pin	Input	Input

Table 5.8.5 Pin States in the STOP Mode in Each State of SYSCR2<DRVE> (3/3)

Pin name	Input/Output	<DRVE>=0	<DRVE>=1
*NMI	Input pin	Input	Input
*PLLSEL	Input pin	Input	Input
*RESET	Input pin	Input	Input
BUSMD	Input pin	Input	Input
ENDIAN	Input pin	Input	Input
BW0~1	Input pin	Input	Input
TEST1~3	Input pin	Input	Input
X1	Input pin	-	-
X2	Output pin	“H” level output	“H” level output

- :Indicates that the input is disabled for the input mode and the input pin and the impedance becomes high for the output mode and the output pin. Note that the input is enabled when the port function register (PxFC) is "1" and the port control register (PxCR) is "0" in case INTx or KWUPx are used for STOP release.

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

Output :The pin is in the output state.

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

6. Exceptions/Interrupts

6.1 Overview

The TMP19A63 device is configured with the following 107 maskable interrupt factors and 14 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: 85 internal factors and 20 external factors (INT0~B,KWUP0~7)

The TMP19A63 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 6.1 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
Other exceptions	0x8000_0180	0xBFC0_0380

(Note) 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 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 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 6.11, NMI Flag Register). 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.

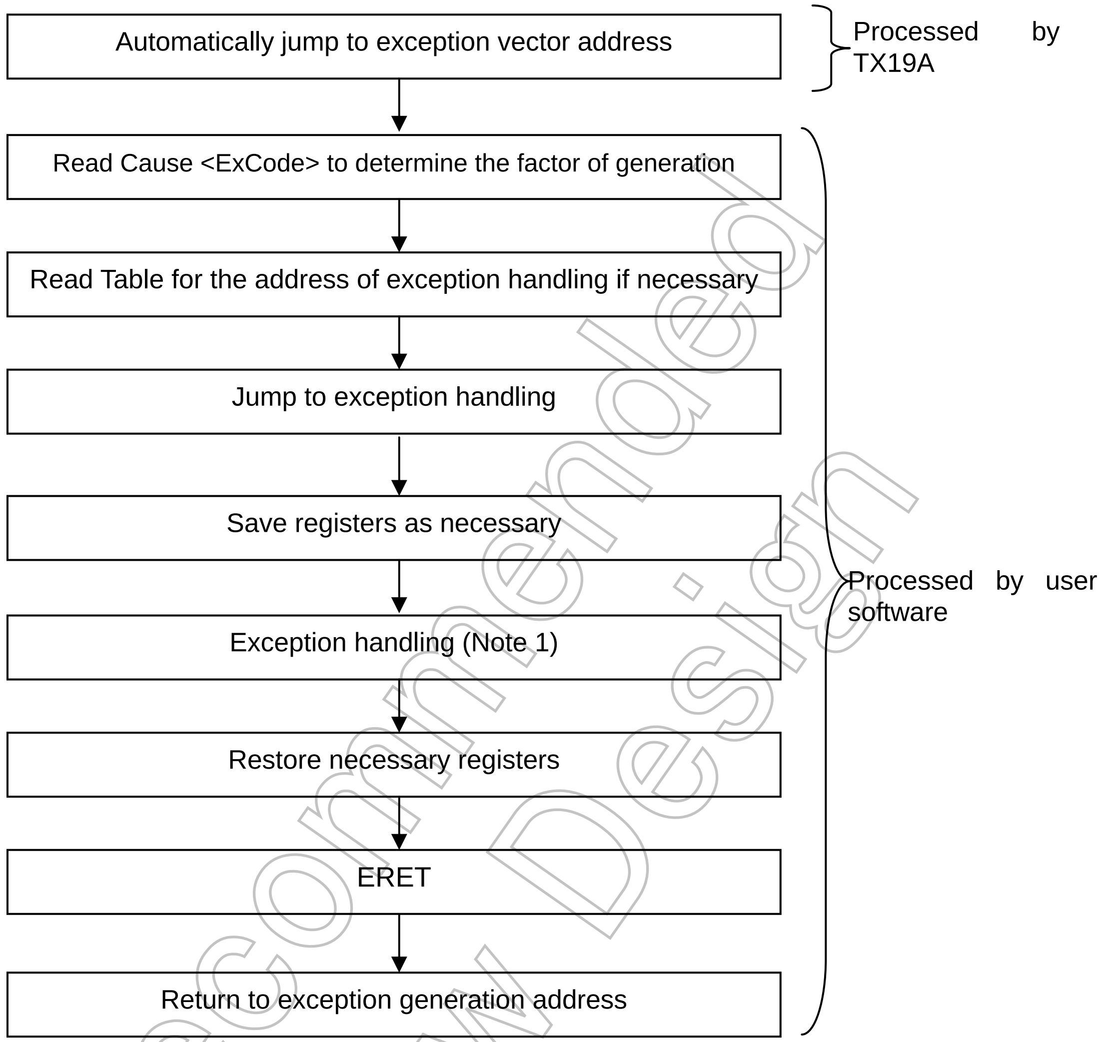


Fig. 6.1 Example Sequence of General Exceptions (Other than Reset Exception and NMI)

(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) "Software interrupt" is different from the idea of "software set" to be used as one of hardware interrupt factors, as described later. The idea of "Software set" is to generate a hardware interrupt by setting "01" to IMR00 <EIM00>.

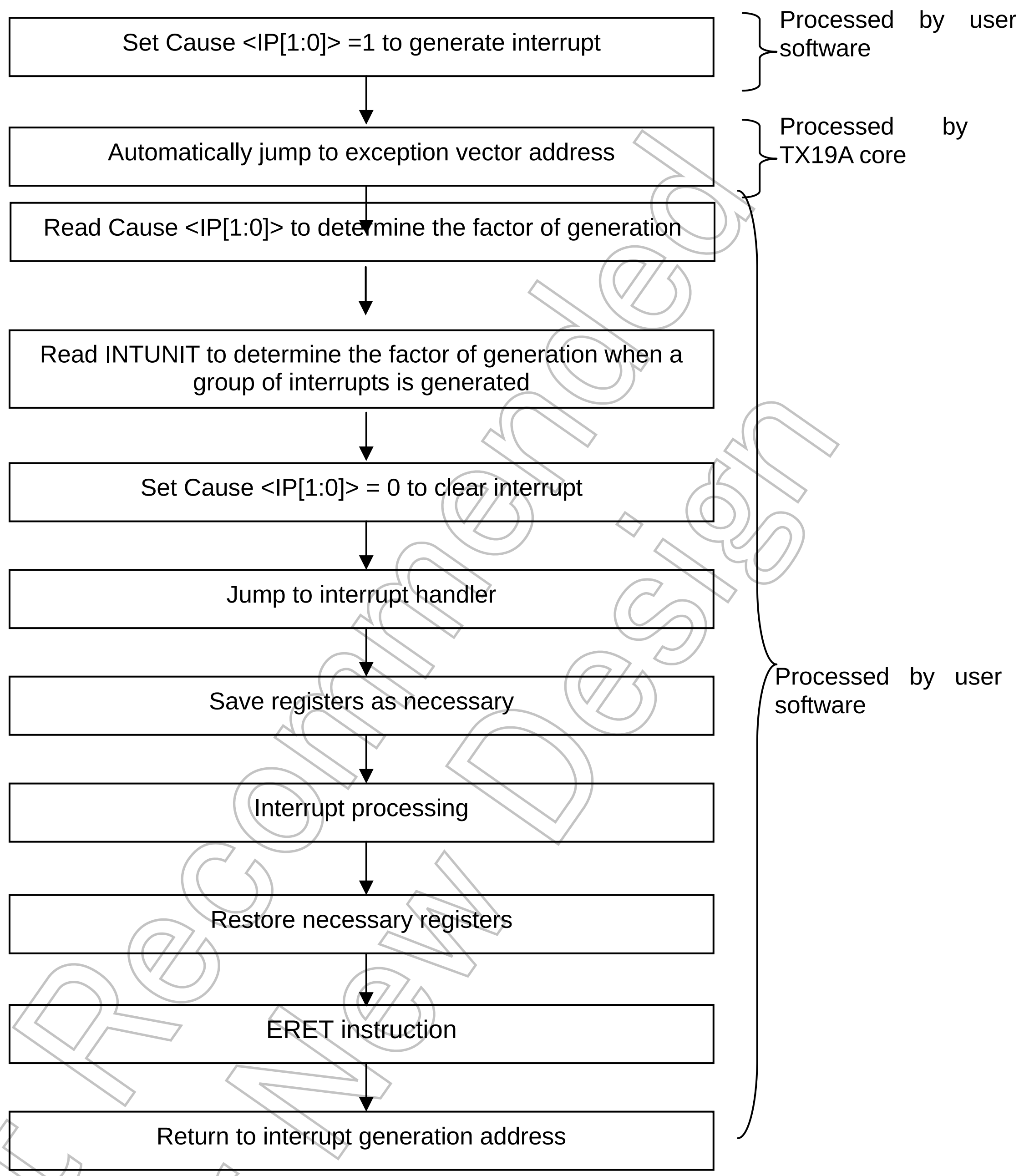


Fig. 6.2 Example of Software Interrupt Operation

(Note) 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 63 factor interrupt requests for which the interrupt controller (INTC) can individually assign one 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 6.2.

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" which is enabled when CP0 register SSCR<SSD>="0". 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.

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 6.9.3 "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.

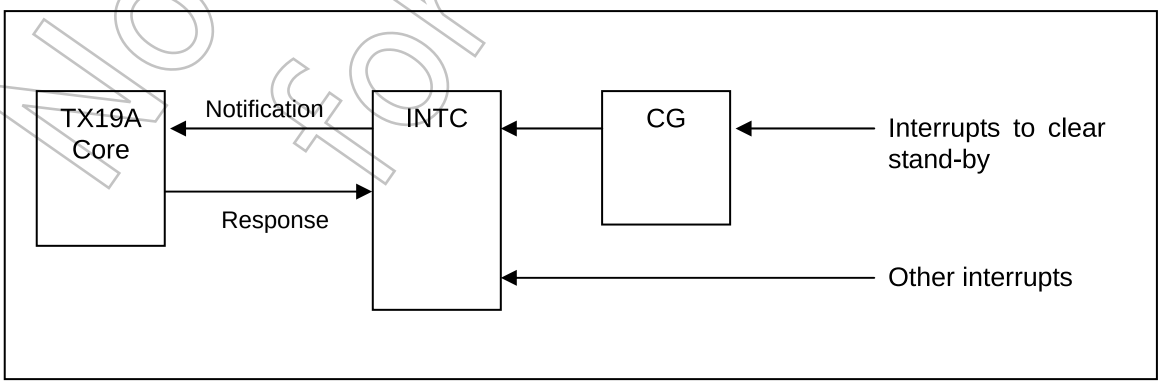


Fig. 6.2 Interrupt Notification Diagram

Table 6.8.2 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		
2	0x008	INT1		
3	0x00C	INT2		
4	0x010	INT3	IMC1	0xFFFF_E004
5	0x014	INT4		
6	0x018	INT5		
7	0x01C	INT6		
8	0x020	INT7	IMC2	0xFFFF_E008
9	0x024	INT8		
10	0x028	INT9		
11	0x02C	INTA		
12	0x030	INTB	IMC3	0xFFFF_E00C
13	0x034	KWUP		
14	0x038	INTRX0 :		
15	0x03C	INTTX0 :		
16	0x040	INTRX1 :	IMC4	0xFFFF_E010
17	0x044	INTTX1 :		
18	0x048	INTRX2 :		
19	0x04C	INTTX2 :		
20	0x050	INTSBIA :	IMC5	0xFFFF_E014
21	0x054	INTADHPA :		
22	0x058	INTADHPB :		
23	0x05C	INTADM :		
24	0x060	INTTB00 :	IMC6	0xFFFF_E018
25	0x064	INTTB08 :		
26	0x068	INTTB12 :		
27	0x06C	INTTB14 :		
28	0x070	INTTB01-07 :	IMC7	0xFFFF_E01C
29	0x074	INTTB09-0F :		
30	0x078	INTTB10-17:		
31	0x07C	INTTB18-1F:		
32	0x080	INTTB20-23:	IMC8	0xFFFF_E020
33	0x084	INTCAPG :		
34	0x088	INTCMPGR :		
35	0x08C	INTTBT :		
36	0x090	Reserved	IMC9	0xFFFF_E024
37	0x094	INTRX3 :		
38	0x098	INTTX3 :		
39	0x09C	INTRX4 :		
40	0x0A0	INTTX4 :	IMCA	0xFFFF_E028
41	0x0A4	INTRX5 :		
42	0x0A8	INTTX5 :		
43	0x0AC	INTRX6 :		
44	0x0B0	INTTX6 :	IMCB	0xFFFF_E02C
45	0x0B4	INTRX7 :		
46	0x0B8	INRTX7 :		
47	0x0BC	INTRX8 :		
48	0x0C0	INTTX8 :	IMCC	0xFFFF_E030
49	0x0C4	INTRX9 :		
50	0x0C8	INTTX9 :		
51	0x0CC	INTSBIB :		
52	0x0D0	INTRXA :	IMCD	0xFFFF_E034
53	0x0D4	INTTXA :		
54	0x0D8	INTDMA0 :		
55	0x0DC	INTDMA1 :		
56	0x0E0	INTDMA2 :	IMCE	0xFFFF_E038
57	0x0E4	INTDMA3 :		
58	0x0E8	INTDMA4 :		
59	0x0EC	INTDMA5 :		
60	0x0F0	INTDMA6 :	IMCF	0xFFFF_E03C
61	0x0F4	INTDMA7 :		
62	0x0F8	INTADA :		
63	0x0FC	INTADB :		

(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 6.8.3 Interrupt Factors to Cancel Stop Mode

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	Reserved	
14	Reserved	
15	Reserved	

* Number 0 to 12 interrupt factors can cancel Stop and Idle modes.

6.8.2 Interrupt Grouping Registers

ADCINT
(0xFFFF_E700)

	7	6	5	4	3	2	1	0
Bit Symbol							INTADMB	INTADMA
Read/Write	R							
After Reset							0	0
Function							:Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

* AD monitoring function for interrupt

	15	14	13	12	11	10	9	8
Bit Symbol							IMINTADMB	IMINTADMA
Read/Write	R/W							
After Reset							0	0
Function							:With MASK 0: No 1: Yes	:With MASK 0: No 1: Yes

TMRBINTA
(0xFFFF_E704)

	7	6	5	4	3	2	1	0
Bit Symbol	INTTB07	INTTB06	INTTB05	INTTB04	INTTB03	TINTB02	INTTB01	
Read/Write	R							
After Reset	0	0	0	0	0	0	0	
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	

	15	14	13	12	11	10	9	8
Bit Symbol	IMINTTB07	IMINTTB06	IMINTTB05	IMINTTB04	IMINTTB03	TIMINTB02	IMINTTB01	
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	

TMRBINTB
(0xFFFF_E708)

	7	6	5	4	3	2	1	0
Bit Symbol	INTTB0F	INTTB0E	INTTB0D	INTTB0C	INTTB0B	TINTB0A	INTTB09	
Read/Write	R							
After Reset	0	0	0	0	0	0	0	
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	

	15	14	13	12	11	10	9	8
Bit Symbol	IMINTTB0F	IMINTTB0E	IMINTTB0D	IMINTTB0C	IMINTTB0B	TIMINTB0A	IMINTTB09	
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	

TMRBINTC
(0xFFFF_E70C)

	7	6	5	4	3	2	1	0
Bit Symbol	INTTB17	INTTB16	INTTB15		INTTB13		INTTB11	INTTB10
Read/Write	R							
After Reset	0	0	0		0		0	0
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes		: Interrupt 0: No 1: Yes		: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

	15	14	13	12	11	10	9	8
Bit Symbol	IMINTTB17	IMINTTB16	IMINTTB15		IMINTTB13		IMINTTB11	IMINTTB10
Read/Write	R/W							
After Reset	0	0	0		0		0	0
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes		: With MASK 0: No 1: Yes		: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes

TMRBINTD
(0xFFFF_E710)

	7	6	5	4	3	2	1	0
Bit Symbol	INTTB1F	INTTB1E	INTTB1D	INTTB1C	INTTB1B	TINTB1A	INTTB19	INTTB18
Read/Write								
After Reset	0	0	0	0	0	0	0	0
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

	15	14	13	12	11	10	9	8
Bit Symbol	IMINTTB1	IMINTTB1	IMINTTB1	IMINTTB1	IMINTTB1	TIMINTB1	IMINTTB1	IMINTTB1
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes

TMRBINTE
(0xFFFF_E714)

	7	6	5	4	3	2	1	0
Bit Symbol	INTCPT 0B	INTCPT 0A	INTCPT 09	INTCPT 08	INTTB23	TINTB22	INTTB21	INTTB20
Read/Write	R							
After Reset					0	0	0	0
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

	15	14	13	12	11	10	9	8
Bit Symbol	IMINT CPT0B	IMINT CPT0A	IMINT CPT09	IMINT CPT08	IMINTTB2 3	TIMINTB2 2	IMINTTB2 1	IMINTTB2 0
Read/Write	R/W							
After Reset					0	0	0	0
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes

CAPINT
(0xFFFF_E718)

	7	6	5	4	3	2	1	0
Bit Symbol			INTCAP B1	INTCAP B0			INTCAP A1	INTCAP A0
Read/Write	R							
After Reset			0	0			0	0
Function			: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes			: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

	15	14	13	12	11	10	9	8
Bit Symbol			IMINTCAP B1	IMINTCAP B0			IMINTCAP A1	IMINTCAP A0
Read/Write	R/W							
After Reset			0	0			0	0
Function			: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes			: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes

CMPINT
(0xFFFF_E71C)

	7	6	5	4	3	2	1	0
Bit Symbol	INTCMP B1	INTCMP B0			INTCMP A1	INTCMP A0		
Read/Write	R							
After Reset	0	0			0	0		
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes			: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes		

	15	14	13	12	11	10	9	8
Bit Symbol	IMINTCM PB1	IMINTCM PB0			IMINTCM PA1	IMINTCM PA0		
Read/Write	R/W							
After Reset	0	0			0	0		
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes			: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes		

TBTINT
(0xFFFF_E720)

	7	6	5	4	3	2	1	0
Bit Symbol	INTCPT 11	INTCPT 10	INTCPT 0F	INTCPT 0E	INTCPT 0D		INTTBTB	INTTBTA
Read/Write	R							
After Reset							0	0
Function	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes		: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

	15	14	13	12	11	10	9	8
Bit Symbol	IMINT CPT11	IMINT CPT10	IMINT CPT0F	IMINT CPT0E	IMINT CPT0D		IMINT TBTB	IMINT TBTA
Read/Write	R/W							
After Reset							0	0
Function	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes		: With MASK 0: No 1: Yes	: With MASK 0: No 1: Yes

KWUPST
(0xFFFF_F910)

	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: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes	: Interrupt 0: No 1: Yes

6.8.3 Detecting Interrupt Requests

Each of interrupt factors has its own interrupt detection sequence as described in Table 6.8.4. Upon detection, an interrupt request is notified to INTC for priority arbitration and then notified to the TX19A processor core. Refer to Table 6.8.5 for the detection level available for each interrupt factor.

Table 6.8.4 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)Other interrupts	INTC	Peripheral circuit → INTC (detection/ arbitration) → TX19A Core

6.8.4 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 6.2.

   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) 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.

6.8.5 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 6.1 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 6.8.2.

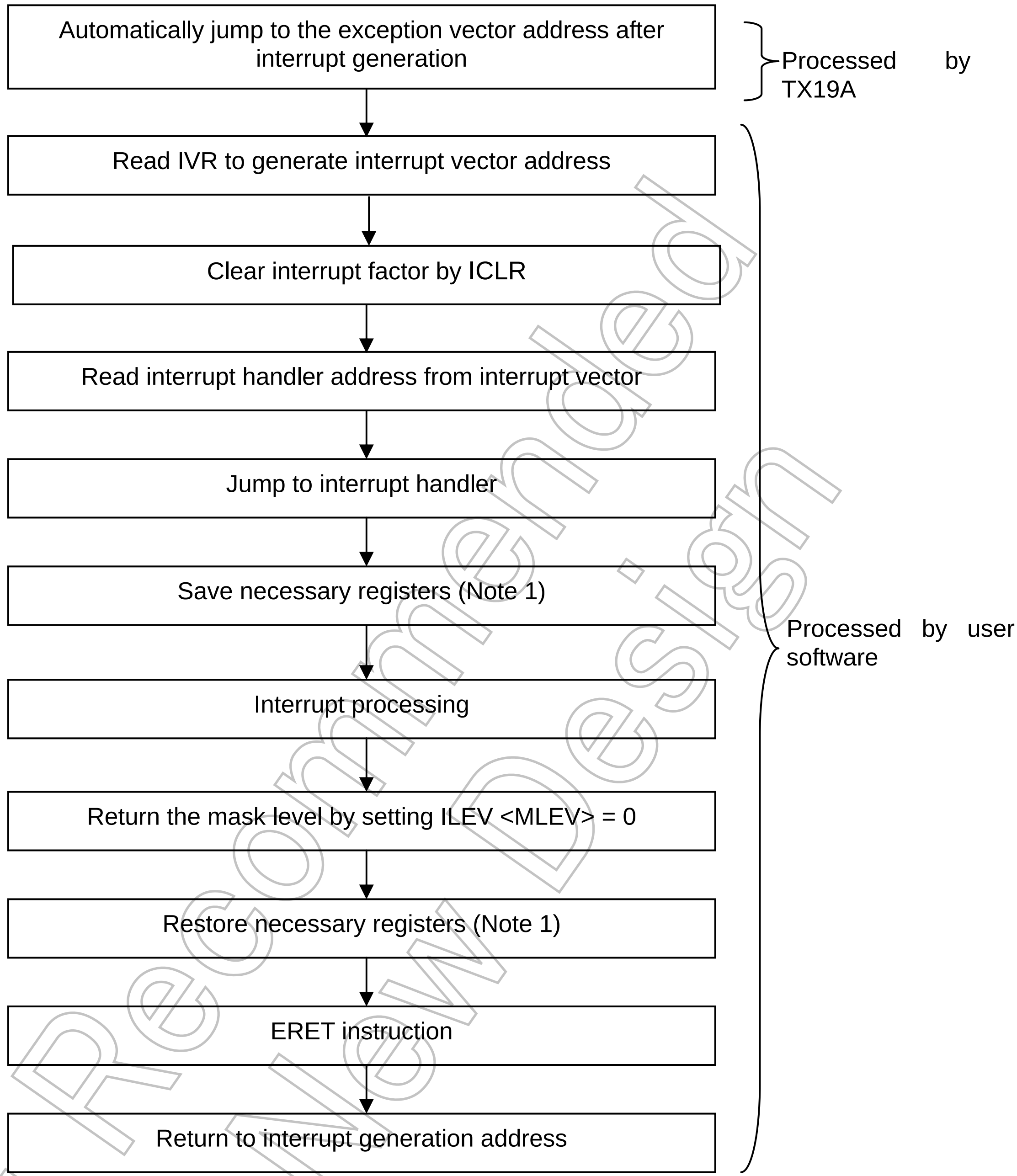


Fig. 6.4 Basic Operation of Hardware Interrupts (Example)

(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.

6.9.1 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 6.9.1.1 “ Common Initialization” and settings specifically required for certain factors and applications are described in Section 6.9.1.2 “Initialization for Individual Interrupt Factors”.

6.9.1.1 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 Vector Table 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) 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.

6.9.1.2 Initialization for Individual Interrupt Factors

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

Table 6.8.5 Registers to be Set for Detecting Interrupts

Interrupt	Detected at	Registers to be Set	Interrupt detection levels available (setting in active condition)
(1) Interrupts from external pins INT0~INTB	INTC	PxFC(PORT) PxCR(PORT) IMCxx(INTC)	With INTC, "L" and "H" levels and falling and rising edges can be set.
	CG	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 CG, "L" and "H" levels and falling/rising edges can be set.
(2) Two-phase counter interrupts	INTC	PxFC(PORT) PxCR(PORT) IMCxx(INTC)	It must be set to rising edge with INTC.
(3) Other interrupts	INTC	IMCxx(INTC)	With INTC, "L" and "H" levels and falling and 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. As for CG edge detection, the edge of the input signal is detected without using internal clock.

(Note 2) 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~INTB

- Use PORT PxCR and PxIE 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 PxPUP 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 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 6.9.4. Registers).

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

Status<IE> ="0"	; Interrupt is disabled
PMCR<PM3C> ="0"	; The port is set to an input port
PMFC<PM3F> ="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) Other hardware interrupts

- Settings are made to use peripheral hardware devices.
- Set INTC IMCxx <EIMxx> (refer to 6.9.4 Registers).

(Note) 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.

6.9.1.3 Interrupt Enable

In order for an interrupt request to be accepted, all the following three parameters must be set to enable the interrupt in addition to the initial settings described in Section 6.9.11 "Initialization for Interrupts".

- Status <ERL> of the CP0 register is set to "0."
- Status <EXL> of the CP0 register is set to "0."
- Status <IE> of the CP0 register is set 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 four methods may be used in setting Status <IE> of the CP0 register to "1."

1. Set Status<IE> of the CP0 register to “1” using the MTC0 instruction (32 bit ISA instruction)
2. Set IER of the CP0 register to any value other than "0" using the MTC0 instruction (32 bit ISA instruction). (Note 1)
3. Set Status<IE> of the CP0 register to “1” using the MTC0 instruction (16 bit ISA instruction).
4. Execute the EI instruction of 16 bit ISA. (Note 2)

(Note 1) This method is recommended for 32 bit ISA because it enables the minimum code increase and high speed processing. If Toshiba C compiler is used, this is executed by the 32 bit ISA instruction "_ _EI ( ) embedded function."

(Note 2) This method is recommended for 16 bit ISA because it enables the minimum code increase and high speed processing. If Toshiba C compiler is used, this instruction is executed by the 16 bit ISA instruction "_ _EI ( ) embedded function."

6.9.1.4 Interrupt Disable

To disable interrupts, either one of the following setting procedures must be performed in addition to the settings described in Section 6.9.1 "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.

Status <ERL> and <EXL> of CP0 register are automatically set by an interrupt or an exception and cleared by ERET instruction. These bits are automatically cleared by ERET instruction. Therefore we recommend setting Status <IE> of CP0 register to “0” to prohibit normal interrupts. Please refer to “6.9.3 Example of Multiple Interrupt Setting” for disabling interrupts using multiple interrupt. Note that one of the following methods may be used in setting Status <IE> of the CP0 register to "0."

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

(Note 1) This method is recommended for 32 bit ISA because it enables the minimum code increase and high speed processing. If Toshiba C compiler is used, this instruction is executed by the 32 bit ISA instruction "_ _DI ( ) embedded function."

(Note 2) This method is recommended for 16 bit ISA because it enables the minimum code increase and high speed processing. If Toshiba C compiler is used, this instruction is executed by the 32 bit ISA instruction "_ _DI ( ) embedded function."

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        ; Interrupt factor is disabled.
sync                     ; Stall until it is write-enabled.
mtc0    r29, IER         ; Interrupt is enabled (Status<IE> =”1”).
```

(Note 1) 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.

6.9.2 Interrupt Processing

This section describes detailed operation of interrupt processing using the basic flow chart of Fig. 6.4.

6.9.2.1 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. 6.5 shows the sequence of accepting interrupts.

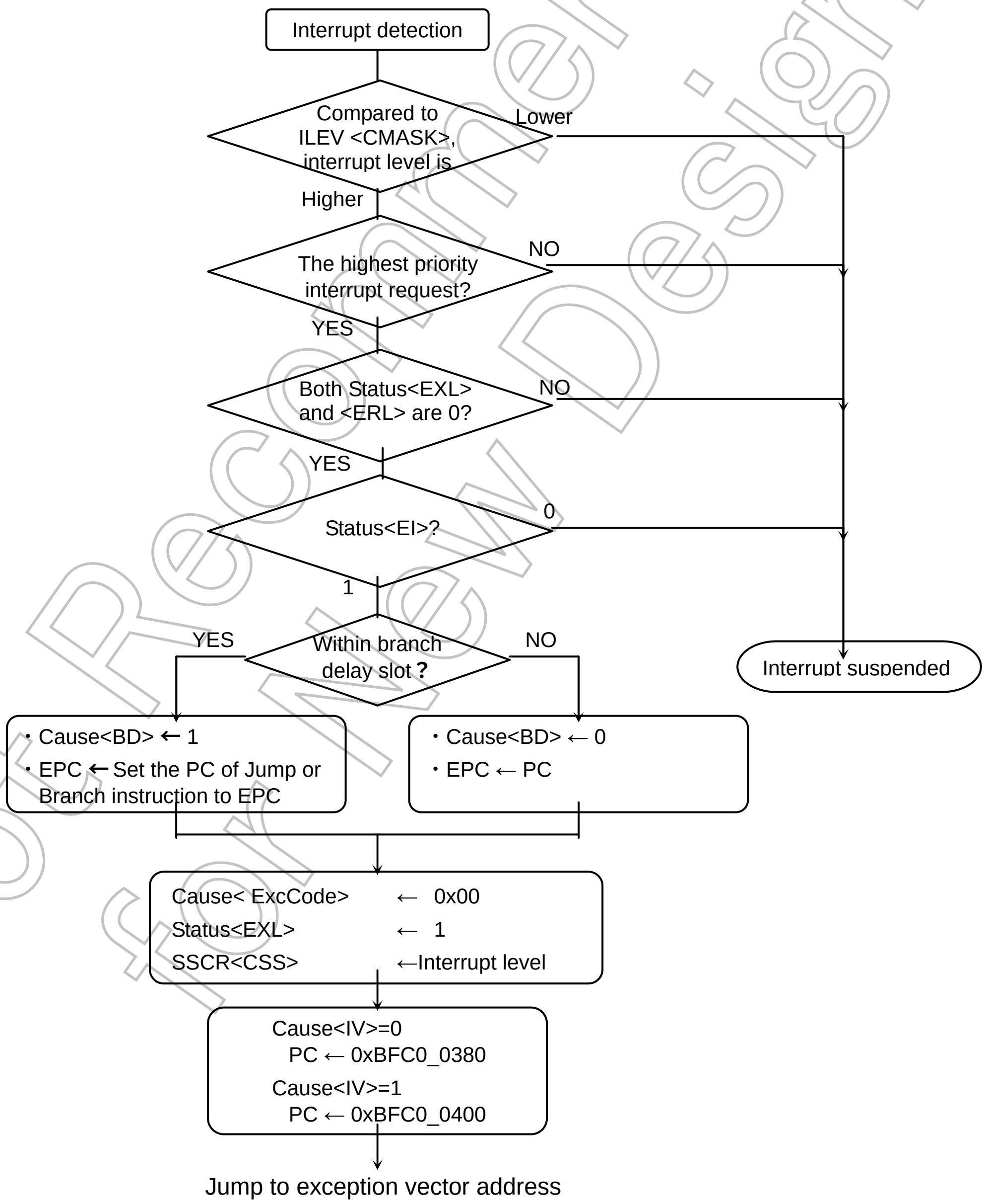


Fig. 6.5 Hardware Process Flow to Accept Interrupts

②Processes 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 6.9.1, “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.

6.9.3 Example of Multiple Interrupt Setting

In "multiple interrupt" processing, a higher interrupt level interrupt is processed while an interrupt is being processed. With TMP19A63, 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) 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 6.9.1.4 "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 6.9.3.1, "Interrupt Control for Multiple Interrupts".

6.9.3.1 Interrupt Control for Multiple Interrupts

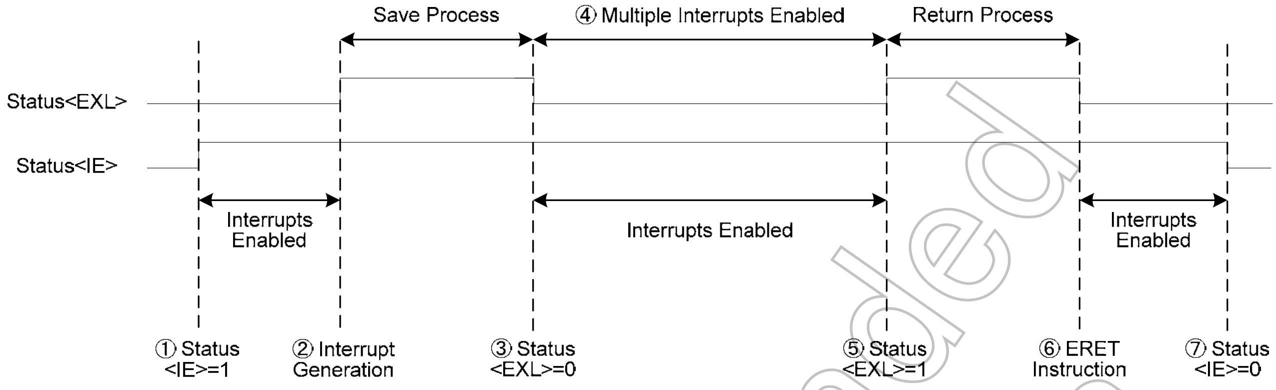


Fig. 6.6 Interrupt Enable/Disable Control Sequence for Multiple Interrupts

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.

6.9.4 Registers

6.9.4.1 Register Map

Table 6.6 INTC Register Map

Address	Register symbol	Register	Corresponding interrupt number
0xFFFF_E000	IMC0	Interrupt mode control register 00	0 ~ 3
0xFFFF_E004	IMC1	Interrupt mode control register 04	4 ~ 7
0xFFFF_E008	IMC2	Interrupt mode control register 08	8 ~ 11
0xFFFF_E00C	IMC3	Interrupt mode control register 12	12 ~ 15
0xFFFF_E010	IMC4	Interrupt mode control register 16	16 ~ 19
0xFFFF_E014	IMC5	Interrupt mode control register 20	20 ~ 23
0xFFFF_E018	IMC6	Interrupt mode control register 24	24 ~ 27
0xFFFF_E01C	IMC7	Interrupt mode control register 28	28 ~ 31
0xFFFF_E020	IMC8	Interrupt mode control register 32	32 ~ 35
0xFFFF_E024	IMC9	Interrupt mode control register 36	36 ~ 39
0xFFFF_E028	IMCA	Interrupt mode control register 40	40 ~ 43
0xFFFF_E02C	IMCB	Interrupt mode control register 44	44 ~ 47
0xFFFF_E030	IMCC	Interrupt mode control register 48	48 ~ 51
0xFFFF_E034	IMCD	Interrupt mode control register 52	52 ~ 55
0xFFFF_E038	IMCE	Interrupt mode control register 56	56 ~ 59
0xFFFF_E03C	IMCF	Interrupt mode control register 60	60 ~ 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.

6.9.4.2 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 6.2 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.

Interrupt Vector Register

		7	6	5	4	3	2	1	0
IVR (0xFFFF_E040)	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								Always reads "0."
		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								

6.9.4.3 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."

Interrupt Level Register

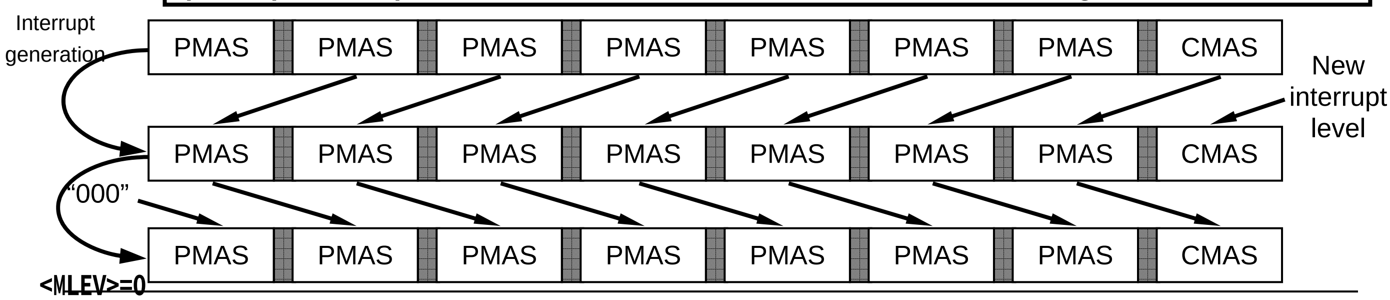
ILEV (0xFFFF_E10C)

	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.



6.9.4.4 Interrupt Mode Control Registers (IMCxx)

IMCxx is comprised of <ILxx>, 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 001~111: 1~7 If DM0 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	15	14	13	12	11	10	9	8
Bit Symbol		EIM11	EIM10	DM1		IL12	IL11	IL10
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **Set it to "0" when using CG.**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 1 is set as the activation factor	Always reads "0."	If DM1 = 0, select the interrupt level for interrupt number 1 (INT0) 000: Disable interrupt 001~111: 1~7 If DM1 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	23	22	21	20	19	18	17	16
Bit Symbol		EIM21	EIM20	DM2		IL22	IL21	IL20
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **Set it to "0" when using CG.**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 2 is set as the activation factor	Always reads "0."	If DM2 = 0, select the interrupt level for interrupt number 2 (INT1) 000: Disable interrupt 001~111: 1~7 If DM2 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	31	30	29	28	27	26	25	24
Bit Symbol		EIM31	EIM30	DM3		IL32	IL31	IL30
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **Set it to "0" when using CG.**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 3 is set as the activation factor	Always reads "0."	If DM3 = 0, select the interrupt level for interrupt number 3 (INT2) 000: Disable interrupt 001~111: 1~7 If DM3 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

IMC1
(0xFFFF_E004)

	7	6	5	4	3	2	1	0
Bit Symbol		EIM41	EIM40	DM4		IL42	IL41	IL40
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”	Selects active state of interrupt request: 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **Set it to “0” when using CG.**		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 001~111: 1~7 If DM4 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

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

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

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

IMC2
(0xFFFF_E008)

	7	6	5	4	3	2	1	0
Bit Symbol		EIM81	EIM80	DM8		IL82	IL81	IL80
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”	Selects active state of interrupt request: 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **Set it to “0” when using CG.**		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 001~111: 1~7 If DM8 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	15	14	13	12	11	10	9	8
Bit Symbol		EIM91	EIM90	DM9		IL92	IL91	IL90
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”	Selects active state of interrupt request: 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **Set it to “0” when using CG.**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 9 is set as the activation factor	Always reads “0.”	If DM9 = 0, select the interrupt level for interrupt number 9 (INT8) 000: Disable interrupt 001~111: 1~7 If DM9 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	23	22	21	20	19	18	17	16
Bit Symbol		EIMA1	EIMA0	DMA		ILA2	ILA1	ILA0
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”	Selects active state of interrupt request: 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **Set it to “0” when using CG.**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number10 is set as the activation factor	Always reads “0.”	If DMA = 0, select the interrupt level for interrupt number 10 (INT9) 000: Disable interrupt 001~111: 1~7 If DMA = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	31	30	29	28	27	26	25	24
Bit Symbol		EIMB1	EIMB0	DMB		ILB2	ILB1	ILB0
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”	Selects active state of interrupt request: 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge **Set it to “0” when using CG.**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 11 is set as the activation factor	Always reads “0.”	If DMB = 0, select the interrupt level for interrupt number 11 (INTA) 000: Disable interrupt 001~111: 1~7 If DMB = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

IMC3
(0xFFFF_E00C)

	7	6	5	4	3	2	1	0
Bit Symbol		EIMC1	EIMC0	DMC		ILC2	ILC1	ILC0
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge **Set it to "0" when using CG.**		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 001~111: 1~7 If DMC = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

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

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

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

		7	6	5	4	3	2	1	0
IMC4 (0xFFFF_E010)	Bit Symbol		EIM101	EIM100	DM10		IL102	IL101	IL100
	Read/Write	R	R/W			R	R/W		
	After Reset	0	0	0	0	0	0	0	0
	Function	Always reads “0.”	Selects active state of interrupt request: 11: Rising edge		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 16 is set as the activation factor	Always reads “0.”	If DM10 = 0, select the interrupt level for interrupt number 16 (INTRX1) 000: Disable interrupt 001~111: 1~7 If DMC = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	Bit Symbol								
	Read/Write	R							
	After Reset	0	0	0	0	0	0	0	0
	Function								
		15	14	13	12	11	10	9	8
	Bit Symbol		EIM111	EIM110	DM11		IL112	IL111	IL110
	Read/Write	R	R/W			R	R/W		
	After Reset	0	0	0	0	0	0	0	0
	Function	Always reads “0.”	Selects active state of interrupt request: 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 17 is set as the activation factor	Always reads “0.”	If DM11= 0, select the interrupt level for interrupt number 17 (INTTX1) 000: Disable interrupt 001~111: 1~7 If DM11 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
		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 is set as the activation factor	Always reads “0.”	If DM12 = 0, select the interrupt level for interrupt number 18 (INTRX2) 000: Disable interrupt 001~111: 1~7 If DM12 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
		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 is set as the activation factor	Always reads “0.”	If DM13 = 0, select the interrupt level for interrupt number 19 (INTTX2) 000: Disable interrupt 001~111: 1~7 If DM13 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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 (INTSBI0) 000: Disable interrupt 001~111: 1~7 If DM14 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads “0.”	If DM15 = 0, select the interrupt level for interrupt number 21 (INTADHPA) 000: Disable interrupt 001~111: 1~7 If DM15 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads “0.”	If DM16 = 0, select the interrupt level for interrupt number 22 (INTADHPB) 000: Disable interrupt 001~111: 1~7 If DM16 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: “H” level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 23 is set as the activation factor	Always reads “0.”	If DM17= 0, select the interrupt level for interrupt number 23 (INTADM) 000: Disable interrupt 001~111: 1~7 If DM17 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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 (INTTB0) 000: Disable interrupt 001~111: 1~7 If DM18 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads "0."	If DM19 = 0, select the interrupt level for interrupt number 25 (INTTB8) 000: Disable interrupt 001~111: 1~7 If DM19 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads "0."	If DM1A = 0, select the interrupt level for interrupt number 26 (INTTB12) 000: Disable interrupt 001~111: 1~7 If DM1A = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads "0."	If DM1B= 0, select the interrupt level for interrupt number 27 (INTTB14) 000: Disable interrupt 001~111: 1~7 If DM1B = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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: 01: "H" level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 28 is set as the activation factor	Always reads "0."	If DM1C = 0, select the interrupt level for interrupt number 28 (INTTBT01-07) 000: Disable interrupt 001~111: 1~7 If DM1C = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: "H" level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 29 is set as the activation factor	Always reads "0."	If DM1D = 0, select the interrupt level for interrupt number 29 (INTTB09-0F) 000: Disable interrupt 001~111: 1~7 If DM1D = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: "H" level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 30 is set as the activation factor	Always reads "0."	If DM1E = 0, select the interrupt level for interrupt number 30 (INTTB10-17) 000: Disable interrupt 001~111: 1~7 If DM1E = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: "H" level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 31 is set as the activation factor	Always reads "0."	If DM1F = 0, select the interrupt level for interrupt number 31 (INTTB18-1F) 000: Disable interrupt 001~111: 1~7 If DM1F = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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: 01: “H” level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 32 is set as the activation factor	Always reads “0.”	If DM20 = 0, select the interrupt level for interrupt number 32 (INTTB20-23) 000: Disable interrupt 001~111: 1~7 If DM20 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: “H” level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 33 is set as the activation factor	Always reads “0.”	If DM21 = 0, select the interrupt level for interrupt number 33 (INTCAPG) 000: Disable interrupt 001~111: 1~7 If DM21 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: “H” level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 34 is set as the activation factor	Always reads “0.”	If DM22 = 0, select the interrupt level for interrupt number 34 (INTCMPGR) 000: Disable interrupt 001~111: 1~7 If DM22 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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: 01: “H” level		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 35 is set as the activation factor	Always reads “0.”	If DM23 = 0, select the interrupt level for interrupt number 35 (INTTBT) 000: Disable interrupt 001~111: 1~7 If DM23 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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								
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”							

	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 is set as the activation factor	Always reads “0.”	If DM25 = 0, select the interrupt level for interrupt number 37 (INTRX3) 000: Disable interrupt 001~111: 1~7 If DM25 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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 is set as the activation factor	Always reads “0.”	If DM26 = 0, select the interrupt level for interrupt number 38 (INTTX3) 000: Disable interrupt 001~111: 1~7 If DM26 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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 is set as the activation factor	Always reads “0.”	If DM27 = 0, select the interrupt level for interrupt number 39 (INTRX4) 000: Disable interrupt 001~111: 1~7 If DM27 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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 is set as the activation factor	Always reads “0.”	If DM28 = 0, select the interrupt level for interrupt number 40 (INTTX4) 000: Disable interrupt 001~111: 1~7 If DM28 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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 is set as the activation factor	Always reads “0.”	If DM29 = 0, select the interrupt level for interrupt number 41 (INTRX5) 000: Disable interrupt 001~111: 1~7 If DM29 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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 is set as the activation factor	Always reads “0.”	If DM2A = 0, select the interrupt level for interrupt number 42 (INTTX5) 000: Disable interrupt 001~111: 1~7 If DM2A = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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 is set as the activation factor	Always reads “0.”	If DM2B = 0, select the interrupt level for interrupt number 43 (INTTX6) 000: Disable interrupt 001~111: 1~7 If DM2B = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(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 is set as the activation factor	Always reads "0."	If DM2C = 0, select the interrupt level for interrupt number 44 (INTTX6) 000: Disable interrupt 001~111: 1~7 If DM2C = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads "0."	If DM2D = 0, select the interrupt level for interrupt number 45 (INTRX7) 000: Disable interrupt 001~111: 1~7 If DM2D = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads "0."	If DM2E = 0, select the interrupt level for interrupt number 46 (INTTX7) 000: Disable interrupt 001~111: 1~7 If DM2E = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
	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 is set as the activation factor	Always reads "0."	If DM2F = 0, select the interrupt level for interrupt number 47 (INTRX8) 000: Disable interrupt 001~111: 1~7 If DM2F = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

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

		7	6	5	4	3	2	1	0
IMCC (0xFFFF_E030)	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 is set as the activation factor	Always reads “0.”	If DM30 = 0, select the interrupt level for interrupt number 48 (INTTX8) 000: Disable interrupt 001~111: 1~7 If DM30 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
		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: 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 49 is set as the activation factor	Always reads “0.”	If DM31 = 0, select the interrupt level for interrupt number 49 (INTRX9) 000: Disable interrupt 001~111: 1~7 If DM31 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
		23	22	21	20	19	18	17	16
	Bit Symbol								
	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 50 is set as the activation factor	Always reads “0.”	If DM32 = 0, select the interrupt level for interrupt number 50(INTTX9) 000: Disable interrupt 001~111: 1~7 If DM32 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		
		31	30	29	28	27	26	25	24
	Bit Symbol		EIM331	EIM330	DM33		IL332	IL331	IL330
	Read/Write	R	R/W			R	R/W		
	After Reset	0	0	0	0	0	0	0	0
	Function	Always reads “0.”	Selects active state of interrupt request: 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 51 is set as the activation factor	Always reads “0.”	If DM33 = 0, select the interrupt level for interrupt number 51 (INTSBI1) 000: Disable interrupt 001~111: 1~7 If DM33 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(Note 1) Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.
(Note 2) The access to the DMAC register by DMAC is prohibited.

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 is set as the activation factor	Always reads "0."	If DM34 = 0, select the interrupt level for interrupt number 52 (INTRXA) 000: Disable interrupt 001~111: 1~7 If DM34 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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 is set as the activation factor	Always reads "0."	If DM35 = 0, select the interrupt level for interrupt number 53 (INTTXA) 000: Disable interrupt 001~111: 1~7 If DM35 = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 54 is set as the activation factor	Always reads "0."	If DM36 = 0, select the interrupt level for interrupt number 54 (INTDMA0) 000: Disable interrupt 001~111: 1~7 If DM36 = 1, select the DMAC channel 001~111: 1~7 000: Not setting allowed		

	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: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 55 is set as the activation factor	Always reads "0."	If DM37 = 0, select the interrupt level for interrupt number 55 (INTDMA1) 000: Disable interrupt 001~111: 1~7 If DM37 = 1, select the DMAC channel 000,002~111:0,2~7 001: Not setting allowed		

(Note 1) Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.
(Note 2) The access to the DMAC register by DMAC is prohibited.

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		
Function	Always reads "0."	Selects active state of interrupt request: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 56 is set as the activation factor	Always reads "0."	If DM38 = 0, select the interrupt level for interrupt number 56 (INTDMA2) 000: Disable interrupt 001~111: 1~7 If DM38 = 1, select the DMAC channel 000,001,011~111: 0,1,3~7 010: Not setting allowed		
	15	14	13	12	11	10	9	8
Bit Symbol		EIM391	EIM390	DM39		IL392	IL391	IL390
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 57 is set as the activation factor	Always reads "0."	If DM39 = 0, select the interrupt level for interrupt number 57 (INTDMA3) 000: Disable interrupt 001~111: 1~7 If DM39 = 1, select the DMAC channel 000~010,100~111: 0~2,4~7 011: Not setting allowed		
	23	22	21	20	19	18	17	16
Bit Symbol		EIM3A1	EIM3A0	DM3A		IL3A2	IL3A1	IL3A0
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."	Selects active state of interrupt request: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 58 is set as the activation factor	Always reads "0."	If DM3A = 0, select the interrupt level for interrupt number 58 (INTDMA4) 000: Disable interrupt 001~111: 1~7 If DM3A = 1, select the DMAC channel 000~011,101~111: 0~3,5~7 100: Not setting allowed		
	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 is set as the activation factor	Always reads "0."	If DM3B = 0, select the interrupt level for interrupt number 59(INTDMA5) 000: Disable interrupt 001~111: 1~7 If DM3B = 1, select the DMAC channel 000~100,110~111: 0~4,6~7 101: Not setting allowed		

(Note 1) Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.
(Note 2) The access to the DMAC register by DMAC is prohibited.

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		
Function	Always reads “0.”	Selects active state of interrupt request: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 60 is set as the activation factor	Always reads “0.”	If DM3C = 0, select the interrupt level for interrupt number 60 (INTDMA6) 000: Disable interrupt 001~111: 1~7 If DM3A = 1, select the DMAC channel 000~101,111: 0~5,7 110: Not setting allowed		

	15	14	13	12	11	10	9	8
Bit Symbol		EIM3D1	EIM3D0	DM3D		IL3D2	IL3D1	IL3D0
Read/Write	R	R/W			R	R/W		
After Reset	0	0	0	0	0	0	0	0
Function	Always reads “0.”	Selects active state of interrupt request: 10: Falling edge **Be sure to set "10."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 61 is set as the activation factor	Always reads “0.”	If DM3D = 0, select the interrupt level for interrupt number 61 (INTDMA7) 000: Disable interrupt 001~111: 1~7 If DM3D = 1, select the DMAC channel 000~110: 0~6 111: Not setting allowed		

	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: 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 62 is set as the activation factor	Always reads “0.”	If DM3E = 0, select the interrupt level for interrupt number 62(INTADA) 000: Disable interrupt 001~111: 1~7 If DM3E = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

	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: 11: Rising edge **Be sure to set "11."**		Set as DMAC activation factor. 0: Non activation factor 1: Interrupt number 63 is set as the activation factor	Always reads “0.”	If DM3F = 0, select the interrupt level for interrupt number 63(INTADB) 000: Disable interrupt 001~111: 1~7 If DM3F = 1, select the DMAC channel 000~011: 0~3 100~111: 4~7		

(Note 1) Default values of EIMxx0 and EIMxx1 are different from the values to be used. Properly set them to the specified values before use.
(Note 2) The access to the DMAC register by DMAC is prohibited.

(Note 1)	**Please ensure that the type of active state is selected before enabling an interrupt request.**
(Note 2)	**When making interrupt requests DMAC activation factors, please ensure that you put the DMAC into standby mode after setting the INTC.**
(Note 3)	**An active condition must be changed after putting the interrupt output of the corresponding device into the state of negate especially when it is changed to the level detection.** **(1) Change the setting from IL="other than 0" to IL="0".** **(2) Change the detection condition (EIM).** **(3) Clear corresponding interrupt by INTCLR.** **(4) Set IL to "other than 0".**

6.9.4.5 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.

Set the IVR <IVR7:0> value that corresponds to the interrupt request that you would like to clear

		7	6	5	4	3	2	1	0
INTCLR	Bit Symbol	EICLR7	EICLR6	EICLR5	EICLR4	EICLR3	EICLR2	EICLR1	EICLR0
(0xFFFF_E060)	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)	**Clear the interrupt request regardless of the active state setting of INTC IMCx <EIMxx>, i.e., either "H" level, "L" level, rising edge, or falling edge, in order to maintain interrupt factors.**
(Note 2)	**Bit manipulation instructions cannot be used to access this register.**
(Note 3)	**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 or disabling interrupts using IMCx <ILxxx> or by canceling the corresponding DMAC activation factors using IMCx<DMxx> before accepting the external transfer requests.**
(Note 4)	**Be sure to clear the corresponding interrupt number with INTCLR after setting IMCx register.**

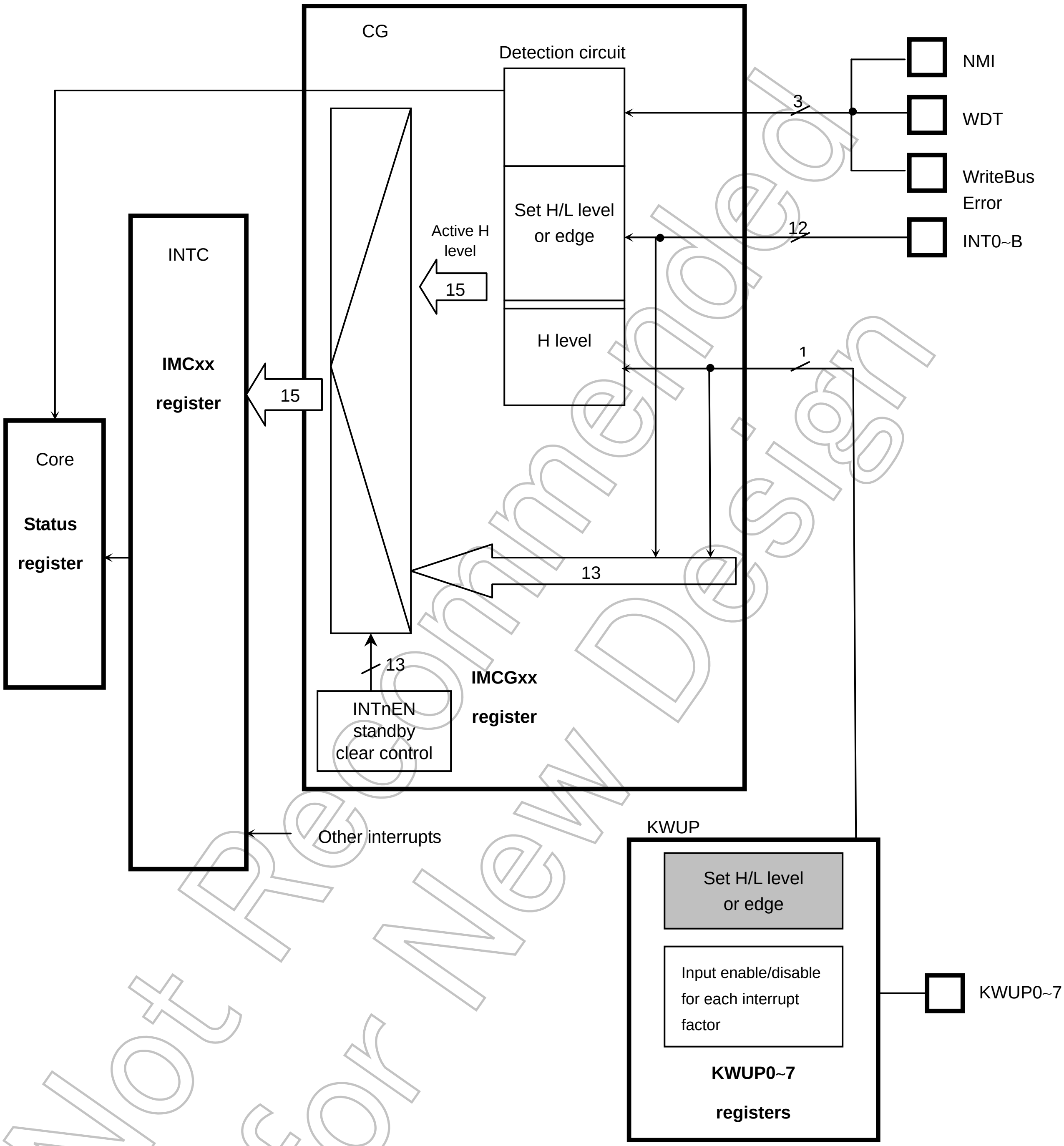


Fig. 6.7 Interrupt Connection Diagram

6.10 INTCG Registers (Interrupts to Clear STOP and IDLE)

INT0 to INTB, KWUP0 to 7 (Interrupts to Clear Stop and Idle modes)

IMCGA
(0xFFFF_EE10)

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

IMCGB
(0xFFFF_EE14)

	7	6	5	4	3	2	1	0
Bit Symbol			EMCG41	EMCG40				INT4EN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT4 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT4 Clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit Symbol			EMCG51	EMCG50				INT5EN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT5 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT5 Clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit Symbol			EMCG61	EMCG60				INT6EN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT6 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT6 Clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
Bit Symbol			EMCG71	EMCG70				INT7EN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT7 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT7 Clear input 0: Disable 1: Enable

IMCGC
(0xFFFF_EE18)

	7	6	5	4	3	2	1	0
Bit Symbol			EMCG81	EMCG80				INT8EN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT8 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT8 Clear input 0: Disable 1: Enable
	15	14	13	12	11	10	9	8
Bit Symbol			EMCG91	EMCG90				INT9EN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INT9 standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INT9 Clear input 0: Disable 1: Enable
	23	22	21	20	19	18	17	16
Bit Symbol			EMCGA1	EMCGA0				INTAEN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INTA standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INTA Clear input 0: Disable 1: Enable
	31	30	29	28	27	26	25	24
Bit Symbol			EMCGB1	EMCGB0				INTBEN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."	Always reads "0."	Set active state of INTB standby clear request. 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge		Always reads "0."	Always reads "0."	Always reads "0."	INTB Clear input 0: Disable 1: Enable

IMCGD (0xFFFF_EE1C)

	7	6	5	4	3	2	1	0
Bit Symbol			EMCGC1	EMCGC0				KWUPEN
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."		Set active state of KWUP standby clear request. 01: "H" level **Be sure to set "01."**		Always reads "0."			KWUP Clear input 0: Disable 1: Enable
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Symbol								
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0	0	0	0
Function	Always reads "0."		Always reads "1."	Always reads "0."				
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Symbol								
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0			0
Function	Always reads "0."		Always reads "1."	Always reads "0."				
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Symbol								
Read/Write	R		R/W		R			R/W
After Reset	0	0	1	0	0			0
Function	Always reads "0."		Always reads "1."	Always reads "0."				

Note: Default values for standby clearing request of IMCGD 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 or Idle 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 mode with TMP19A63, 13 factors, i.e., INT0 to INTB and KWUP0 to 7 are available as clearing interrupts. Whether or not INT0 to INTB are to be used as Stop/ Idle clearing interrupts as well as active state edge/level selection is set with CG.

(Note 4) Among the above 13 factors to be assigned as Stop/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 KWUP0 to 7 are to be used as normal interrupts, set the active level by KWUPSTn and set High level with INTC. No CG setting is necessary.

Interrupt factors other than those assigned as Stop/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: 0100: INT4 1001: INT9 1110: 1111:			
	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 13 factors that are assigned to clear Stop/ Idle modes,**

① **For KWUP, use KWUPS**

② **For INT0 to INTB, use the EICRCG register in the above CG block and then use the INTCLR register in theINTC block.**

6.11 NMI Flag Register

NMIFLG
(0xFFFF_EE24)

	7	6	5	4	3	2	1	0
Bit Symbol						NMI	WDT	WBER
Read/Write	R							
After Reset	0	0	0	0	0	0	0	0
Function	Always reads "0."					NMI factor 1: NMI generated by input from NMI pin	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."							

(Note) WDT and WBER are cleared to "0" when they are read.

6.12 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.

6.12.1 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>, the register bank will not be switched. The users need to program an additional process for saving the contents of the register.
- 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, the register bank will not be switched. The users need to program an additional process for saving the contents of the register..
- 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).

6.12.2 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 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 registers

Px :Port register

To read/ write port data.

PxCR :Output control register

Need to enable the input with PxIE register even when input is set.

PxFCn :Function register

To set functions. An assigned function can be activated by setting “1”.

PxOD :Open drain control register

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

PxPUP :Pull-up control register

To control program pull-ups.

PxSEL :Serial setting register

Needs to be set when using serial function.

PxIE :Input control enable register

To control inputs. “0”cannot be set as a default to avoid through current.

All the ports other than P0 and P1 require the setting. “1” is input if IE=”0”.

7.2 Port 0 (P00~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. Outputs can be set 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 output disable 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, all bits of P0FC1 are set to "1."

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

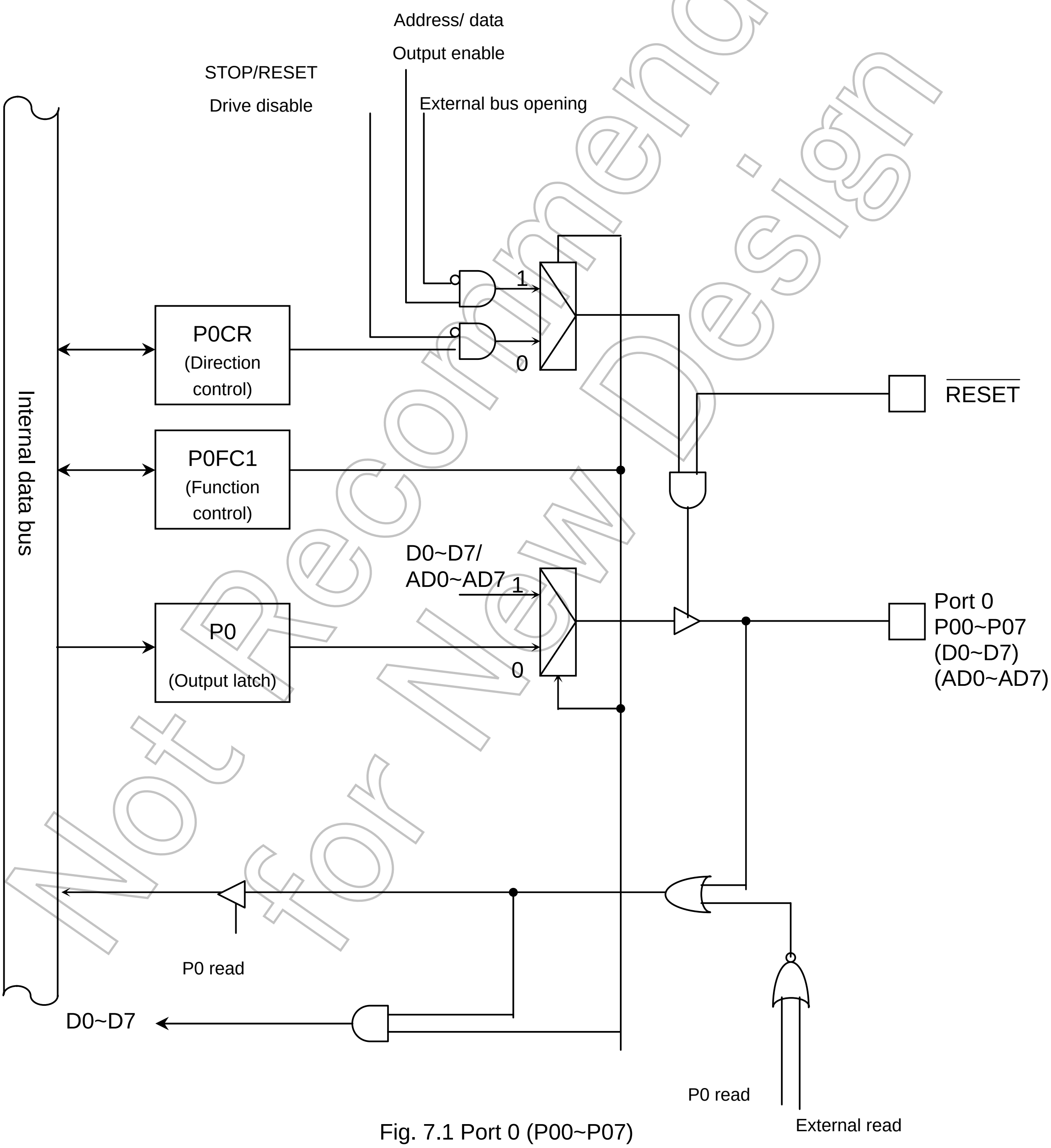


Fig. 7.1 Port 0 (P00~P07)

Port 0 register

P0 (0xFFFF_F000)

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

Port 0 control register

P0CR (0xFFFF_F001)

	7	6	5	4	3	2	1	0
Bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: Output disable 1: Output enable							

Port 0 function register 1

P0FC1 (0xFFFF_F002)

	7	6	5	4	3	2	1	0
Bit Symbol	P07FC1	P06FC1	P05FC1	P04FC1	P03FC1	P02FC1	P01FC1	P00FC1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: External bus setting							

7.3 Port 1 (P10~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. Output can be set by using the control register P1CR and the function registers P1FC1 and P1FC2. A reset allows all bits of the output latch P1, P1CR, P1FC1 and P1FC2 to be cleared to "0" and the port 1 to be put in output disable 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, P1FC1 and P1FC2 must be provisioned to allow the port 1 to function as either an address bus or an address data bus.

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

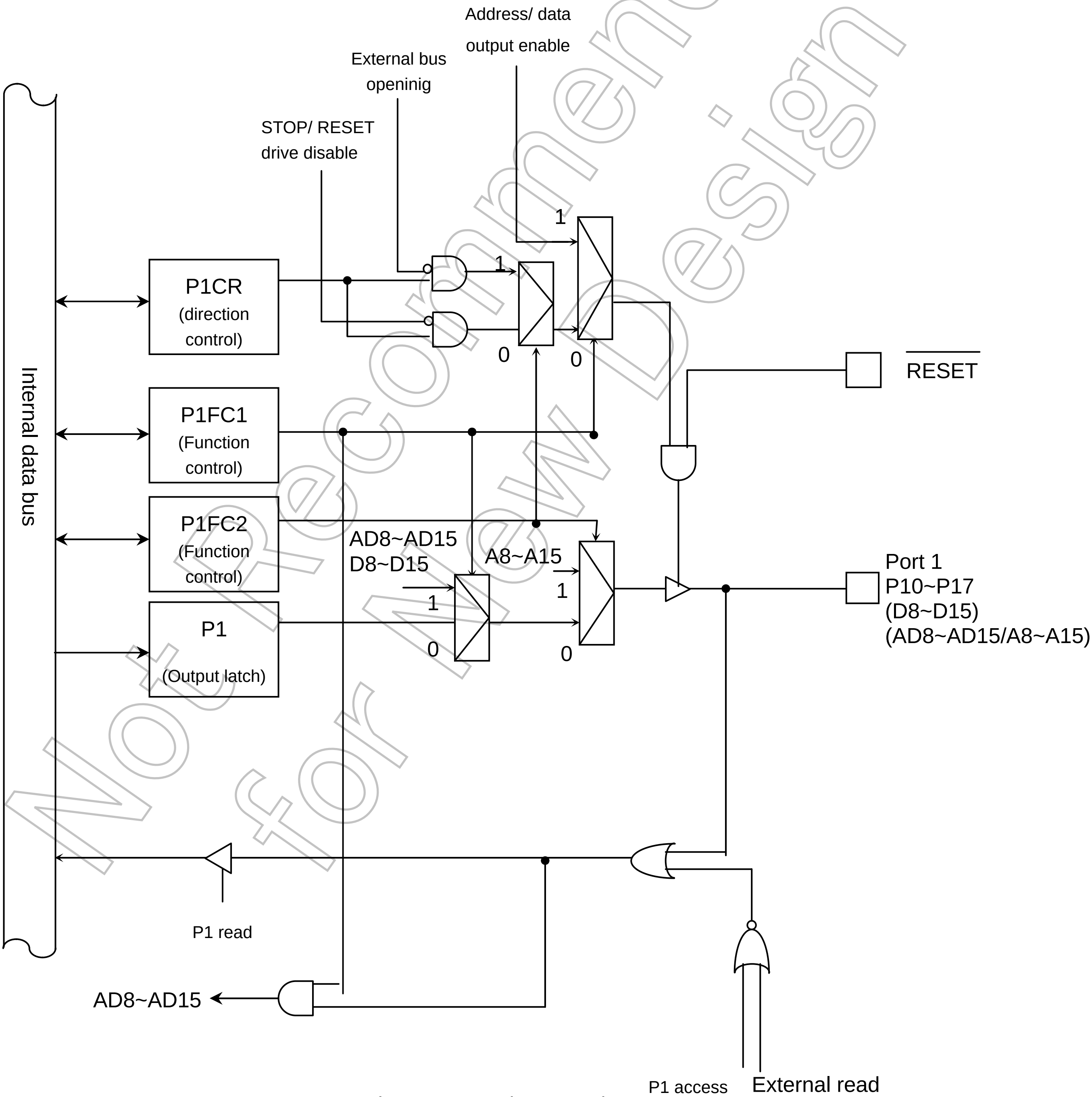


Fig. 7.2 Port 1 (P10~P17)

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_F011)

	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	0: output disable 1: output enable							

Port 1 function register 1

P1FC1 (0xFFFF_F012)

	7	6	5	4	3	2	1	0
Bit Symbol	P17F1	P16F1	P15F1	P14F1	P13F1	P12F1	P11F1	P10F1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: external bus setting (AD/D8~AD/D15)							

Port 1 function register2

P1FC2 (0xFFFF_F013)

	7	6	5	4	3	2	1	0
Bit Symbol	P17F2	P16F2	P15F2	P14F2	P13F2	P12F2	P11F2	P10F2
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: external bus setting (A8 ~A15)							

Note) You cannot set “1” to P1FC1 and F1FC2 simultaneously.

7.4 Port 2 (P20~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. Outputs can be set by using the control register P2CR and the function registers P2FC1 and P2FC2. A reset allows all bits of the output latch P2 to be set to "1," all bits of P2CR, P2FC1 and P2FC2 to be cleared to "0," and the port 2, to be an input port though it is input/output disabled.

Besides the general-purpose input/output port function, the port 2 performs another function: A0 through A7 function as an address bus and A16 through A23 function as another address bus. To access external memory, registers P2CR, P2FC1 and P2FC2 must be provisioned to allow the port 2 to function as an address bus.

If the BUSMD pin is set to "L" level during a reset, the port 2 is put in separate bus 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).

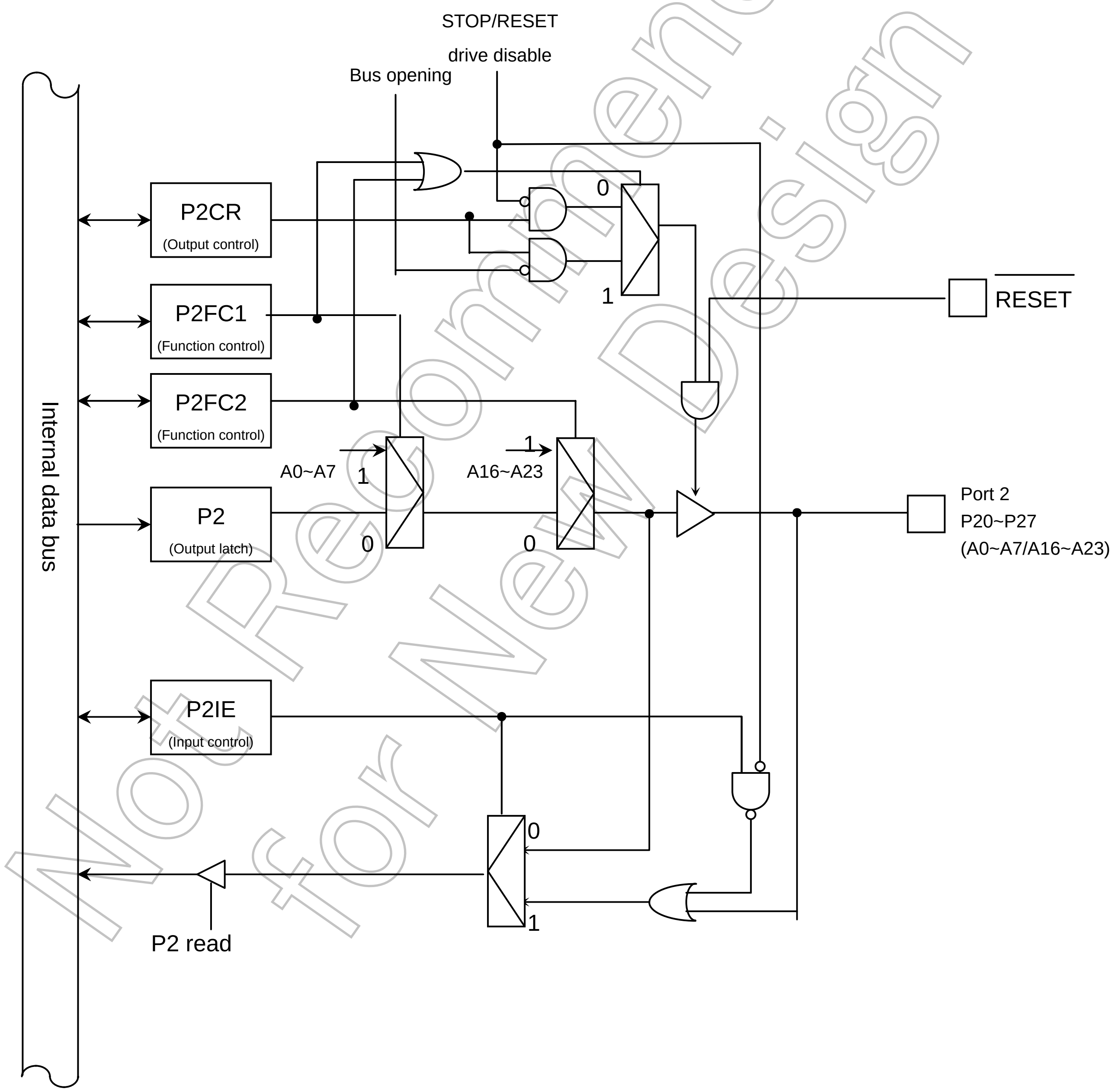


Fig. 7.3 Port 2(P20~P27)

Port 2 register

		7	6	5	4	3	2	1	0
P2	Bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
(0xFFFF_F020)	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port 2 control register

		7	6	5	4	3	2	1	0
P2CR	Bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
(0xFFFF_F021)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: output disable 1: output enable							

Port 2 function register 1

		7	6	5	4	3	2	1	0
P2FC1	Bit Symbol	P27F1	P26F1	P25F1	P24F1	P23F1	P22F1	P21F1	P20F1
(0xFFFF_F022)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: external bus setting (A0~A7)							

Port 2 Function register 2

		7	6	5	4	3	2	1	0
P2FC2	Bit Symbol	P27F2	P26F2	P25F2	P24F2	P23F2	P22F2	P21F2	P20F2
(0xFFFF_F023)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: PORT 1: external bus setting (A16~A23)							

Port 2 Input enable control register

		7	6	5	4	3	2	1	0
P2IE	Bit Symbol	P27IE	P26IE	P25IE	P24IE	P23IE	P22IE	P21IE	P20IE
(0xFFFF_F02E)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.5 Port 3 (P30~P37)

The port 3 is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register P3CR and the function register P3FC1.

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.

Set IE to input mode when using WAIT/BUSREQ input function.

-

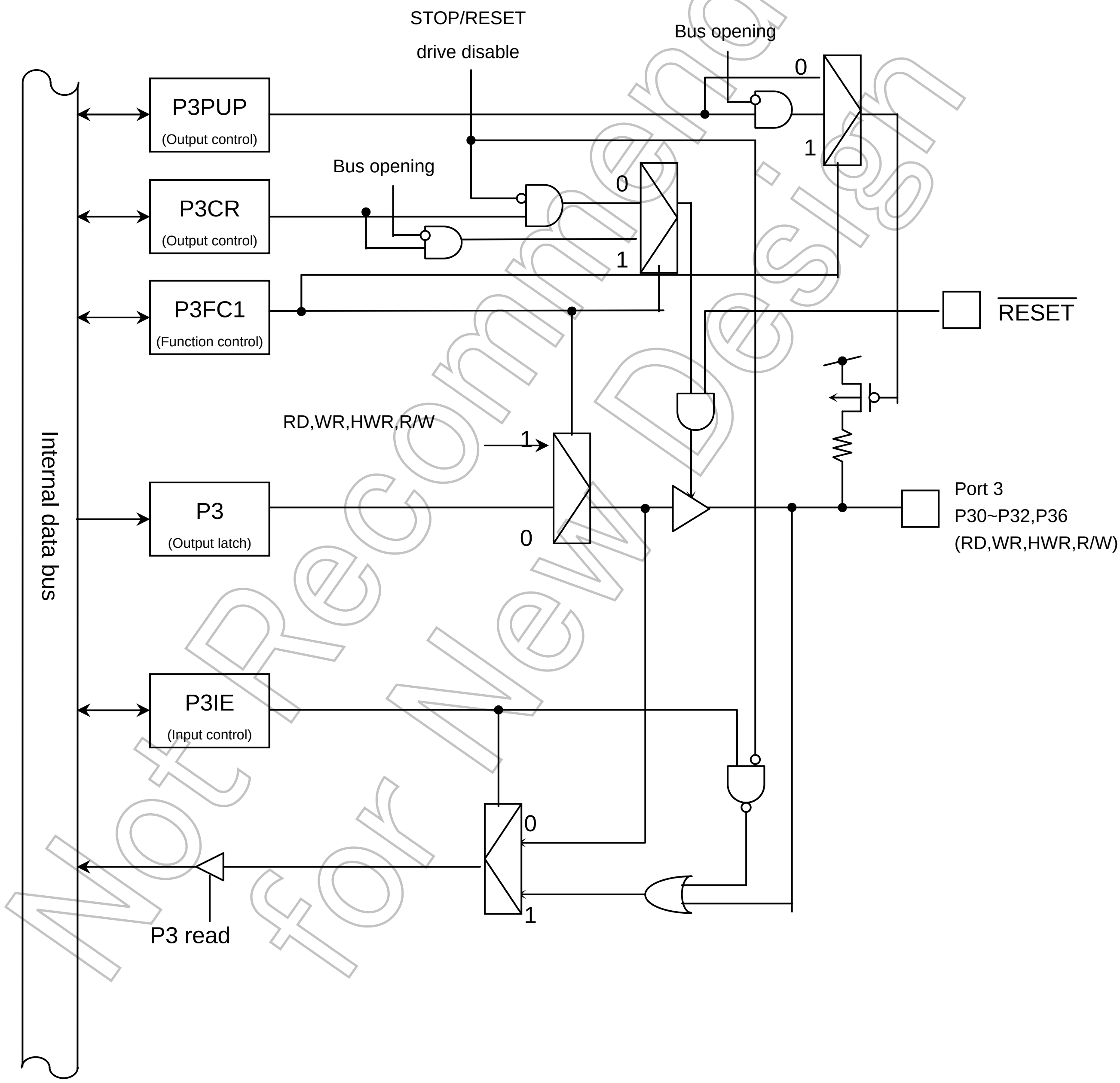


Fig. 7.4 Port 3 (P30~P32, P36)

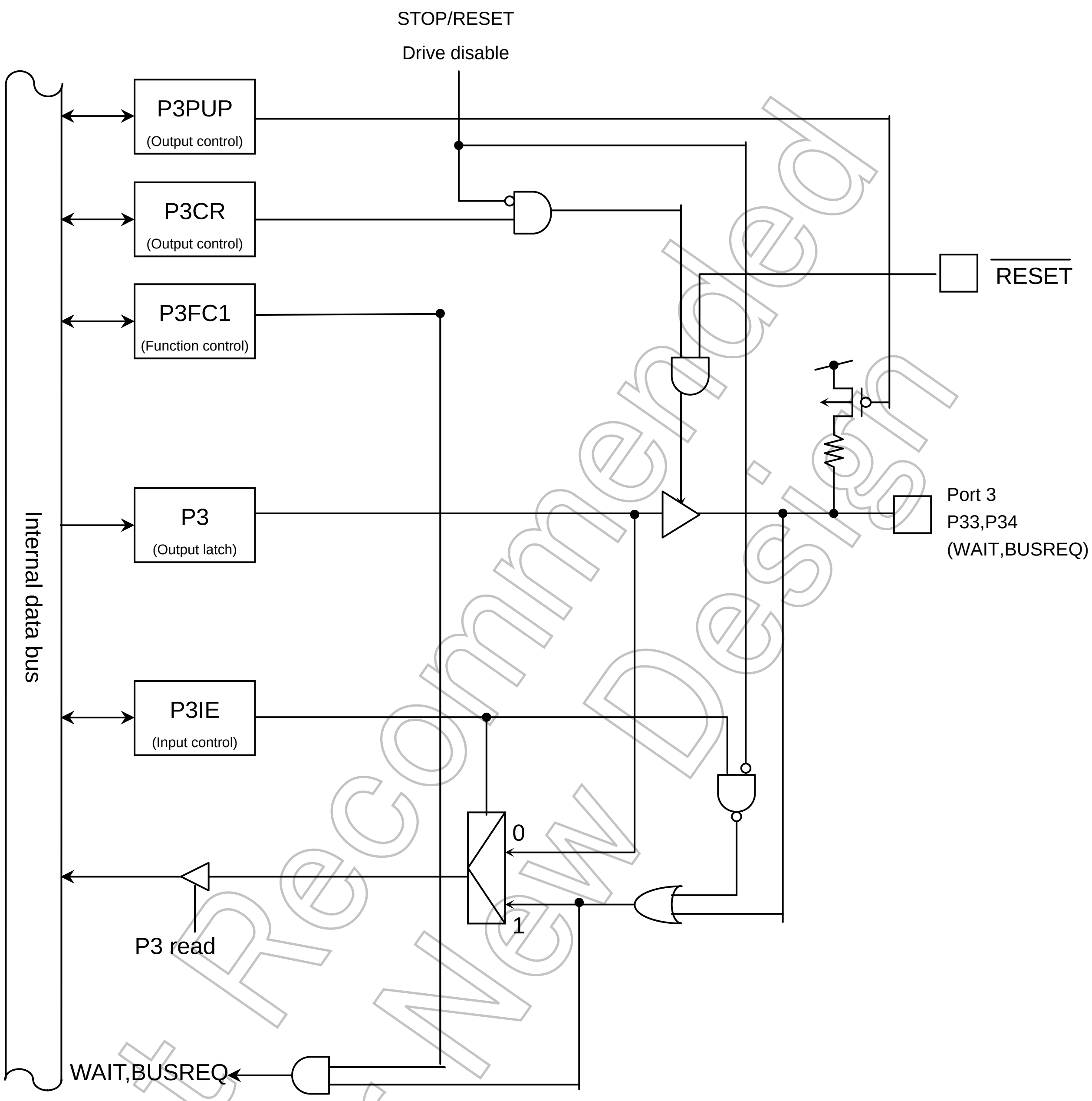


Fig. 7.5 Port 3 (P33,P34)

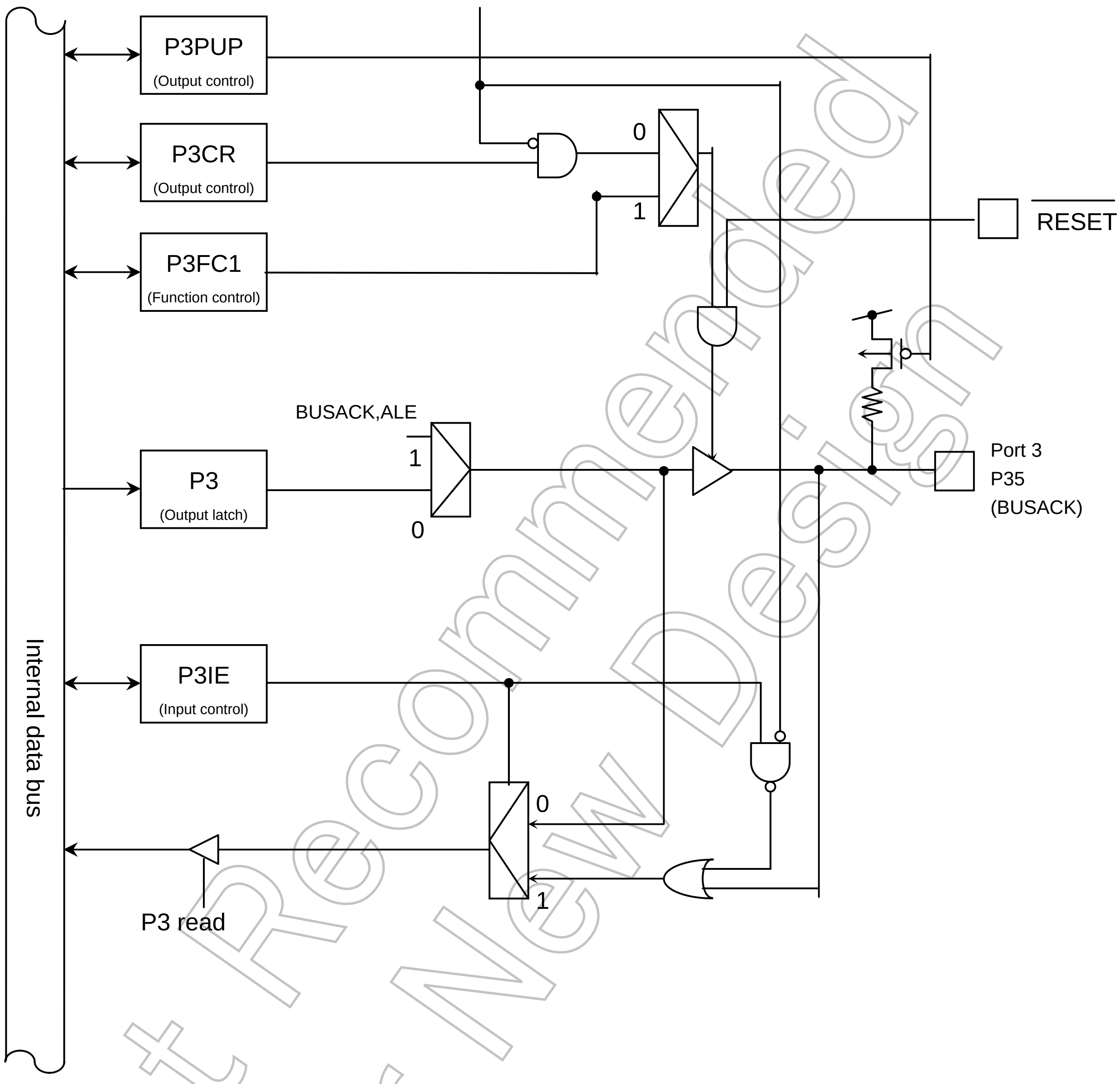


Fig. 7.6 Port 3 (P35)

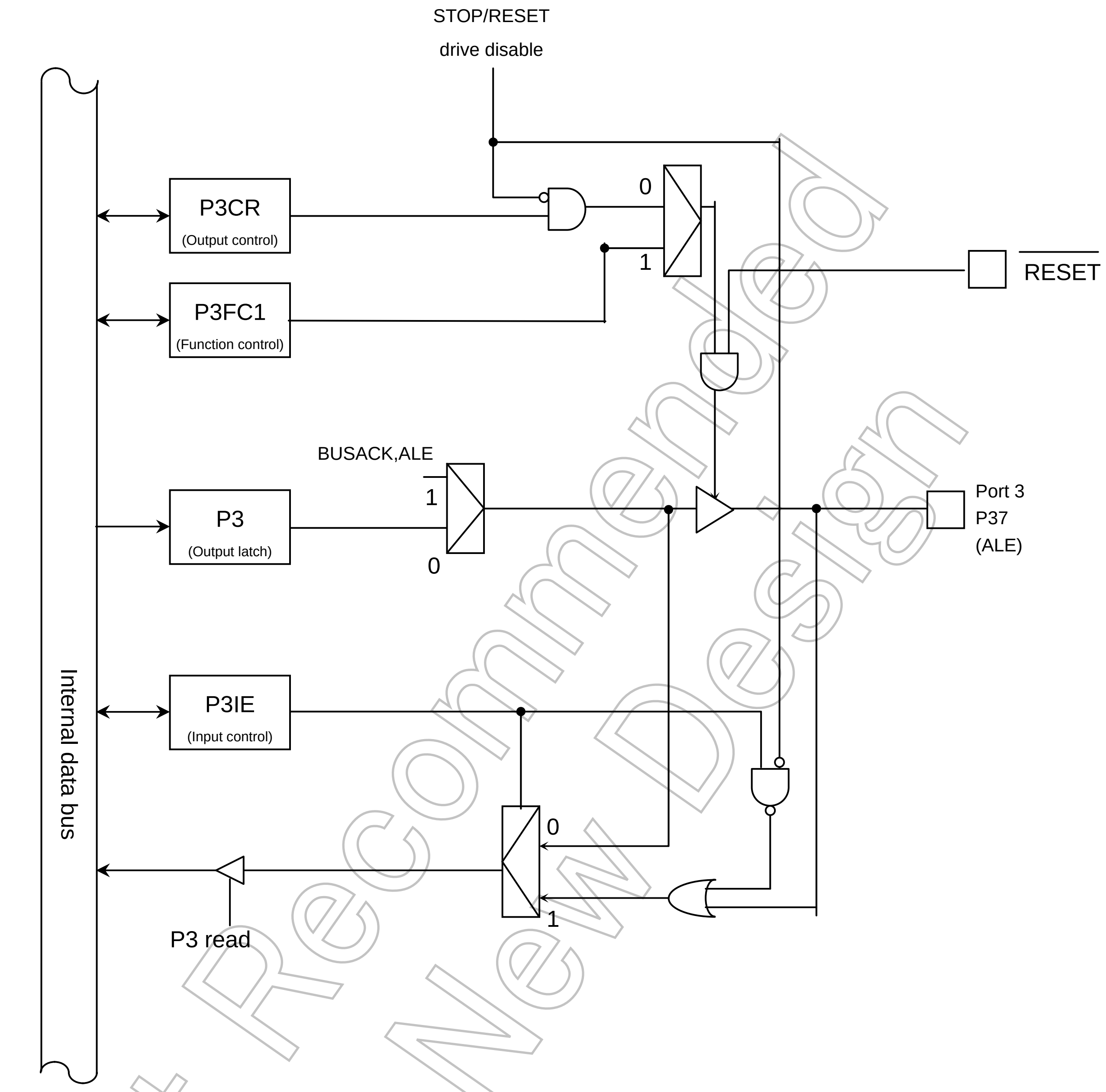


Fig. 7.7 Port 3 (P37)

Port 3 register

P3 (0xFFFF_F030)		7	6	5	4	3	2	1	0
	Bit Symbol	P37	P36	P35	P34	P33	P32	P31	P30
	Read/Write	R/W							
	After reset	0	Input mode (output latch register is set to “1.”)						

Port 3 control register

P3CR (0xFFFF_F031)		7	6	5	4	3	2	1	0
	Bit Symbol	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: output disable 1: output enable							

Port 3 function register 1

P3FC1 (0xFFFF_F032)		7	6	5	4	3	2	1	0
	Bit Symbol	P37F1	P36F1	P35F1	P34F1	P33F1	P32F1	P31F1	P30F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:ALE	0:PORT 1R/W	0:PORT 1:BUSACK	0:PORT 1:BUSREQ	0:PORT /WAIT 1:RDY	0:PORT 1:HWR	0:PORT 1:WR	0:PORT 1:RD

Port 3 Pull-up control register

P3PUP (0xFFFF_F03B)		7	6	5	4	3	2	1	0
	Bit Symbol	P37UP	P36UP	P35UP	P34UP	P33UP	P32UP	P31UP	P30UP
	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

Port 3 Input enable control register

P3IE (0xFFFF_F03E)		7	6	5	4	3	2	1	0
	Bit Symbol	P37IE	P36IE	P35IE	P34IE	P33IE	P32IE	P31IE	P30IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.6 Port 4 (P40~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. Outputs can be set by using the control register P4CR and the function register P4FC1.

Besides the general-purpose input/output port function, the port 4 performs other functions: P40 through P45 output the chip select signal ($\overline{CS0}$ to $\overline{CS5}$), P46 functions as the SCOUT output pin for outputting internal clocks.

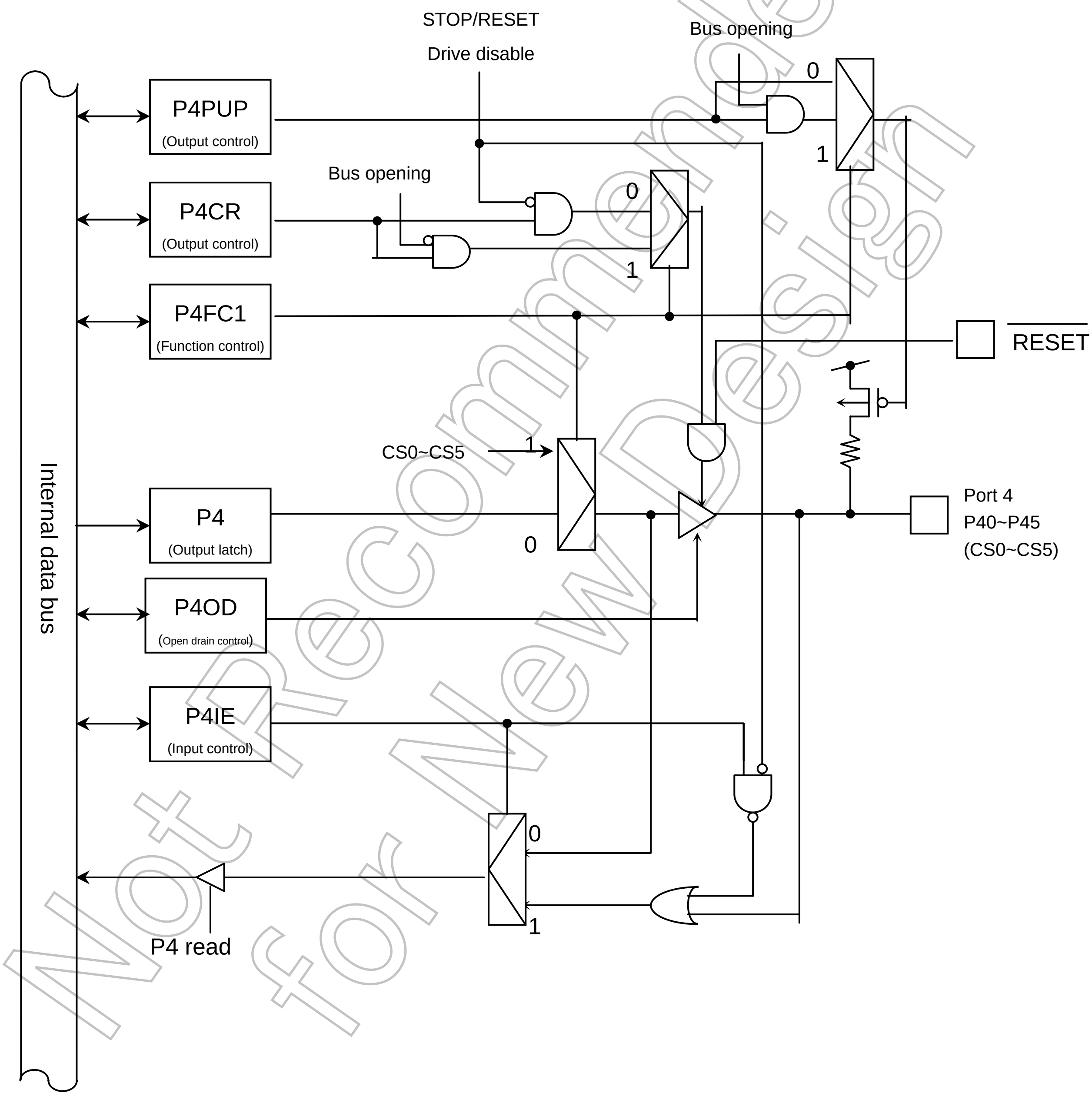


Fig. 7.8 Port 4 (P40~P45)

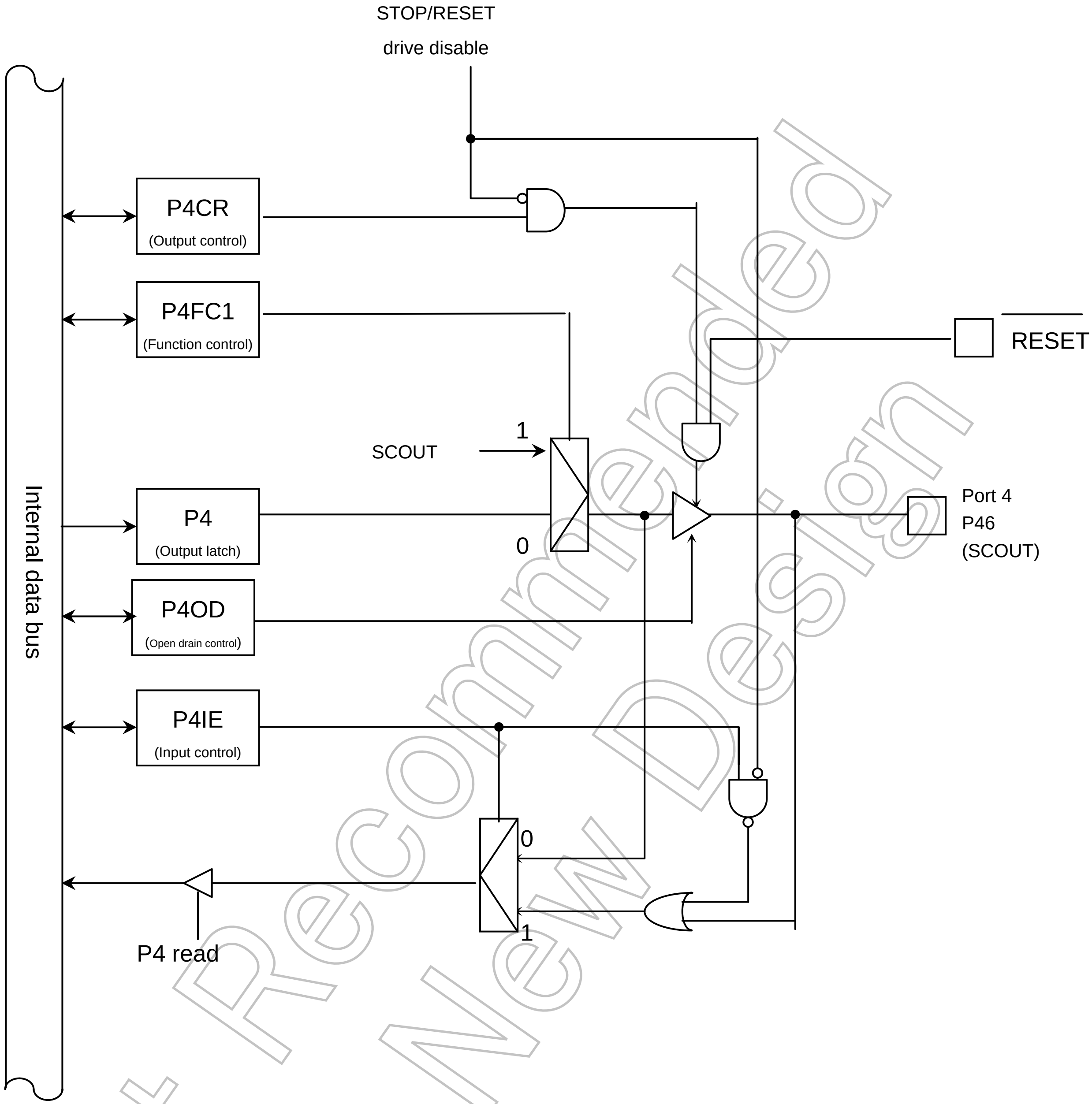


Fig. 7.9 Port 4 (P46)

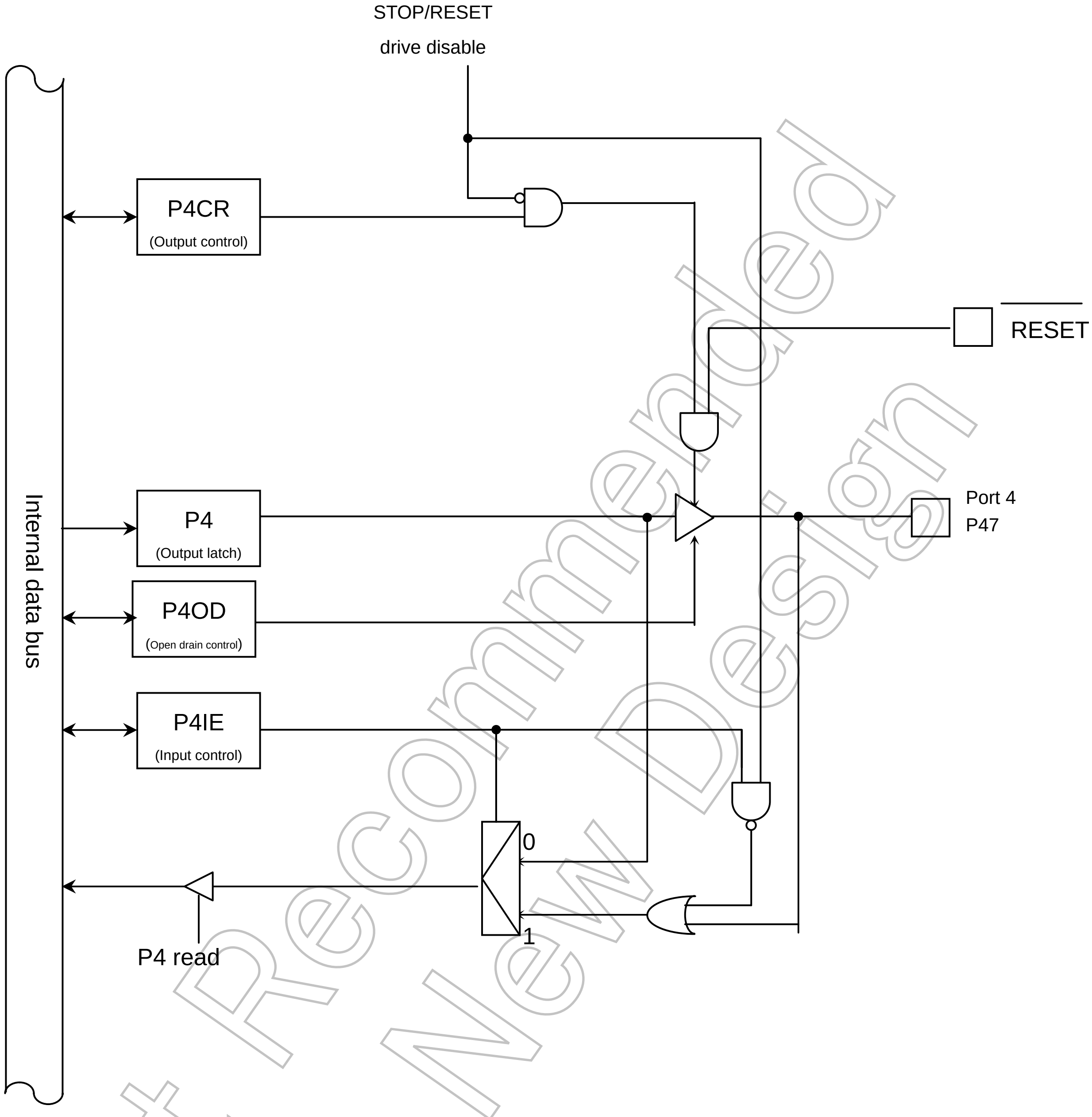


Fig. 7.10 Port 4 (P47)

Port 4 register

P4 (0xFFFF_F040)

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

Port 4 control register

P4CR (0xFFFF_F041)

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

Port 4 function register 1

P4FC1 (0xFFFF_F042)

	7	6	5	4	3	2	1	0
Bit Symbol	P47F1	P46F1			P43F1	P42F1	P41F1	P40F1
Read/Write	R/W		R/W		R/W			
After reset	0	0	0	0	0	0	0	0
Function		0: PORT 1: SCOUT	0: PORT 1: CS5	0: PORT 1: CS4	0: PORT 1: CS3	0: PORT 1: CS2	0: PORT 1: CS1	0: PORT 1: CS0

Port 4 Pull-up control register

P4PUP (0xFFFF_F04B)

	7	6	5	4	3	2	1	0
Bit Symbol	P47UP	P46UP	P45UP	P44UP	P43UP	P42UP	P41UP	P40UP
Read/Write			R/W					
After reset			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

Port 4 Input enable control register

P4IE (0xFFFF_F04E)

	7	6	5	4	3	2	1	0
Bit Symbol	P47IE	P46IE	P45IE	P44IE	P43IE	P42IE	P41IE	P40IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled

7.6 Port 5 (P50~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. Outputs can be set by using the function register P5FC1 and the control register P5CR. A reset allows all bits of the output latch P5 to be set to "1," all bits of P5CR and P5FC1 to be cleared to "0," and the port 5 to be put in output disable mode.

The port 5 also functions as an address bus (A0 through A7). To access external memory, P5CR and P5FC1 must be provisioned to allow the port 5 to function as an address bus.

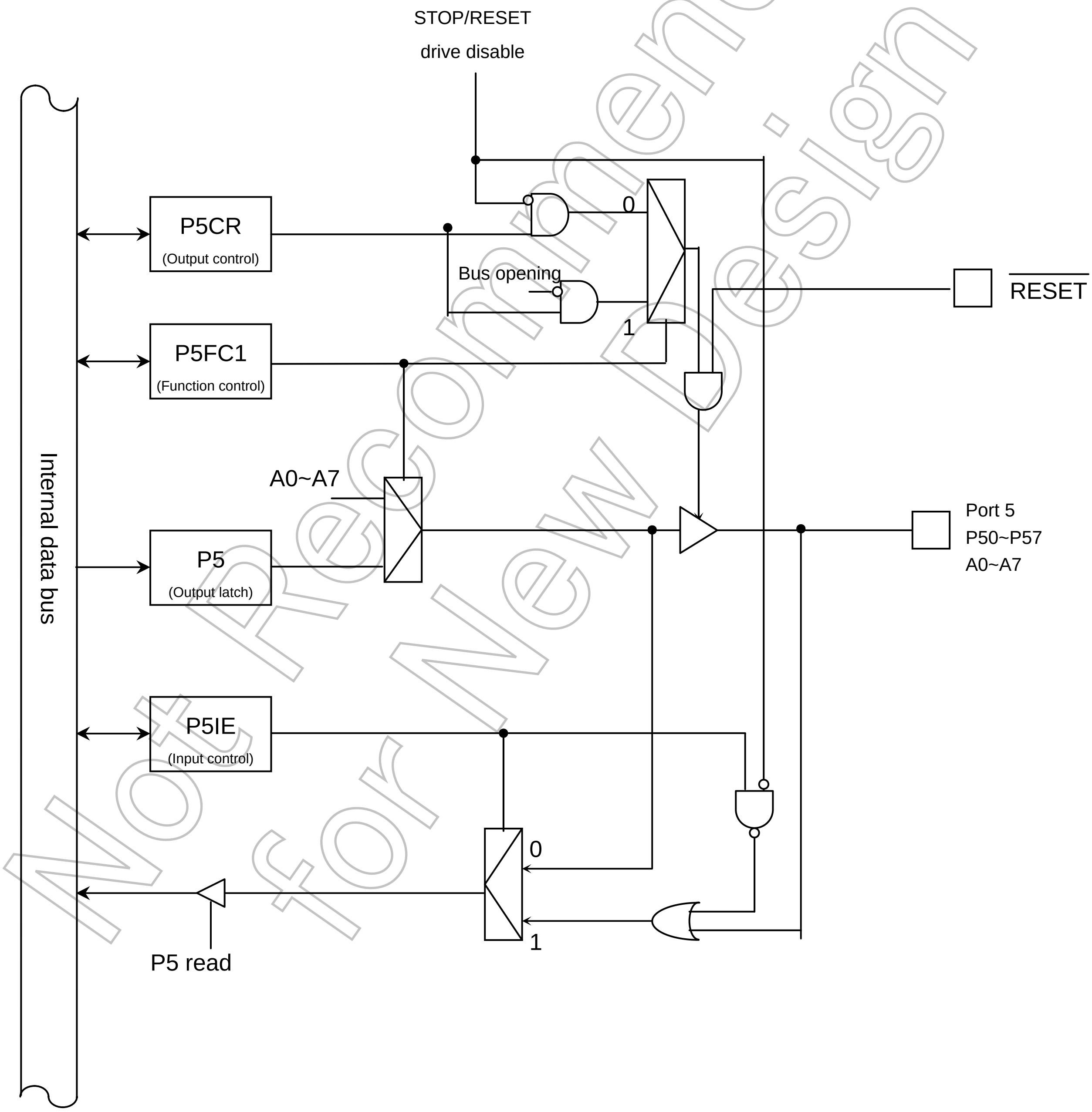


Fig. 7.11 Port 5 (P50~P57)

Port 5 register

P5 (0xFFFF_F050)

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

Port 5 control register

P5CR (0xFFFF_F051)

	7	6	5	4	3	2	1	0
Bit Symbol	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: output disable 1: output enable							

Port 5 function register 1

P5FC1 (0xFFFF_F052)

	7	6	5	4	3	2	1	0
Bit Symbol	P57F1	P56F1	P55F1	P54F1	P53F1	P52F1	P51F1	P50F1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: external bus setting (A0~A7)							

Port 5 input enable control register

P5IE (0xFFFF_F05E)

	7	6	5	4	3	2	1	0
Bit Symbol	P57IE	P56IE	P55IE	P54IE	P53IE	P52IE	P51IE	P50IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled	Input 0: disabled 1:enabled

7.7 Port 6 (P60~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. Outputs can be set by using the function register P6FC1 and the control register P6CR. A reset allows all bits of the output latch P6 to be set to "1," all bits of P6CR and P6FC1 to be cleared to "0," and the port 6 to be put in output disable mode.

The port 6 also functions as an address bus (A8 through A15). To access external memory, P6CR and P6FC1 through P6FC4 must be provisioned to allow the port 6 to function as an address bus.

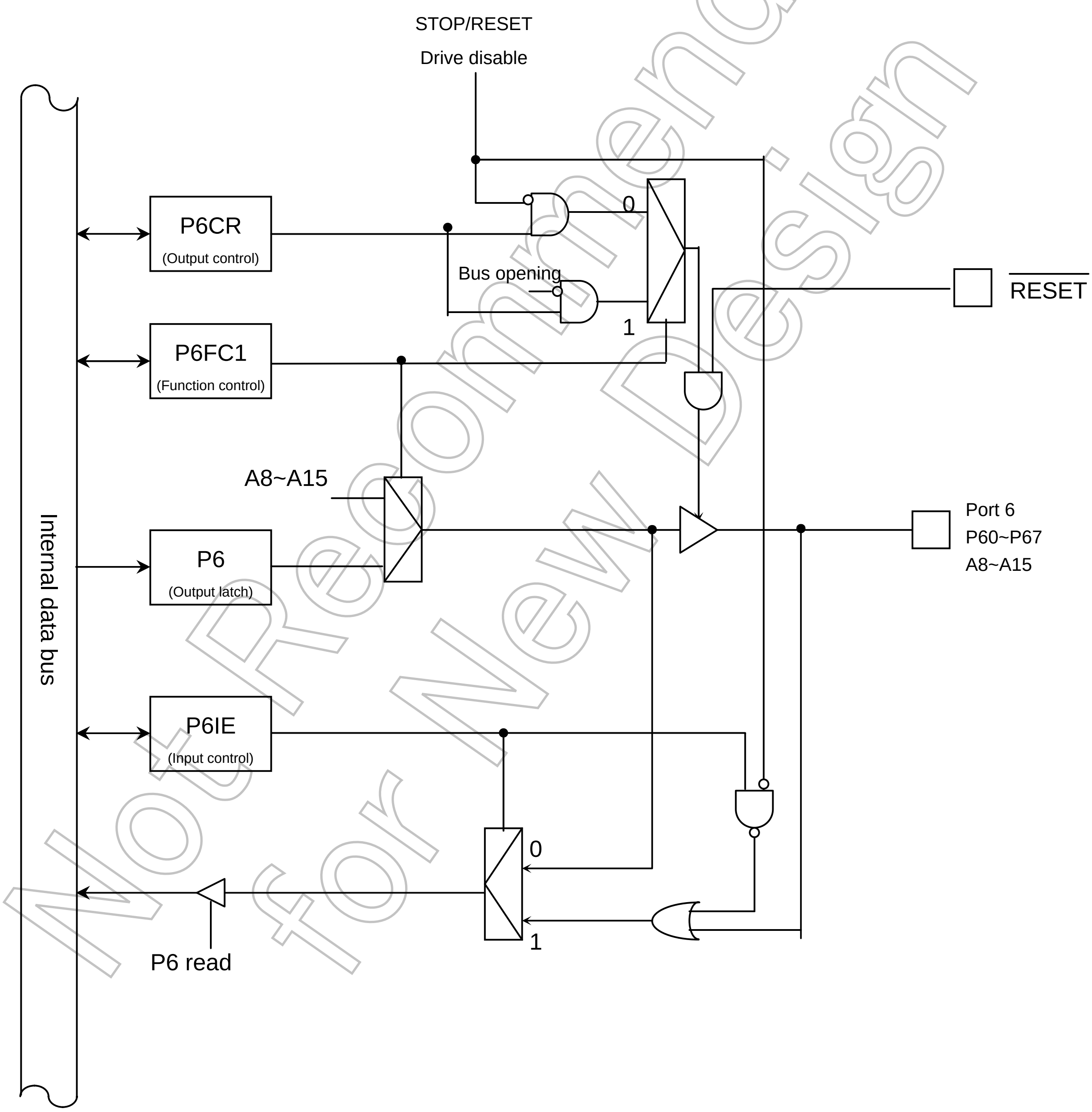


Fig. 7.12 Port 6 (P60~P67)

Port 6 register

P6 (0xFFFF_F060)

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

Port 6 control register

P6CR (0xFFFF_F061)

	7	6	5	4	3	2	1	0
Bit Symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: output disable 1: output enable							

Port 6 function register 1

P6FC1 (0xFFFF_F062)

	7	6	5	4	3	2	1	0
Bit Symbol	P67F1	P66F1	P65F1	P64F1	P63F1	P62F1	P61F1	P60F1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: PORT 1: external bus setting (A8~A15)							

Port 6 input enable control register

P6IE (0xFFFF_F06E)

	7	6	5	4	3	2	1	0
Bit Symbol	P67IE	P66IE	P65IE	P64IE	P63IE	P62IE	P61IE	P60IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled	Input 0:disabled 1: enabled

7.8 Port 7 (P70~P77)

The port 7 is an 8-bit, analog input port for the A/D converter. Although the port 7 is an input port during a reset, any inputs are disabled.

Set the corresponding input enable control register when you use the port 7 as an input port.

Set the register to be input disabled when you use it as an AD function port.

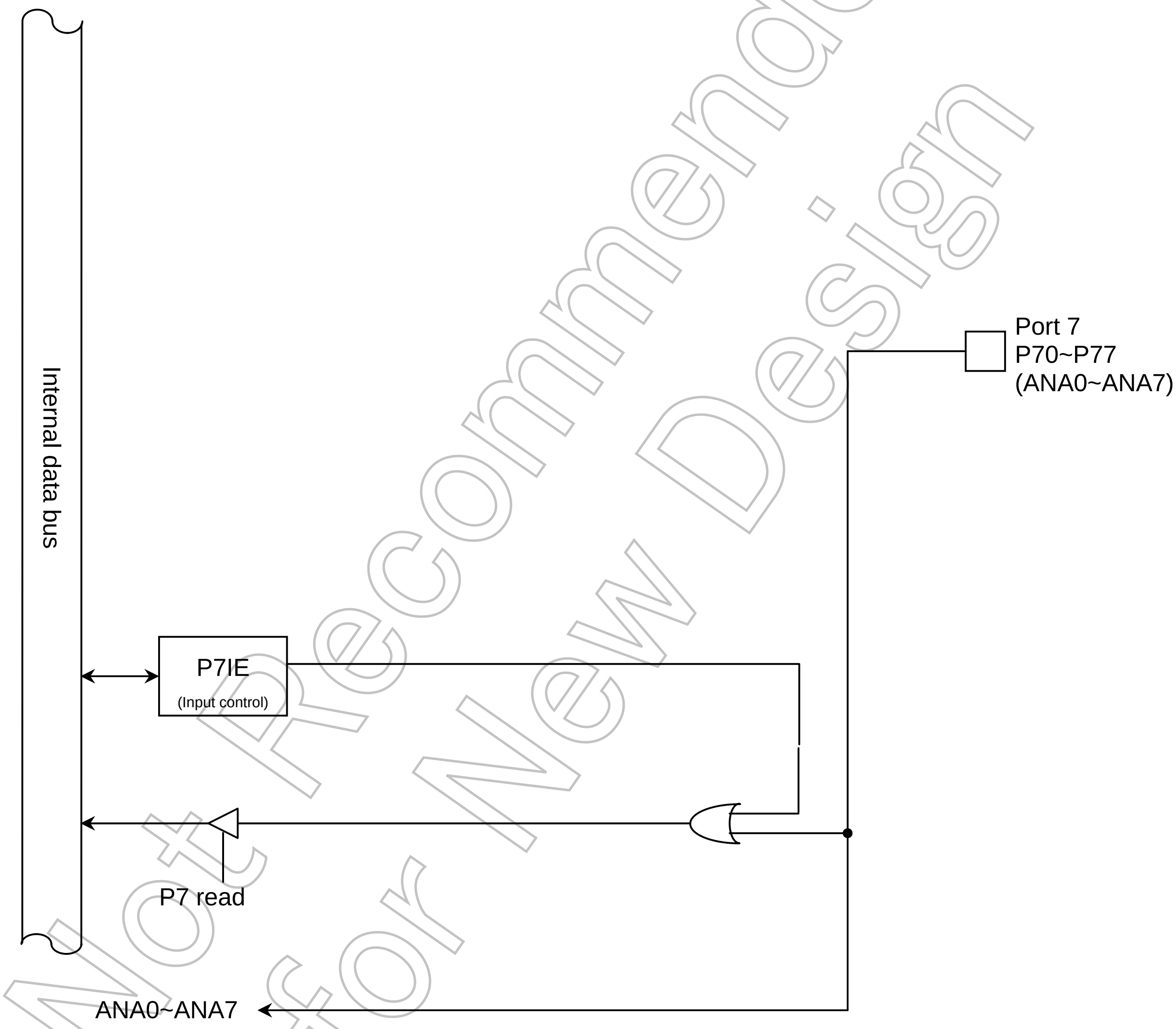


Fig. 7.13 Port 7 (P70~P77)

Port 7 register

P7 (0xFFFF_F070)		7	6	5	4	3	2	1	0
	Bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
	Read/Write	R							
	After reset	Pin condition can be read.							

Port 7 Input enable control register

P7IE (0xFFFF_F07E)		7	6	5	4	3	2	1	0
	Bit Symbol	P77IE	P76IE	P75IE	P74IE	P73IE	P72IE	P71IE	P70IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled	Input 0:disabled 1:enabled

7.9 Port 8 (P80~P87)

The port 8 is an 8-bit, analog input port for the A/D converter.

Although the port 8 is an input port during a reset, any inputs are disabled. Set the corresponding input enable control register when you use the port 8 as an input port.

Set the register to be input disabled when you use it as an AD function port.

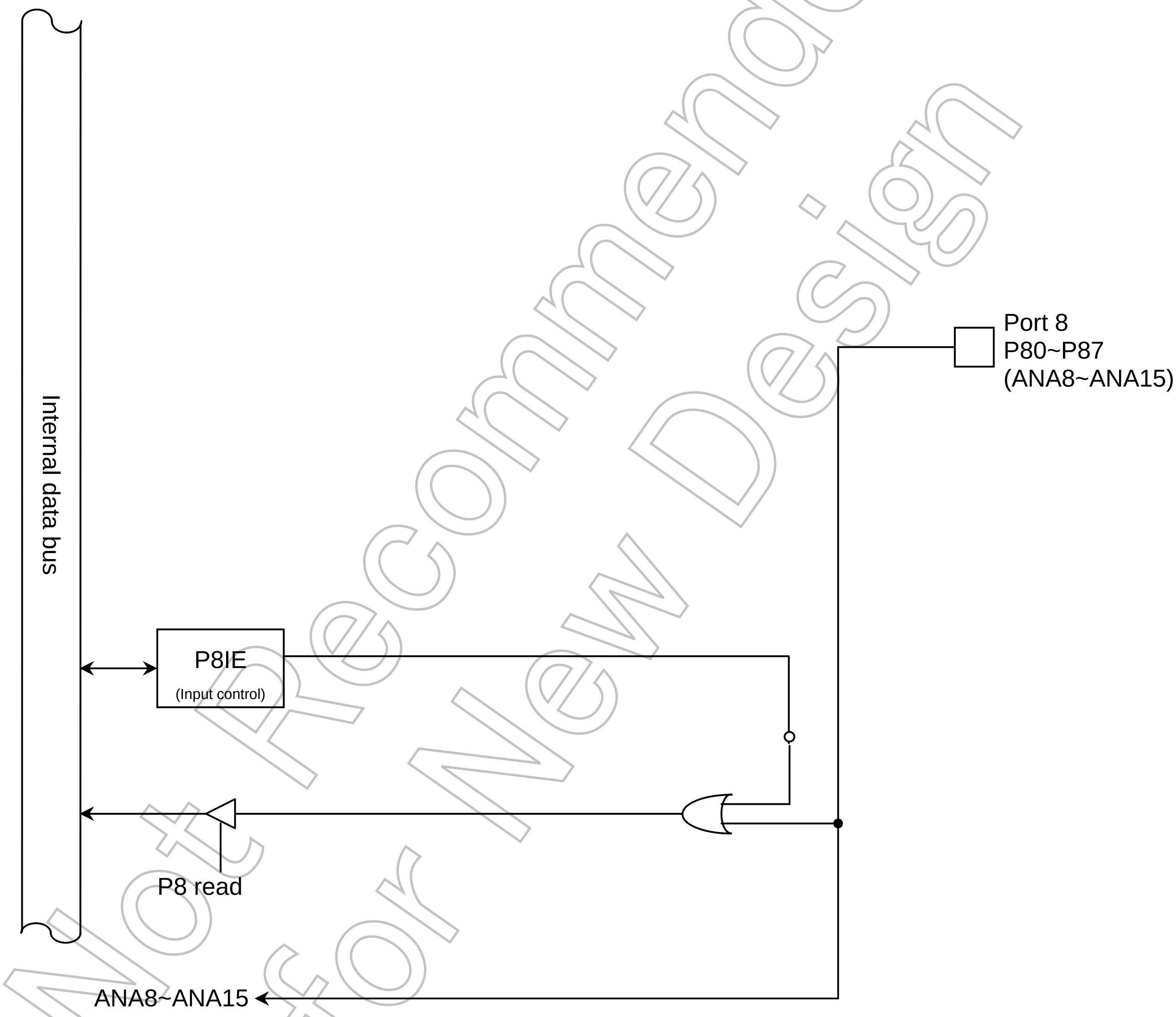


Fig. 7.14 Port 8 (P80~P87)

Port 8 register

P8 (0xFFFF_F080)		7	6	5	4	3	2	1	0
	Bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
	Read/Write	R							
	After reset	Pin condition can be read.							

Port 8 Input enable control register

P8IE (0xFFFF_F08E)		7	6	5	4	3	2	1	0
	Bit Symbol	P87IE	P86IE	P85IE	P84IE	P83IE	P82IE82	P81IE	P80IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.10 Port 9 (P90~P97)

The port 9 is an 8-bit, analog input port for the A/D converter.

Although the port 9 is an input port during a reset, any inputs are disabled. Set the corresponding input enable control register when you use the port 9 as an input port.

Set the register to be input disabled when you use it as an AD function port.

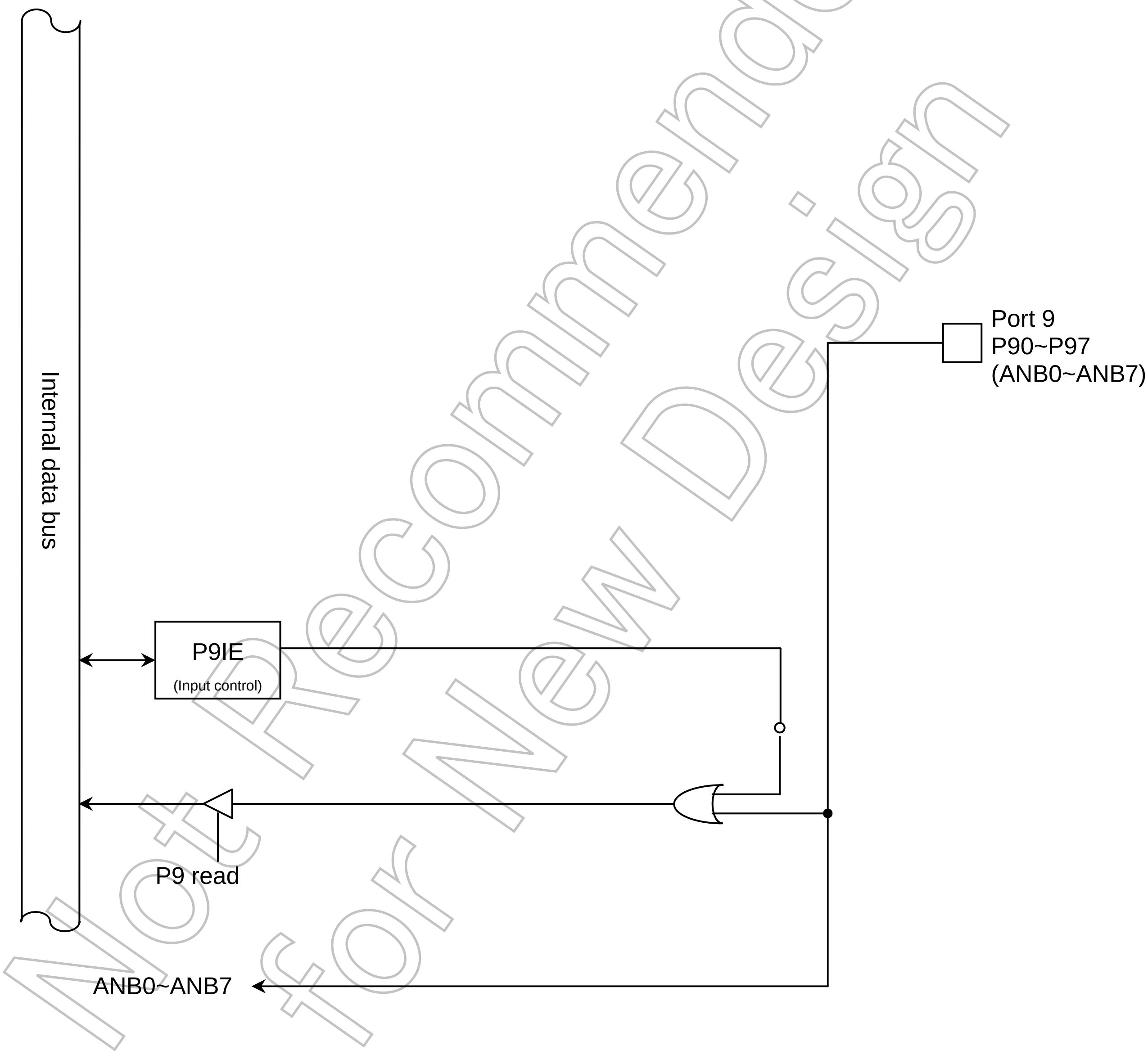


Fig. 7.15 Port 9 (P90~P97)

Port 9 register

		7	6	5	4	3	2	1	0
P9 (0xFFFF_F090)	Bit Symbol	P97	P96	P95	P94	P93	P92	P91	P90
	Read/Write	R							
	After reset	Pin condition can be read.							

Port 9 Input enable control register

		7	6	5	4	3	2	1	0
P9IE (0xFFFF_F09E)	Bit Symbol	P97IE	P96IE	P95IE	P94IE	P93IE	P92IE	P91IE	P90IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.11 Port A (PA0~PA7)

The port A is an 8-bit, analog input port for the A/D converter.

Although the port A is an input port during a reset, any inputs are disabled. Set the corresponding input enable control register when you use the port A as an input port.

Set the register to be input disabled when you use it as an AD function port.

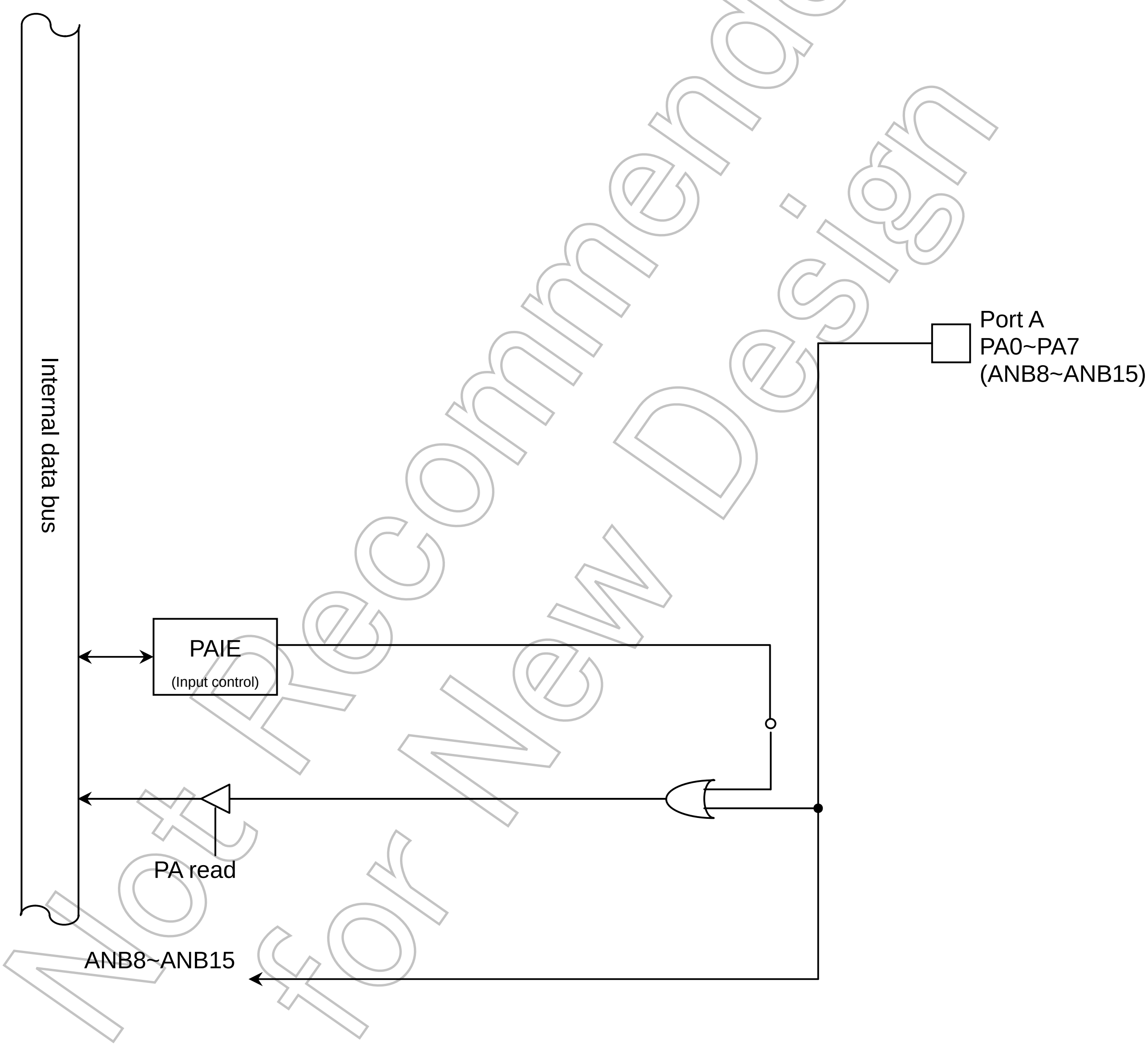


Fig. 7.16 Port A (PA8~PA15)

Port A register

PA (0xFFFF_F0A0)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Read/Write	R							
	After reset	Pin condition can be read.							

Port A Input enable control register

PAIE (0xFFFF_F0AE)		7	6	5	4	3	2	1	0
	Bit Symbol	PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.12 Port B (PB0~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. Outputs can be set by using the function register PBFC1 and the control register PBCR. A reset allows all bits of the output latch PB to be set to "0," all bits of PBCR and PBFC1 to be cleared to "0," and the port B to be put in output disable mode.

The port 6 also has 16-bit timer input function (PB0 through PB7).

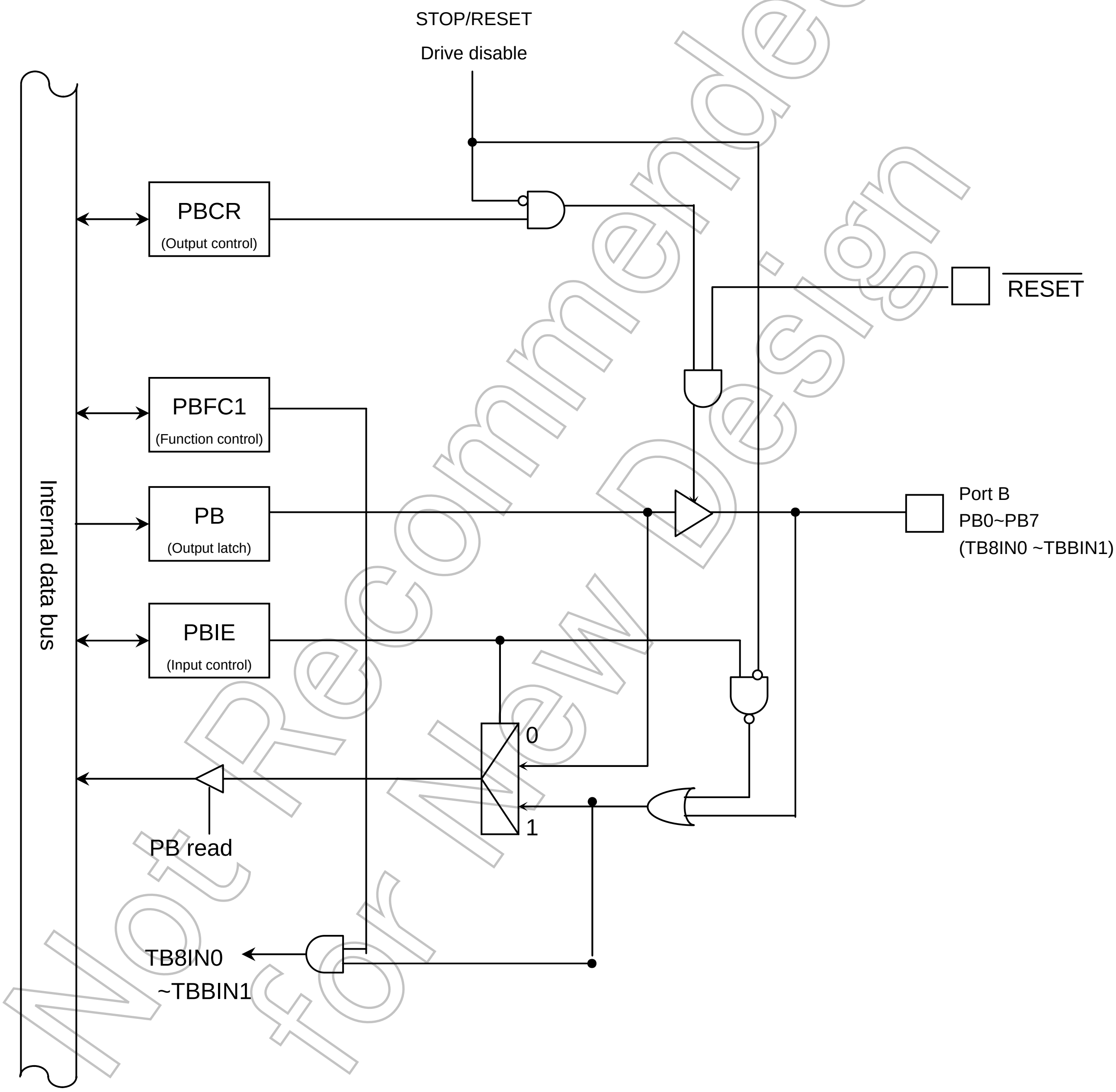


Fig. 7.17 Port B (PB0~PB7)

Port B register

PB (0xFFFF_F0B0)		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_F0B1)		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: output disable 1: output enable							

Port B function register 1

PBFC1 (0xFFFF_F0B2)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7F1	PB6F1	PB5F1	PB4F1	PB3F1	PB2F1	PB1F1	PB0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:TBBIN1	0:PORT 1:TBBIN0	0:PORT 1:TBAIN1	0:PORT 1:TBAIN0	0:PORT 1:TB9IN1	0:PORT 1:TB9IN0	0:PORT 1:TB8IN1	0:PORT 1: TB8IN0

Port B Input enable control register

PBIE (0xFFFF_F0BE)		7	6	5	4	3	2	1	0
	Bit Symbol	PB7IE	PB6IE	PB5IE	PB4IE	PB3IE	PB2IE	PB1IE	PB0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.13 Port C (PC0~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. Outputs can be set by using the function register PCFC and the control register PCCR. A reset allows all bits of the output latch PC, PCCR and PCFC1 to be cleared to "0," and the port C to be put in output disable mode.

The port 6 also has 16-bit timer input function (PC0 through PC7).

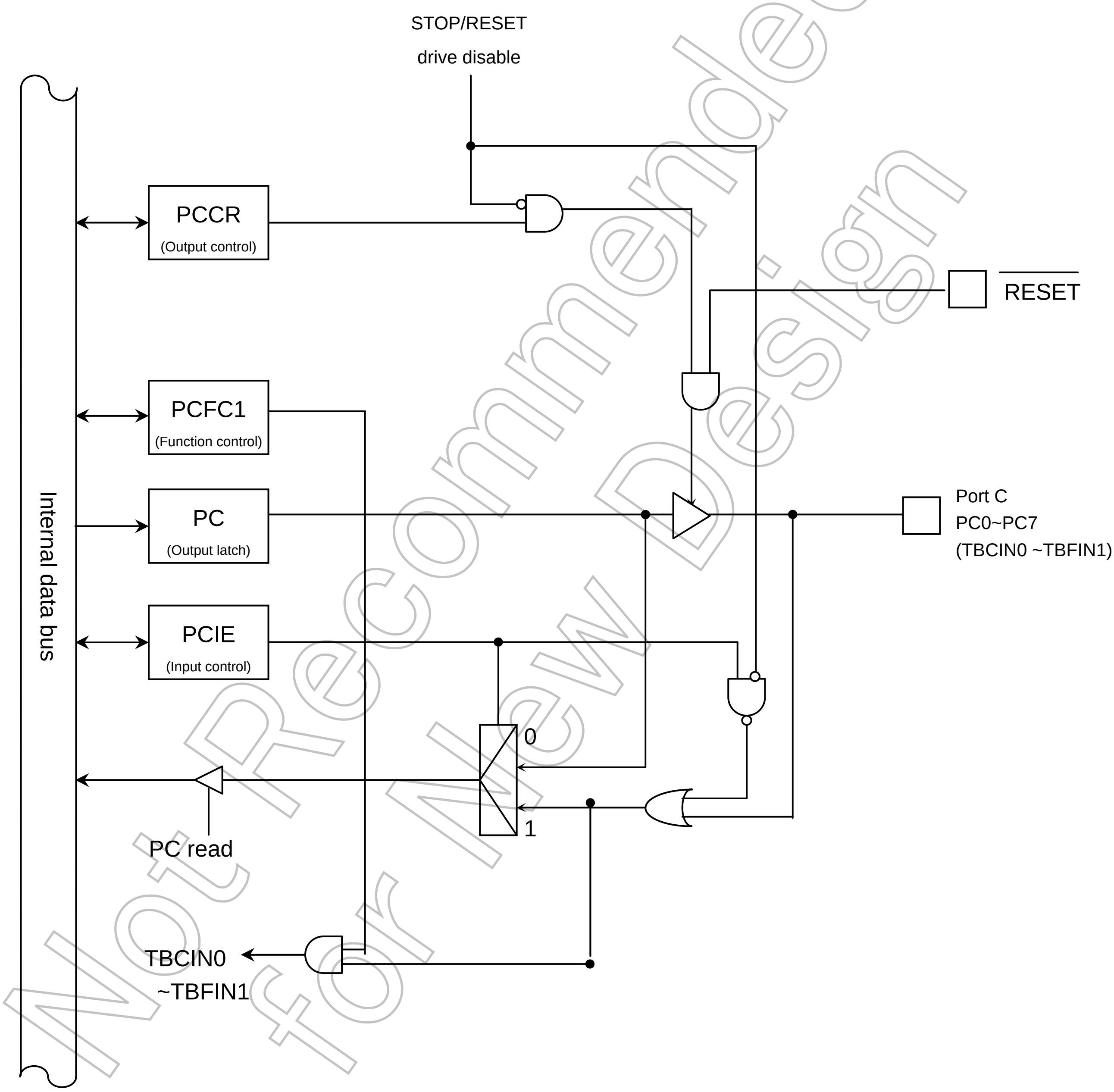


Fig. 7.18 Port C (PC0~PC7)

Port C register

PC (0xFFFF_F0C0)		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_F0C1)		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: output disable 1: output enable							

Port C function register 1

PCFC1 (0xFFFF_F0C2)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7F1	PC6F1	PC5F1	PC4F1	PC3F1	PC2F1	PC1F1	PC0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:TBFIN1	0:PORT 1:TBFIN0	0:PORT 1:TBEIN1	0:PORT 1:TBEIN0	0:PORT 1:TBDIN1	0:PORT 1:TBDIN0	0:PORT 1:TBCIN1	0:PORT 1: TBCIN0

Port C input enable control register

PCIE (0xFFFF_F0CE)		7	6	5	4	3	2	1	0
	Bit Symbol	PC7IE	PC6IE	PC5IE	PC4IE	PC3IE	PC2IE	PC1IE	PC0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.14 Port D (PD0~PD7)

Port D is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the function register PDFC1 and the control register PDCR. A reset allows all bits of the output latch PD to be set to "0," all bits of PDCR and PCFC1 to be cleared to "0," and the port D to be put in output disable mode.

Besides the input/output port function, the port D performs other functions: PD0 through PD5 input a 16-bit timer and PD6 and PD7 output a 16-bit timer.

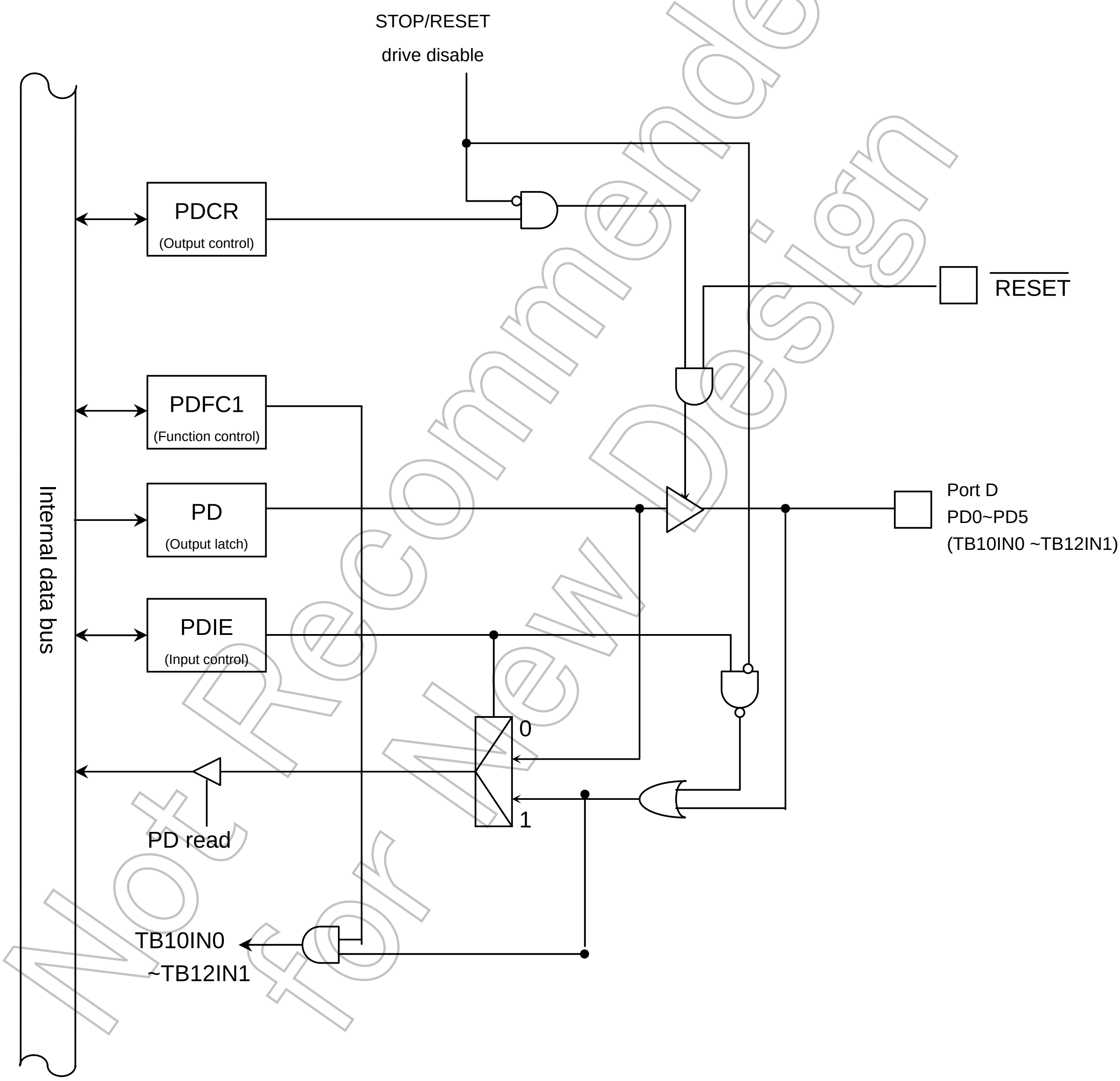


Fig. 7.19 Port D (PD0~PD5)

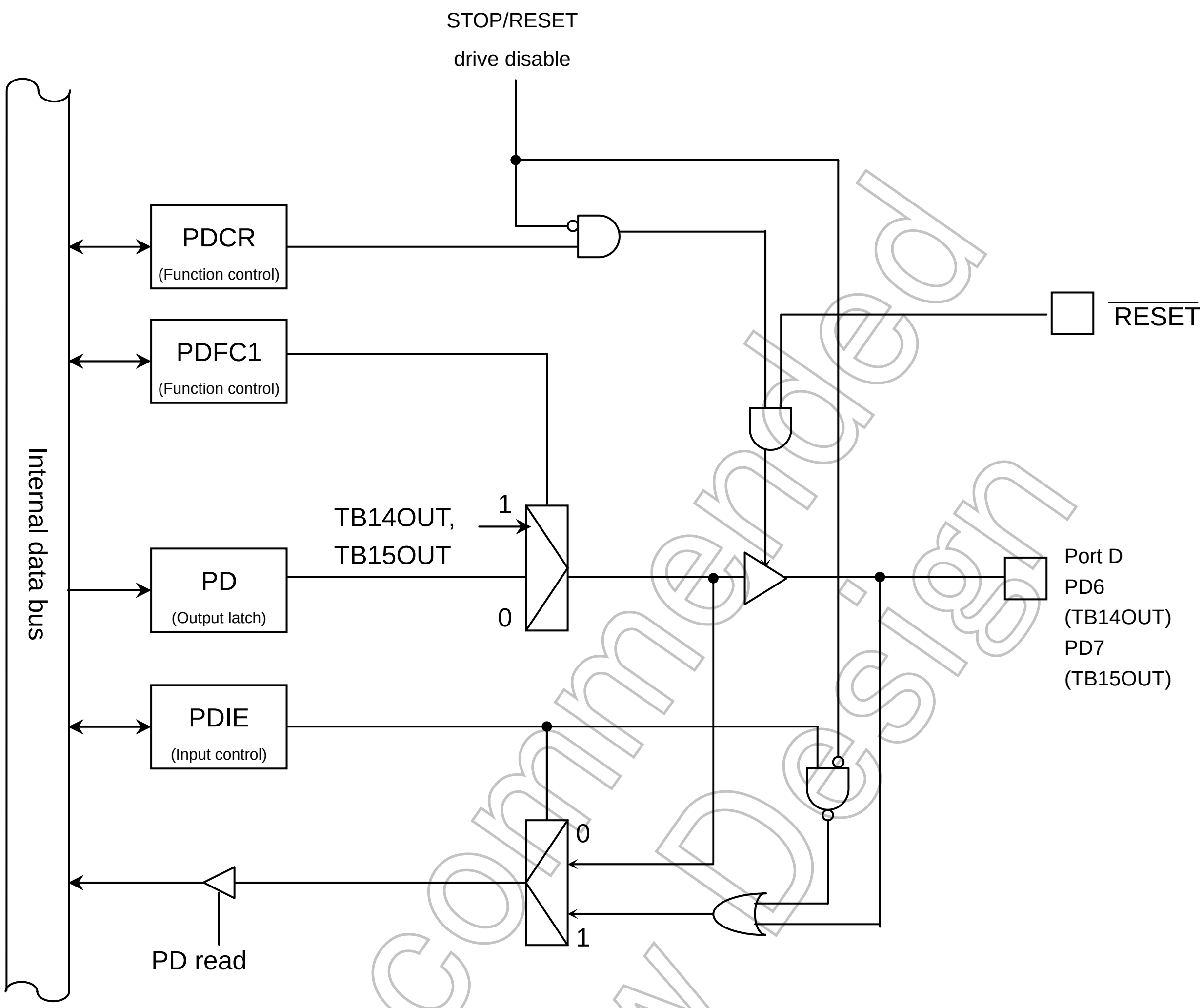


Fig. 7.20 Port D (PD6,PD7)

Port D register

PD (0xFFFF_F0D0)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Read/Write	R/W							
	After reset	Input mode (output latch register is set to “1.”)							

Port D control register

PDCR (0xFFFF_F0D1)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7C	PD6C	PD5C	PD4C	PD3C	PD2C	PD1C	PD0C
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: output disable 1: output enable							

Port D function register 1

PDFC1 (0xFFFF_F0D2)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7F1	PD6F1	PD5F1	PD4F1	PD3F1	PD2F1	PD1F1	PD0F1
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:PORT 1:TB15OUT	0:PORT 1:TB14OUT	0:PORT 1:TB12IN1	0:PORT 1:TB12IN0	0:PORT 1:TB11IN1	0:PORT 1:TB11IN0	0:PORT 1:TB10IN1	0:PORT 1: TB10IN0

Port D input enable control register

PDIE (0xFFFF_F0DE)		7	6	5	4	3	2	1	0
	Bit Symbol	PD7IE	PD6IE	PD5IE	PD4IE	PD3IE	PD2IE	PD1IE	PD0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.15 Port E (PE0~PE7)

Port E is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the function register PEFC1 and the control register PECR. A reset allows all bits of the output latch PE, PECR and PEFC1 to be cleared to "0," and the port E to be put in output disable mode.

Besides the input/output port function, the port E performs other functions: PE0 through PE4 output a 16-bit timer, PE5 and PE6 have I2C function and PE5 through PE7 have SIO function.

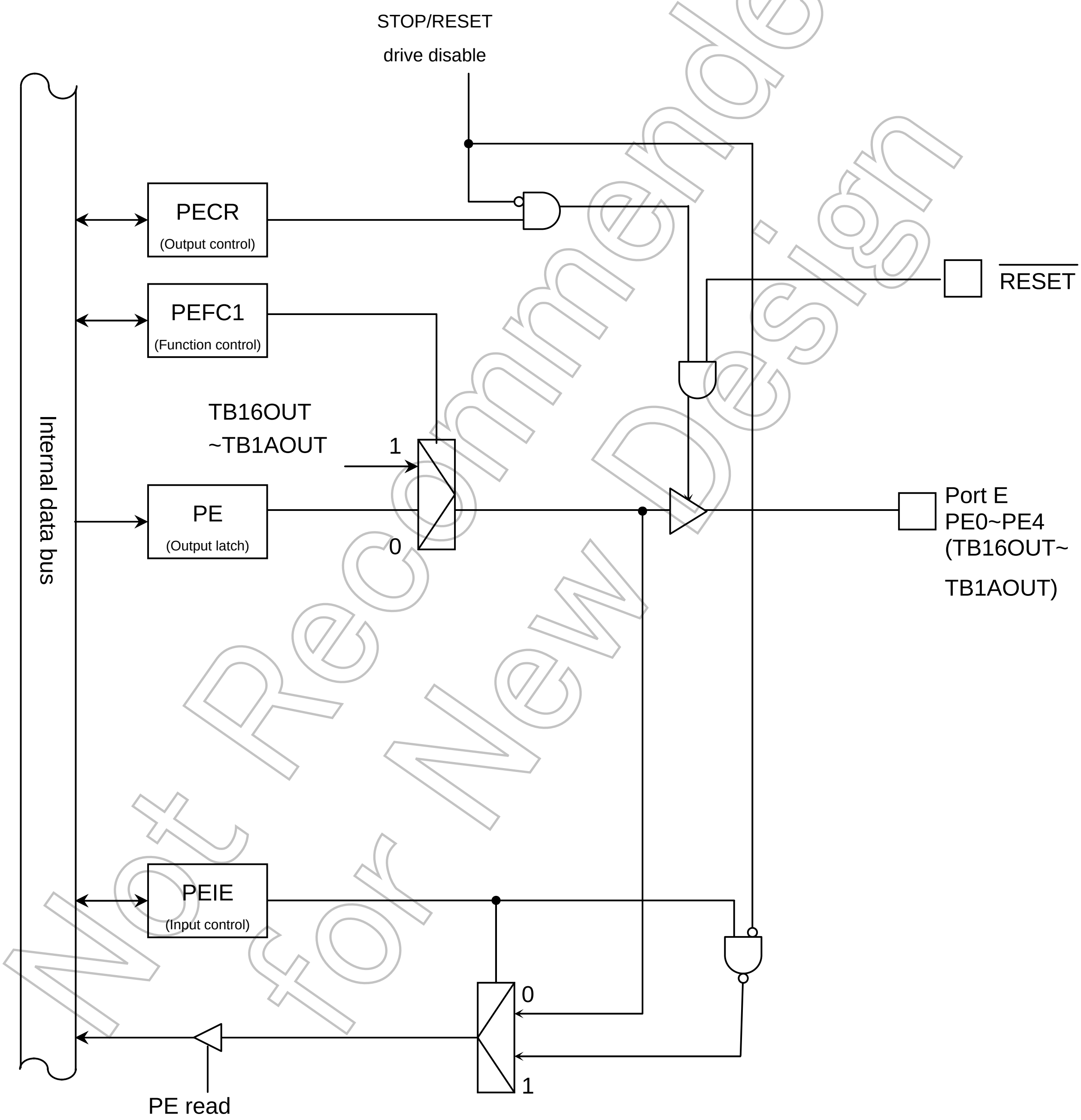


Fig. 7.21 Port E (PE0~PE4)

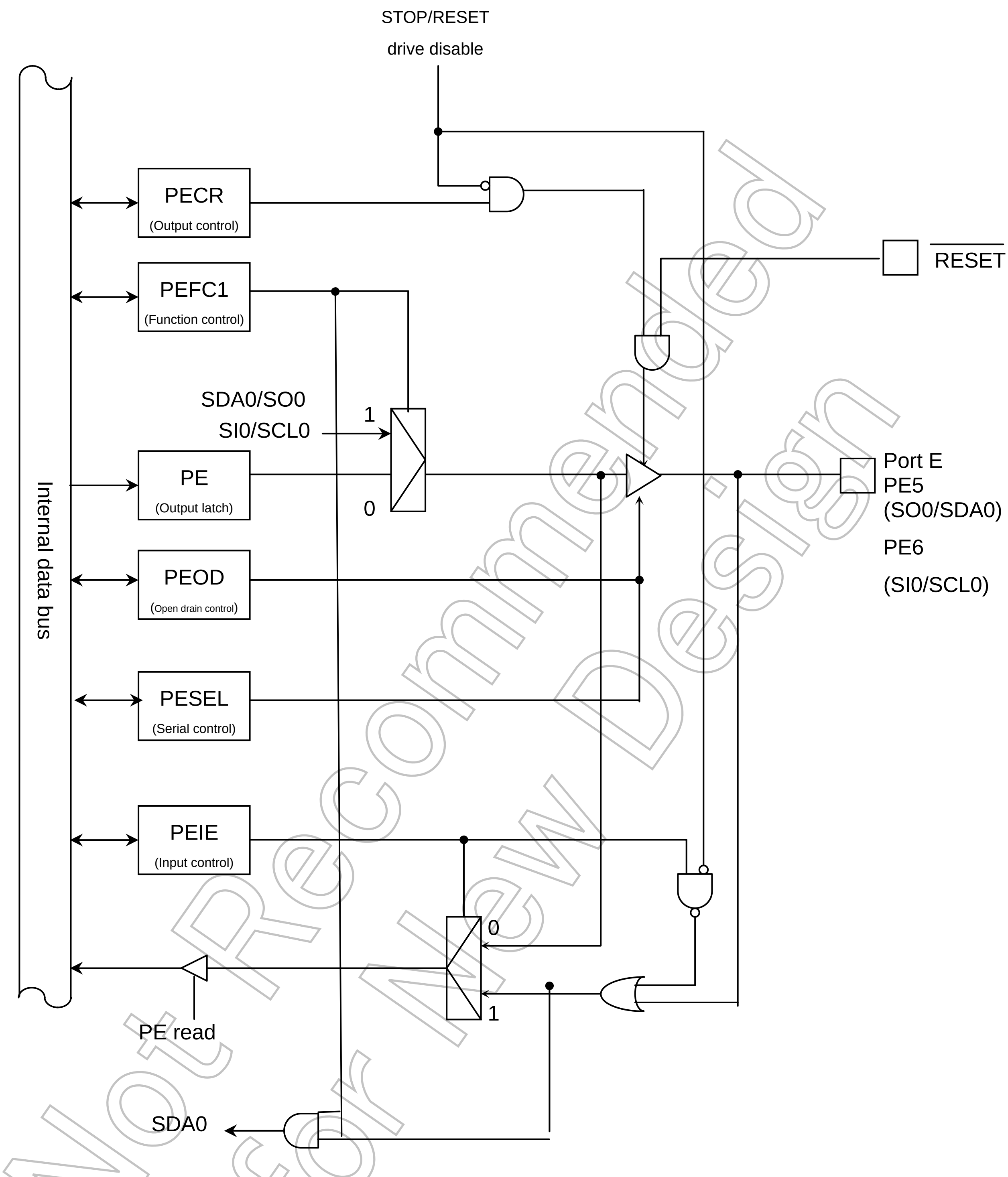


Fig. 7.22 Port E (PE5,PE6)

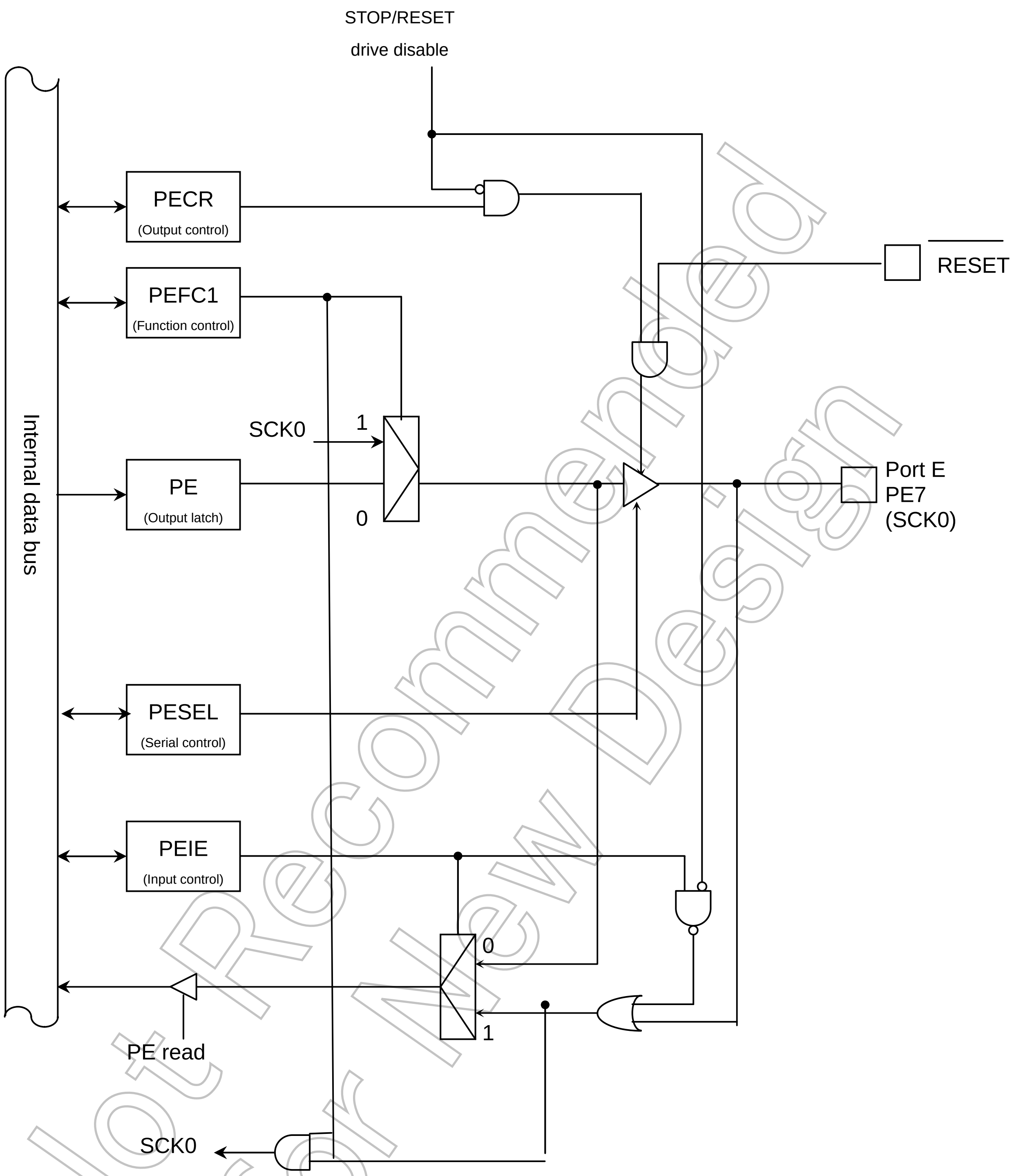


Fig 7.23 Port E (PE7)

Port E register

PE (0xFFFF_F0E0)

	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_F0E1)

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

Port E function register 1

PEFC1 (0xFFFF_F0E2)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7F1	PE6F1	PE5F1	PE4F1	PE3F1	PE2F1	PE1F1	PE0F1
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function	0:Port 1:SCK0	0:Port 1:SCL0	0:Port 1:SDA0	0:Port 1:TB1AOUT	0:Port 1:TB19OUT	0:Port 1:TB18OUT	0:Port 1:TB17OUT	0:Port 1:TB16OUT

Port E open drain (OD) control register

PEOD (0xFFFF_F0EA)

	7	6	5	4	3	2	1	0
Bit Symbol		PE6OD	PE5OD					
Read/Write	R/W							
After reset		0	0					
Function		0:CMOS 1:OD	0:CMOS 1:OD					

Port E select control register

PESEL (0xFFFF_F0ED)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7SEL	PE6SEL	PE5SEL					
Read/Write	R/W							
After reset	0	0	0	0	0	0		0
Function	SIO0 0:off 1:SCK0	SIO0 0: off 1:SI0	SIO0 0: off 1:SO0					

Port Einput control register

PEIE (0xFFFF_F0EE)

	7	6	5	4	3	2	1	0
Bit Symbol	PE7IE	PE6IE	PE5IE	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.16 Port F (PF0~PF7)

Port F is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the function register PFFCx and the control register PFCR. A reset allows all bits of PF, PFCR and PFFC1 to be cleared to "0," and the port F to be put in output disable mode.

Besides the input/output port function, the port F performs other functions: PF0 through PF2, PF4 through PF6 have a serial communication function (SIO/UART ch0 and ch1).

If the port F is used as UART/SIO function port, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set as the port, please do not set any other function.

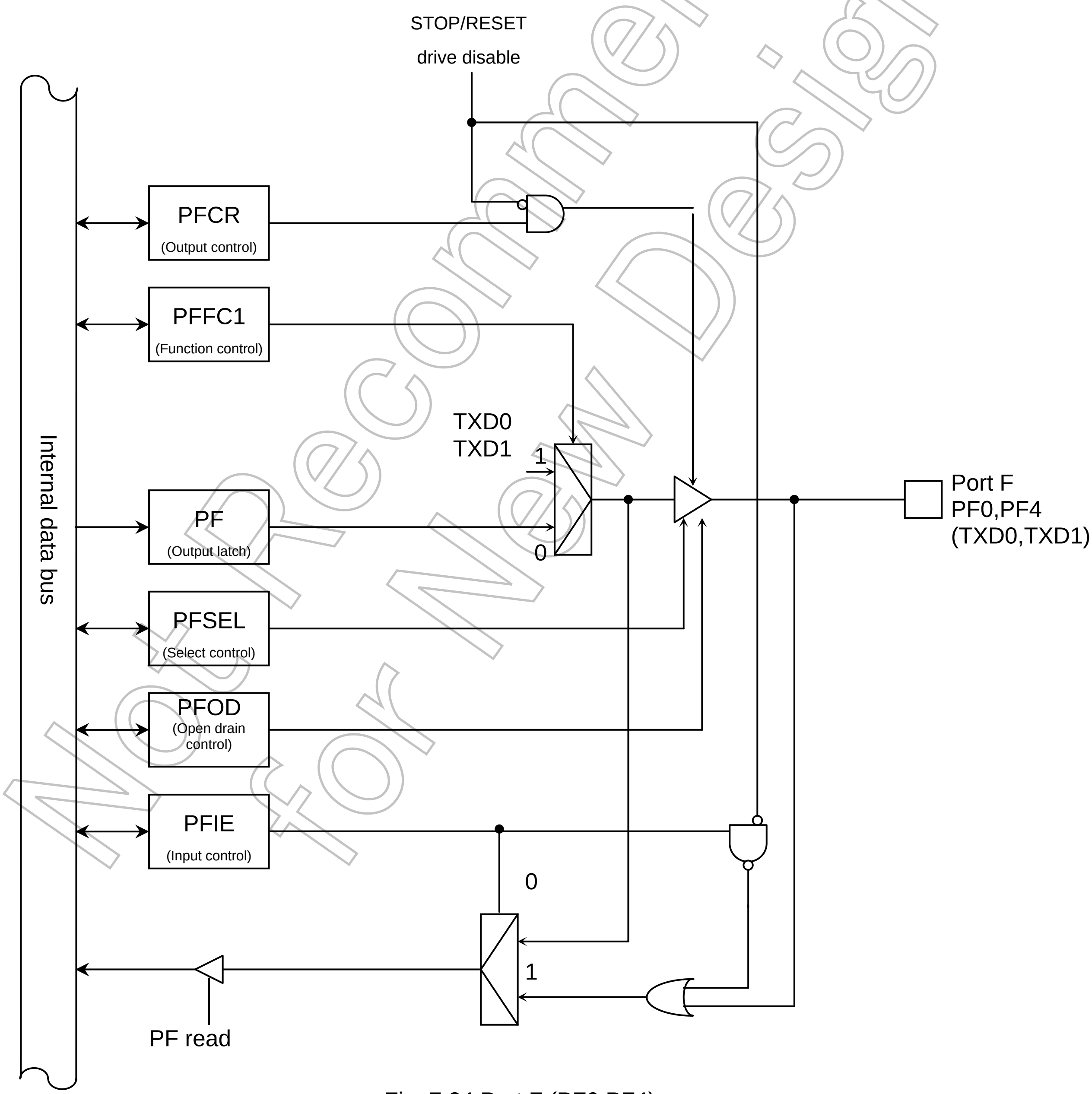


Fig. 7.24 Port F (PF0,PF4)

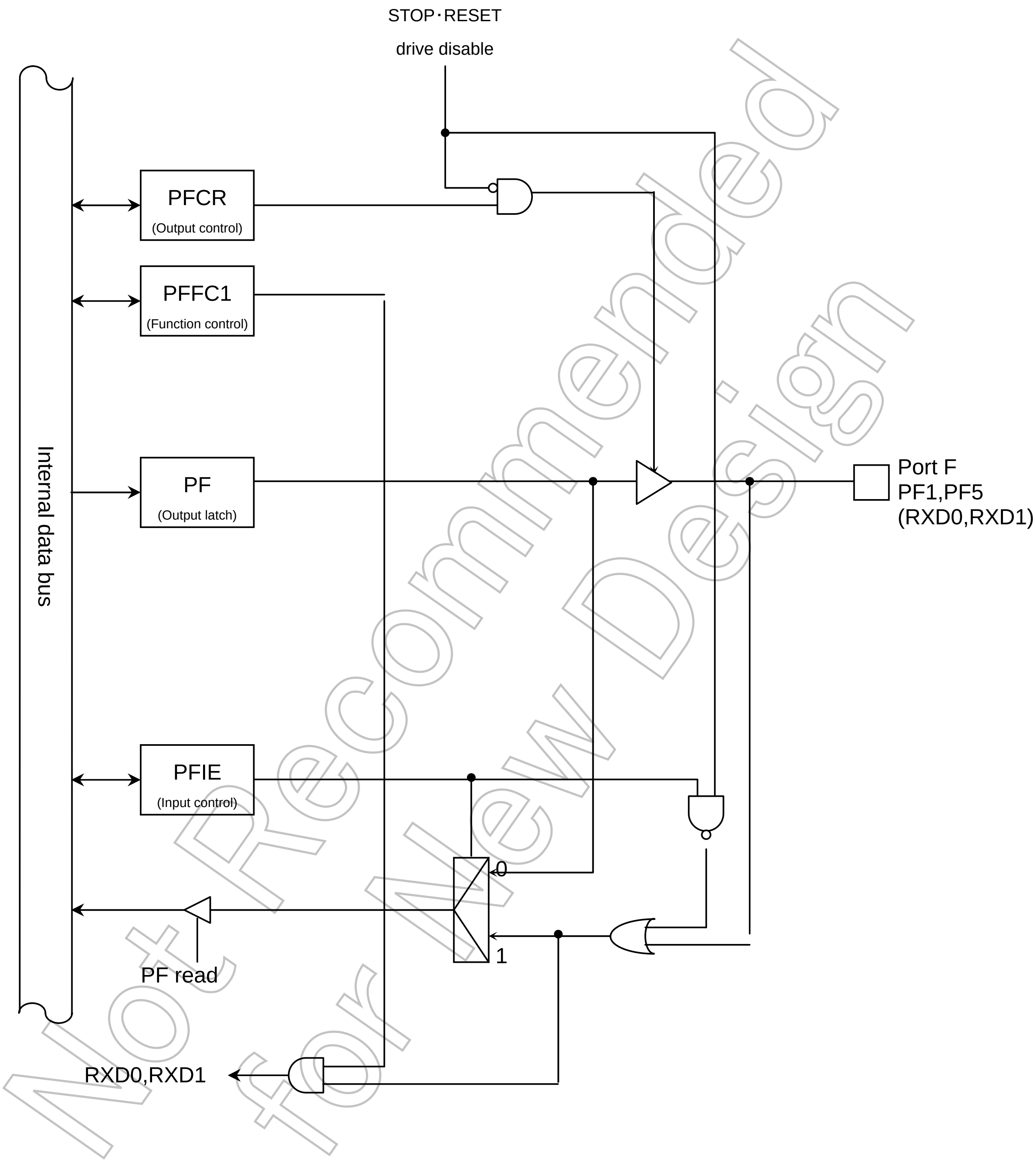


Fig. 7.25 Port F (PF1,PF5)

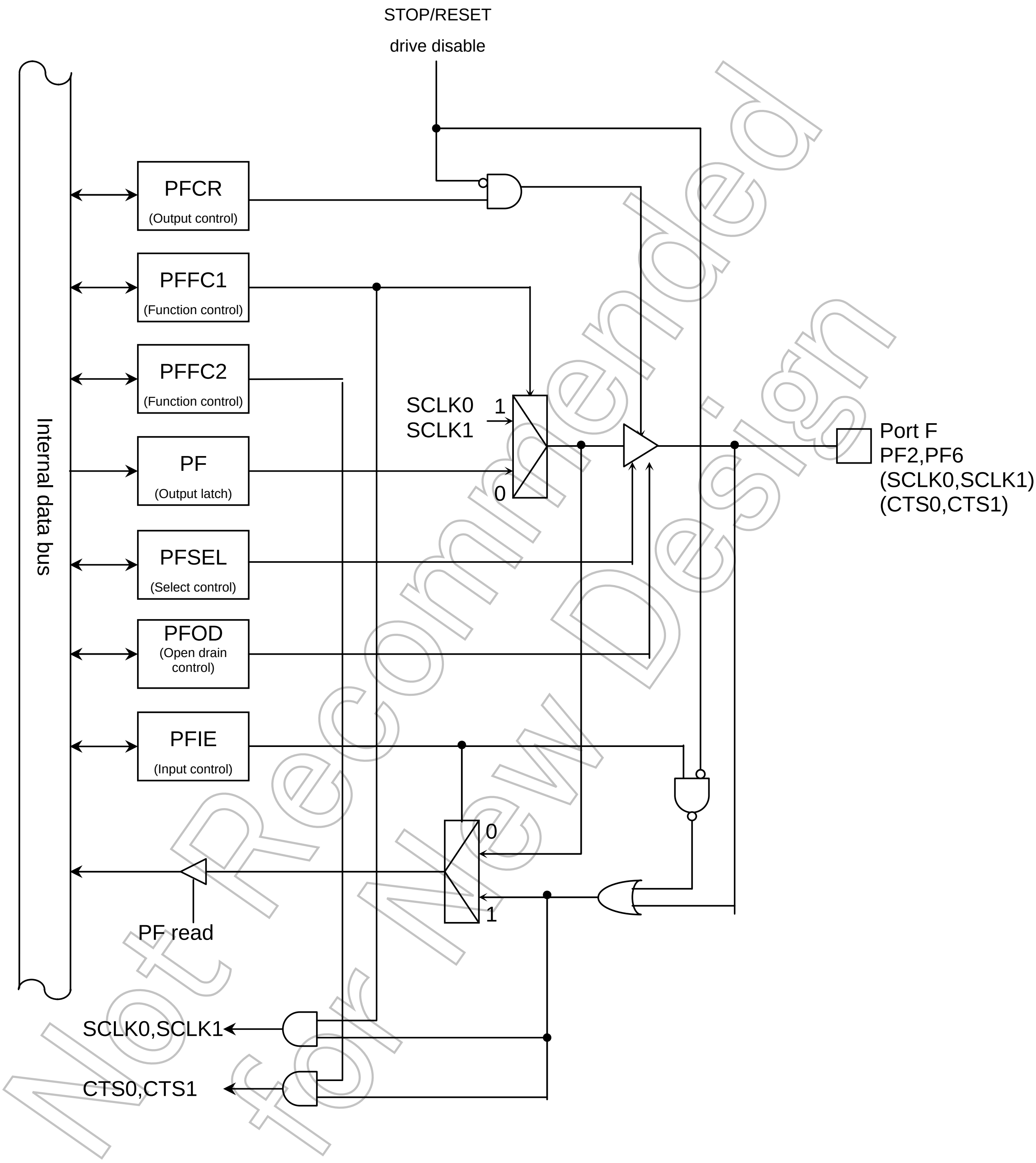


Fig. 7.26 Port F (PF2,PF6)

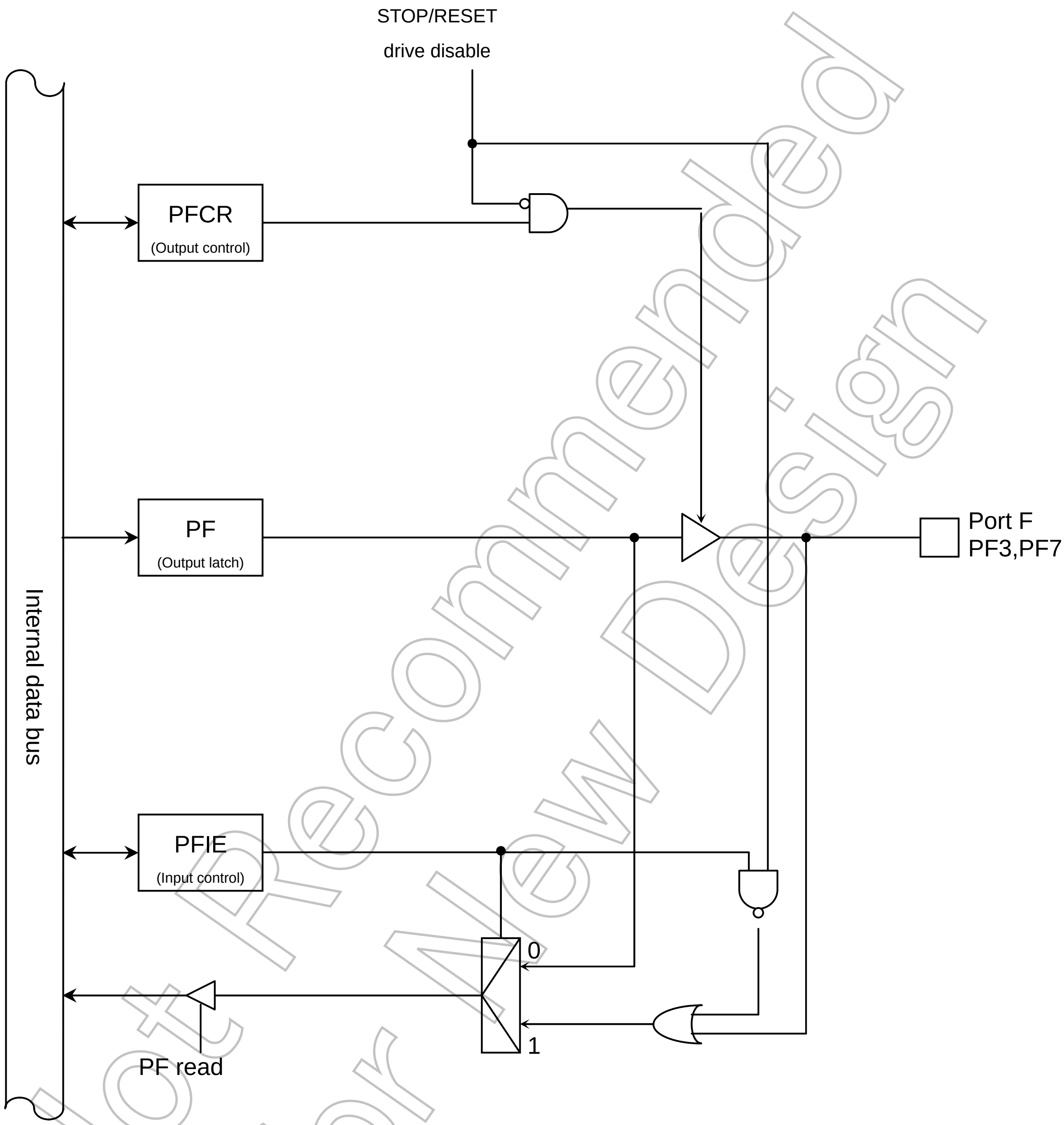


Fig. 7.27 Port F (PF3,PF7)

Port F register

PF (0xFFFF_F0F0)		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_F0F1)		7	6	5	4	3	2	1	0
	Bit Symbol	PF7C	PF6C	PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
	Read/Write	R/ W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:Input 1: Output							

Port F function register 1

PFFC1 (0xFFFF_F0F2)		7	6	5	4	3	2	1	0
	Bit Symbol		PF6F1	PF5F1	PF4F1		PF2F1	PF1F1	PF0F1
	Read/Write	R / W							
	After reset		0	0	0		0	0	0
	Function		0:Port 1:SCLK1	0:Port 1:RXD1	0:Port 1:TXD1		0:Port 1:SCLK0	0:Port 1:RXD0	0:Port 1:TXD0

Port F function register 2

PFFC2 (0xFFFF_F0F3)		7	6	5	4	3	2	1	0
	Bit Symbol		PF6F2				PF2F2		
	Read/Write	R / W							
	After reset		0				0		
	Function		0:Port 1:CTS 1				0:Port 1:CTS0		

Port F open drain (OD) control register

PFOD (0xFFFF_F0FA)		7	6	5	4	3	2	1	0
	Bit Symbol		PF6OD		PF4OD		PF2OD		PF0OD
	Read/Write	R/W							
	After reset		0		0		0		0
	Function		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD

Port F select control register

PFSEL (0xFFFF_F0FD)		7	6	5	4	3	2	1	0
	Bit Symbol		PF6SEL		PF4SEL		PF2SEL		PF0SEL
	Read/Write	R/W							
	After reset	0	0	0	0	0	0		0
	Function		SCLK1 0: off 1:SCLK		TXD1 0: off 1:TXD		SCLK0 0: off 1:SCLK		TXD0 0: off 1:TXD

Port F input control register

PFIE (0xFFFF_F0FE)

	7	6	5	4	3	2	1	0
Bit Symbol	PF7IE	PF6IE	PF5IE	PF4IE	PF3IE	PF2IE	PF1IE	PF0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.17 Port G (PG0~PG7)

The port G is a general-purpose, 8-bit input/output port. For this port, outputs can be specified in units of bits by using the control register PGCR and the function register PGFCx. A reset allows all bits of the PG, PGCR, PGFC1 and PGFC2 to be cleared to "0," and the port G to be put in output disable mode.

Besides the input/output port function, the port C performs other function: PG0 through PG2 and PG4 through PG6 have a serial communication function (SIO/UART ch2 and ch3).

If the port G is used as UART/SIO function port, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set as the port, please do not set any other function.

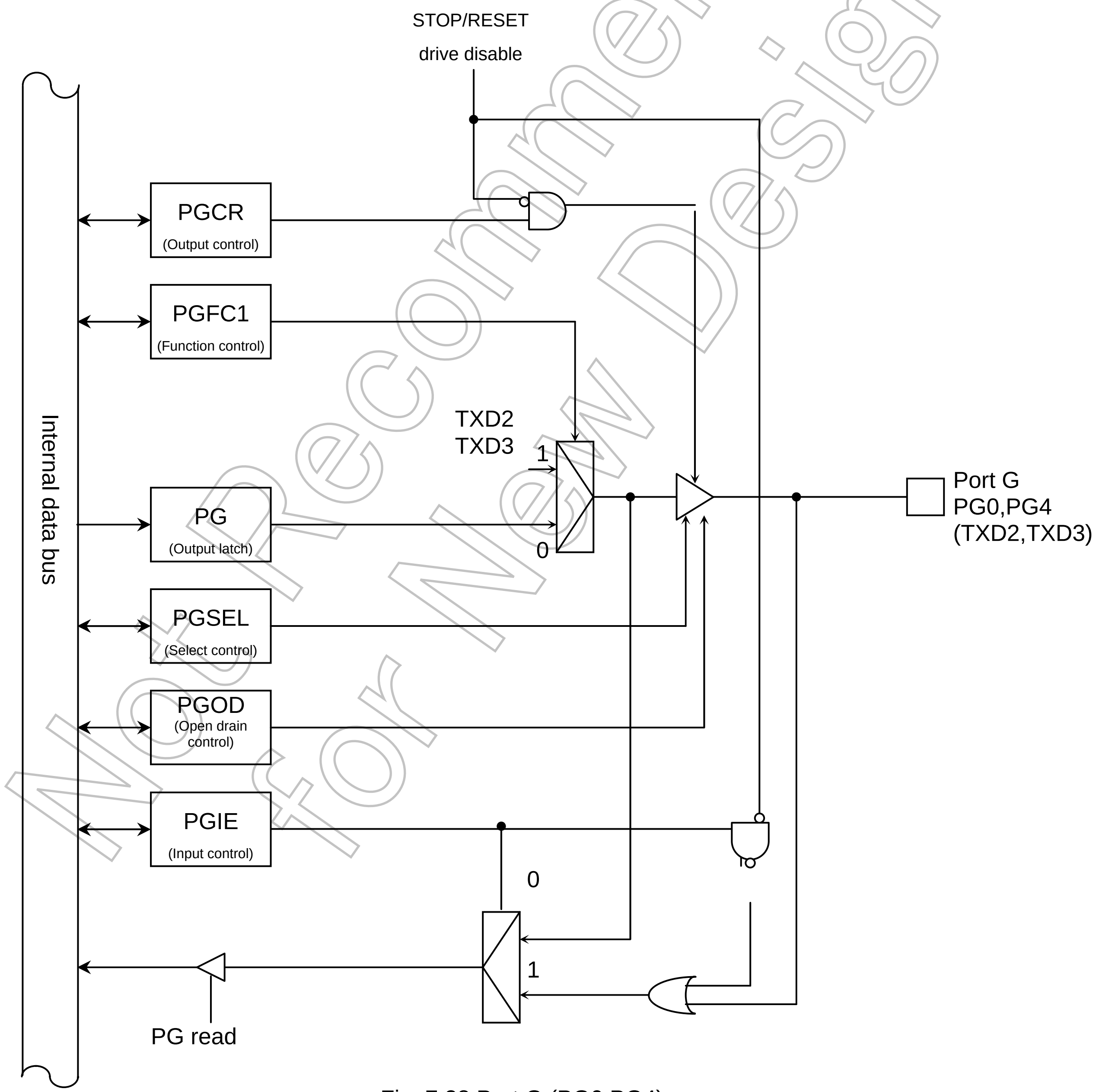


Fig. 7.28 Port G (PG0,PG4)

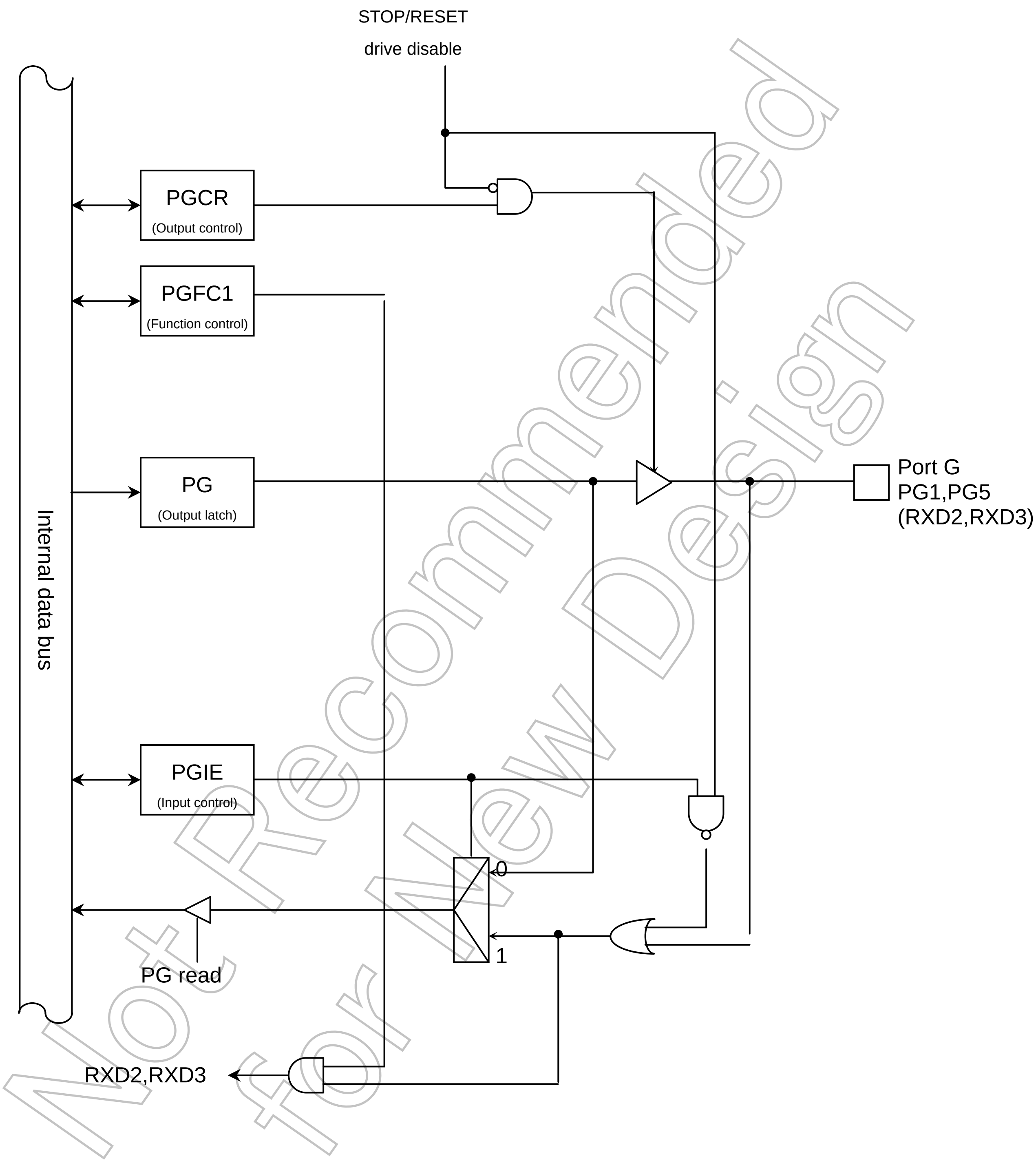


Fig. 7.29 Port G (PG1,PG5)

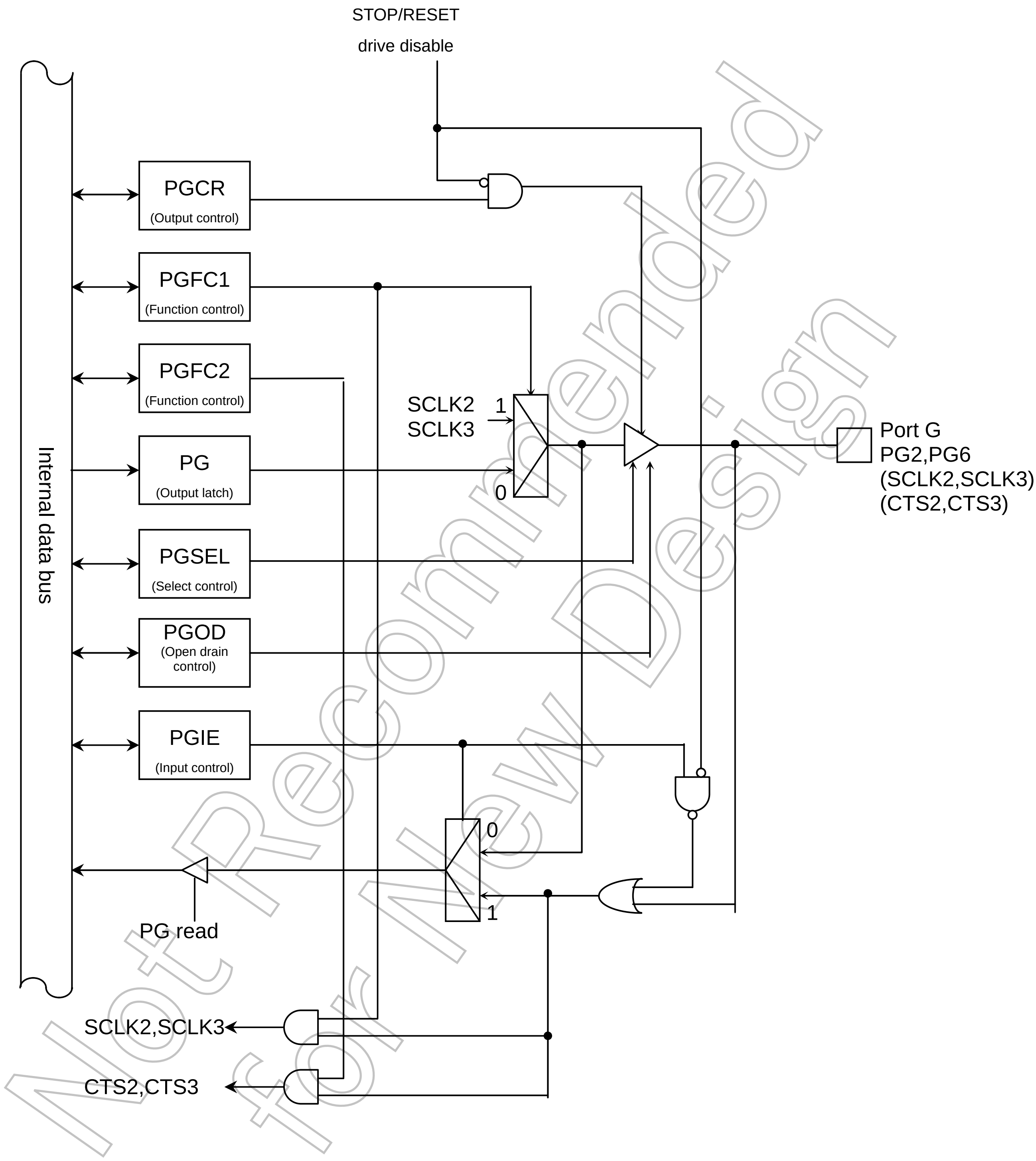


Fig. 7-30 Port G (PG2,PG6)

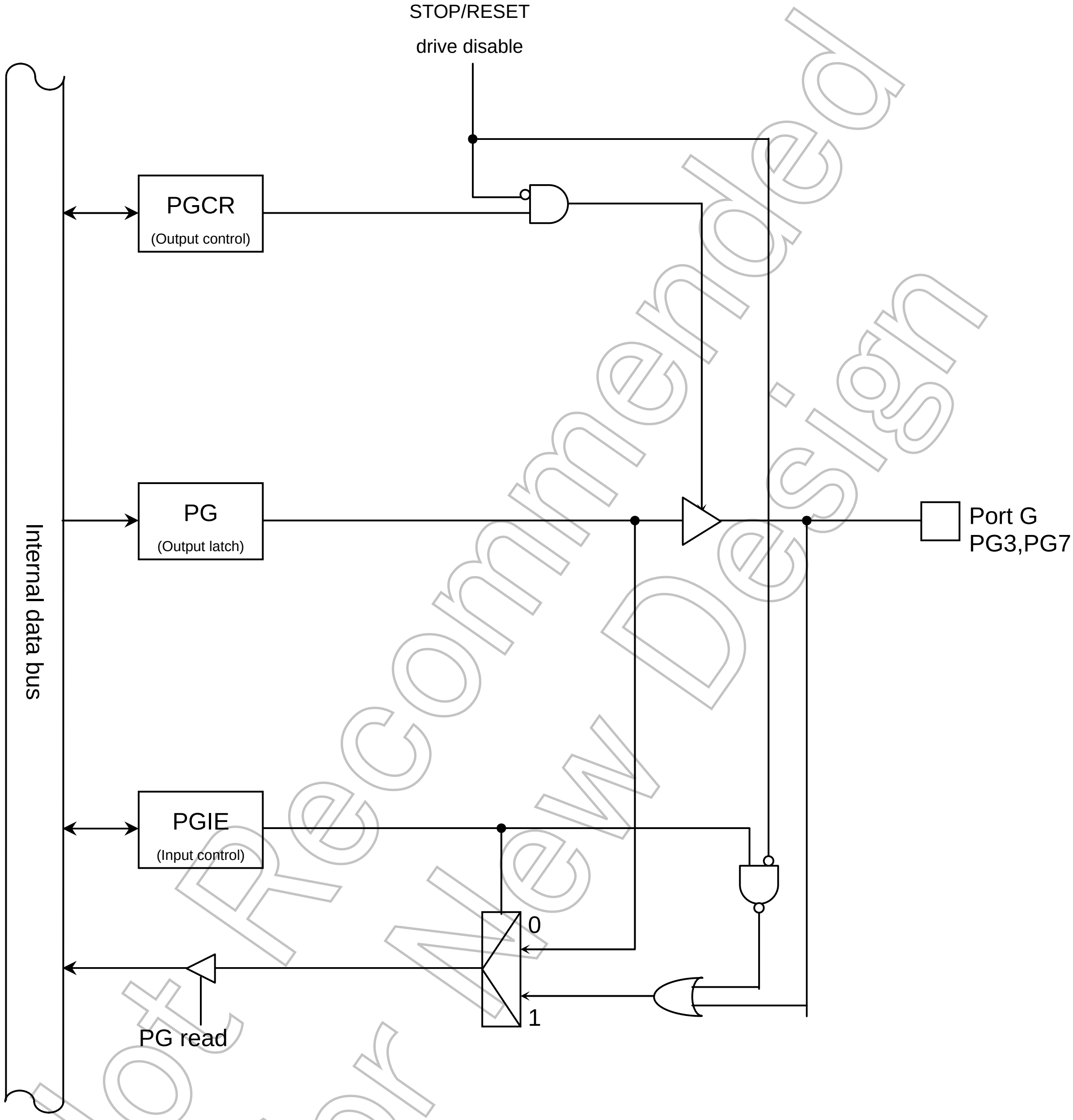


Fig. 7.31 Port G (PG3,PG7)

Port G register

PG (0xFFFF_F100)

	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_F101)

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

Port G function register 1

PGFC1 (0xFFFF_F102)

	7	6	5	4	3	2	1	0
Bit Symbol	PG7F1	PG6F1	PG5F1	PG4F1	PG3F1	PG2F1	PG1F1	PG0F1
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function	0:Port 1:TBTIN2	0:Port 1:SCLK3	0:Port 1:RXD3	0:Port 1:TXD3	0:Port 1:TBTIN1	0:Port 1:SCLK2	0:Port 1:RXD2	0:Port 1:TXD2

Port G function register 2

PGFC2 (0xFFFF_F103)

	7	6	5	4	3	2	1	0
Bit Symbol		PG6F2				PG2F2		
Read/Write	R / W							
After reset		0				0		
Function		0: Port 1: CTS 3				0: Port 1: CTS2		

Port G open drain (OD) control register

PGOD (0xFFFF_F10A)

	7	6	5	4	3	2	1	0
Bit Symbol		PG6OD		PG4OD		PG2OD		PGF0OD
Read/Write	R/W							
After reset		0		0		0		0
Function		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD

Port G select control register

PGSEL (0xFFFF_F10D)

	7	6	5	4	3	2	1	0
Bit Symbol		PG6SEL		PG4SEL		PG2SEL		PG0SEL
Read/Write	R/W							
After reset	0	0	0	0	0	0		0
Function		SCLK3 0: off 1:SCLK		TXD3 0: off 1:TXD		SCLK2 0: off 1:SCLK		TXD2 0: off 1:TXD

Port G input control register

		7	6	5	4	3	2	1	0
PGIE	Bit Symbol	PG7IE	PG7IE	PG5IE	PG4IE	PG3IE	PG2IE	PG1IE	PG0IE
(0xFFFF_F10E)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.18 Port H (PH0~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. Outputs can be set by using the control register PHCR and the function register PHFCx. A reset allows all bits of the PH to be set to "1," all bits of PHCR, PHFC1 and PHFC2 to be cleared to "0," and the port H to be put in output disable mode.

Besides the input/output port function, the port H performs other functions: PH0 through PH2 and PH4 through PH6 have a serial communication function (SIO/UART ch4 and ch5). PH3 and PH7 input external interrupt (INT9 and INTA).

If the port H is used as UART/SIO function port, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set as the port, please do not set any other function.

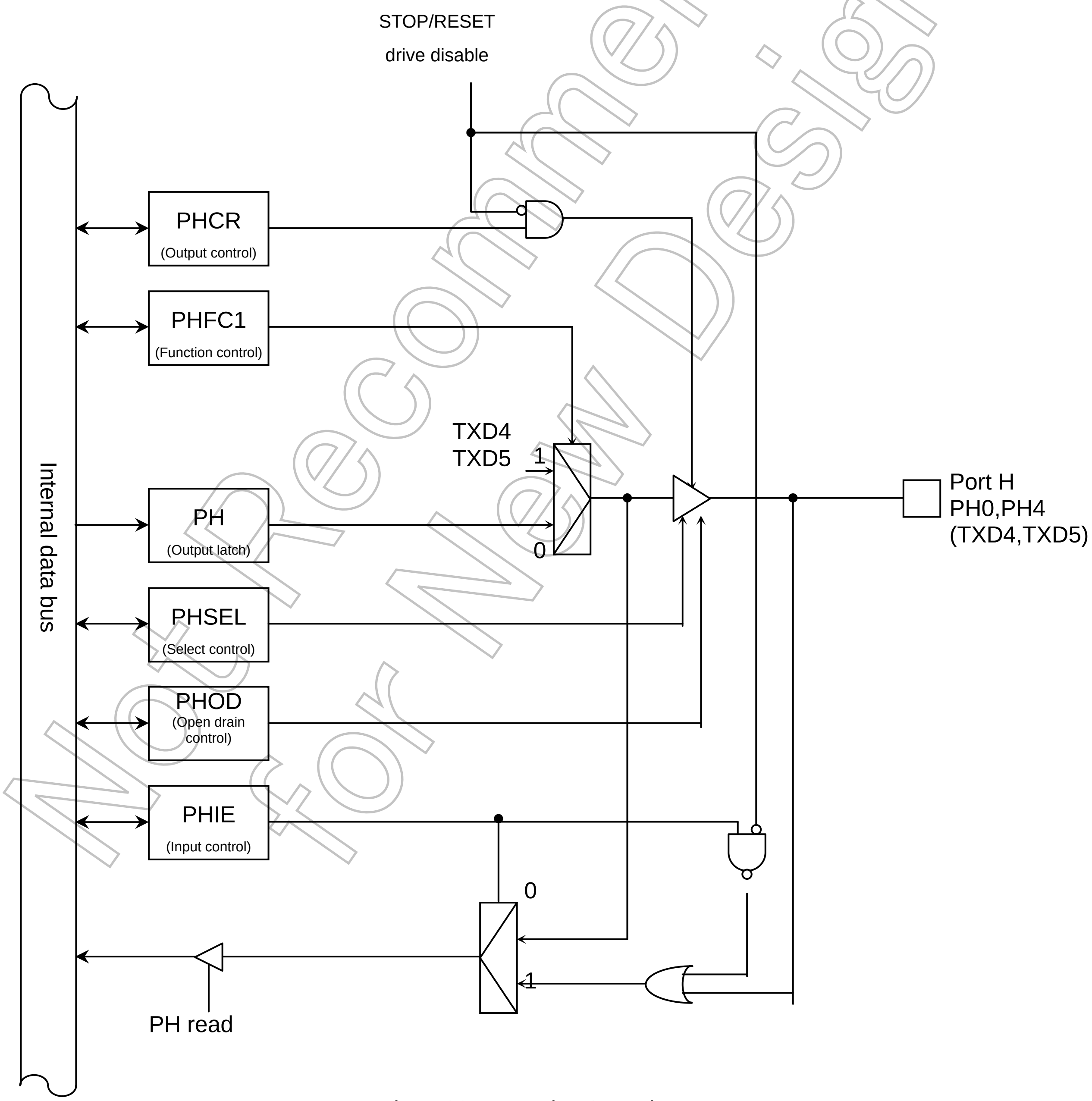


Fig. 7.32 Port H (PH0,PH4)

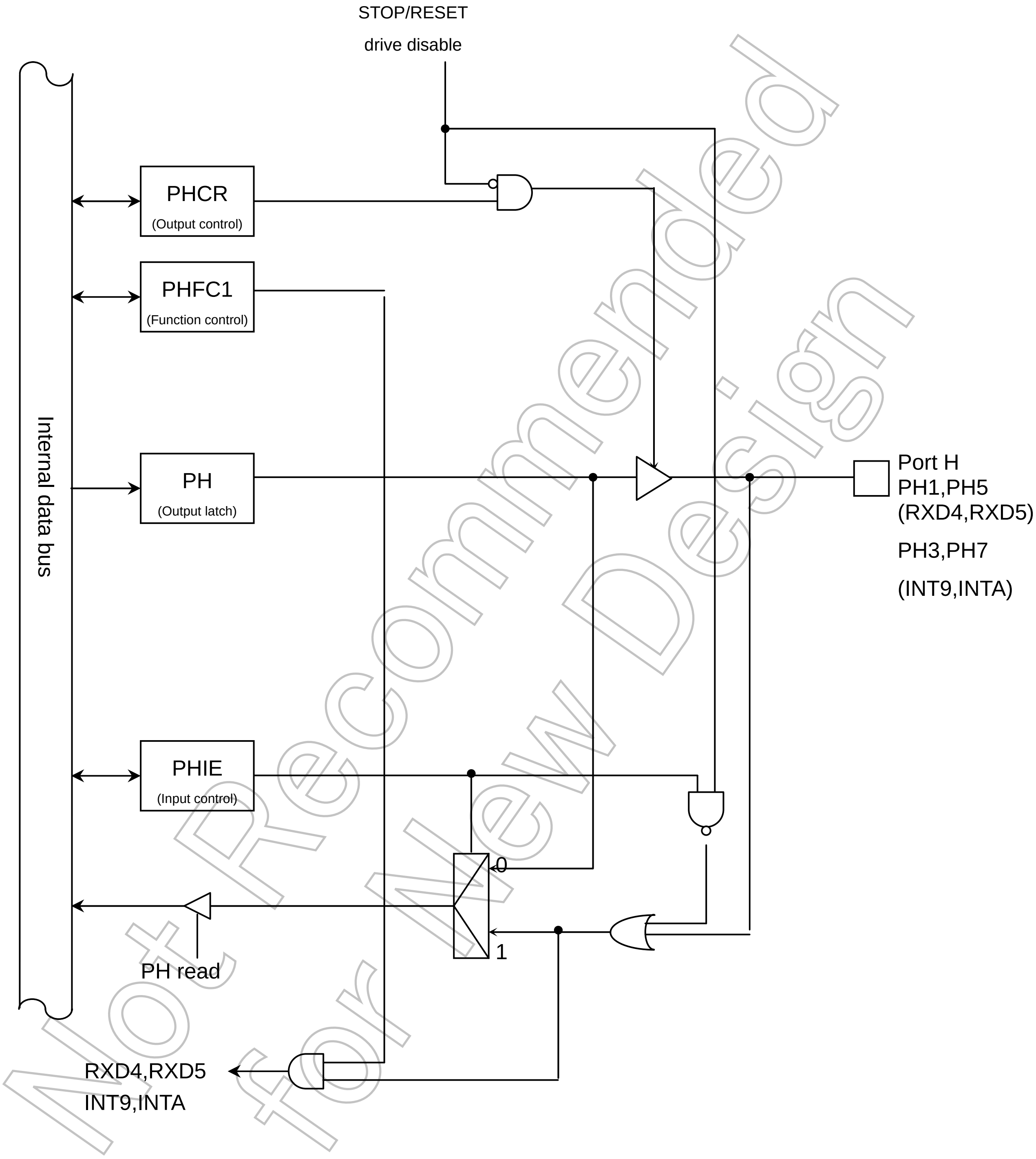


Fig. 7.33 Port H (PH1,PH5,PH3,PH7)

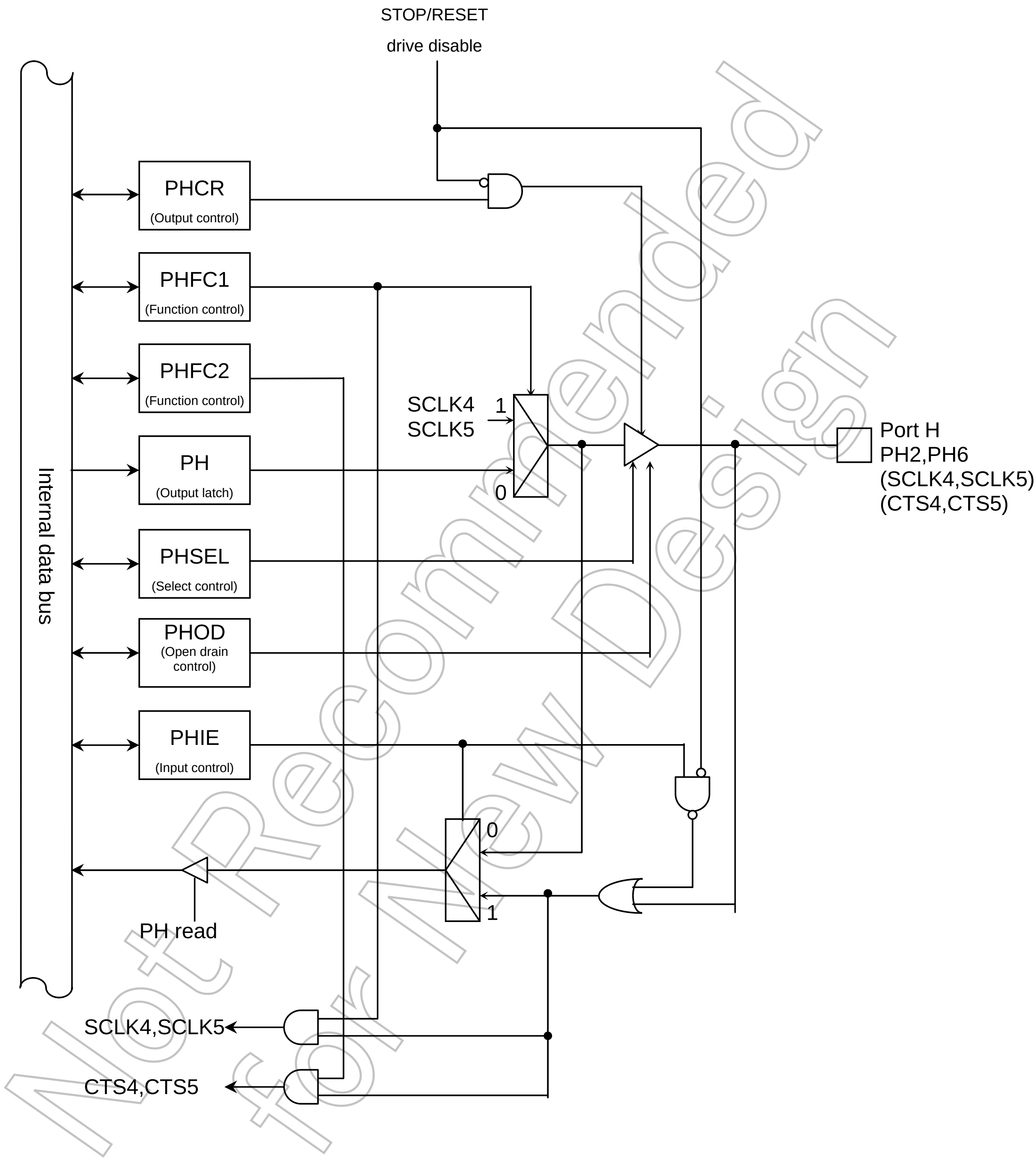


Fig. 7.34 Port H (PH2,PH6)

Port H register

PH (0xFFFF_F110)

	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_F111)

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

Port H function register 1

PHFC1 (0xFFFF_F112)

	7	6	5	4	3	2	1	0
Bit Symbol	PH7F1	PH6F1	PH5F1	PH4F1	PH3F1	PH2F1	PH1F1	PH0F1
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function	0:Port 1:INTA	0:Port 1:SCLK5	0:Port 1:RXD5	0:Port 1:TXD5	0:Port 1:INT9	0:Port 1:SCLK4	0:Port 1:RXD4	0:Port 1:TXD4

Port H function register 2

PHFC2 (0xFFFF_F113)

	7	6	5	4	3	2	1	0
Bit Symbol		PH6F2				PH2F2		
Read/Write	R / W							
After reset		0				0		
Function		0:Port 1:CTS 5				0:Port 1:CTS4		

Port H open drain (OD) control register

PHOD (0xFFFF_F11A)

	7	6	5	4	3	2	1	0
Bit Symbol		PH6OD		PH4OD		PH2OD		PH0OD
Read/Write	R/W							
After reset		0		0		0		0
Function		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD

Port H select control register

PHSEL (0xFFFF_F11D)

	7	6	5	4	3	2	1	0
Bit Symbol		PH6SEL		PH4SEL		PH2SEL		PH0SEL0
Read/Write	R/W							
After reset	0	0	0	0	0	0		0
Function		SCLK5 0: off 1:SCLK		TXD5 0: off 1:TXD		SCLK4 0: off 1:SCLK		TXD4 0: off 1:TXD

Port H input control register

		7	6	5	4	3	2	1	0
PHIE	Bit Symbol	PH7IE	PH6IE	PG5IE	PH4IE	PH3IE	PH2IE	PH1IE	PH0IE
(0xFFFF_F11E)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.19 Port I (PI0~PI7)

The port I is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PICR and the function register PIFCx. A reset allows all bits of the PI to be set to "1," all bits of PICR, PIFC1 and PIFC2 to be cleared to "0," and the port I to be put in output disable mode.

Besides the input/output port function, the port I performs other functions: PI0 through PI2 and PI4 through PI6 have a serial communication function (SIO/UART ch6 and ch7). PI3 input external interrupt (INTB).

If the port I is used as UART/SIO function port, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set as the port, please do not set any other function.

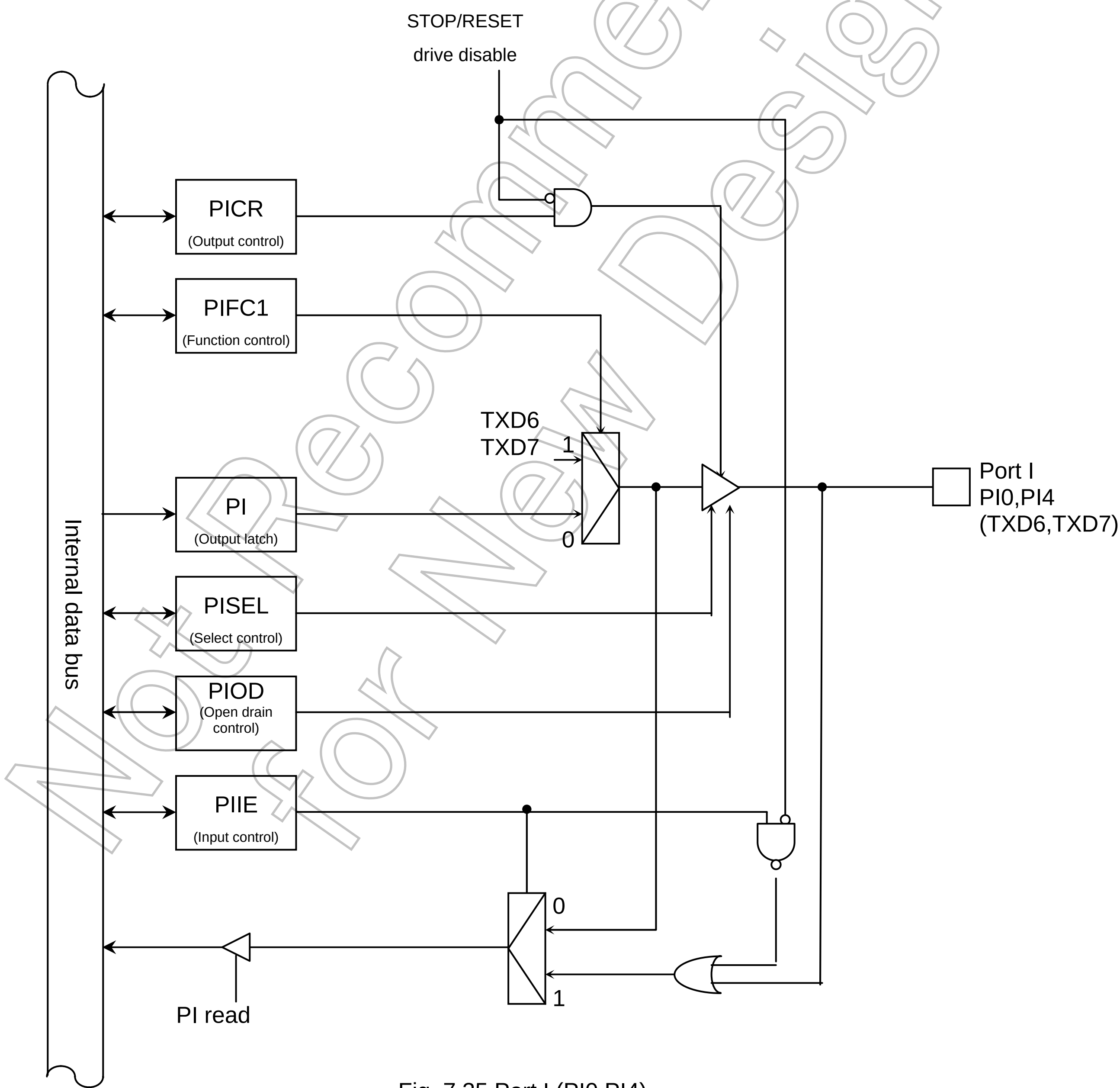


Fig. 7.35 Port I (PI0,PI4)

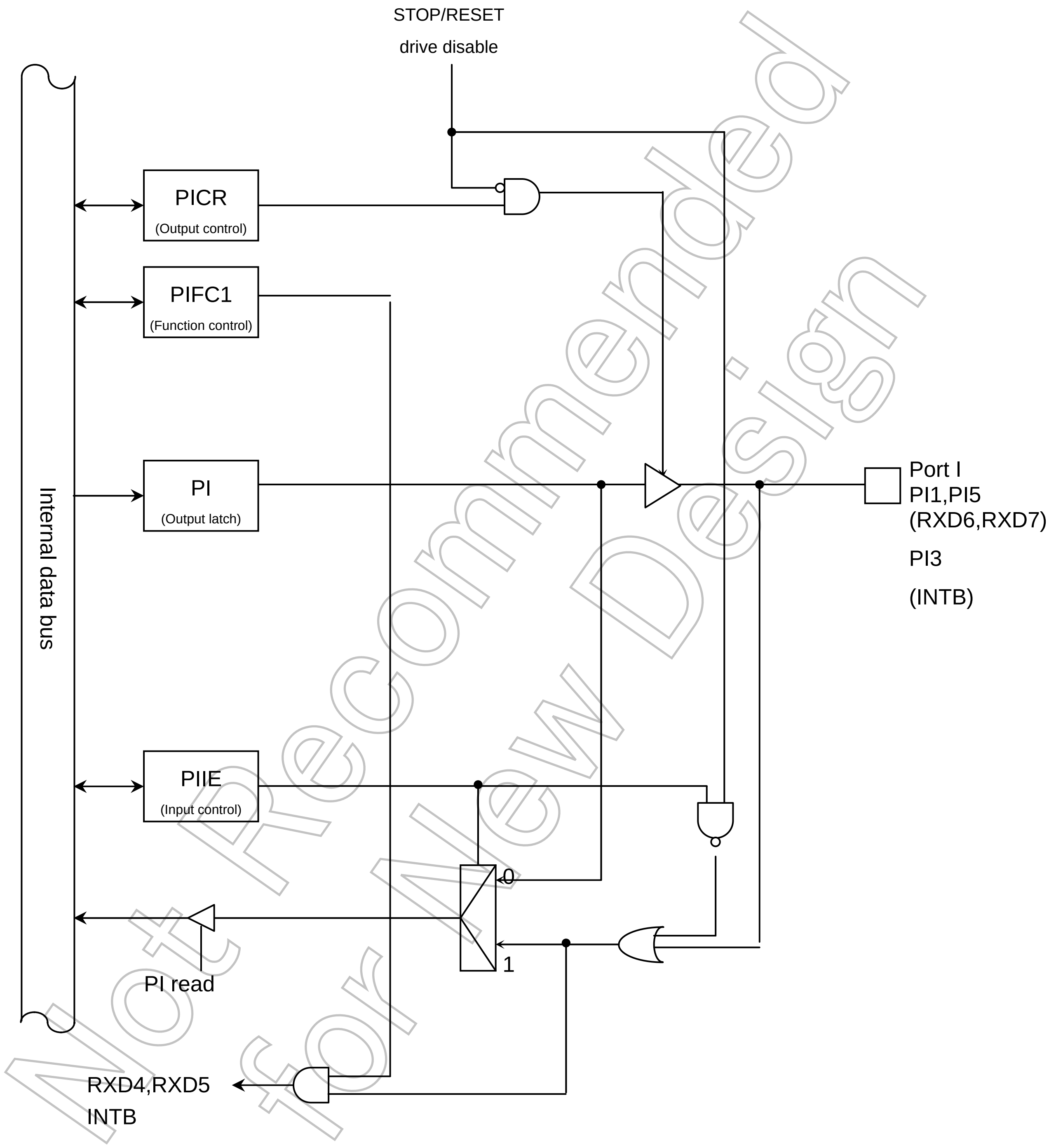


Fig. 7.36 Port I (PI1,PI5,PI3)

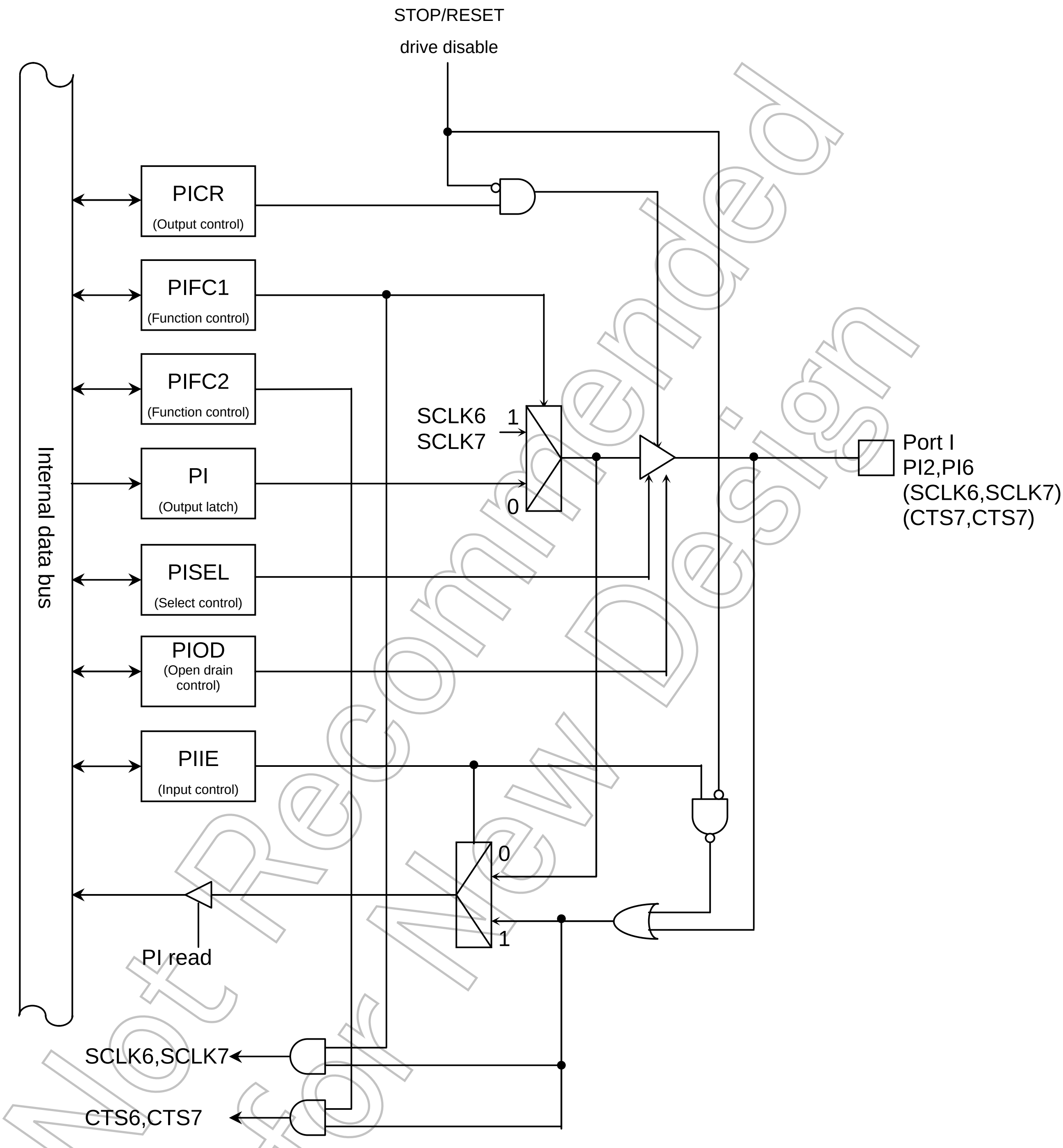


Fig. 7.37 Port I (PI2,PI6)

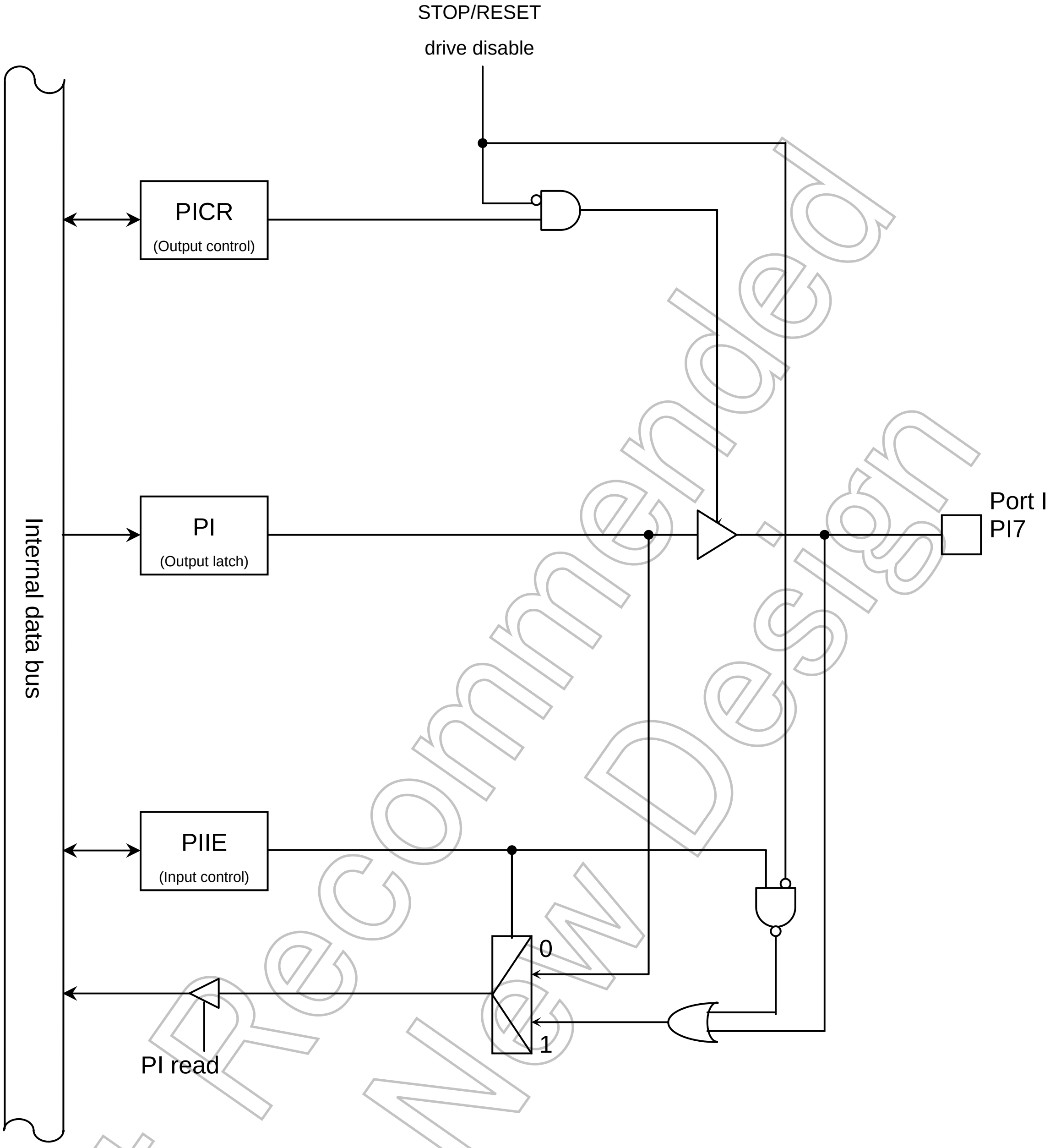


Fig. 7.38 Port I (PI7)

Port I register

PI (0xFFFF_F120)

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

Port I control register

PICR (0xFFFF_F121)

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

Port I function register 1

PIFC1 (0xFFFF_F122)

	7	6	5	4	3	2	1	0
Bit Symbol	PI7F1	PI6F1	PI5F1	PI4F1	PI3F1	PI2F1	PI1F1	PI0F1
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function		0:Port 1:SCLK7	0:Port 1:RXD7	0:Port 1:TXD7	0:Port 1:INTB	0:Port 1:SCLK6	0:Port 1:RXD6	0: Port 1:TXD6

Port I function register 2

PIFC2 (0xFFFF_F123)

	7	6	5	4	3	2	1	0
Bit Symbol		PI6F2				PI2F2		
Read/Write	R / W							
After reset		0				0		
Function		0: Port 1:CTS 7				0:Port 1:CTS6		

Port I open drain (OD) control register

PIOD (0xFFFF_F12A)

	7	6	5	4	3	2	1	0
Bit Symbol		PI6OD		PI4OD		PI2OD		PI0OD
Read/Write	R/W							
After reset		0		0		0		0
Function		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD		0:CMOS 1: OD

Port I select control register

PISEL (0xFFFF_F12D)

	7	6	5	4	3	2	1	0
Bit Symbol		PI6SEL		PI4SEL		PI2SEL		PI0SEL
Read/Write	R/W							
After reset	0	0	0	0	0	0		0
Function		SCLK7 0: off 1:SCLK		TXD7 0:off 1:TXD		SCLK6 0:off 1:SCLK		TXD6 0: off 1:TXD

Port I input control register

		7	6	5	4	3	2	1	0
PIIE	Bit Symbol	PI7IE	PI6IE	PI5IE	PI4IE	PI3IE	PI2IE	PI1IE	PI0IE
(0xFFFF_F12E)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.20 Port J (PJ0~PJ7)

The port J is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PJCR and the function register PJFCx. A reset allows all bits of the PJ, PJCR, PJFC1 and PJFC2 to be cleared to "0," and the port J to be put in output disable mode.

Besides the input/output port function, the port J performs other functions: PJ0 through PJ2 and PJ5 through PJ7 have a serial bus I/F function (SBI2). PJ3 and PJ4 input 32-bit timer capture trigger (TC0IN and TC1IN).

If the port J is used as a port UART/SIO function or serial bus I/F function, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set as the port, please do not set any other function.

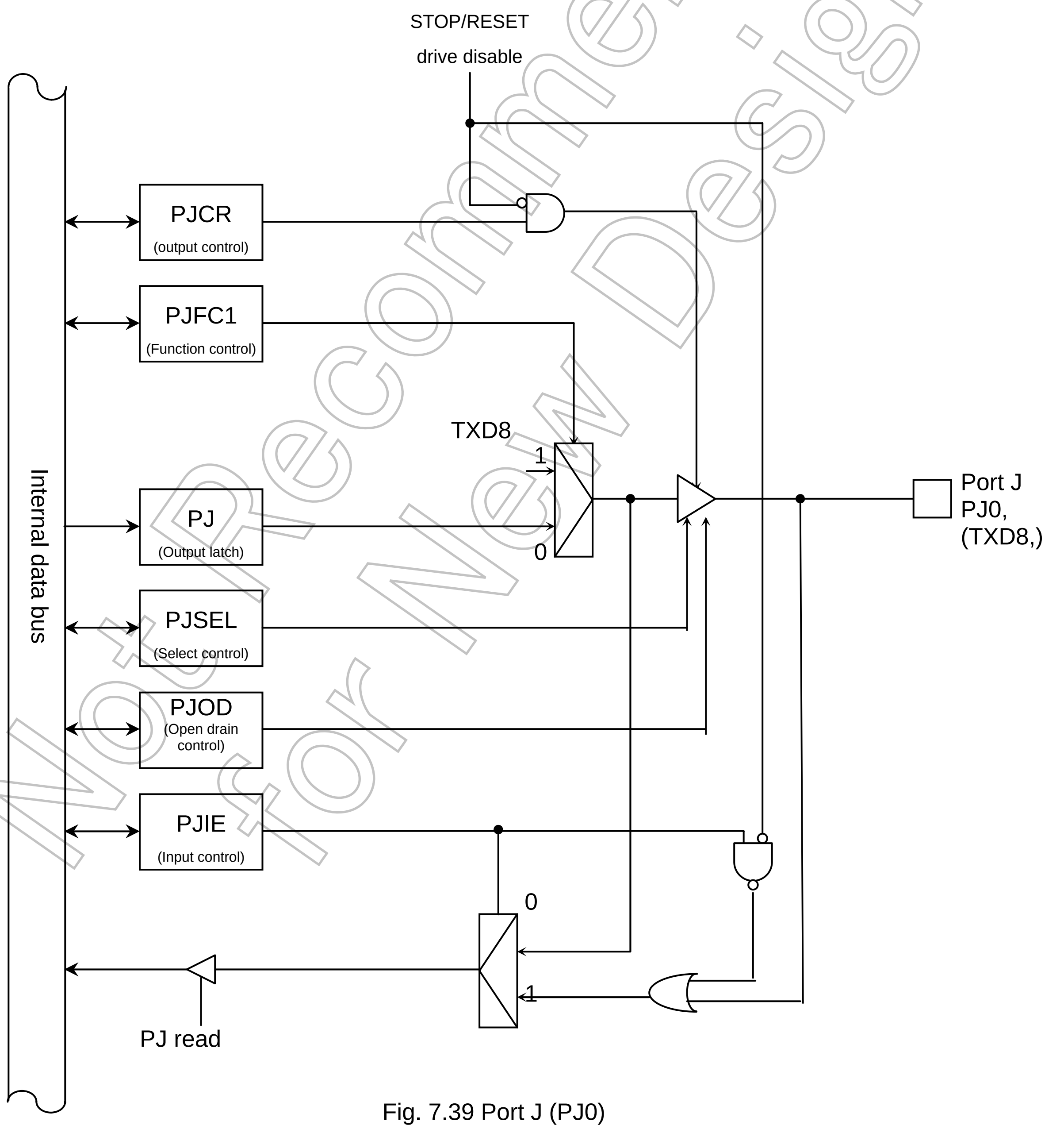


Fig. 7.39 Port J (PJ0)

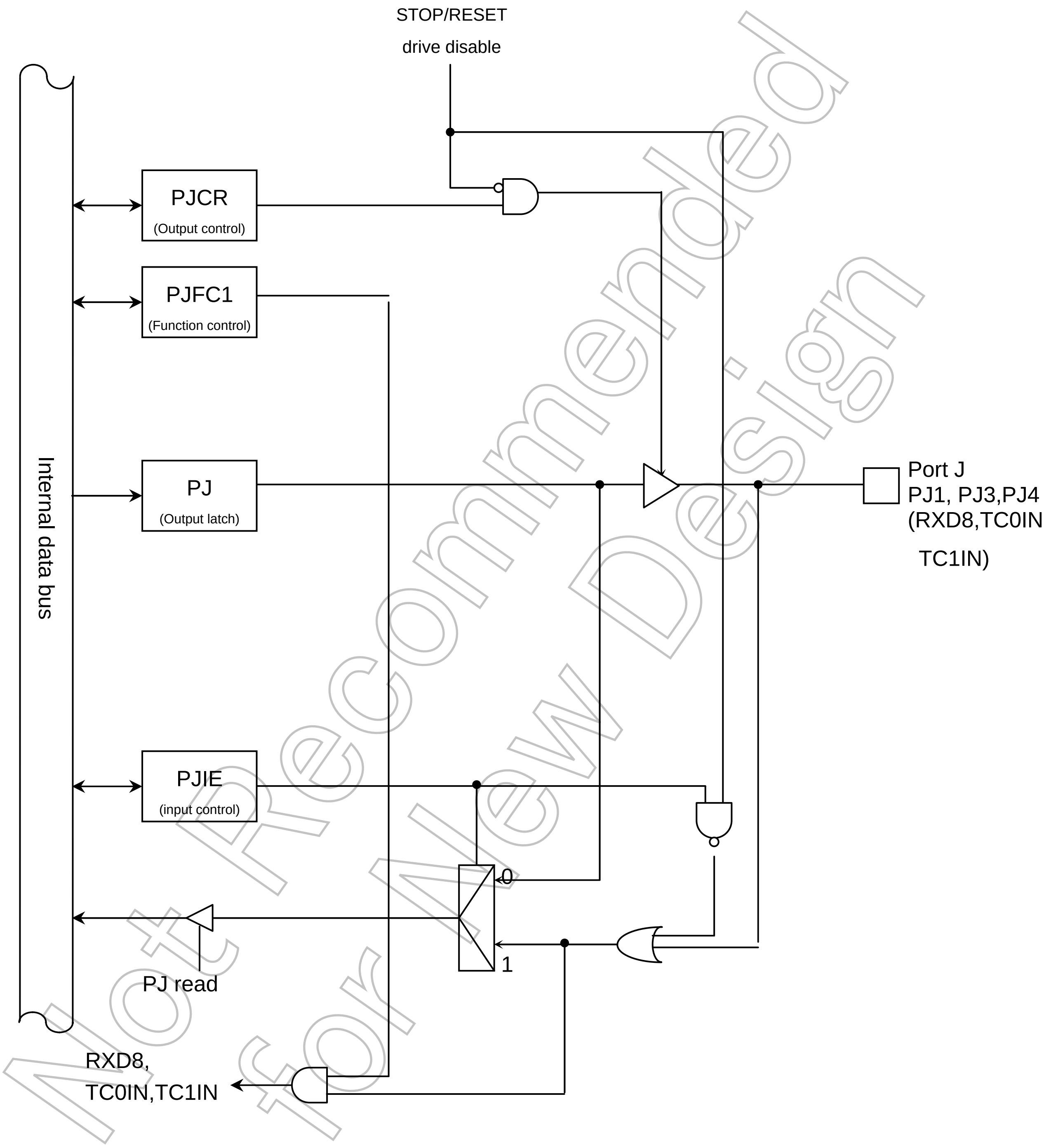


Fig. 7.40 Port J (PJ1,PJ3,PJ4)

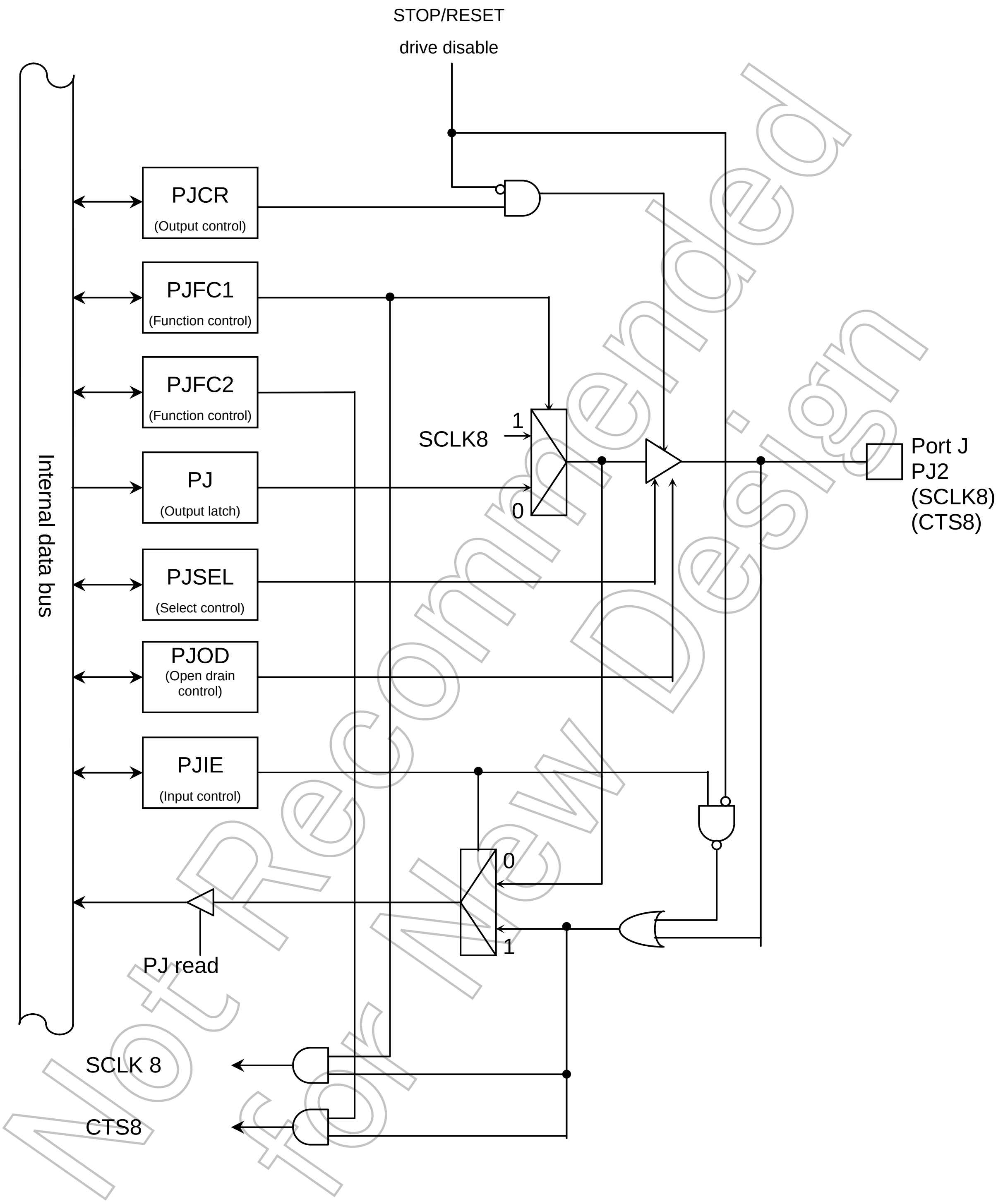


Fig. 7.41 Port J (PJ2)

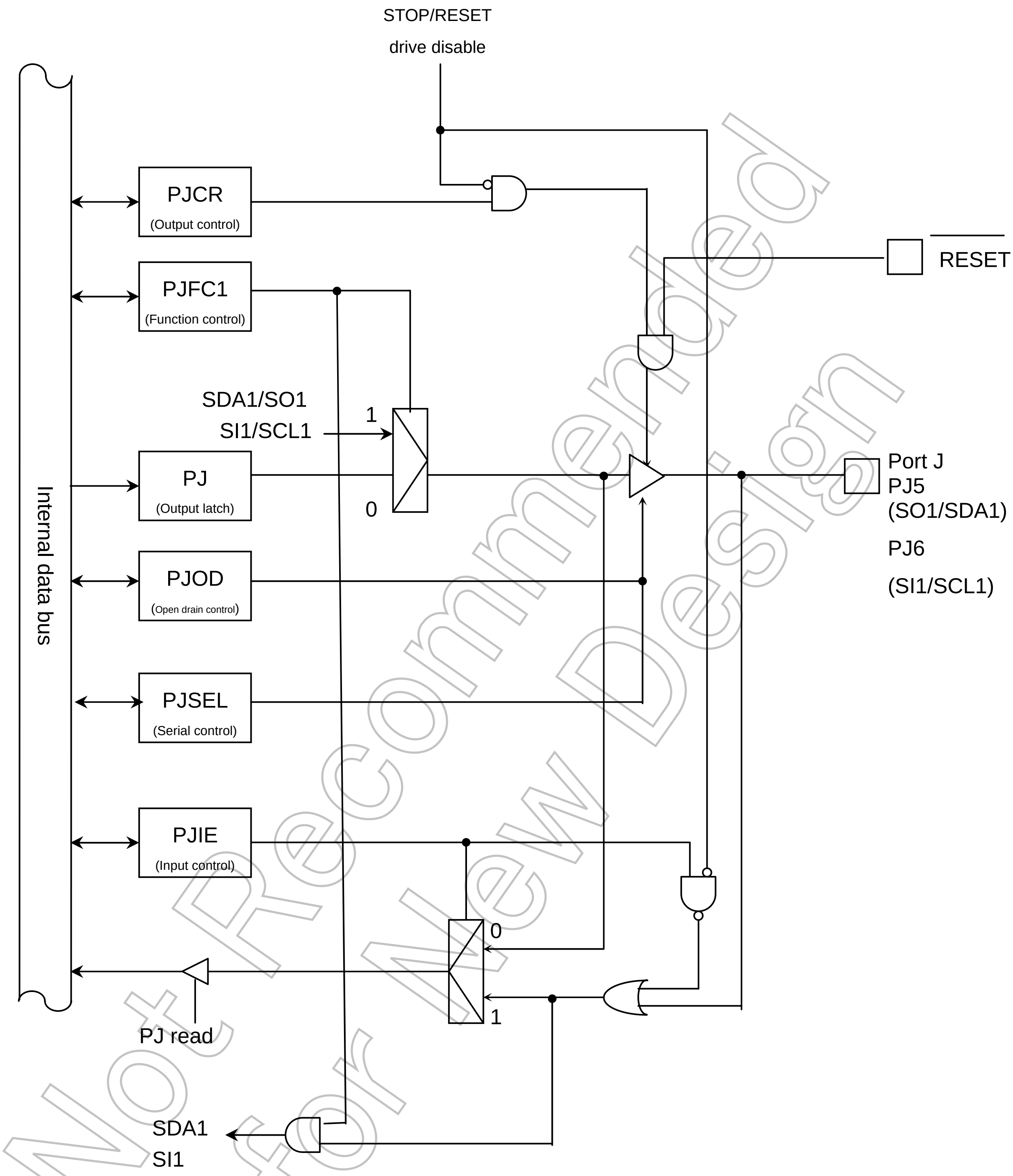


Fig. 7.42 Port J (PJ5,PJ6)

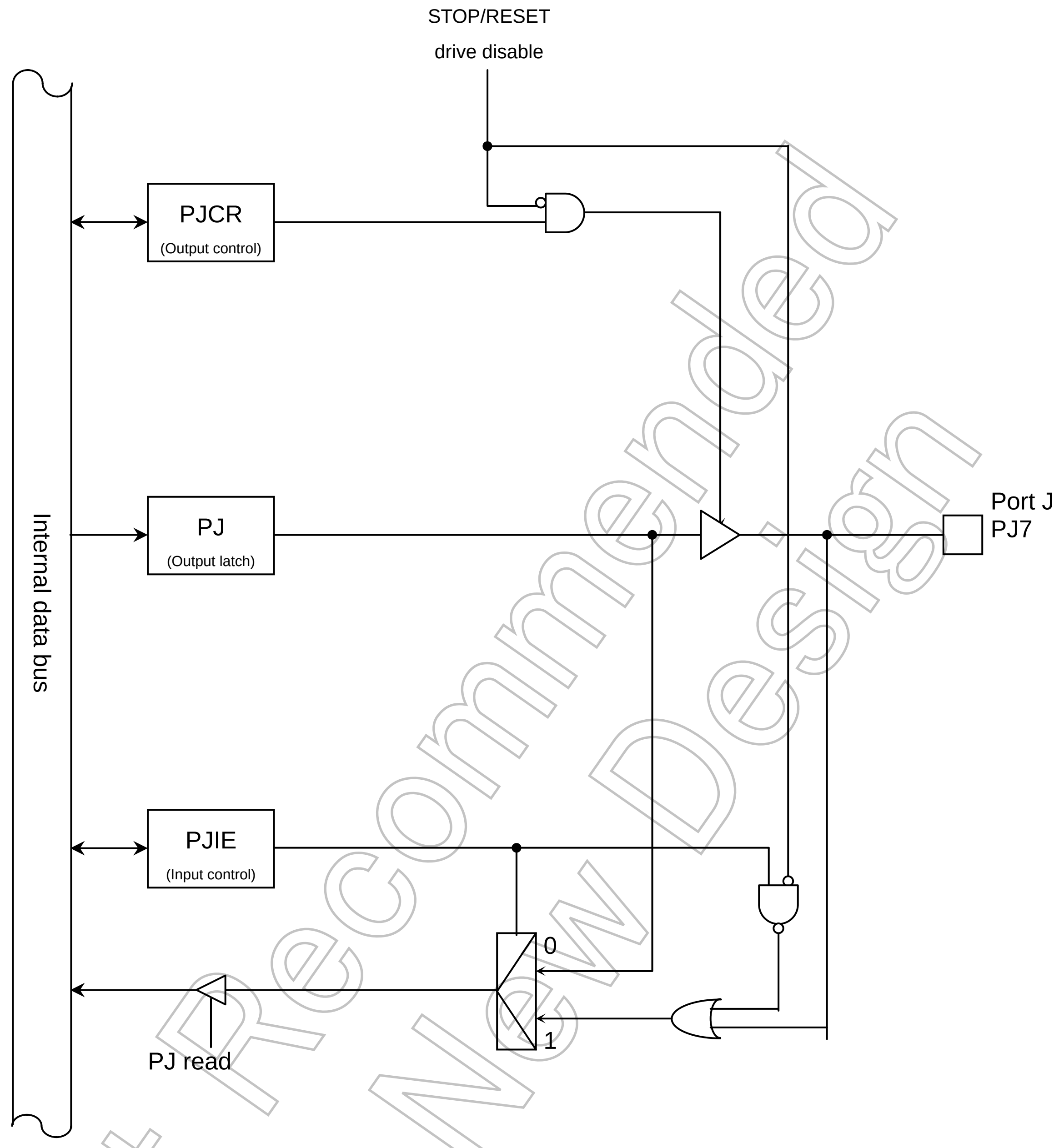


Fig. 7.43 Port J (PJ7)

Port J register

PJ (0xFFFF_F130)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
	Read/Write	R/W							
	After reset	Input mode (output latch register is set to “1.”)							

Port J control register

PJCR (0xFFFF_F131)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7C	PJ6C	PJ5C	PJ4C	PJ3C	PJ2C	PJ1C	PJ0C
	Read/Write	R/ W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: input 1: output							

Port J function register 1

PJFC1 (0xFFFF_F132)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7F1	PJ6F1	PJ5F1	PJ4F1	PJ3F1	PJ2F1	PJ1F1	PJ0F1
	Read/Write	R / W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:Port 1:SCK1	0:Port 1:SI1	0:Port 1:SO1	0:Port 1:TC1IN	0:Port 1:TC0IN	0:Port 1:SCLK8	0:Port 1:RXD8	0:Port 1:TXD8

Port J function register 2

PJFC2 (0xFFFF_F133)		7	6	5	4	3	2	1	0
	Bit Symbol						PJ2F2		
	Read/Write	R / W							
	After reset						0		
	Function						0:Port 1:CTS8		

Port J open drain (OD) control register

PJOD (0xFFFF_F13A)		7	6	5	4	3	2	1	0
	Bit Symbol		PJ6OD	PJ5OD			PJ2OD		PJ0OD
	Read/Write	R/W							
	After reset		0	0			0		0
	Function		0:CMOS 1:OD	0:CMOS 1:OD			0:CMOS 1:OD		0:CMOS 1:OD

Port J select control register

PJSEL (0xFFFF_F13D)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7SEL	PJ6SEL	PJ5SEL	PJ4SEL		PJ2SEL		PJ0SEL
	Read/Write	R/W							
	After reset	0	0	0	0	0	0		0
	Function	SIO1 0: off 1:SCK1	SIO1 0: off 1:SI1	SIO1 0: off 1:SO1			SCLK6 0: off 1:SCLK		TXD6 0: off 1:TXD

Port J input control register

PJIE (0xFFFF_F13E)		7	6	5	4	3	2	1	0
	Bit Symbol	PJ7EI	PJ6EI	PJ5EI	PJ4EI	PJ3EI	PJ2EI	PJ1EI	PJ0EI
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.7 Port K (PK0~PK7)

The port K is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PKCR and the function register PKFCx. A reset allows all bits of the PK, PKCR and PKFC1 to be set to "0," and the port K to be put in output disable mode.

Besides the input/output port function, the port K has Key on wake-up function (KEY0 through KEY7).

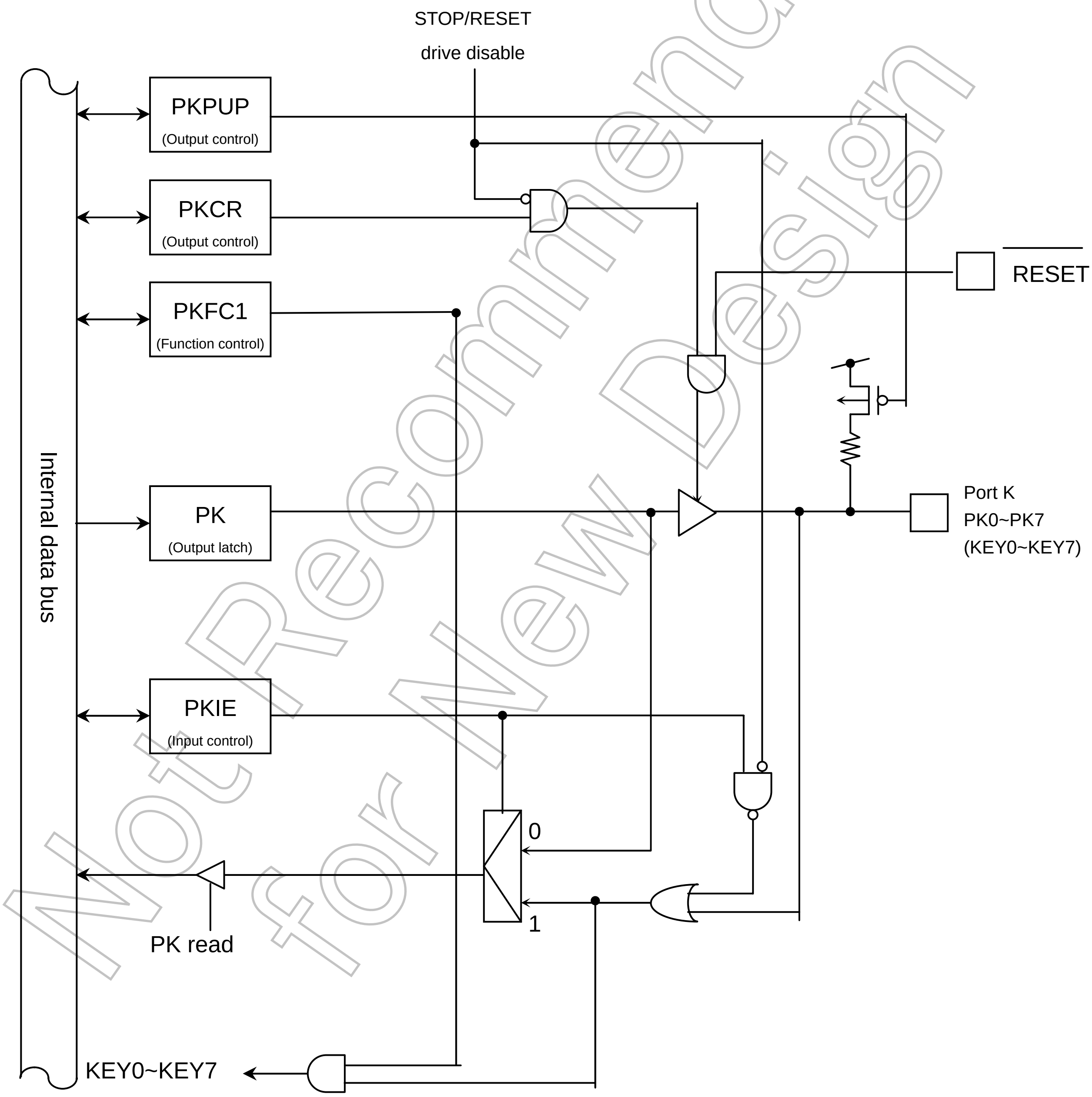


Fig. 7.44 Port K (PK0~PK7)

Port K register

PK (0xFFFF_F140)

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

Port K control register

PKCR (0xFFFF_F141)

	7	6	5	4	3	2	1	0
Bit Symbol	PK7C	PK6C	PK5C	PK4C	PK3C	PK2C	PK1C	PK0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0: output disable 1: output enable							

Port K function register 1

PKFC1 (0xFFFF_F142)

	7	6	5	4	3	2	1	0
Bit Symbol	PK7F1	PK6F1	PK5F1	PK4F1	PK3F1	PK2F1	PK1F1	PK0F1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:Port 1:KEY7	0:Port 1:KEY6	0:Port 1:KEY5	0:Port 1:KEY4	0:Port 1:KEY3	0:Port 1:KEY2	0:Port 1:KEY1	0:Port 1:KEY0

Port K Pull-up control register

PKPUP (0xFFFF_F14B)

	7	6	5	4	3	2	1	0
Bit Symbol	PK7UP	PK6UP	PK5UP	PK4UP	PK3UP	PK2UP	PK1UP	PK0UP
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 K Input control register

PKIE (0xFFFF_F14E)

	7	6	5	4	3	2	1	0
Bit Symbol	PK7EI	PK6EI	PK5EI	PK4EI	PK3EI	PK2EI	PK1EI	PK0EI
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.22 Port L (PL0~PL7)

The port L is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PLCR and the function register PLFCx. A reset allows all bits of the PL, PLCR, PLFC1 and PLFC2 to be cleared to "0," and the port L to be put in output disable mode.

Besides the input/output port function, the port L performs other functions: PL0 and PL1 input 32-bit timer capture trigger. PL3 and PL7 output 32-bit timer compare match. PL4 through PL6 have UART/SION function.

If the port L is used as a port UART/SIO function, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set as the port, please do not set any other function.

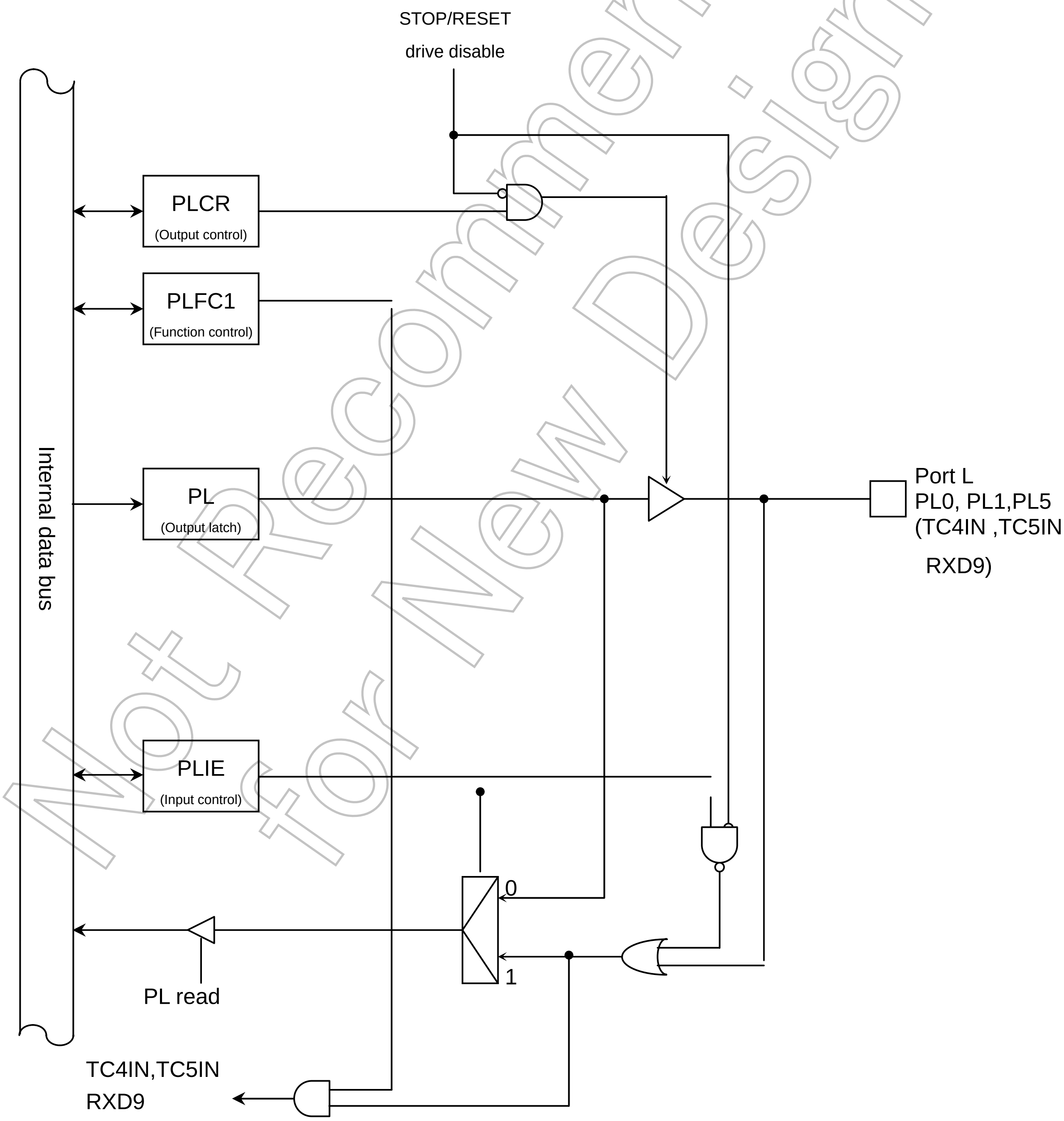


Fig. 7.45 Port L (PL0,PL1)

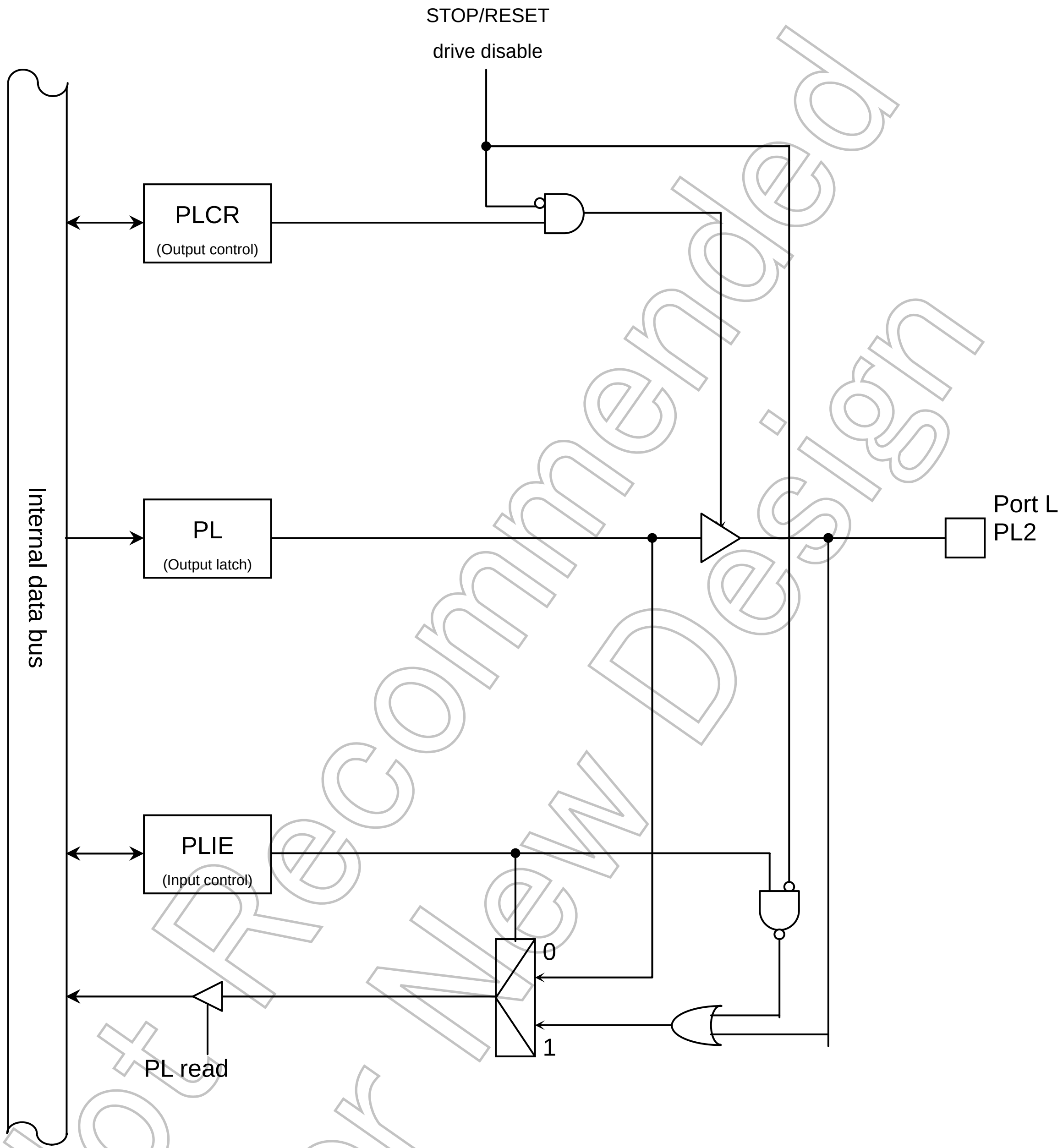


Fig. 7.46 Port L (PL2)

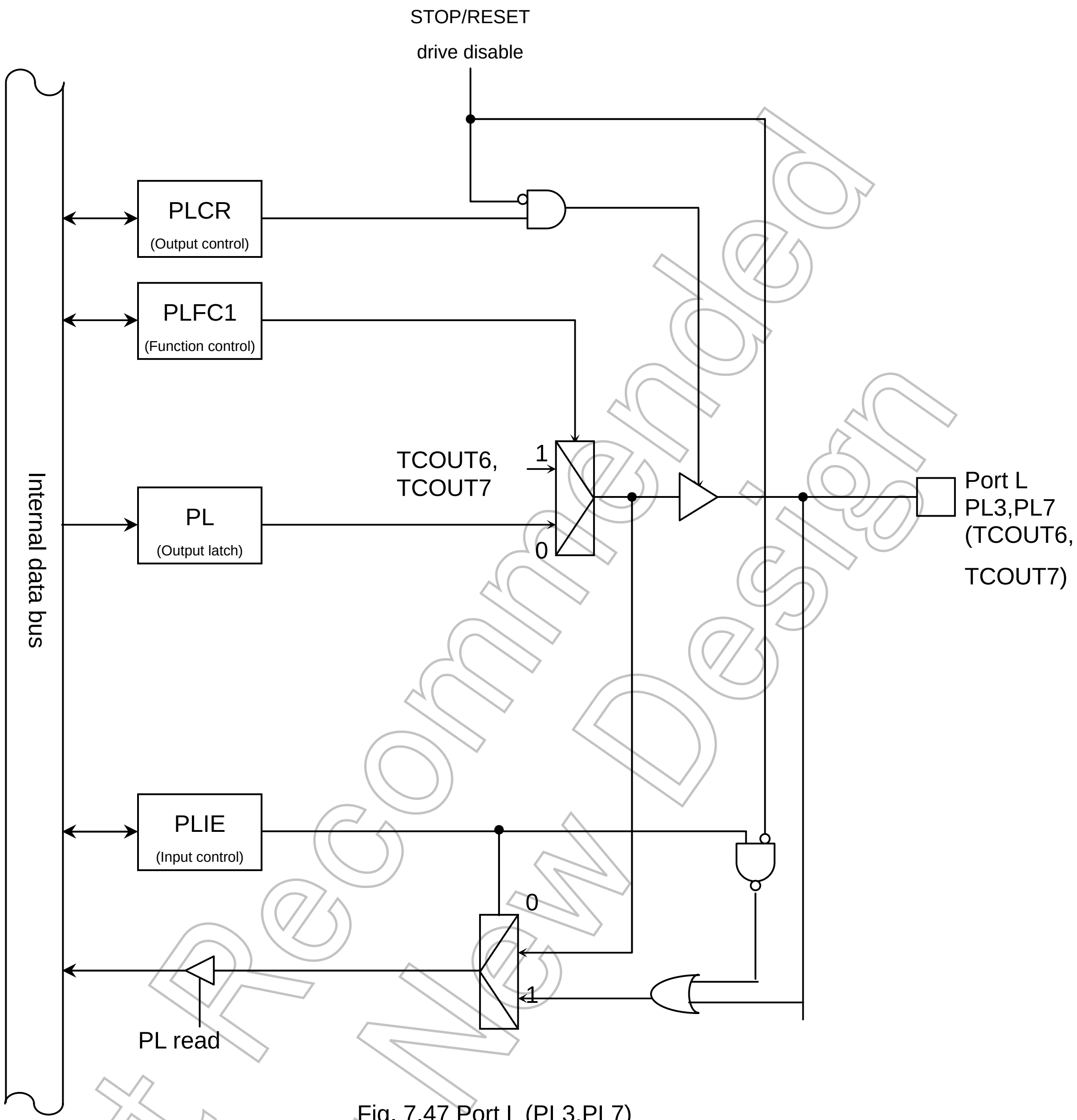


Fig. 7.47 Port L (PL3,PL7)

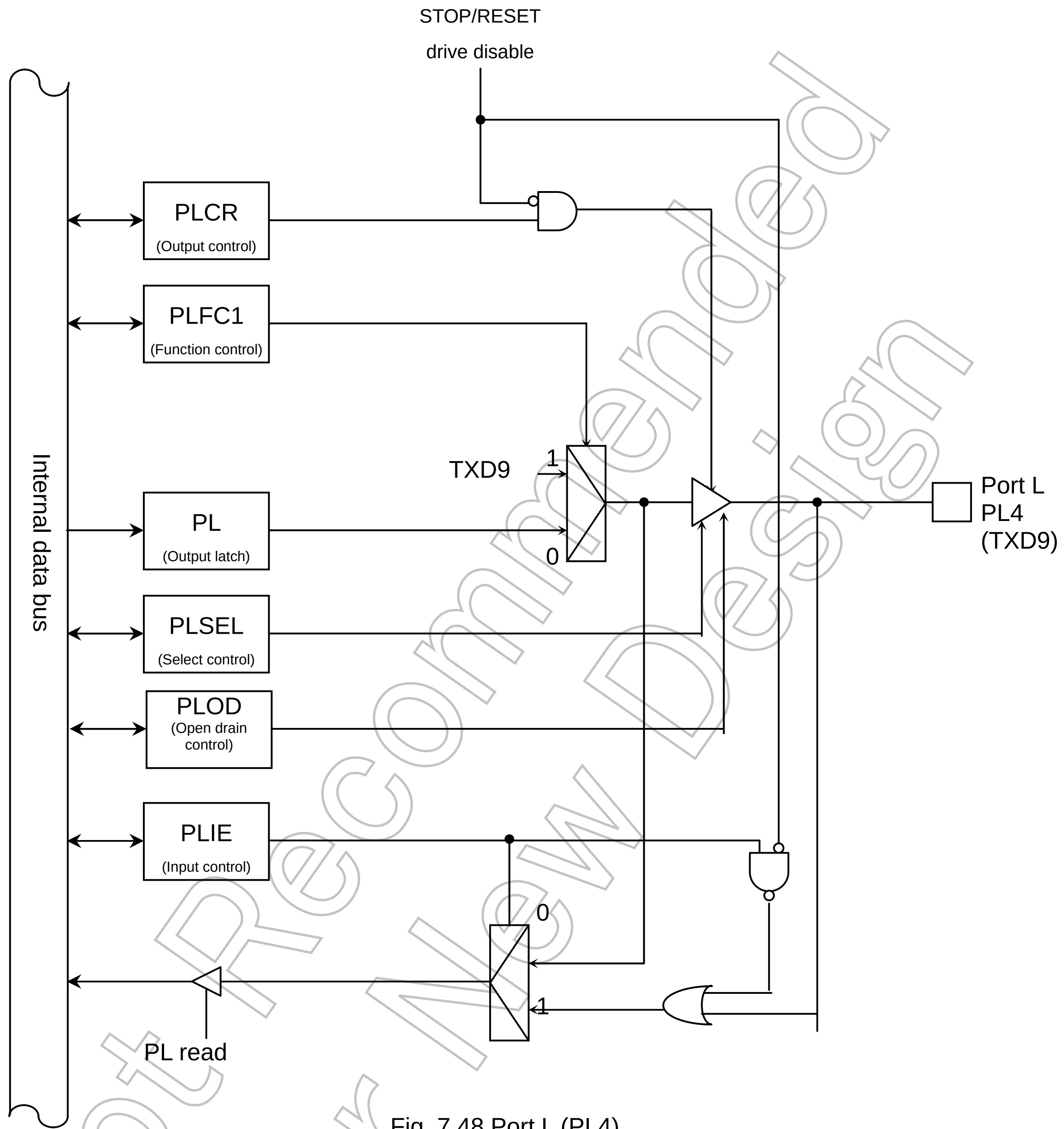


Fig. 7.48 Port L (PL4)

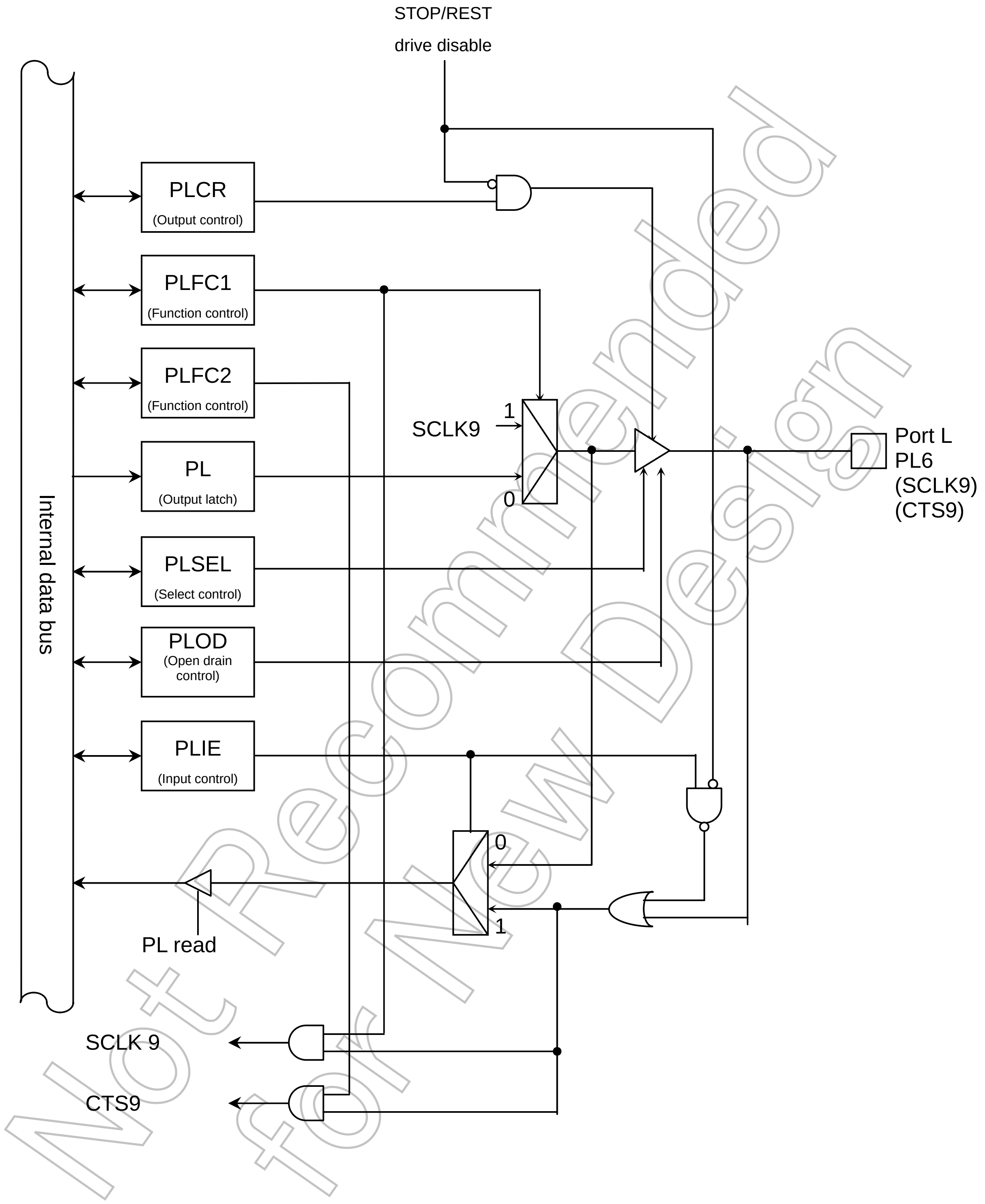


Fig. 7.49 Port L (PL6)

Port L register

PL (0xFFFF_F150)

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

Port L control register

PLCR (0xFFFF_F151)

	7	6	5	4	3	2	1	0
Bit Symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
Read/Write	R/ W							
After reset	0	0	0	0	0	0	0	0
Function	0: input 1: output							

Port L function register 1

PLFC1 (0xFFFF_F152)

	7	6	5	4	3	2	1	0
Bit Symbol	PL7F1	PL6F1	PL5F1	PL4F1	PL3F1	PL2F1	PL1F1	PL0F1
Read/Write	R / W							
After reset	0	0	0	0	0	0	0	0
Function	0:Port 1:TCOUT7	0:Port 1:SCLK9	0:Port 1:RXD9	0:Port 1:TXD9	0:Port 1:TCOUT6		0:Port 1:TC5IN	0:Port 1:TC4IN

Port L function register 2

PLFC2 (0xFFFF_F153)

	7	6	5	4	3	2	1	0
Bit Symbol		PL6F2						
Read/Write	R / W							
After reset		0						
Function		0:Port 1:CTS9						

Port L open drain (OD) control register

PLOD (0xFFFF_F15A)

	7	6	5	4	3	2	1	0
Bit Symbol		PL6OD		PL4OD				
Read/Write	R/W							
After reset		0		0				
Function		0:CMOS 1: OD		0:CMOS 1: OD				

Port L select control register

PLSEL (0xFFFF_F15D)

	7	6	5	4	3	2	1	0
Bit Symbol		PL6SEL		PL4SEL				
Read/Write	R/W							
After reset		0		0				
Function		SCLK7 0: off 1:SCLK		TXD7 0: off 1:TXD				

Port L input control register

		7	6	5	4	3	2	1	0
PLIE	Bit Symbol	PL7IE	PL6IE	PL5IE	PL4IE	PL3IE	PL2IE	PL1IE	PL0IE
(0xFFFF_F15E)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.23 Port M (PM0~PM7)

The port M is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PMCR and the function register PMFCx. A reset allows all bits of the PM, PMCR, PMFC1 to be cleared to "0," and the port M to be put in output disable mode.

Besides the input/output port function, the port M performs other functions: PM0 through PM5 input external interrupt (INT0~INT5). PM6 and PM7 output 32-bit timer compare match.

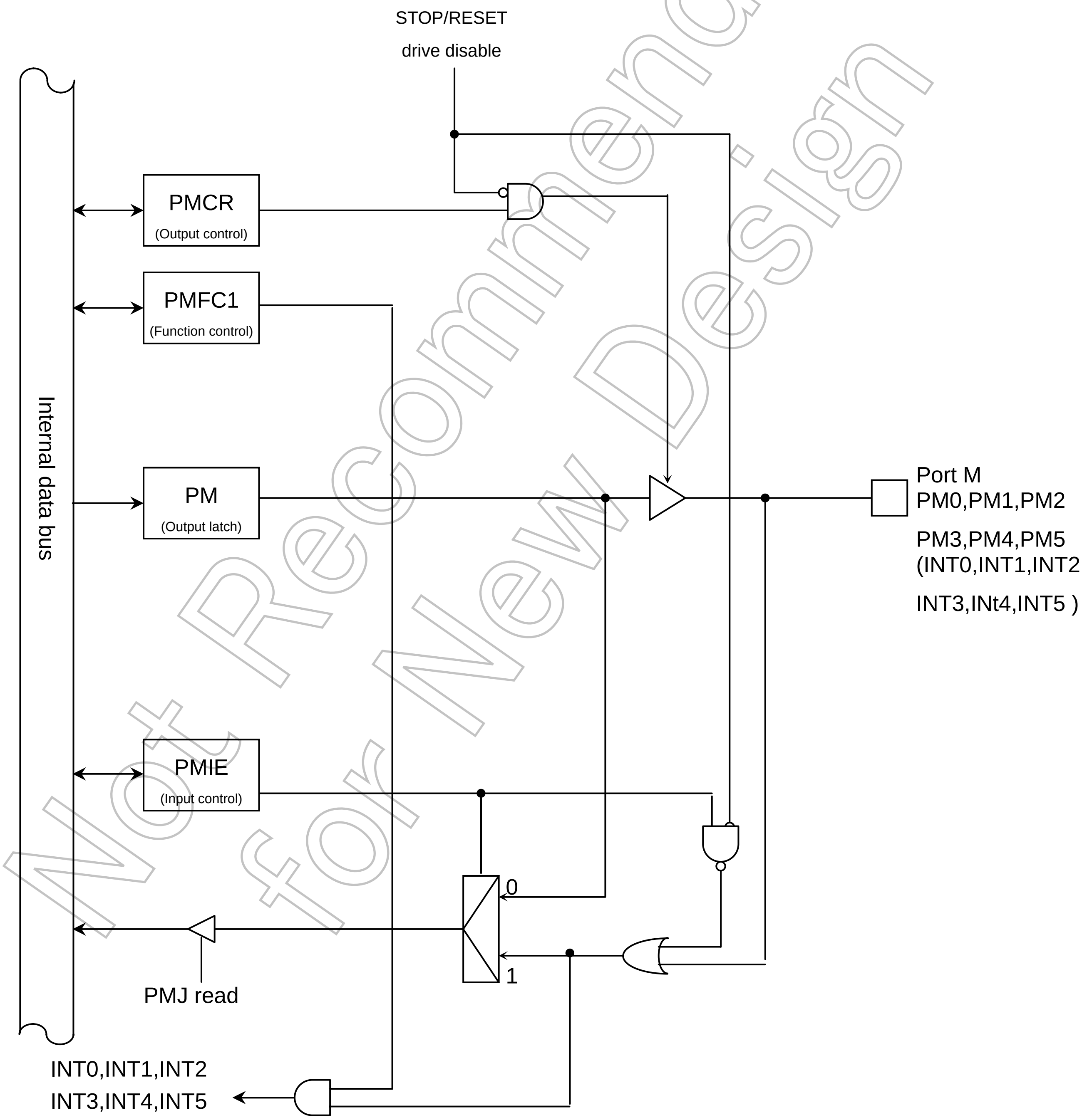


Fig. 7.50 Port M (PM0,PM1,PM2,PM3,PM4,PM5)

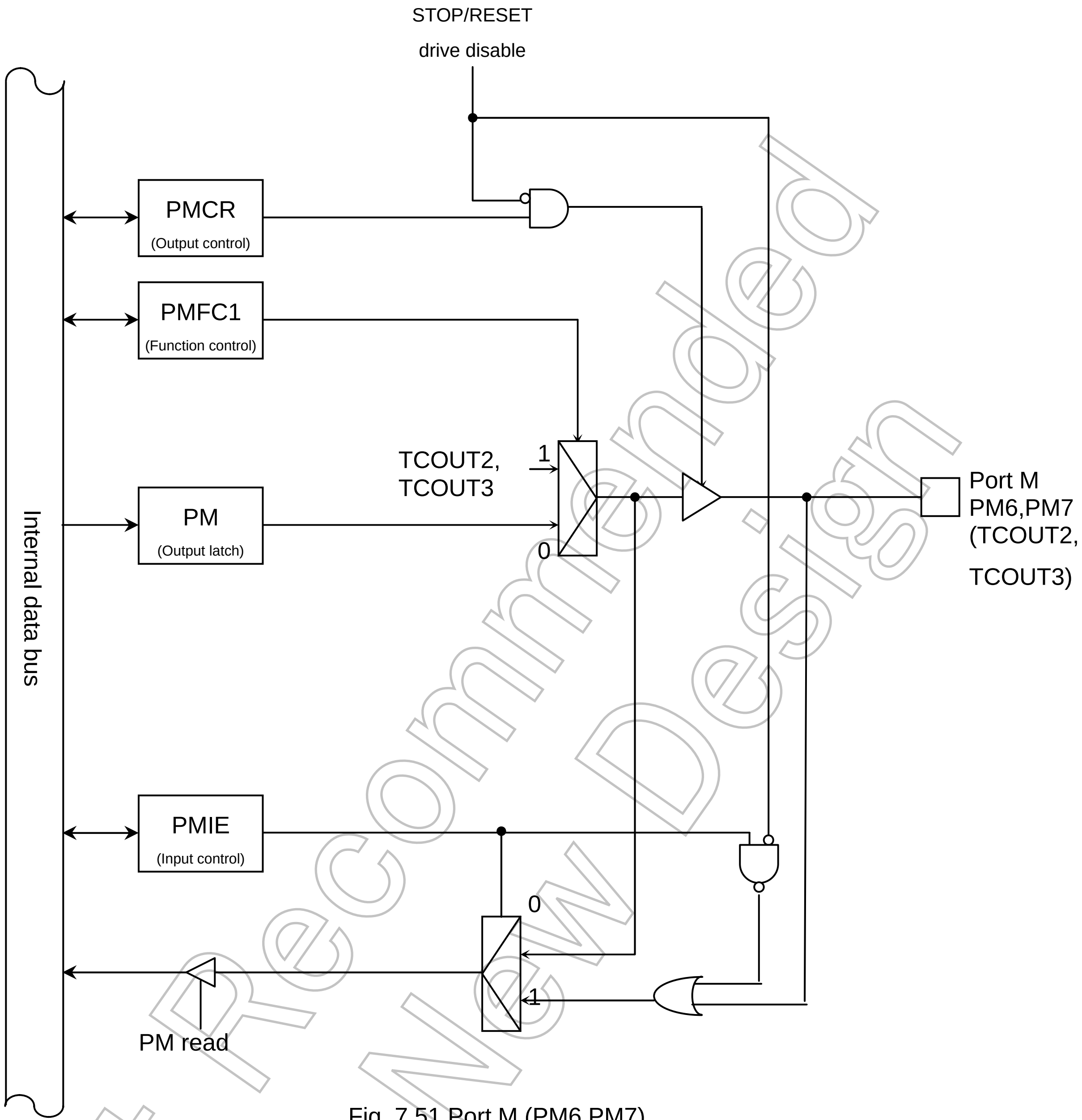


Fig. 7.51 Port M (PM6,PM7)

Port M register

PM (0xFFFF_F160)

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

Port M control register

PMCR (0xFFFF_F161)

	7	6	5	4	3	2	1	0
Bit Symbol	PM7C	PM6C	PM5C	PM4C	PM3C	PM2C	PM1C	PM0C
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:output disable 1: output enable							

Port M function register 1

PMFC1 (0xFFFF_F162)

	7	6	5	4	3	2	1	0
Bit Symbol	PM7F1	PM6F1	PM5F1	PM4F1	PM3F1	PM2F1	PM1F1	PM0F1
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	0:Port 1:TCOUT3	0:Port 1:TCOUT2	0:Port 1:INT5	0:Port 1:INT4	0:Port 1:INT3	0:Port 1:INT2	0:Port 1:INT1	0:Port 1:INT0

Port M input control register

PMIE (0xFFFF_F16E)

	7	6	5	4	3	2	1	0
Bit Symbol	PEIM7	PEIM6	PEIM5	PEIM4	PEIM3	PEIM2	PEIM1	PEIM0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.23 Port N (PN0~PN7)

The port N is a general-purpose, 8-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PNCR and the function register PNFCx. A reset allows all bits of the PN, PNCR, PNFC1 and PNFC2 to be cleared to "0," and the port N to be put in output disable mode.

Besides the input/output port function, the port N performs other functions: PN0 through PN2 input external interrupt (INT6~INT8). PN3 and PN7 input A/D converter external start request. PL4 through PL6 have UART/SIO function.

If the port N is used as a port UART/SIO function, select the function with the function register of the corresponding port and set the open drain control register along with the select control register. When the function registers 1 and 2 are set, the setting in the register 1 is prioritized.

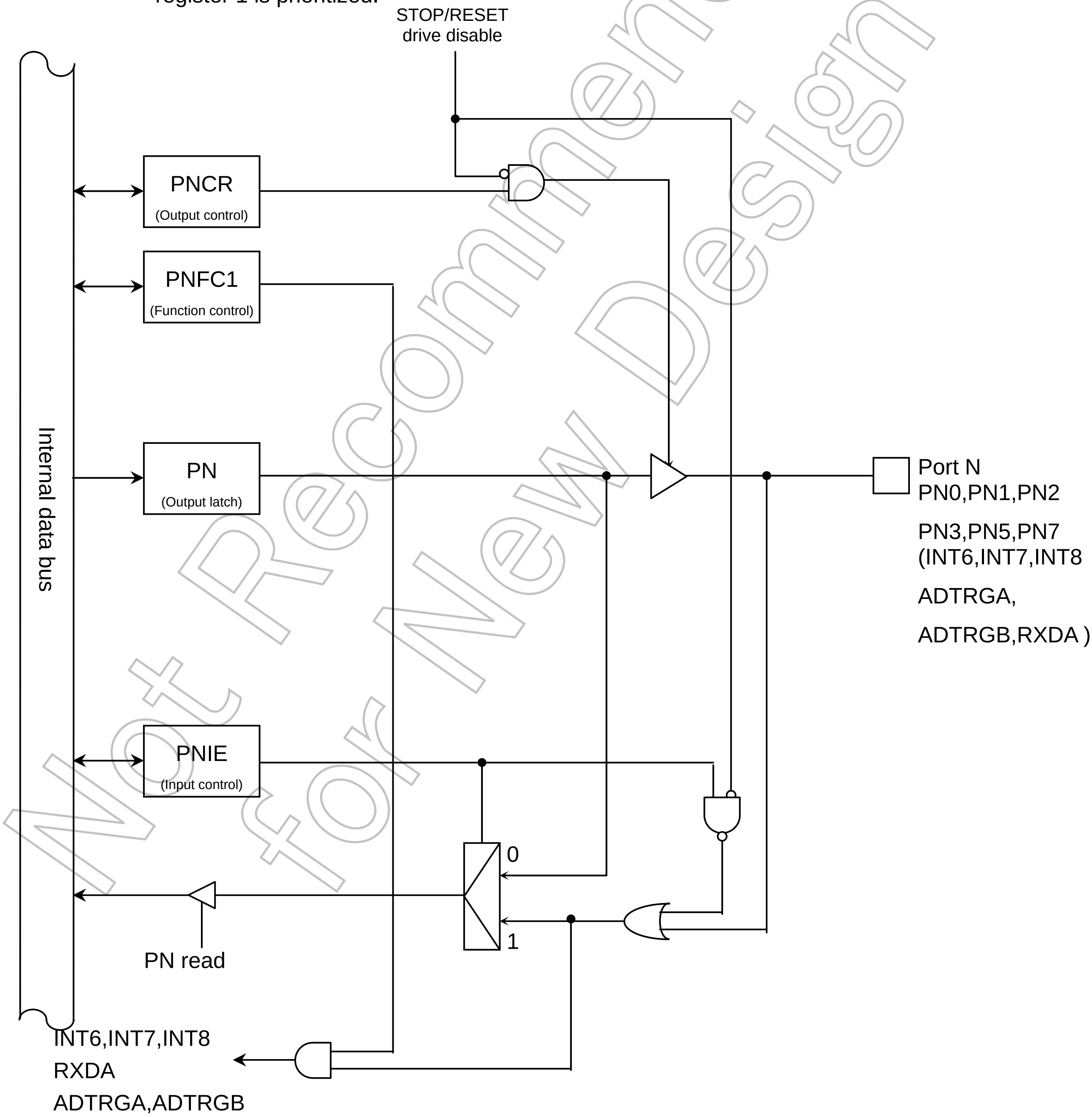


Fig. 7.52 Port N (PN0,PN1,PN2,PN3,PN5,PN7)

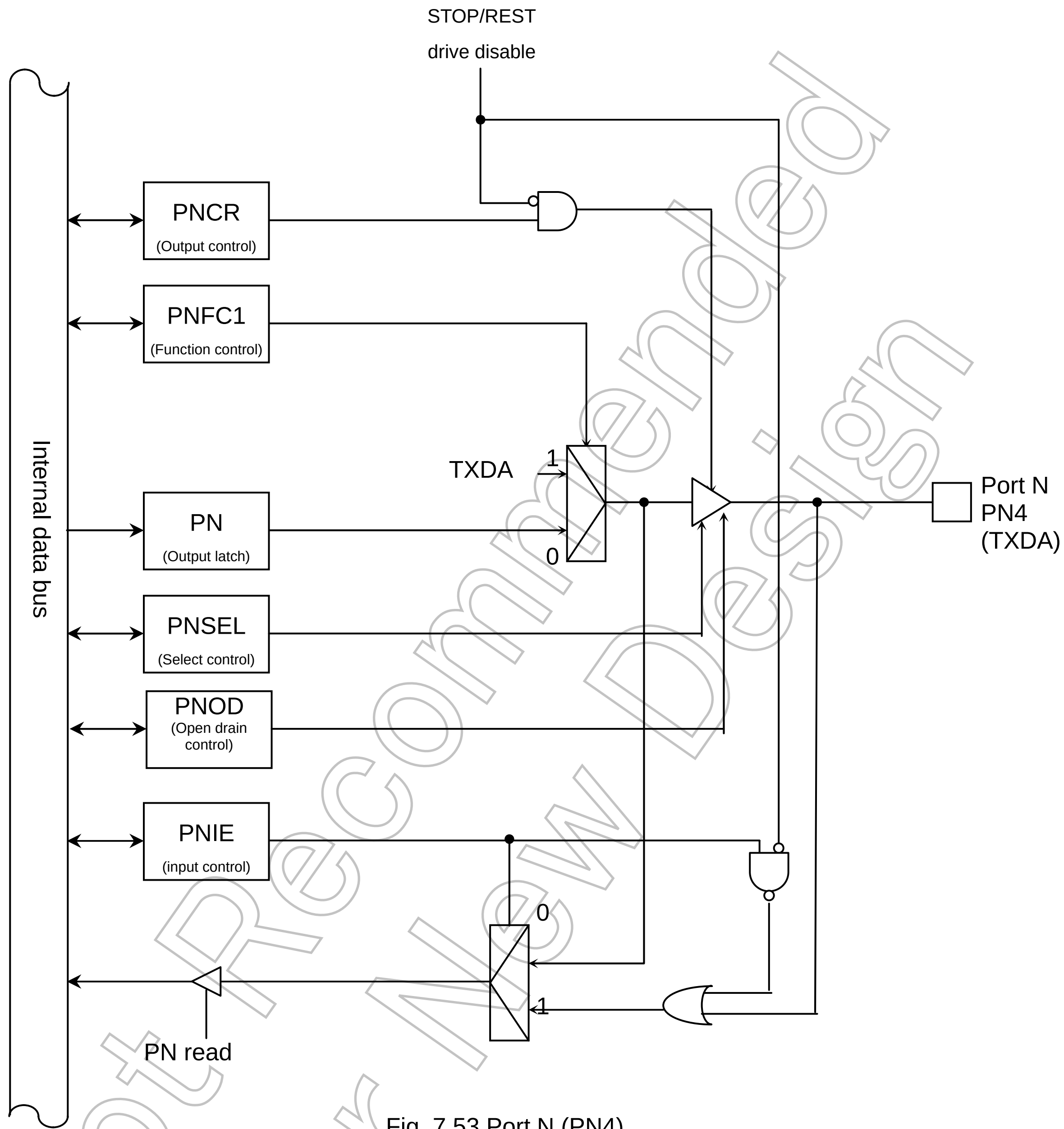


Fig. 7.53 Port N (PN4)

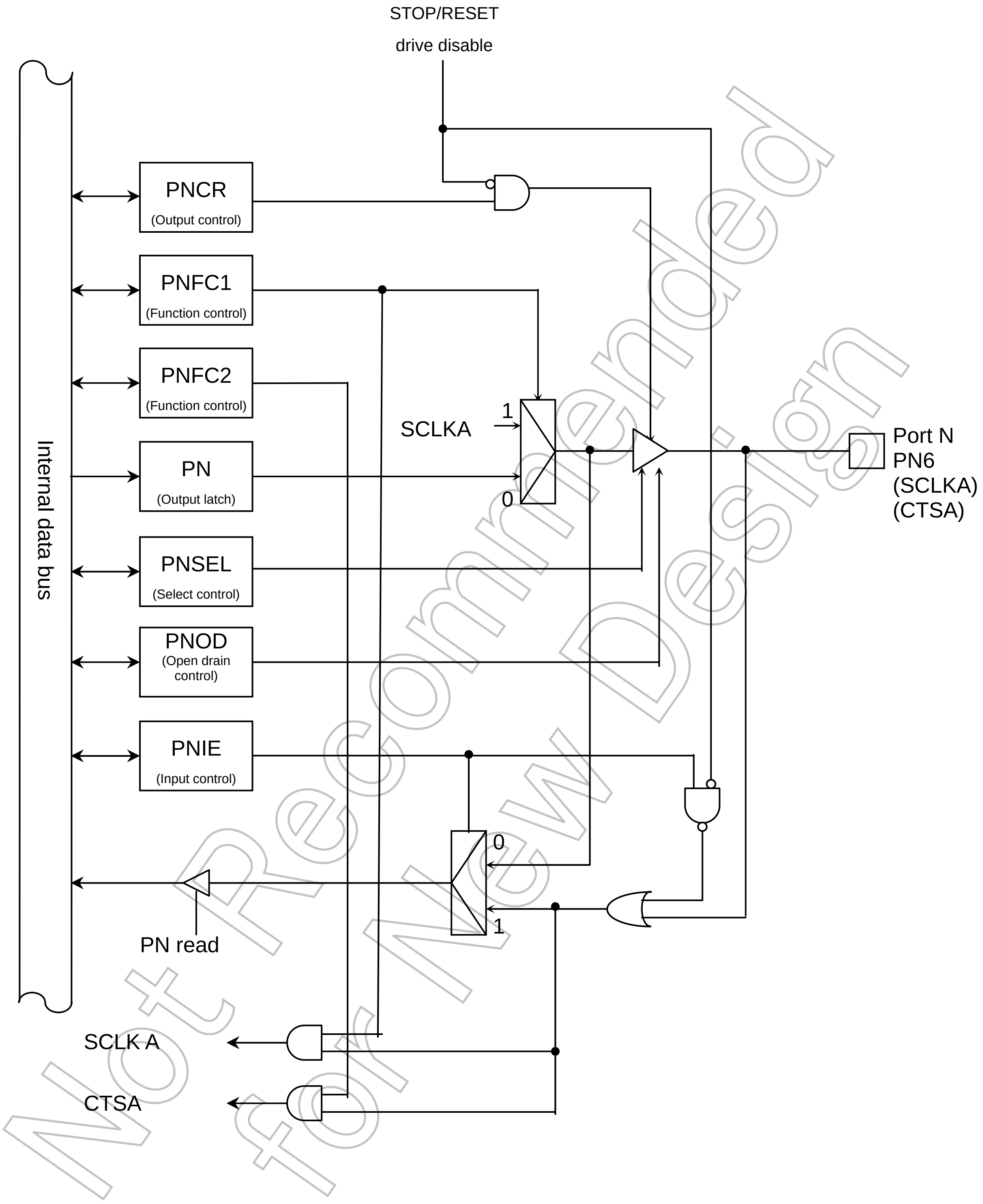


Fig. 7.54 Port N (PN6)

Port N register

PN (0xFFFF_F170)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7	PN6	PN5	PN4	PN3	PN2	PN1	PN0
	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port N control register

PNCR (0xFFFF_F171)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7C	PN6C	PN5C	PN4C	PN3C	PN2C	PN1C	PN0C
	Read/Write	R/ W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: input 1: output							

Port N function register 1

PNFC1 (0xFFFF_F172)		7	6	5	4	3	2	1	0
	Bit Symbol	PN7F1	PN6F1	PN5F1	PN4F1	PN3F1	PN2F1	PN1F1	PN0F1
	Read/Write	R / W							
	After reset	0	0	0	0	0	0	0	0
	Function	0:Port 1:ADTRGB	0:Port 1:SCLKA	0:Port 1:RXDA	0:Port 1:TXDA	0:Port 1:ADTRGA	0:Port 1:INT8	0:Port 1:INT7	0:Port 1:INT6

Port N function register 2

PNFC2 (0xFFFF_F173)		7	6	5	4	3	2	1	0
	Bit Symbol		PN6F2						
	Read/Write	R / W							
	After reset		0						
	Function		0:Port 1:CTSA						

Port N open drain (OD) control register

PNOD (0xFFFF_F17A)		7	6	5	4	3	2	1	0
	Bit Symbol		PN6OD		PN4OD				
	Read/Write	R/W							
	After reset		0		0				
	Function		0:CMOS 1:OD		0:CMOS 1:OD				

Port N select control register

PNSEL (0xFFFF_F17D)		7	6	5	4	3	2	1	0
	Bit Symbol		PN6SEL		PN4SEL				
	Read/Write	R/W							
	After reset		0		0				
	Function		SCLKA 0: off 1:SCLK		TXDA 0: off 1:TXD				

Port N input control register

		7	6	5	4	3	2	1	0
PNIE	Bit Symbol	PN7IE	PN6IE	PN5IE	PN4IE	PN3IE	PN2IE	PN1IE	PN0IE
(0xFFFF_F17E)	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.24 Port O (PO0~PO7)

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

Besides the input/output port function, the port O performs other functions: PO0 through PO7 function as a signal for DSU-ICE (TPD0~TPD7). If the port O is used for the DSU-ICE signal, it cannot be used as the input/output mode.

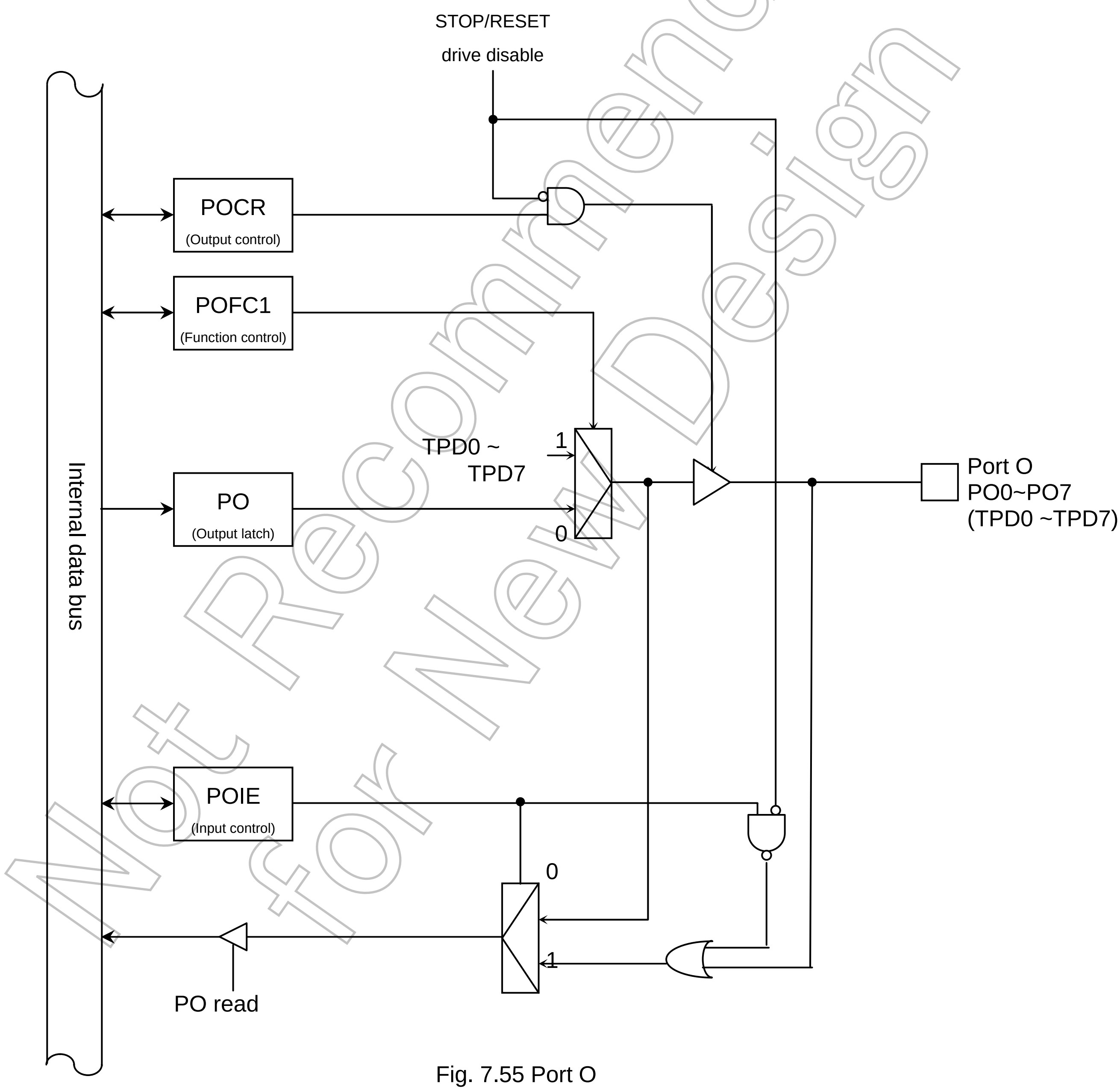


Fig. 7.55 Port O

Port O register

PO (0xFFFF_F180)

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

Port O control register

POCR (0xFFFF_F181)

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

Port O input control register

POIE (0xFFFF_F18E)

	7	6	5	4	3	2	1	0
Bit Symbol	PO7IE	PO6IE	PO5IE	PO4IE	PO3IE	PO2IE	PO1IE	PO0IE
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.25 Port P (PP0~PP7)

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

Besides the input/output port function, the port P performs other functions: PP0 through PP7 function as a signal for DSU-ICE (TPD0~TPD7/ TPC0~TPC7). If the port P is used for the DSU-ICE signal, it cannot be used as the input/output mode.

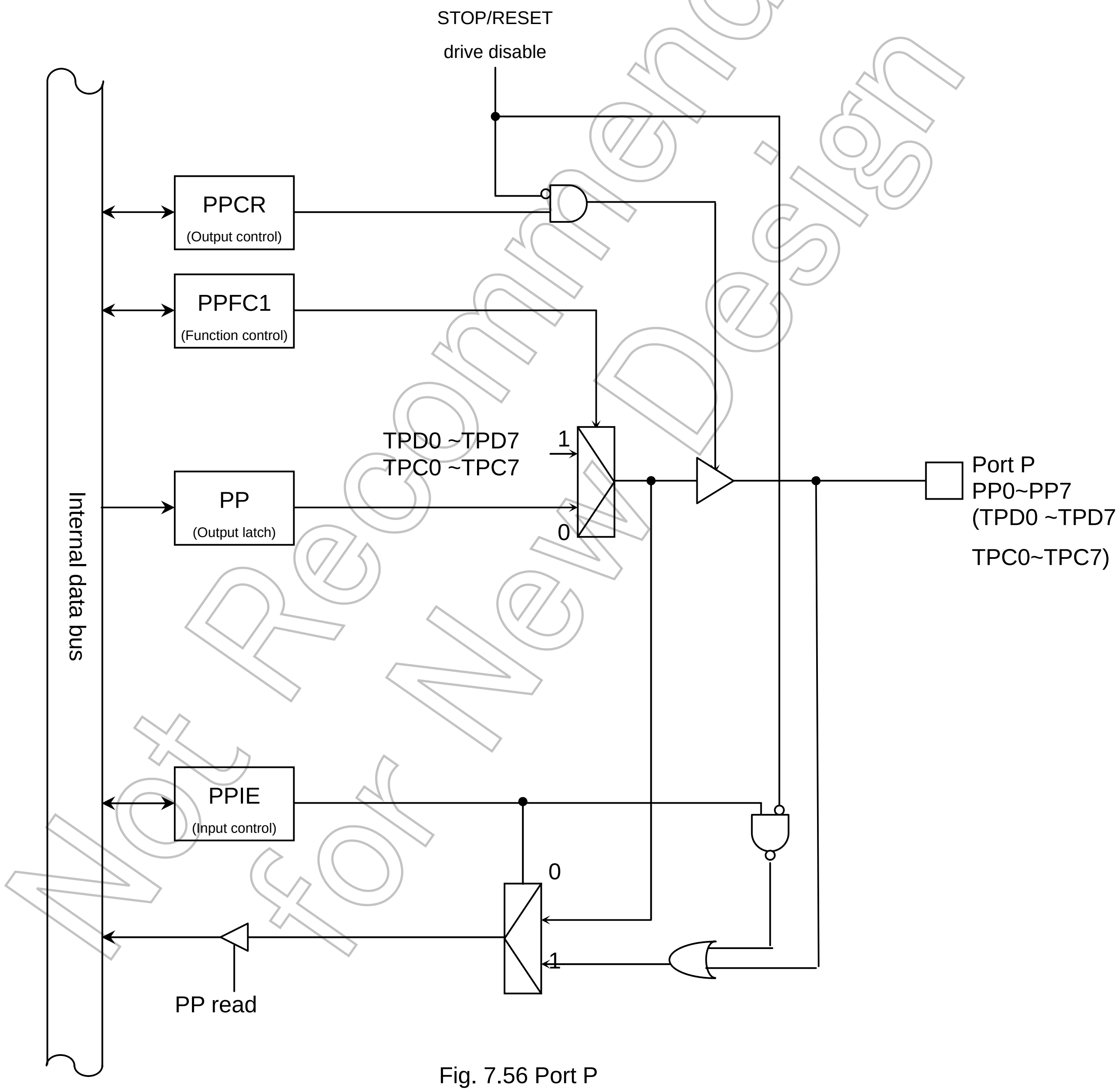


Fig. 7.56 Port P

Port P register

PP (0xFFFF_F190)		7	6	5	4	3	2	1	0
	Bit Symbol	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
	Read/Write	R/W							
	After reset	Input mode (output latch register is set to “1.”)							

Port P control register

PPCR (0xFFFF_F191)		7	6	5	4	3	2	1	0
	Bit Symbol	PP7C	PP6C	PP5C	PP4C	PP3C	PP2C	PP1C	PP0C
	Read/Write	R/ W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: input 1: output							

Port P input control register

PPIE (0xFFFF_F18E)		7	6	5	4	3	2	1	0
	Bit Symbol	PP7IE	PP6IE	PP5IE	PP4IE	PP3IE	PP2IE	PP1IE	PP0IE
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

7.26 Port Q (PQ0~PQ3)

The port Q is a general-purpose, 4-bit input/output port. For this port, inputs and outputs can be specified in units of bits. Outputs can be set by using the control register PQCR and the function register PQFC1. A reset allows all bits of the PQ, PQCR and PQFC1 to be cleared to "0," and the port Q to be put in output disable mode.

Besides the input/output port function, the port Q performs other functions: PQ0 and PQ2 function as DREQ2 and DREQ3. PQ1 and PQ3 function as DACK2 and DACK3.

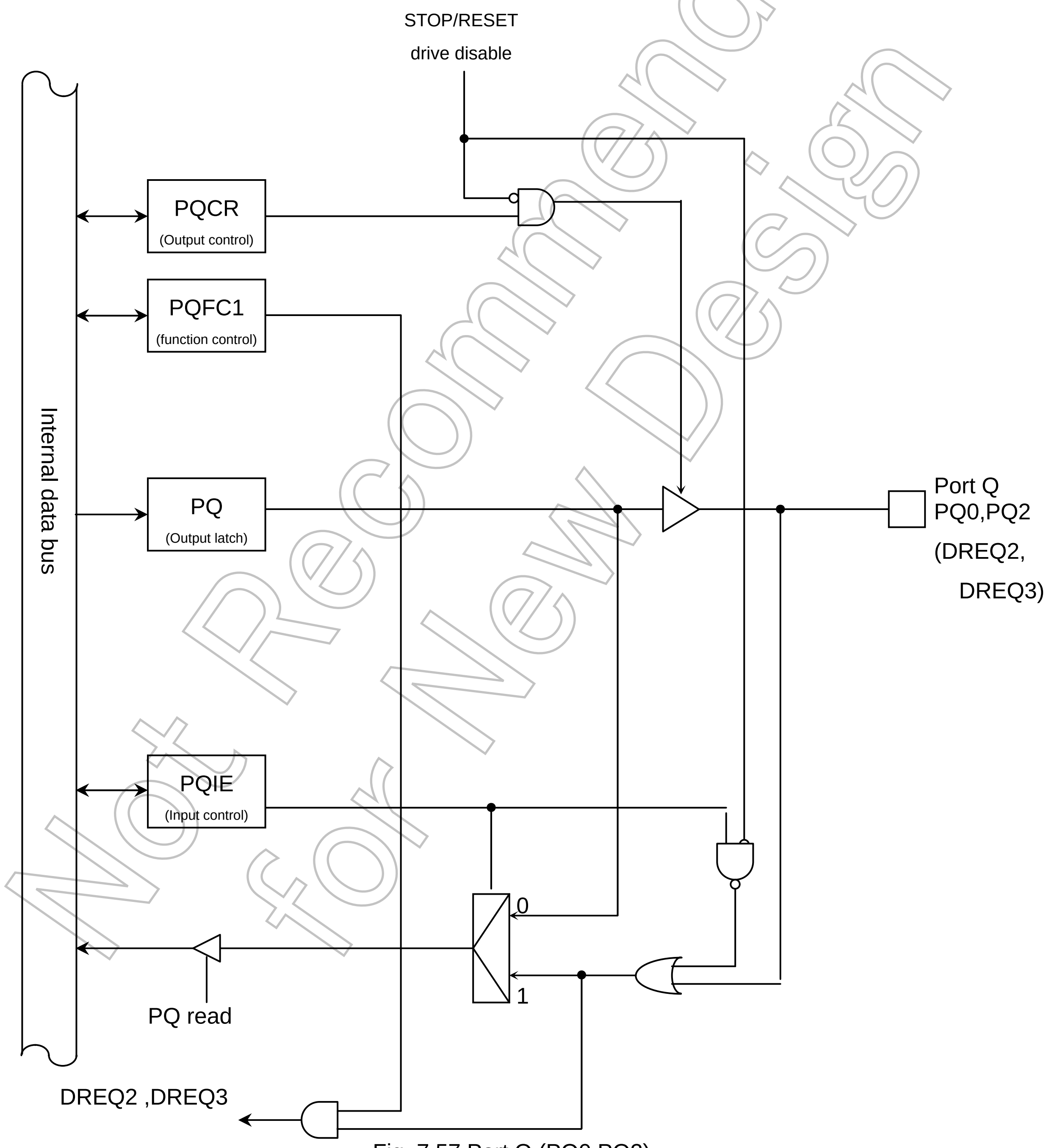


Fig. 7.57 Port Q (PQ0,PQ2)

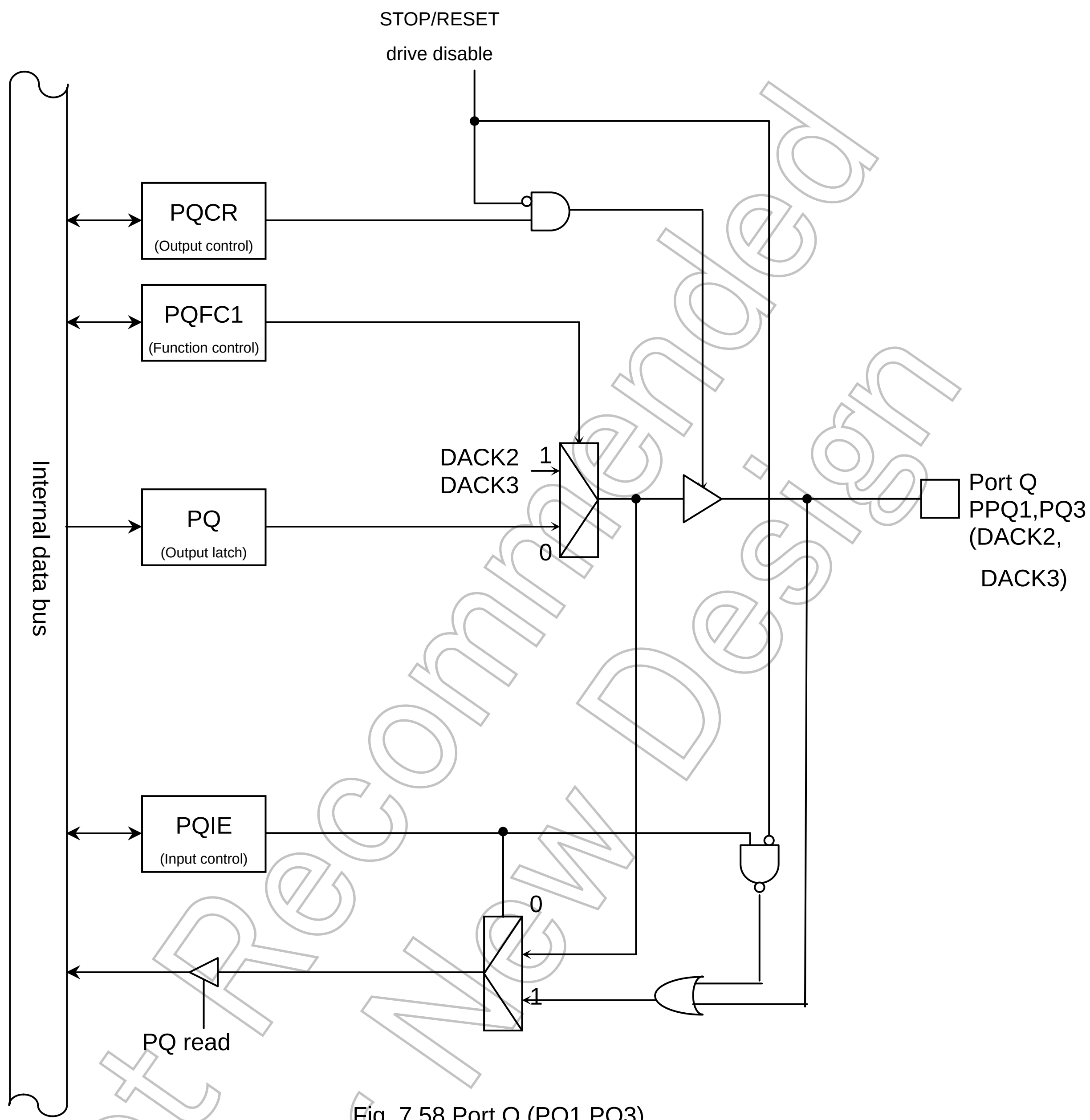


Fig. 7.58 Port Q (PQ1,PQ3)

Port Q register

PQ (0xFFFF_F1A0)		7	6	5	4	3	2	1	0
	Bit Symbol					PQ3	PQ2	PQ1	PQ0
	Read/Write	R/W							
	After reset	Input mode (output latch register is set to "1.")							

Port Q control register

PQCR (0xFFFF_F1A1)		7	6	5	4	3	2	1	0
	Bit Symbol					PQ3C	PQ2C	PQ1C	PQ0C
	Read/Write	R/ W							
	After reset					0	0	0	0
	Function	0: input 1:output							

Port Q function register 1

PQFC1 (0xFFFF_F1A2)		7	6	5	4	3	2	1	0
	Bit Symbol					PQ3F1	PQ2F1	PQ1F1	PQ0F1
	Read/Write	R / W							
	After reset					0	0	0	0
	Function					0:Port 1:DACK3	0:Port 1:DREQ3	0:Port 1:DACK2	0:Port 1:DREQ2

Port Q input control register

PQIE (0xFFFF_F1AE)		7	6	5	4	3	2	1	0
	Bit Symbol					PQ3IE	PQ2IE	PQ1IE	PQ0IE
	Read/Write	R/W							
	After reset					0	0	0	0
	Function					Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled	Input 0: disabled 1: enabled

8.External Bus Interface

The TMP19A63 has a built-in external bus interface function to connect to external memory, I/Os, etc. This interface consists of an external bus interface circuit (EBIF), a chip selector (CS) and a wait controller.

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

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

• External bus mode

Selectable address, data separator bus mode and multiplex mode

• Wait function

This function can be enabled for each block.

- A wait of up to 7 clocks can be automatically inserted.
- A wait can be inserted via the $\overline{\text{WAIT}}$ / $\overline{\text{RDY}}$ pin.

(Data bus width

Either an 8- or 16-bit width can be set for each block.

(Recovery cycle (read/write)

If external bus cycles occur continuously, 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 31 clocks can be inserted and this dummy cycle can be specified for each block.

• Bus arbitration function

.8.1 Address and Data Pins

(1) Address and data pin settings

The TMP19A63 can be set to either separate bus or multiplexed bus mode. Setting the BUSMD pin to the "L" level at a reset activates the separate bus mode, and setting the pin to the "H" level activates the multiplexed bus mode. Port pins 0, 1, 2, 5 and 6, which are to be connected to external devices (memory), are used as address buses, data buses and address/data buses (see Table 8.1).

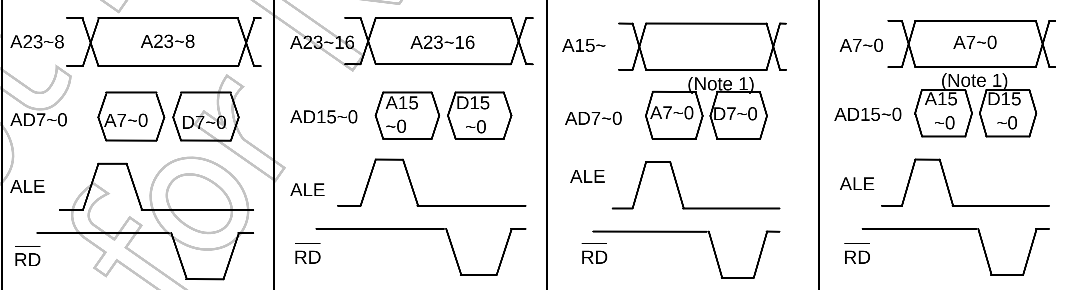
Table 8.1.1 Bus Mode, Address and Data Pins

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

To access an external device, set the address and data bus functions by using the port control register (PnCR) and the port function register (PnFC).

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

Table 8.1.2 Address and Data Pins in the Multiplex Mode

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

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

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

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

(2) Address HOLD when an internal area is accessed

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

8.2 Data Format

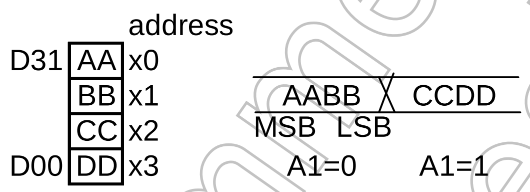
Internal registers and external bus interfaces of the TMP19A63 are configured as described below.

(1) Big-endian mode

① Word access

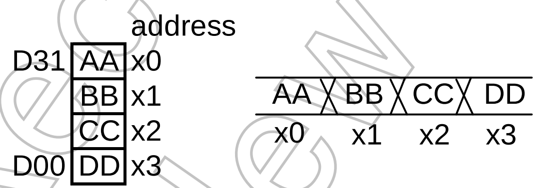
- 16-bit bus width

Internal registers External buses



- 8-bit bus width

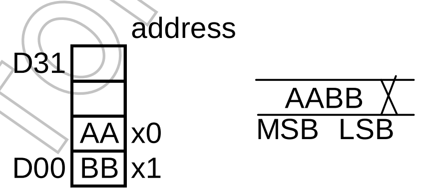
Internal registers External buses



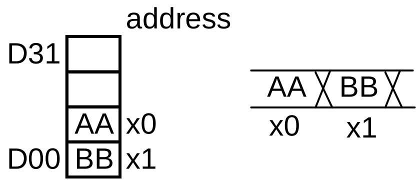
② Half word access

- 16-bit bus width

Internal registers External buses

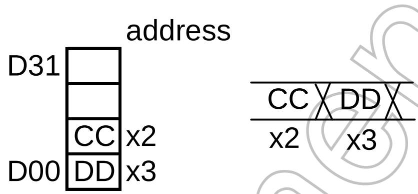


- 8-bit bus width



Internal registers External buses

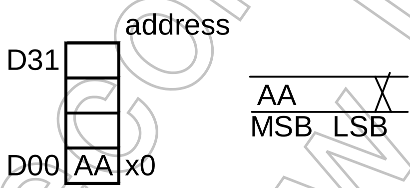
Internal registers External buses

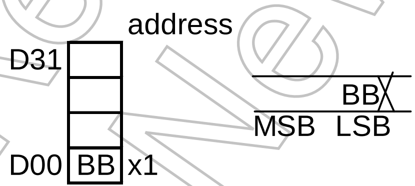


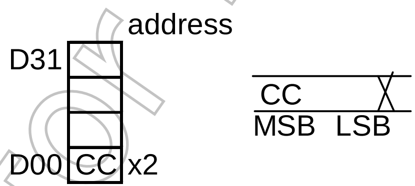
③ Byte access

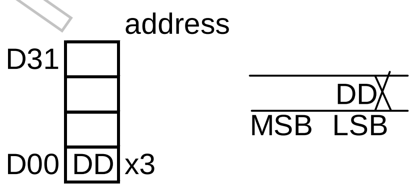
- 16-bit bus width

Internal registers External buses

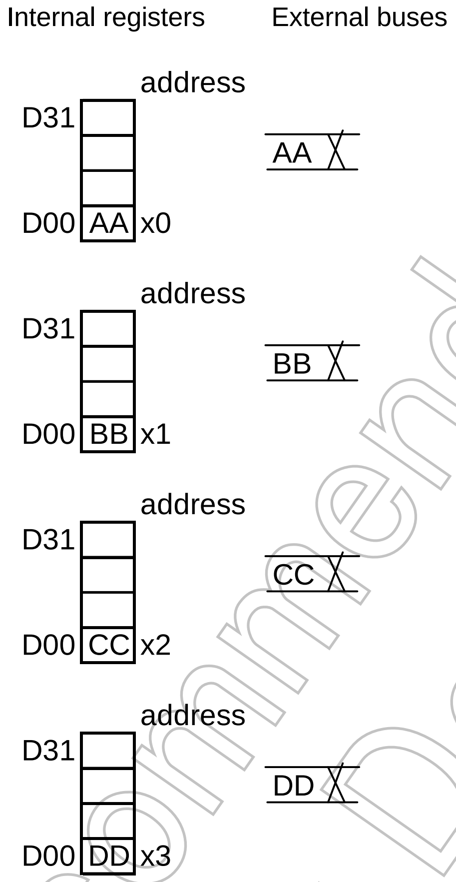








- 8-bit bus width

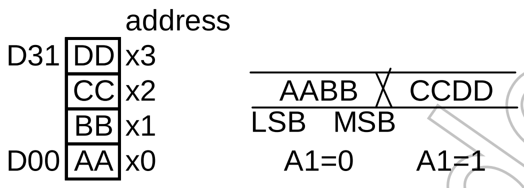


(2) Little-endian mode

① Word access

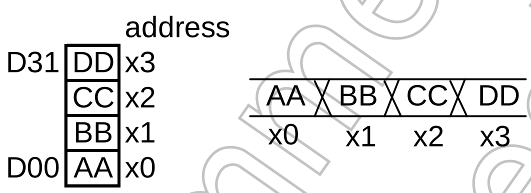
- 16-bit bus width

Internal registers External buses



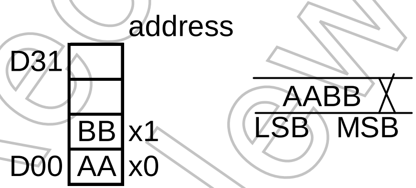
- 8-bit bus width

Internal registers External buses

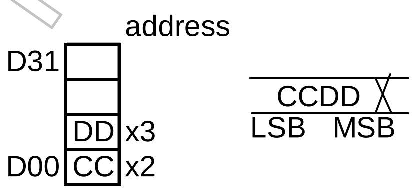


② Half word access

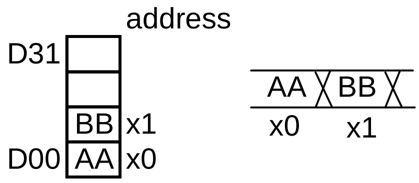
- 16-bit bus width



Internal registers External buses

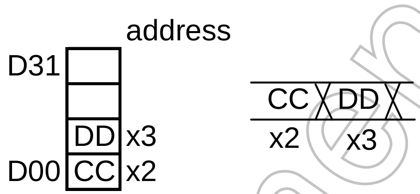


- 8-bit bus width



Internal registers External buses

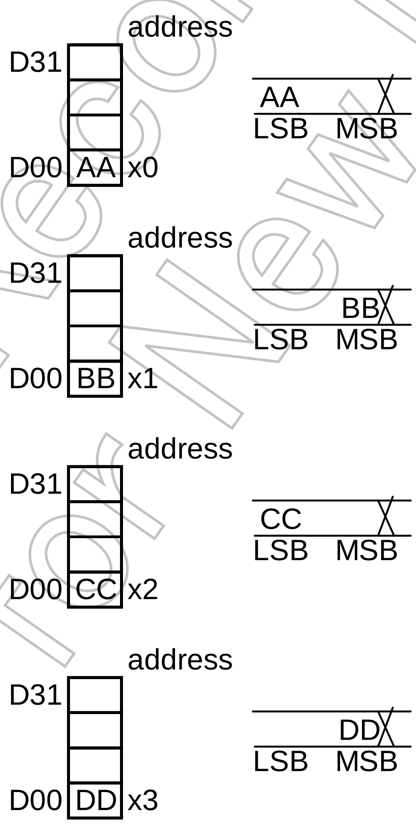
Internal registers External buses



③ Byte access

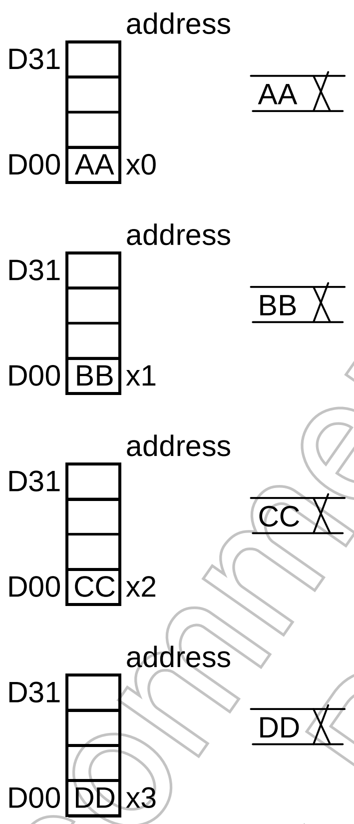
- 16-bit bus width

Internal registers External buses



- 8-bit bus width

Internal registers External buses



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 TMP19A63 basically consists of three clock pulses and a wait can be inserted as mentioned later. The basic clock of an external bus cycle is the same as the internal system clock.

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

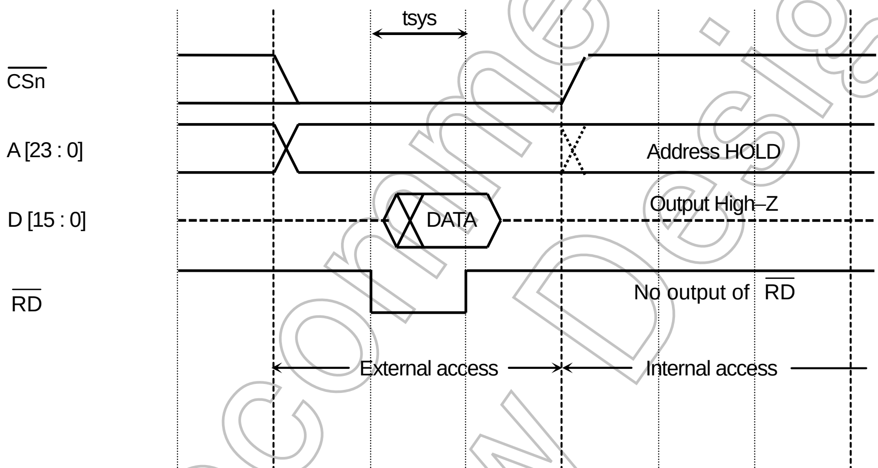


Fig. 8.3.1 Read Operation Timing Diagram

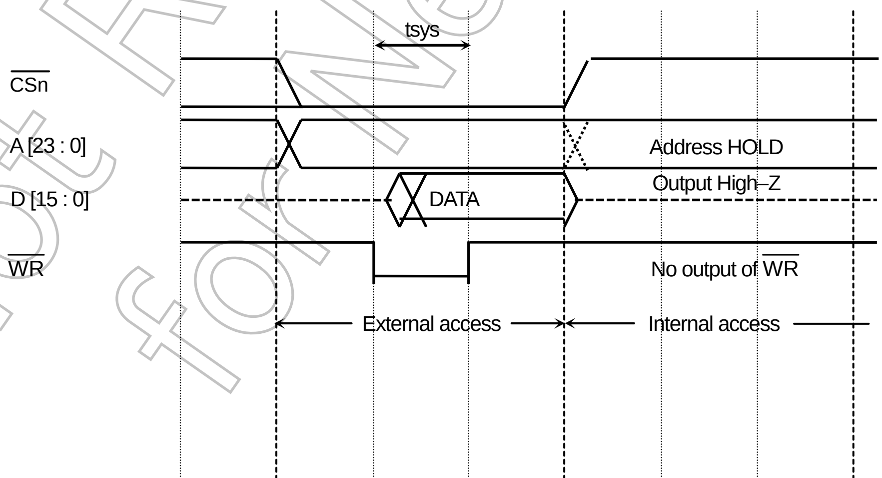


Fig. 8.3.2 Write Operation Timing Diagram

(2) Wait timing

A wait cycle can be inserted for each block by using the chip selector (CS) and wait controller. The following three types of wait can be inserted:

① A wait can be inserted via the $\overline{\mathrm{WAIT}}$ pin

(2+2N, 3+2N, 4+2N, 5+2N, 6+2N and 7+2N

Note: 2N is the number of external waits that can be inserted.)

② A wait can be inserted via the $\overline{\mathrm{RDY}}$ pin

(2+2N, 3+2N, 4+2N, 5+2N, 6+2N and 7+2N

Note: 2N is the number of external waits that can be inserted.)

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

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

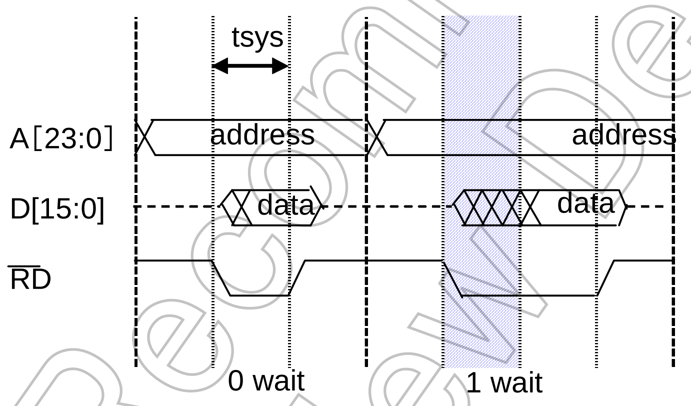


Fig. 8.3.3 Read Operation Timing Diagram (0 Wait and 1 Wait Automatically Inserted)

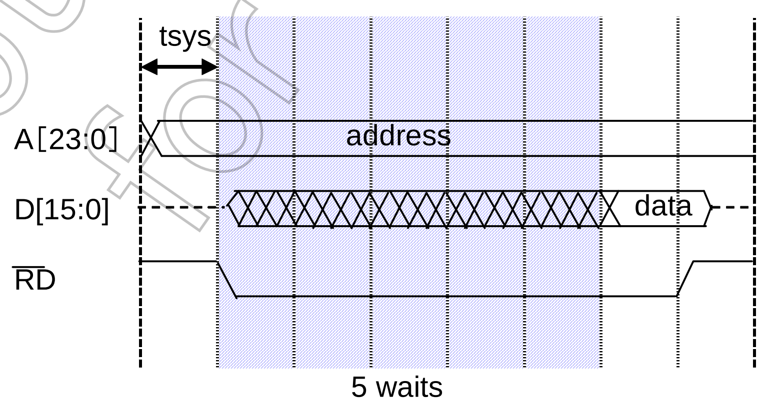
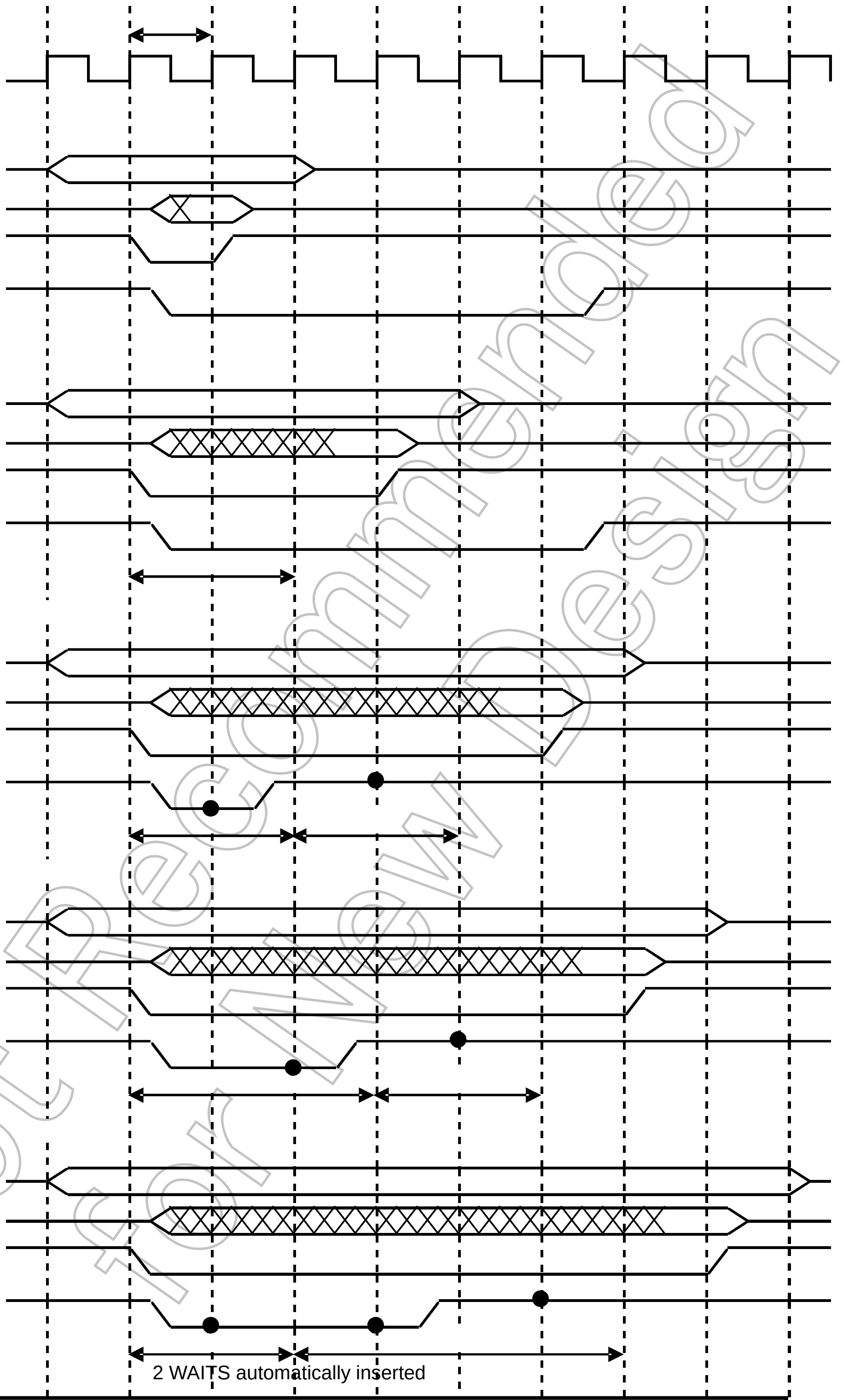


Fig. 8.3.4 Read Operation Timing Diagram (5 Waits Automatically Inserted)

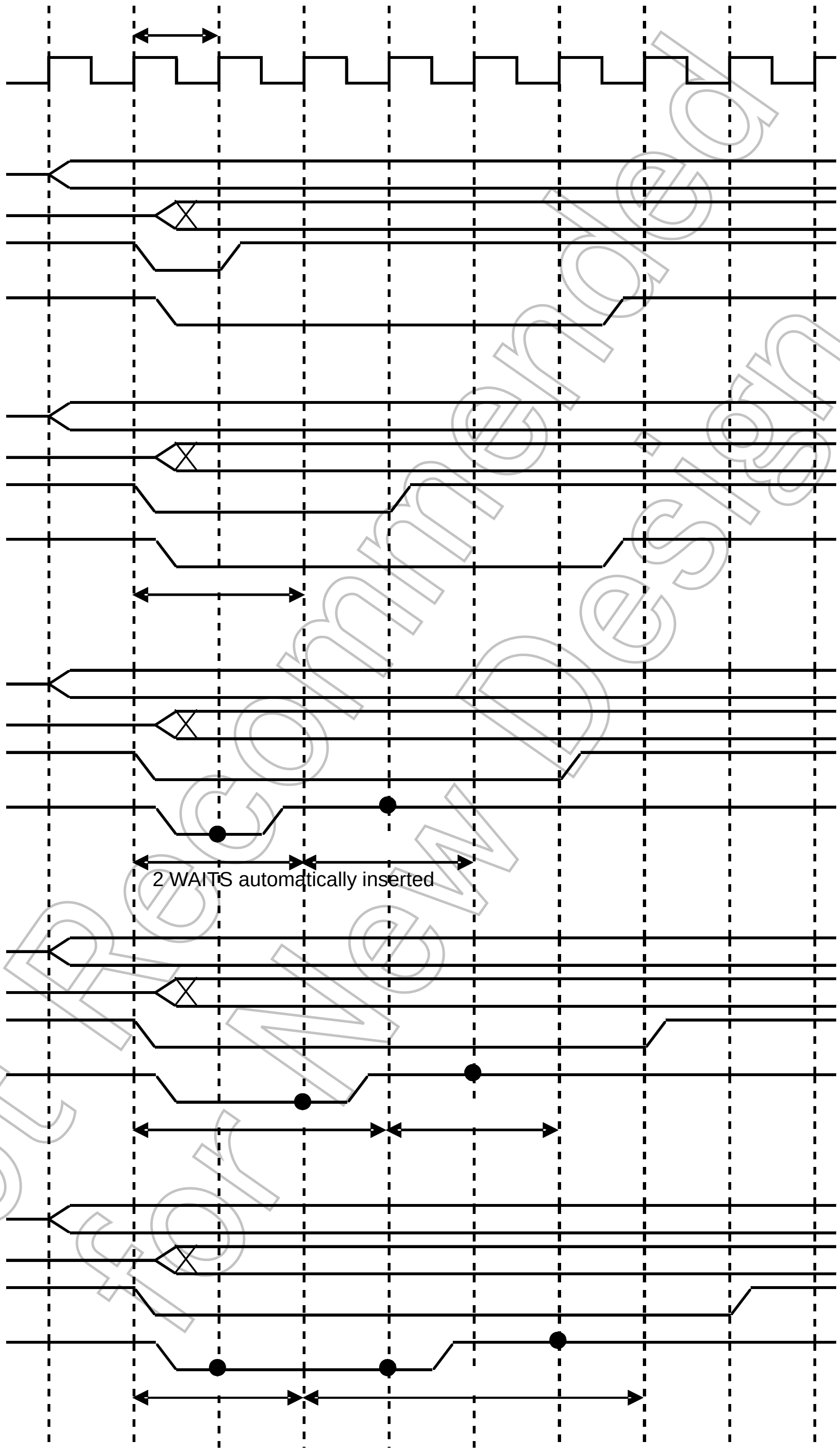
Fig. 8.3.5 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the separate bus mode.



● --- 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. 8.3.5 Read Operation Timing Diagram

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



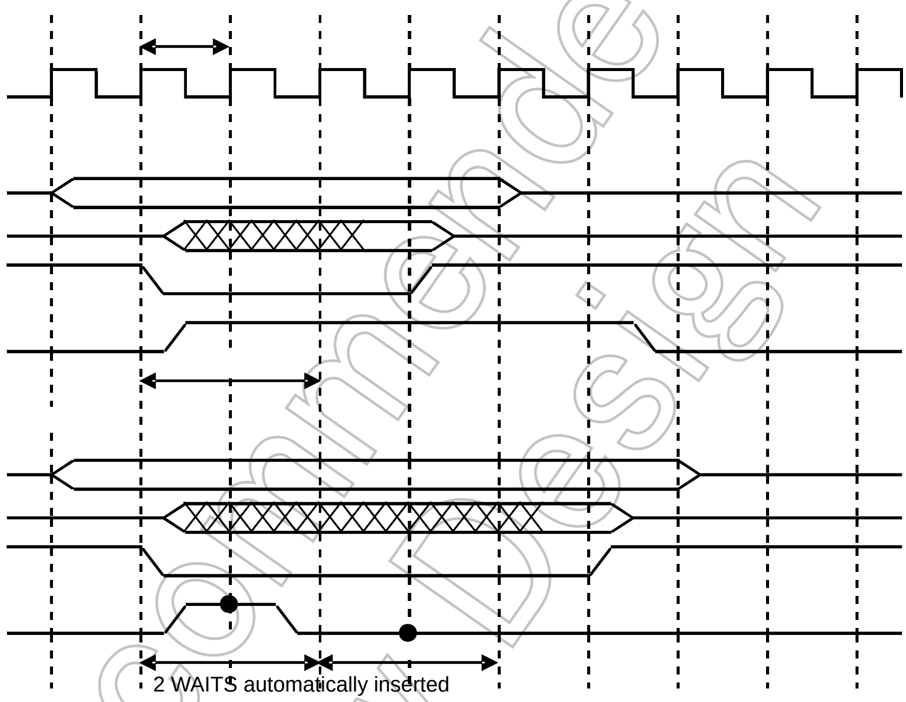
● --- External wait sampling point
External wait sampling points take place before a cycle of waits automatically inserted is finished and before a 2N_wait cycle is finished as shown above. The same applies to combinations of other numbers of waits.

Fig. 8.3.6 Write Operation Timing Diagram

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

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

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



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

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

(3) Time that it takes before ALE is asserted

When the external bus of the TMP19A63 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 at a point of two system (internal) clocks after an address is established. If <ALESEL> is cleared to "0," the $\overline{RD}$ or $\overline{WR}$ signal is asserted at a point of one system (internal) clock after an address is established. This assert setting cannot be established for each block in an external area and the same setting is commonly used in an external address space.

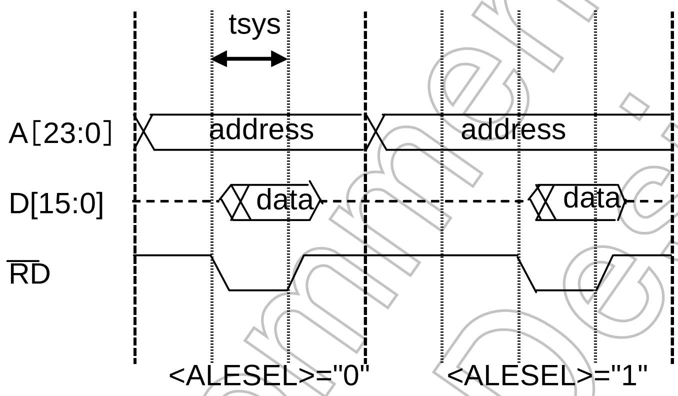


Fig. 8.3.8 SYSCR3 <ALESEL> Set Value and External Bus Operation

(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. 8.3.9 shows the timing of recovery time insertion.

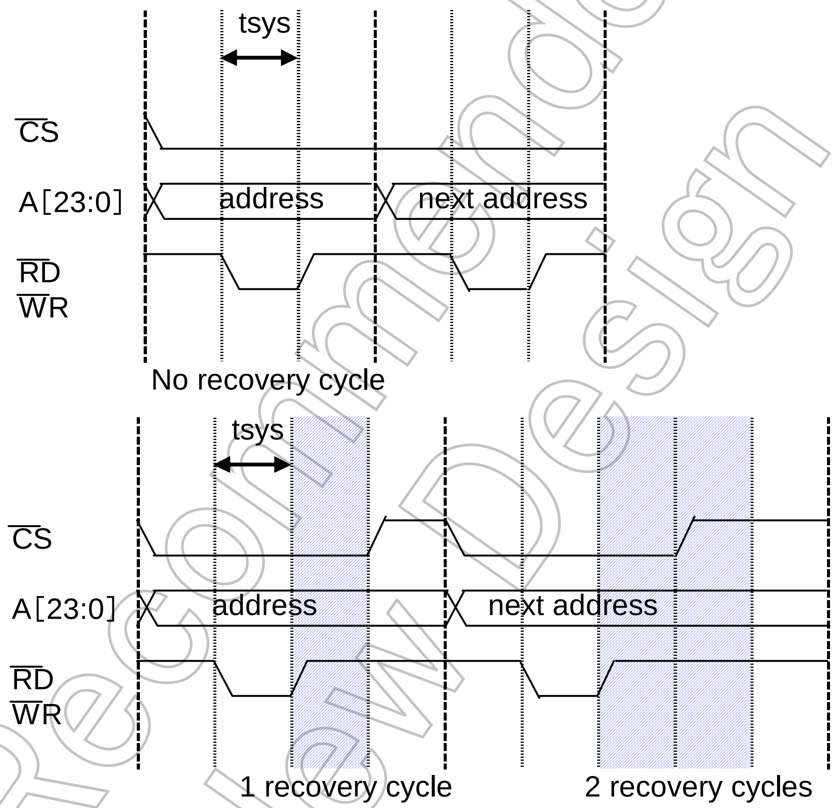


Fig. 8.3.9 Timing of Recovery Time Insertion

(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. 8.3.10 shows the timing of recovery time insertion.

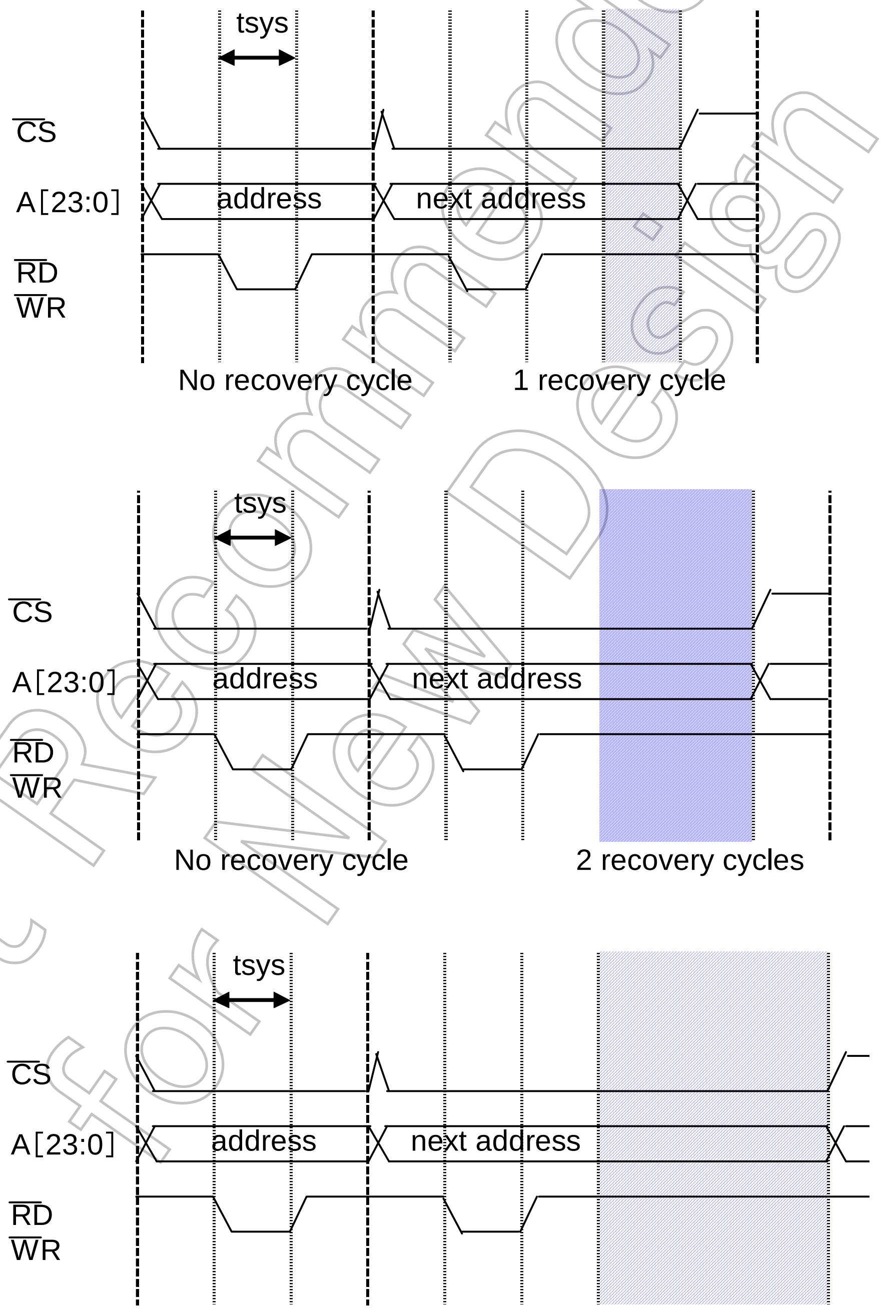


Fig. 8.3.10 CS Timing of Recovery Time Insertion

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 TMP19A63 basically consists of three clock pulses and a wait can be inserted as mentioned later. The basic clock of an external bus cycle is the same as the internal system clock.

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

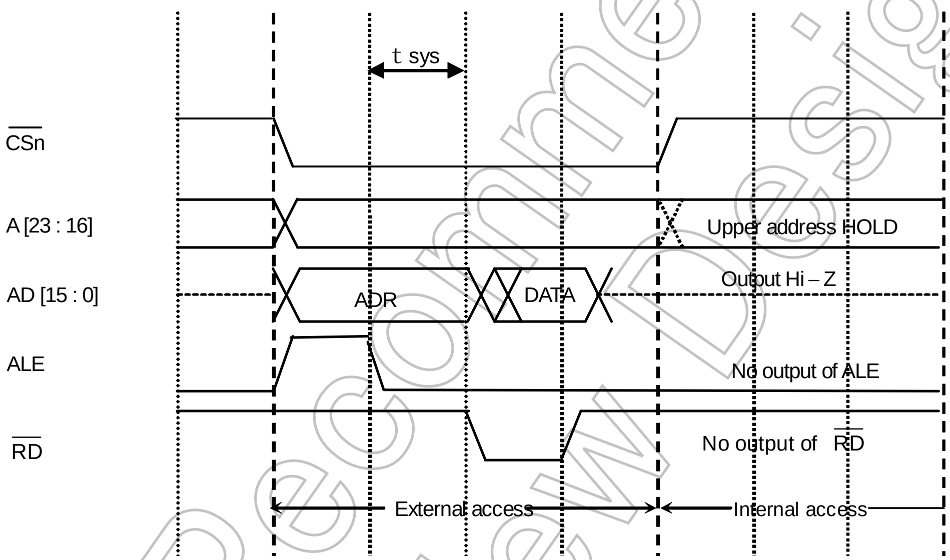


Fig. 8.4.1 Read Operation Timing Diagram

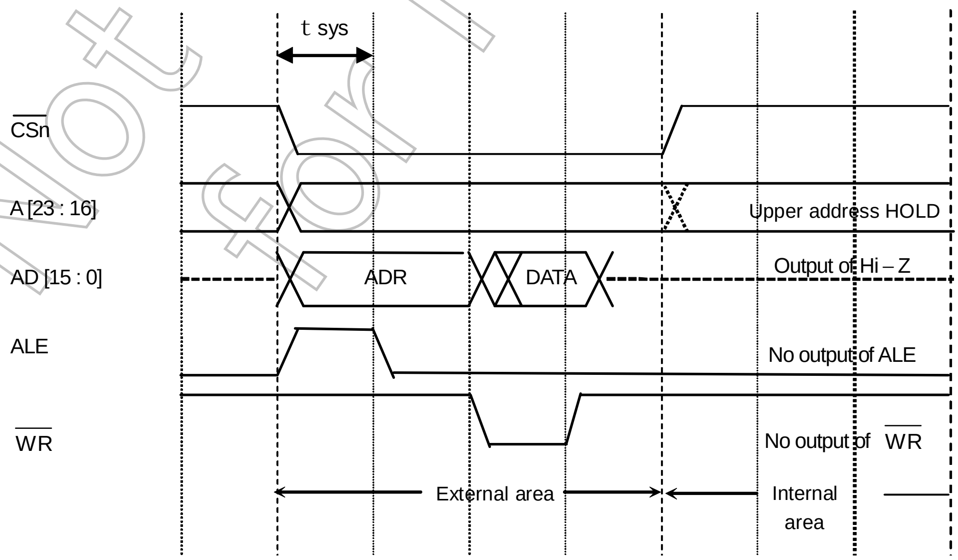


Fig. 8.4.2 Write Operation Timing Diagram

(2) Wait Timing

A wait cycle can be inserted for each block by using the chip selector (CS) and wait controller. The following three types of wait can be inserted:

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

② A wait can be inserted via the $\overline{\text{WAIT}}$ pin.

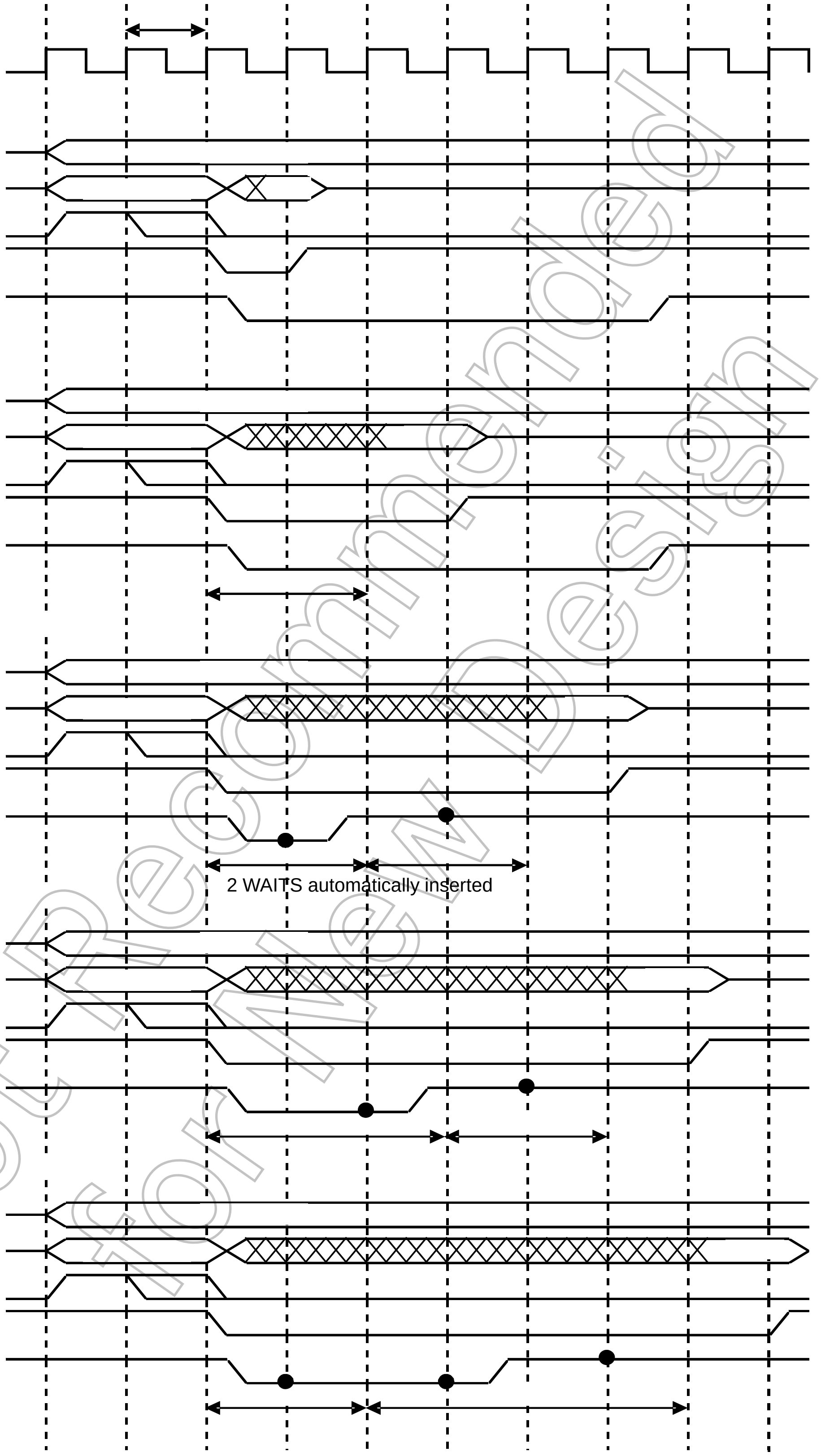
(2+2N, 3+2N, 4+2N, 5+2N, 6+2N and 7+2N
Note: 2N is the number of external waits that can be inserted.)

③ A wait can be inserted via the $\overline{\text{RDY}}$ pin.

(2+2N, 3+2N, 4+2N, 5+2N, 6+2N and 7+2N
Note: 2N is the number of external waits that can be inserted.)

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

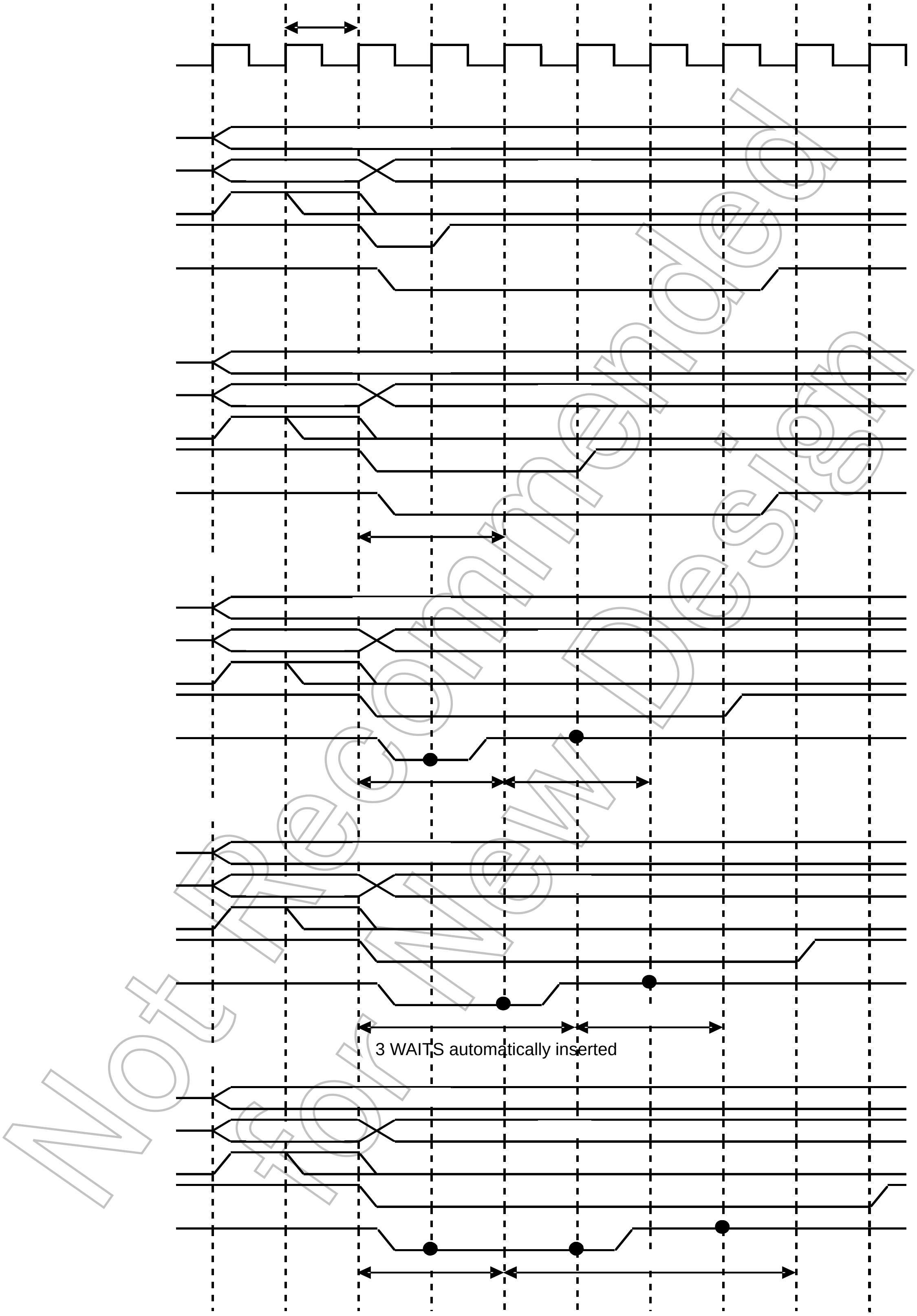
Fig. 8.4.3 shows the read operation timing when 0 wait, waits automatically inserted, and waits automatically inserted + external waits are inserted in the multiplexed bus mode.



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

Fig. 8.4.3 Read Operation Timing Diagram

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



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

Fig. 8.4.4 Write Operation Timing Diagram

(3) Time for ALE to be asserted

As an ALE assertion time, either 1 clock or 2 clocks can be selected. The setting bit is located in the system clock control register. The default is 2 clocks. This assert setting cannot be established for each block in an external area and the same setting is commonly used in an external address space.

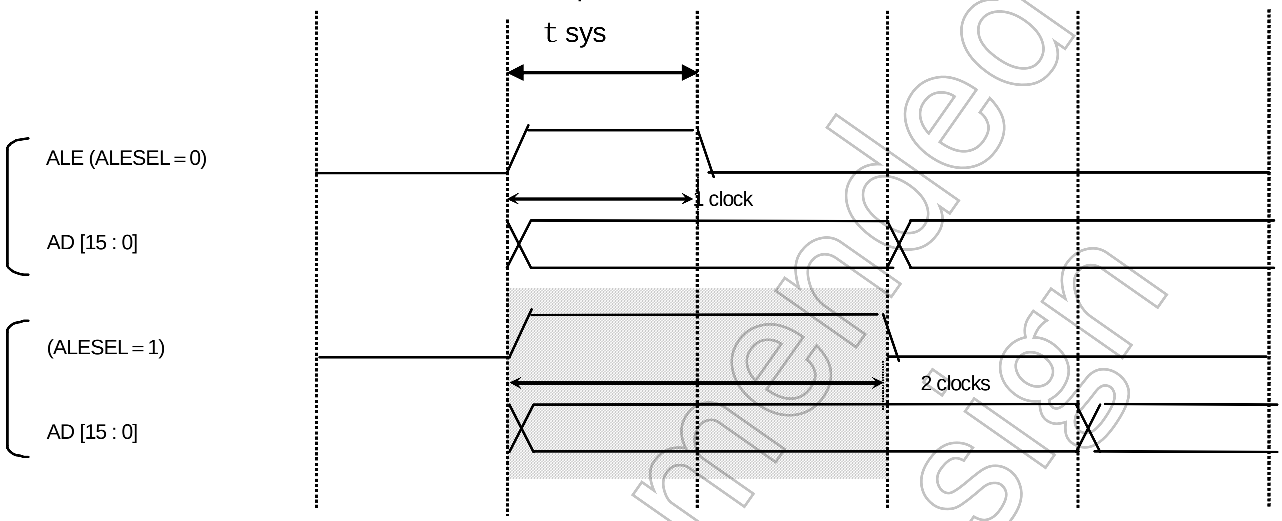


Fig. 8.4.5 Time for ALE to be asserted

Fig. 8.4.13 shows the ALE timings with 1 clock or 2 clocks.

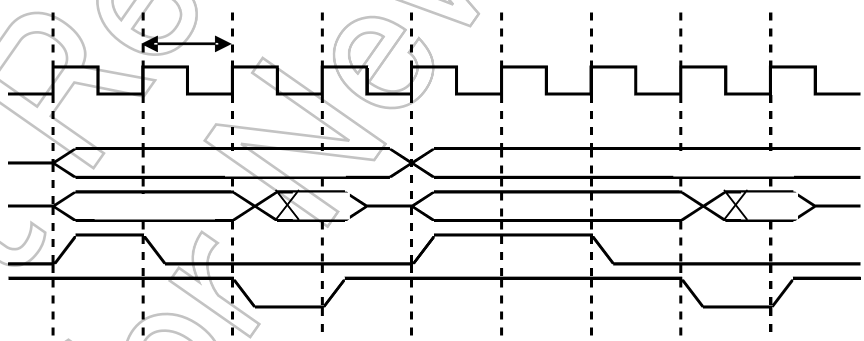


Fig. 8.4.6 Read Operation Timing (the ALE timings with 1 clock or 2 clocks)

(4) Read and Write Recovery Time

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

A dummy cycle can be inserted in both a read and a write cycle. The dummy cycle insertion setting can be made in the chip selector and wait controller registers, BmnCS<BnWCV> (write recovery cycle) and <BnRCV> (read recovery cycle). As for the number of dummy cycles, one or two system clocks (internal) can be specified for each block. Fig. 8.4.7 shows the timing of recovery time insertion.

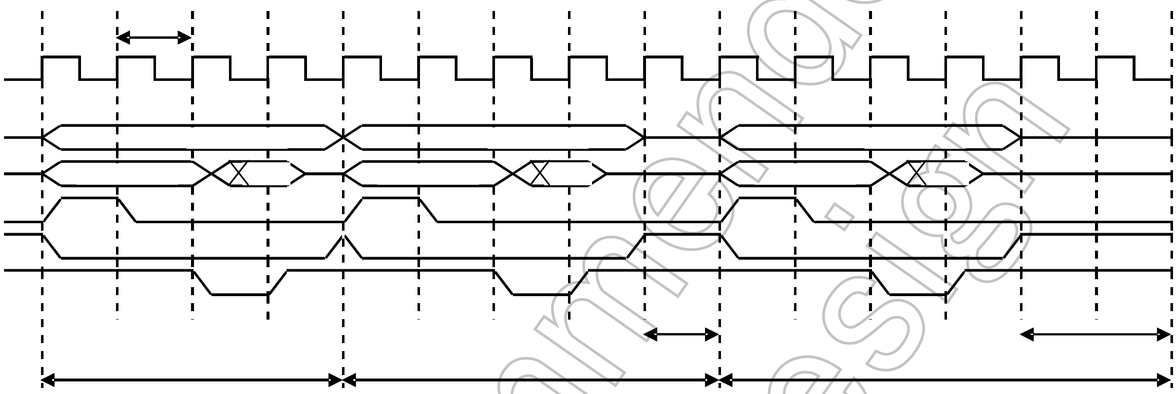


Fig. 8.4.7 Timing of Recovery Time Insertion

(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>. The number of dummy cycles can be specified by one internal system clock for each block. Fig. 8.4.8 shows the timing of recovery time insertion.

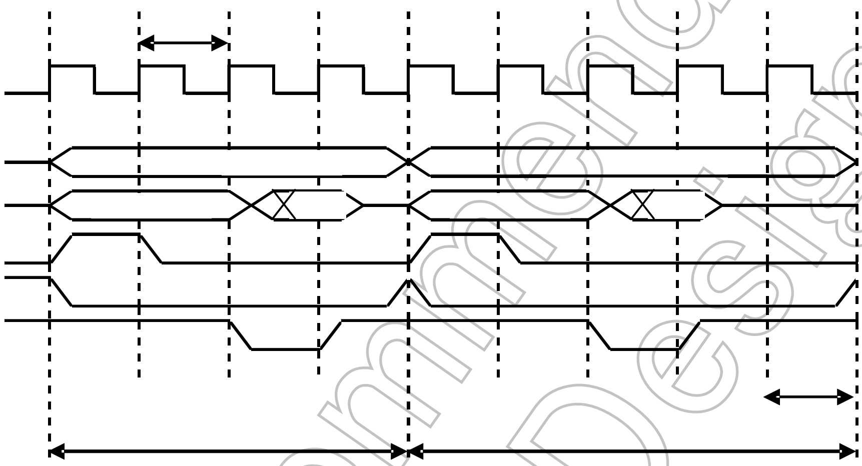


Fig. 8.4.8 Timing of Recovery Time Insertion

8.5 Bus Arbitration

The TMP19A63 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 TMP19A63 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 only for TMP19A63 external buses, 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 bus cycle has to wait until the external bus master releases the bus. For this reason, if the $\overline{\text{BUSRQ}}$ remains active, the TMP19A63 may lock.

(2) Acquisition of bus control authority

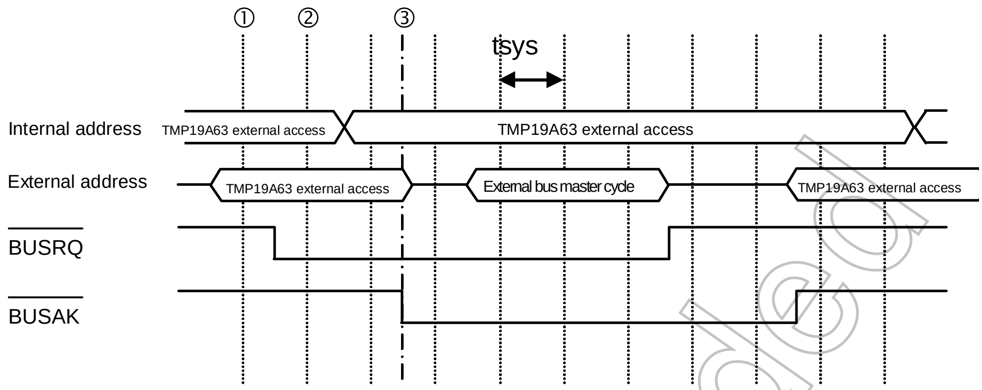
The external bus master requests the TMP19A63 for bus control authority by asserting the $\overline{\text{BUSRQ}}$ signal. The TMP19A63 samples the $\overline{\text{BUSRQ}}$ signal at the break of external bus cycles on the internal buses (G-BUS) and determines whether or not to give the bus control authority to the external bus master. When it gives the bus control authority to the external bus master, it asserts the $\overline{\text{BUSAK}}$ signal. At the same time, it makes address buses, data buses and bus control signals ($\overline{\text{RD}}$ and $\overline{\text{WR}}$) in a state of high impedance. (The internal pull-up is enabled for the $\text{R}/\overline{\text{W}}$, $\overline{\text{HWR}}$ and $\overline{\text{CSx}}$.)

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

If access to external areas occurs consecutively on the TMP19A63, a dummy cycle can be inserted. Again, requests for buses are accepted at the break of external bus cycles on the internal buses (G-BUS). During a dummy cycle, the next external bus cycle is already started on the internal buses. Therefore, even if the $\overline{\text{BUSRQ}}$ signal is asserted during a dummy cycle, the bus is not released until the next external bus cycle is completed.

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

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



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

② The TMP19A63 recognizes that the $\overline{\text{BUSRQ}}$ is at the "L" level, and releases the bus at the end of the bus cycle.

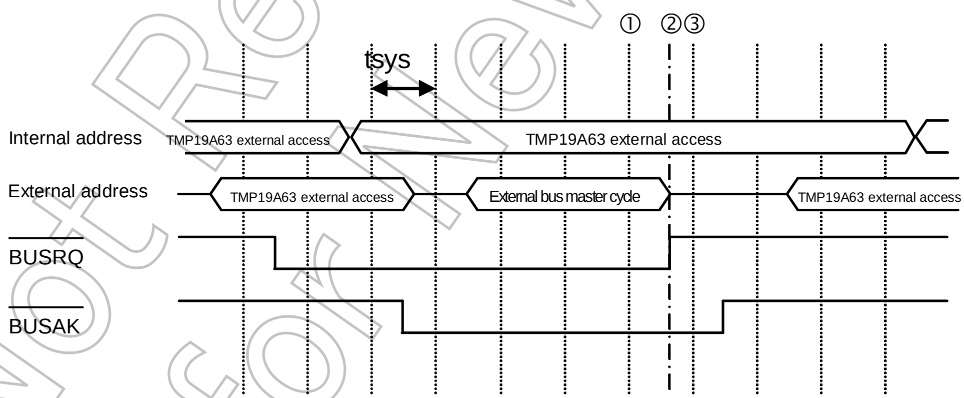
③ When the bus is completed, the TMP19A63 asserts $\overline{\text{BUSAK}}$. The external bus master recognizes that the $\overline{\text{BUSAK}}$ is at the "L" level, and acquires the bus control authority to start bus operations.

Fig. 8.5.1 Bus Control Authority Acquisition Timing

(3) Release of bus control authority

The external bus master releases the bus control authority when it becomes unnecessary. If the external bus master no longer needs the bus control authority having been obtained already, it negates the $\overline{\text{BUSRQ}}$ signal and returns the bus control authority to the TMP19A63.

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



① The external bus master has the bus control authority.

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

③ The TMP19A63 recognizes that the $\overline{\text{BUSRQ}}$ is at the "H" level, and negates the $\overline{\text{BUSAK}}$.

Fig. 8.5.2 Timing of Releasing Bus Control Authority

9. The Chip Selector and Wait Controller

The TMP19A63 can be connected to external devices (I/O devices, ROM and SRAM).

6-block address spaces (CS0 through CS5) can be established in the TMP19A63 and three parameters can be specified for each address and other address spaces: data bus width, the number of waits and the number of dummy cycles.

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

The specification of the spaces CS0 through CS5 is to be performed with a combination of base addresses (BAn, n=0 to 5) and mask addresses (MAn, n=0 to 5) using the base and mask address setting registers (BMA0 through 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 B45CS).

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

9.1 Specifying Address Spaces

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

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

9.1.1 Base and Mask Address Setting Registers

Fig. 9.1.1 through Fig. 9.1.3 show base and mask address setting registers. For base addresses (BA0 through BA5), a start address in the space CS0 through CS5 is specified. In each bus cycle, the chip selector and wait controller compare values in their registers with addresses though those addresses with address bits masked by the mask address (MA0 through MA5) are not compared. The size of an address space is determined by the mask address setting.

(1) Base addresses

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

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

(2) Mask addresses

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

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

CS0 space and CS1 space: A29 through A14

CS2 space through CS5 space: A30 through A15

(Note 1)	**Address settings must be made using physical addresses.**
(Note 2)	**Make sure to write "0" for all the bits to be specified for BMAn when CS is not used. There are cases that decoding is executed internally when CS is not used.**
(Note 3)	**Mapping I/O, ROM and RAM to CS space is prohibited. Otherwise, access to externalspace and internal space occurs simultaneously.**

Base and mask address setting registers BMA0 (0xFFFF_E400H)~BMA5 (0xFFFF_E41CH)

BMA0 (0xFFFF_E400H)

	7	6	5	4	3	2	1	0
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
Symbol	MA0							
Read/Write	R/W							
After reset	0	0	0	0	0	0	1	1
Function	Make sure to write “0.”						CS0 space size setting 0: Address for comparison	
	23	22	21	20	19	18	17	16
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
Symbol	BA0							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	A31 to A24 to be set as a start address							

BMA1 (0xFFFF_E404H)

	7	6	5	4	3	2	1	0
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
Symbol	MA1							
Read/Write	R/W							
After reset	0	0	0	0	0	0	1	1
Function	Make sure to write “0.”						CS1 space size setting 0: Address for comparison	
	23	22	21	20	19	18	17	16
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
Symbol	BA1							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	A31 to A24 to be set as a start address							

(Note) Make sure to 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 TMP19A63 is 16 MB and so bits 10 through 15 must be set to "0" as addresses A24 through A29 are not masked.

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

BMA2
(0xFFFF_E408H)

	7	6	5	4	3	2	1	0
Symbol	MA2							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS2 space size setting 0: Address for comparison							
	15	14	13	12	11	10	9	8
Symbol	MA2							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure to write “0.”							CS2 space size setting 0: Address for comparison
	23	22	21	20	19	18	17	16
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
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_E40CH)

	7	6	5	4	3	2	1	0
Symbol	MA3							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS3 space size setting 0: Address for comparison							
	15	14	13	12	11	10	9	8
Symbol	MA3							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure to write “0.”							CS3 space size setting 0: Address for comparison
	23	22	21	20	19	18	17	16
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
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) 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 TMP19A63 is 16 MB and so bits 9 through 15 must be set to "0" as addresses A24 through A30 are not masked.

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

BMA4 (0xFFFF_E410H)

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

BMA5 (0xFFFF_E41CH)

	7	6	5	4	3	2	1	0
Symbol	MA5							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	CS5 space size setting 0: Address for comparison							
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Symbol	MA5							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	1
Function	Make sure to write "0."							CS5 space size setting 0: Address for comparison
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Symbol	BA4							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A23 to A16 to be set as a start address							
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Symbol	BA4							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	A31 to A24 to be set as a start address							

(Note) Make sure to write "0" for bits 9 through 15 for BMA4 and BMA5.
The size of both the CS4 and CS5 spaces can be a minimum of 32 KB to a maximum of 2 GB.
The external address space of the TMP19A63 is 16 MB and so bits 9 through 15 must be set to "0" as addresses A24 through A30 are not masked.

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

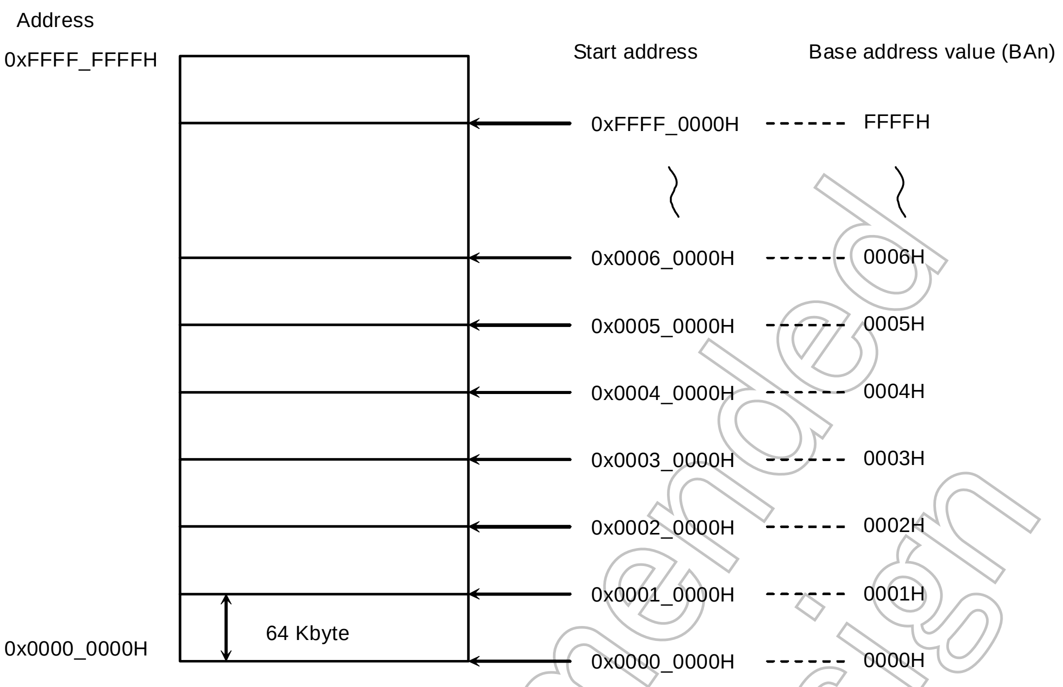


Fig. 9.1.4 Start and Base Address Register Values

9.1.2 How to Define Start Addresses and Address Spaces

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

31															16	15															0
BA0																MA0															
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
C				0				0				0				0				0				0				3			

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

In the base address (BA0), specify "0xC000" that corresponds to higher 16 bits of a start address, while in the mask address (MA0), specify whether a comparison of addresses in the space A29 through A14 is to be made or not. Set bits 15 to 2 of the mask address (MA0) to "0" in order for a comparison of A31 and A30 to be made definitely and to ensure a comparison of A29 through A16. A comparison of A31 and A30 is always executed.

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

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

31															16	15															0
BA2																MA2															
0	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1				F				D				0				0				0				1				F			

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

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

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_0000～0x1FDF_FFFF is designated as a CS2 space.

After a reset, the CS0, CS1, and CS2 through CS5 spaces are disabled.

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

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

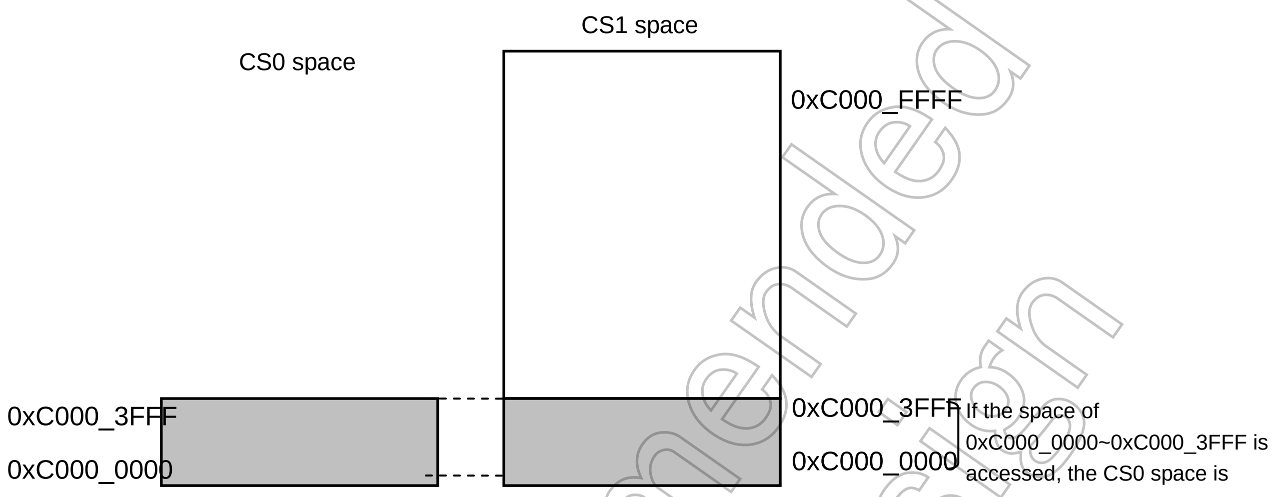


Table 9.1.1 CS Space and Space Sizes

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

9.2 The Chip Selector and Wait Controller

Fig. 9.2.1 through Fig. 9.2.4 show the chip selector and wait controller registers. For each address space (spaces CS0 through CS5 and other address spaces), each chip selector and wait controller register (B01CS through B45CS) 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 selected by priority (priority order: CS0>CS1>CS2>CS3>CS4>CS5).

B01CS
(FFFFE480H)

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

	15	14	13	12	11	10	9	8
Bit symbol	B0CSCV		B0WCV		B0E		B0RCV	
Read/Write	R/W		R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function	Specify the number of dummy cycles to be inserted. (CS0 recovery time) 11: Disable 10: Disable 01: 1 cycle 00: None		Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable		CS0Enable 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable	

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

	31	30	29	28	27	26	25	24
Bit symbol	B1CSCV		B1WCV		B1E		B1RCV	
Read/Write	R/W		R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function	Specify the number of dummy cycles to be inserted. (CS1 recovery time) 11: Disable 10: Disable 01: 1 cycle 00: None		Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable		CS1Enable 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable	

Fig. 9.2.1 Chip Selector and Wait Controller Registers

		7	6	5	4	3	2	1	0
B23CS (0xFFFF_E484H)	Bit symbol	B2OM			B2BUS	B2W			
	Read/Write	R/W		R	R/W	R/W			
	After reset	0	0	0	0	0	1	0	1
	Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (Automatic wait insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (External wait input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			
		15	14	13	12	11	10	9	8
	Bit symbol	B2CSCV		B2WCV		B2E		B2RCV	
	Read/Write	R/W		R/W		R/W	R	R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	Specify the number of dummy cycles to be inserted. (CS2 recovery time) 11: Disable 10: Disable 01: 1 cycle 00: None		Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable		CS2Enable 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable	
		23	22	21	20	19	18	17	16
	Bit symbol	B3OM			B3BUS	B3W			
	Read/Write	R/W		R	R/W	R/W			
	After reset	0	0	0	0	0	1	0	1
	Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (Automatic wait insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (External wait input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			
		31	30	29	28	27	26	25	24
	Bit symbol	B3CSCV		B3WCV		B3E		B3RCV	
	Read/Write	R/W		R/W		R/W	R	R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	Specify the number of dummy cycles to be inserted. (CS3 recovery time) 11: Disable 10: Disable 01: 1 cycle 00: None		Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable		CS3Enable 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable	

Fig. 9.2.2 Chip Selector and Wait Controller Registers

B45CS
(0xFFFF_E488H)

	7	6	5	4	3	2	1	0
Bit symbol	B4OM			B4BUS	B4W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (Automatic wait insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (External wait input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			
	15	14	13	12	11	10	9	8
Bit symbol	B4CSCV		B4WCV		B4E		B4RCV	
Read/Write	R/W		R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function	Specify the number of dummy cycles to be inserted. (CS4 recovery time) 11: Disable 10: Disable 01: 1 cycle 00: None		Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable		CS4Enable 0: Disable 1: Enable		Specify the number of dummy cycles to be inserted. (read, recovery time) 00: 2 cycles 01: 1 cycle 10: None 11: Disable	
	23	22	21	20	19	18	17	16
Bit symbol	B5OM			B5BUS	B5W			
Read/Write	R/W		R	R/W	R/W			
After reset	0	0	0	0	0	1	0	1
Function	Select the chip selector output waveform. 00: ROM/RAM Do not make any other settings.			Select data bus width. 0: 16bit 1: 8bit	Specify the number of waits. (Automatic wait insertion) 0000: 0WAIT 0001: 1WAIT 0010: 2WAIT 0011: 3WAIT 0100: 4WAIT 0101: 5WAIT 0110: 6WAIT 0111: 7WAIT (External wait input) 1010: (2+2N) WAIT 1011: (3+2N) WAIT 1100: (4+2N) WAIT 1101: (5+2N) WAIT 1110: (6+2N) WAIT 1111: (7+2N) WAIT 1000,1001: reserved			
	31	30	29	28	27	26	25	24
Bit symbol	B5CSCV		B5WCV		B5E		B5RCV	
Read/Write	R/W		R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0
Function	Specify the number of dummy cycles to be inserted. (CS5 recovery time) 11: Disable 10: Disable 01: 1 cycle 00: None		Specify the number of dummy cycles to be inserted. (write, recovery time) 00: 2 cycles 01: 1 cycle 10: None Disable		CS5Enable Disable Enable		Specify the number of dummy cycles to be inserted. (read recovery time) 00: 2 cycles 01: 1 cycle 10: None Disable	

Fig. 9.2.3 Chip Selector and Wait Controller Registers

A reset of the TMP19A63 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 TMP19A63 has a built-in 8-channel DMA Controller (DMAC).

10.1 Features

The DMAC of the TMP19A63 has the following features:

(1) DMA with 8 independent channels
(eight interrupt factors, 0: INTDMA0 through INTDMA7)

(2) Two types of requests for bus control authority: With and without snoop requests
Transfer requests: Internal requests (software initiated)/external requests (external interrupts, interrupt requests given by internal peripheral I/Os, and requests given by the $\overline{\text{DREQ}}$ pin)
Requests given by the $\overline{\text{DREQ}}$ pin: Level mode

(3) Transfer mode: Dual address mode

(4) Transfer devices: Memory space transfer

(5) 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

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

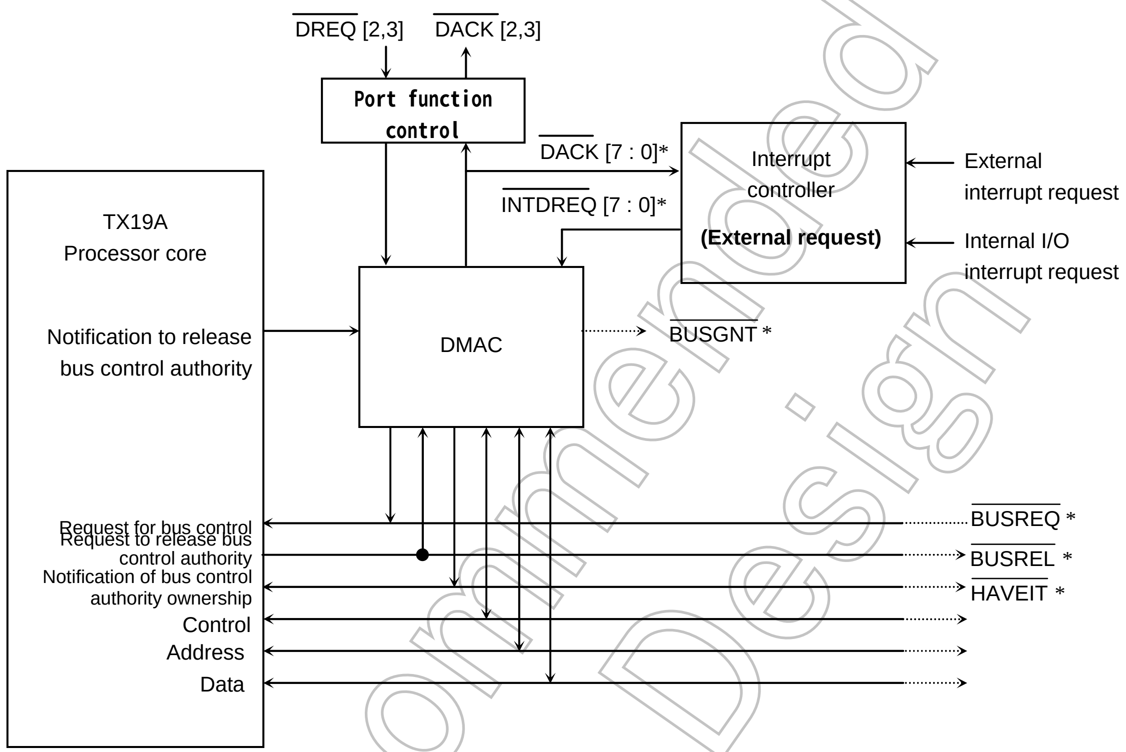
(7) Channel priority: Fixed (in ascending order of channel numbers)

(8) Endian switchover function

10.2 Configuration

10.2.1 Internal Connections of the TMP19A63

Fig. 10.1 shows the internal connections with the DMAC in the TMP19A63.



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

Fig. 10.1 DMAC Connections in the TMP19A63

The DMAC has eight DMA channels. Each of these channels handles the data transfer request signal (INTDREQn) from the interrupt controller and the acknowledgment signal (DACKn) generated in response to INTDREQn ("n" is a channel number from 0 to 7). External pins (DREQ3 and DREQ2) are internally wired to allow them to function as pin of the port Q. To use them as a pin of the port Q, they must be selected by setting the function control register PQFC to an appropriate setting.
Pins handle the data transfer request from external pins DREQ3 and DREQ2 and acknowledge signal output supplied through external pins, DACK3 and DACK2. Channel 0 is given higher priority than channel 1. Channel 1 is given higher priority than channel 2. Channel 2 is given 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 the snoop function. For further information on the snoop function, refer to 10.2.3 “Snoop Function”.

In the DMAC, bus control authority can be select from SREQ and GREQ depend 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.

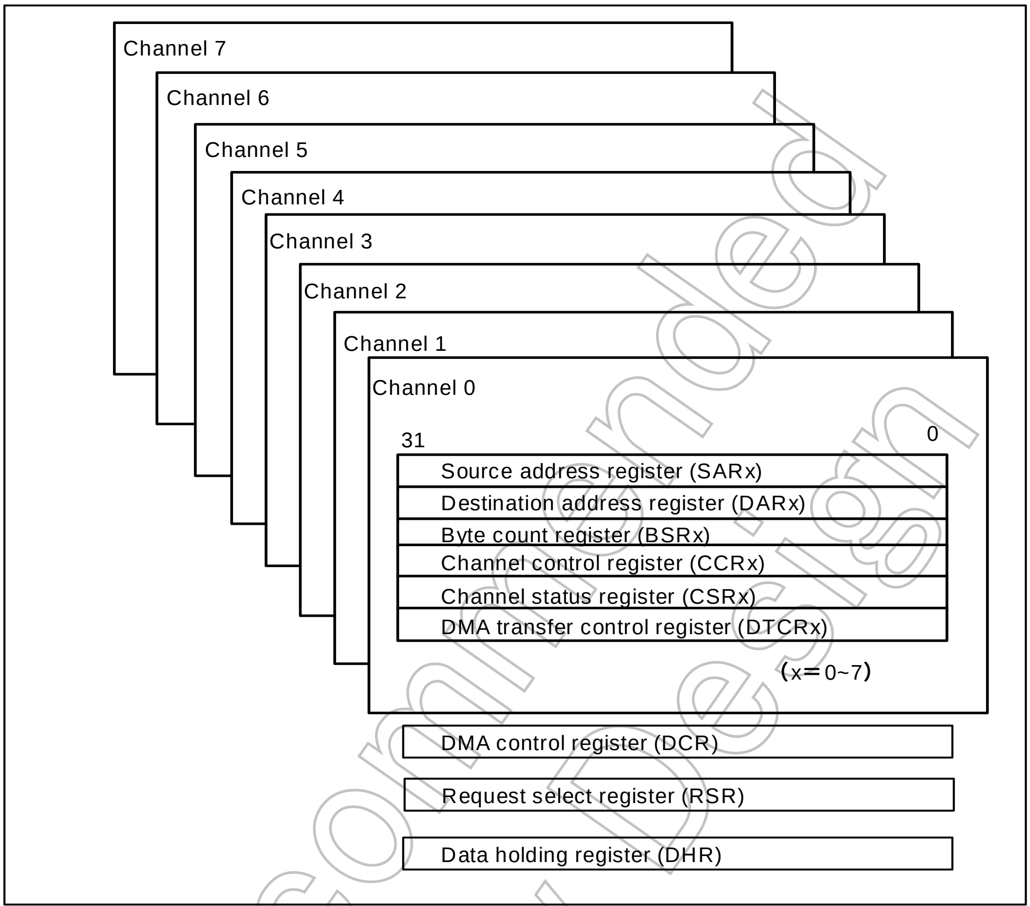


Fig.10.2 DMAC Internal Blocks

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 by way of the G-Bus, bus operations cannot be executed and, as a result, the pipeline stalls unless the DMAC accept a bus control release request.

(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 1

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)

Table 10.2 DMAC Registers 2

Address	Register symbol	Register name
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)
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					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 four 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 Rst2 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
4	Rst4	Reset 4	Performs a software reset of the DMAC channel 4. If the Rst2 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 Rst2 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 Rst2 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 Rst2 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 2

Fig. 10.3 DMA Control Register (DCR)

(Note 1)	**If a write to the DCR register occurs during 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 is strictly prohibited.**

10.3.2 Channel Control Registers (CCRn)

CCRn
(0xFFFF_E200H)
(0xFFFF_E220H)
(0xFFFF_E240H)
(0xFFFF_E260H)
(0xFFFF_E280H)
(0xFFFF_E2A0H)
(0xFFFF_E2C0H)
(0xFFFF_E2E0H)

	7	6	5	4	3	2	1	0
Bit symbol	SAC	DIO	DAC		TrSiz		DPS	
Read/Write	R/W	R/W	R/W		R/W		R/W	
After reset	0							
Function	See detailed description							
	15	14	13	12	11	10	9	8
Bit symbol		ExR	PosE	Lev	SReq	RelEn	SIO	SAC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0							
Function	Always set this bit to “0”.	See detailed description						
	23	22	21	20	19	18	17	16
Bit symbol	NIEn	AblEn					Big	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1			0			1	0
Function	See detailed description		Always set this bit to “0”.				See detailed description	Always set this bit to “0”.
	31	30	29	28	27	26	25	24
Bit symbol	Str							
Read/Write	W							W
After reset	0							
Function	See detailed description							Always set this bit to “0”.

Fig. 10.4 Channel Control Register (CCRn) (1/2)

Bit	Mnemonic	Field name	Description
31	Str	Channel start	Start (initial value:–) Starts channel operation. If this bit is set to 1, the channel goes into a standby mode and starts to transfer data in response to a transfer request. Only a write of 1 is valid to the Str bit and a write of 0 is ignored. A read always returns 0. 1: Starts channel operation
24	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
23	NIEn	Normal completion interrupt enable	Normal Completion Interrupt Enable (initial value: 1) 1: Normal completion interrupt enable 0: Normal completion interrupt disable
22	AbIEn	Abnormal completion interrupt enable	Abnormal Completion Interrupt Enable (initial value:1) 1: Abnormal completion interrupt enable 0: Abnormal completion interrupt disable
21	—	(Reserved)	This is a reserved bit. Although it's initial value is "1," always set this bit to "0".
20	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
19	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
18	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
17	Big	Big-endian	Big Endian (initial value: 1) 1: A channel operates by big-endian 0: A channel operates by little-endian
16	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
15	—	(Reserved)	This is a reserved bit. Always set this bit to "0."
14	ExR	External request mode	External Request Mode (initial value: 0) Selects a transfer request mode. (only for 2ch and 3ch) 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 signal level or signal change for recognizing the external transfer request. 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. This function cannot be used if SREQ is generated since the TX19A processor core cannot make a bus control release request. 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 at the break of bus operation. 0: The bus control release request is not acknowledged.
9	SIO	Transfer type selection	Source Type: continuous (initial value: 0) Specifies the transfer type. 1: Single transfer 0: Continuous transfer
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	(Reserved)	Always set this bit to “0”.
5 : 4	DAC	Destination address count	Destination Address Count (initial value: 00) Specifies the manner of change in a destination address. 1x: Address fixed 01: Address decrease 00: Address increase
3 : 2	TrSiz	Transfer unit	Transfer Size (initial value: 00) Specifies the amount of data to be transferred in response to one transfer request. 11: 8 bits (1 byte) 10: 16 bits (2 bytes) 0x: 32 bits (4bytes)
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 (2bytes) 0x: 32 bits (4 bytes)

Fig. 10.4 Channel Control Register (CCRn) (2/2)

(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> size to be the same as the device port size <DPS>.

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

10.3.3 Request Select Register (RSR)

RSR (0xFFFF_E304H)

	7	6	5	4	3	2	1	0
Bit symbol					ReqS3	ReqS2		
Read/Write					R/W	R/W		
After reset	0							
Function	Always set this bit to "0".				See detailed description	See detailed description	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								
Read/Write								
After reset	0							
Function								
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write								
After reset	0							
Function								

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

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

Fig. 10.5 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. 10.6 Channel Status Register (CSRn) (1/2)

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

Fig. 10.6 Channel Status Register (CSRn) (2/2)

10.3.5 Source Address Registers (SARn)

SARn
(0xFFFF_E208H)
(0xFFFF_E228H)
(0xFFFF_E248H)
(0xFFFF_E268H)
(0xFFFF_E288H)
(0xFFFF_E2A8H)
(0xFFFF_E2C8H)
(0xFFFF_E2E8H)

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

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

Fig. 10.7 Source Address Register (SARn)

10.3.6 Destination Address Register (DARn)

DARn
(0xFFFF_E20CH)
(0xFFFF_E22CH)
(0xFFFF_E24CH)
(0xFFFF_E26CH)
(0xFFFF_E28CH)
(0xFFFF_E2ACH)
(0xFFFF_E2CCH)
(0xFFFF_E2ECH)

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

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

Fig. 10.8 Destination Address Register (DARn)

10.3.7 Byte Count Registers (BCRn)

BCRn
(0xFFFF_E210H)
(0xFFFF_E230H)
(0xFFFF_E250H)
(0xFFFF_E270H)
(0xFFFF_E290H)
(0xFFFF_E2B0H)
(0xFFFF_E2D0H)
(0xFFFF_E2F0H)

	7	6	5	4	3	2	1	0
Bit symbol	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
Read/Write	R/W							
After reset	0							
Function	See detailed description							
	15	14	13	12	11	10	9	8
Bit symbol	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8
Read/Write	R/W							
After reset	0							
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								

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

Fig. 10.9 Byte Count Register (BCRn)

10.3.8 DMA Transfer Control Register (DTCRn)

DTCRn
(0xFFFF_E218H)
(0xFFFF_E238H)
(0xFFFF_E258H)
(0xFFFF_E278H)
(0xFFFF_E298H)
(0xFFFF_E2B8H)
(0xFFFF_E2D8H)
(0xFFFF_E2F8H)

	7	6	5	4	3	2	1	0
Bit symbol			DACM			SACM		
Read/Write			R/W			R/W		
After reset	0							
Function			See detailed description			See detailed description		
	15	14	13	12	11	10	9	8
Bit symbol								
Read/Write								
After reset	0							
Function								
	23	22	21	20	19	18	17	16
Bit symbol								
Read/Write								
After reset	0							
Function								
	31	30	29	28	27	26	25	24
Bit symbol								
Read/Write								
After reset	0							
Function								

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

Fig. 10.10 DMA transfer control register (DTCRn)

10.3.9 Data Holding Register (DHR)

DHR
(0xFFFF_E30CH)

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

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

Fig. 10.11 Data Holding Register (DHR)

10.4 Functions

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

10.4.1 Overview

(1) Source and destination

The DMAC handles data transferred within memory space. A device where the data is output is called a source device and a device where the data is input is called a destination device. The memory device can be designated as a source or destination device.

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 $\overline{\text{TrSiz}}$) if a single transfer is designated to select a transfer type (SIO BIT). On the other hand, the $\overline{\text{DACKn}}$ signal is asserted only when the number of bytes transferred (value set in the BCRn register) becomes "0" at a continuous transfer. Therefore, one transfer request allows data to be transferred successively without a pause.

For example, if data is transferred between an internal I/O and the internal (external) memory of the TMP19A63, a request made by the internal I/O to the DMAC is cleared after completion of each data transfer. The transfer operation is always put in a standby mode for the next transfer request unless the number of bytes transferred (value set in the BCRn register) becomes "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; therefore this function cannot be used.

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) Stop the DMAC before putting the TX19A into IDLE (doze) mode while snoop function is active.

(3) Transfer request modes

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

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

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

(4) Address mode

The DMAC of the TMP19A63 provides only one address mode, a dual address mode. A single address mode is not available. In the dual address mode, data can be transferred within memory space. Source and destination device addresses are output by the DMAC. To access an I/O device, the DMAC asserts the $\overline{\text{DACKn}}$ 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. error occurrence). An interrupt signal can be generated upon completion of data transfer.

Fig. 10.12 shows the state transitions of channel operation.

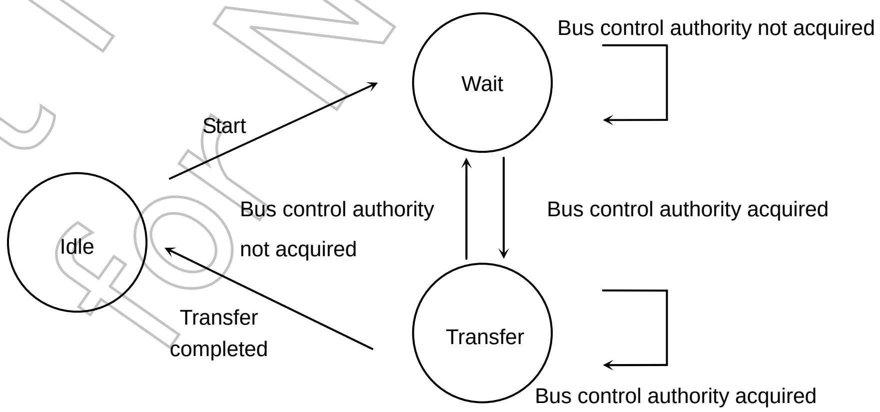


Fig. 10.12 Channel Operation State Transition

(6) Combinations of transfer modes

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

<table>
<tr><th>Transfer request</th><th>Edge/level</th><th>Address mode</th><th>Transfer type</th></tr>
<tr><td>Internal</td><td>—</td><td rowspan="6">Dual</td><td rowspan="2">Continuous</td></tr>
<tr><td rowspan="2">External</td><td rowspan="2">“L” level ($\overline{\text{INTDREQ}}$n)</td></tr>
<tr><td>Single</td></tr>
<tr><td rowspan="3">External</td><td rowspan="2">“L” level ($\overline{\text{DREQ}}$n)</td><td>Continuous</td></tr>
<tr><td rowspan="2">Single</td></tr>
<tr><td>Falling edge ($\overline{\text{DREQ}}$n)</td></tr>
</table>

(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. 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 for the source address or DACM for the destination address in the DTCRn register. To specify the bit position for counting a source 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 increases or decreases as per normal. 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 the bus control authority and starts to transfer data if the transfer request is generated after the start of channel operation.

- Internal transfer request

A transfer request is generated immediately if the Str bit of CCR is set to "1" when the ExR bit of CCRn is "0". 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 unless transition to a channel with higher priority or transition of the bus control authority to a bus master with higher priority occurs.

Continuous transfer is only available with internal transfer request.

- External transfer request

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

The TMP19A63 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. 32, 16 or 8 bits can be selected.

See the next page for the detailed descriptions on transfer requests using $\overline{\text{INTDREQn}}$ and $\overline{\text{DREQn}}$.

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

The $\overline{\text{DACKn}}$ signal can clear a transfer request made by the interrupt controller. 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 INTDREQn is cleared upon completion of one data transfer from one transfer request. On the other hand, at the continuous transfer, it can be transferred successively in response to a transfer request because INTDREQn is not cleared until the number of bytes transferred (value set in the BCRn register) becomes "0."

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

② A transfer request made by an external device

External pins ($\overline{\text{DREQ3}}$ and $\overline{\text{DREQ2}}$) are internally wired to allow them to function as pin of the port Q. 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 continuous transfer, only the "L" level mode can be used. At 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 negated 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 the control authority and goes into a standby mode.

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

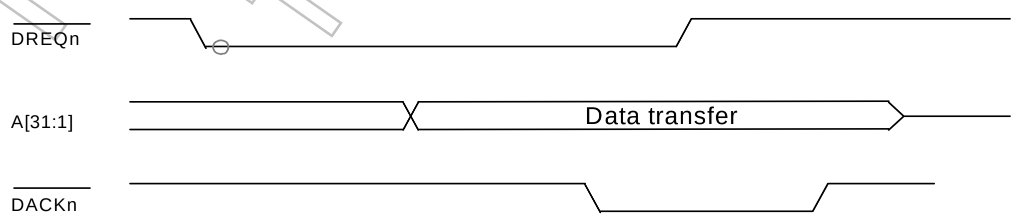


Fig. 10.13 Transfer Request Timing (Level Mode)

– Edge mode

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

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

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

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

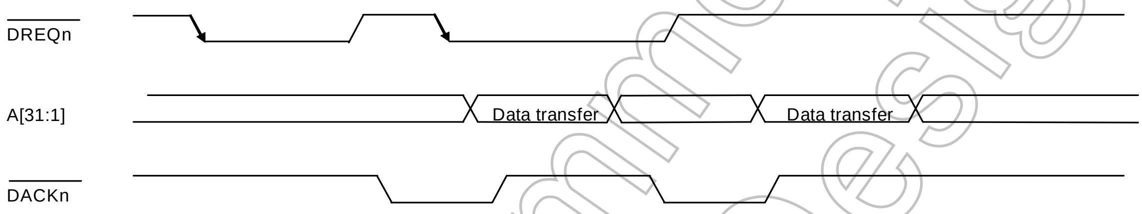


Fig. 10.14 Transfer Request Timing (Edge Mode)

10.4.3 Address Mode

In the address mode, you can specify 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. The former is called as the dual address mode, and the latter is called as the single address mode. For TMP19A63, only the dual address mode is available.

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

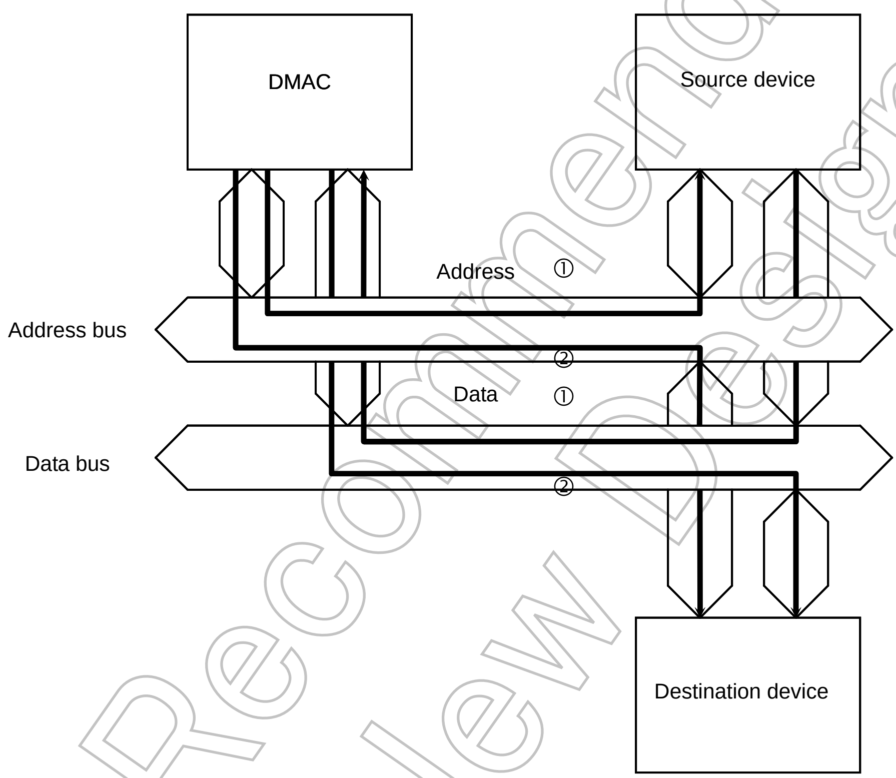


Fig. 10.15 Basic Concept of Data Transfer in the Dual Address Mode

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

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

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

10.4.4 Channel Operation

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

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

Channel operation is completed either normally or abnormally (e.g. occurrence of an error). One of the conditions 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 the channel is put into a standby mode if no error is detected. If an error is detected, the channel is gone into the abnormal completion. When a channel goes into a standby mode, the Act bit of the CSRn of that channel becomes "1".

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

Completion of channel operation

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

Channel operation does not start and the completion of operation is considered to be abnormal completion if "1" is set to the Str bit of the CCRn register when the NC or AbC bit of the CSRn register is "1,"

Normal completion

Channel operation is considered to have been completed normally in the case shown below. For the normally completed channel operation, it needs to be completed after the transfer of a unit of data (value specified in the TrSiz field of CCRn) is 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 were 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" appears later in this document.

10.4.5 Order of Priority of Channels

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

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

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

<u>Interrupts</u>

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, INTDMA1: 1ch, INTDMA1: 2ch, INTDMA1: 3ch

- Normal completion interrupt

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

- Abnormal completion interrupt

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

10.5 Timing Diagrams

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

10.5.1 Dual Address Mode

- Continuous transfer

Fig. 10.16 shows an example of the timing with which 16-bit data is transferred from one external memory (16-bit width) to another (16-bit width). Data is actually transferred successively until BCRn becomes "0."

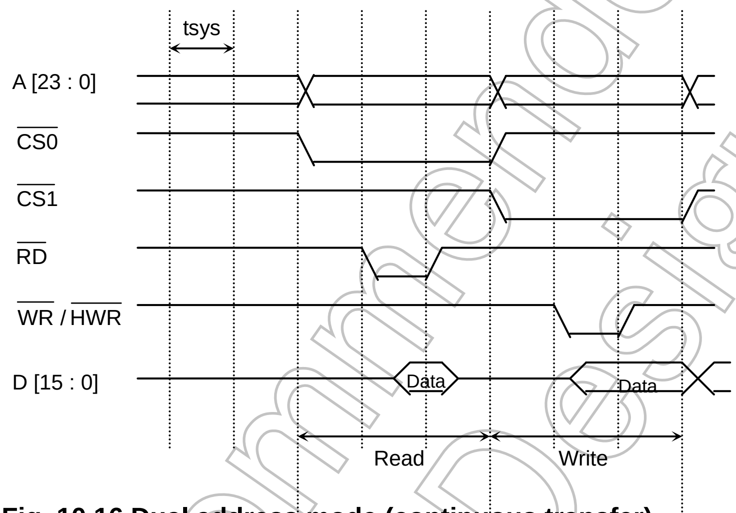


Fig. 10.16 Dual address mode (continuous transfer)

- Single transfer (1)

Fig. 10.17 shows an example of the timing if the unit of data to be transferred is set to 16 bits and the device port size is set to 16 bits.

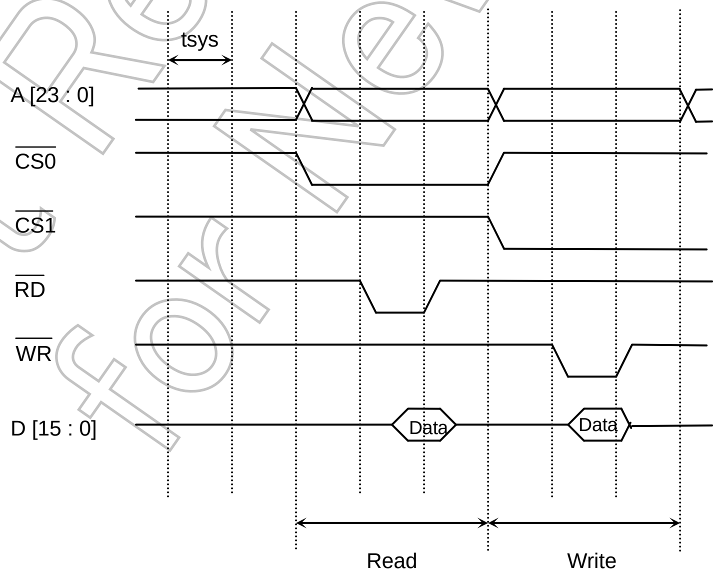


Fig. 10.17 Dual address mode (single transfer)

- Single transfer (2)

Fig. 10.18 shows an example of the single transfer timing if the unit of data to be transferred is set to 16 bits and the device port size is set to 16 bits.

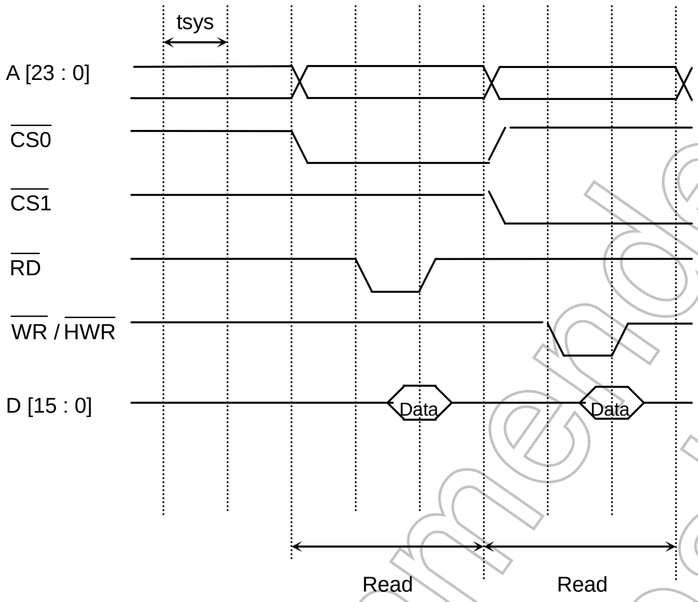


Fig. 10.18 Dual address mode (single transfer)

10.5.2 DREQn-Initiated Transfer Mode

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

  Fig. 10.19 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

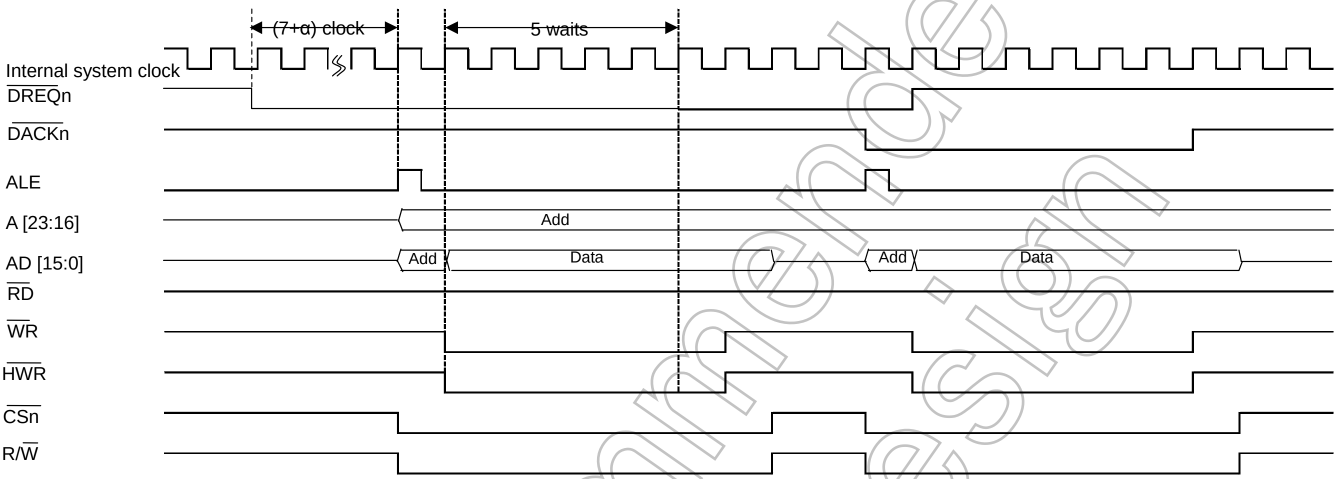


Fig. 10.19 Level Mode (from Internal RAM to External Memory)

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

  Fig. 10.20 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bit width) to internal RAM.

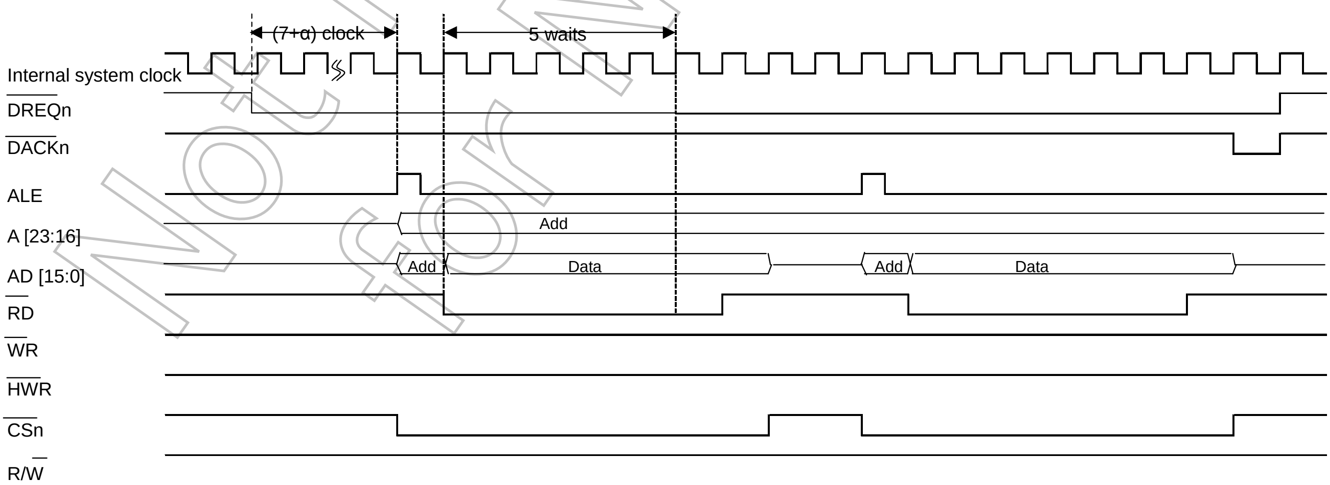


Fig. 10-20 Level Mode (from External Memory to Internal RAM)

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

Fig. 10.21 shows two timing cycles in which 16-bit data is transferred twice from internal RAM to external memory (16-bit width).

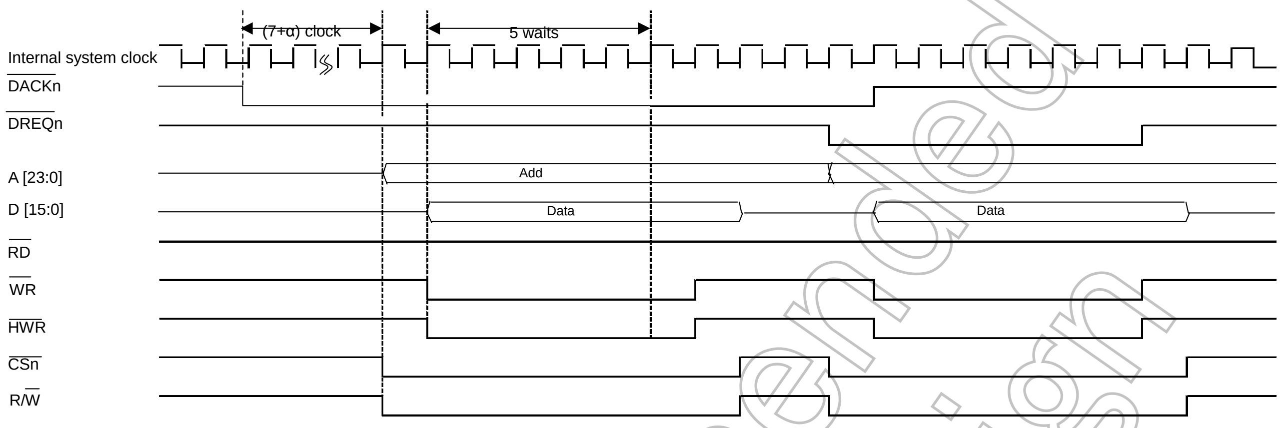


Fig. 10.21 Level Mode (Internal RAM to External Memory)

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

Fig. 10.22 shows two timing cycles in which 16-bit data is transferred twice from external memory (16-bit width) to internal RAM.

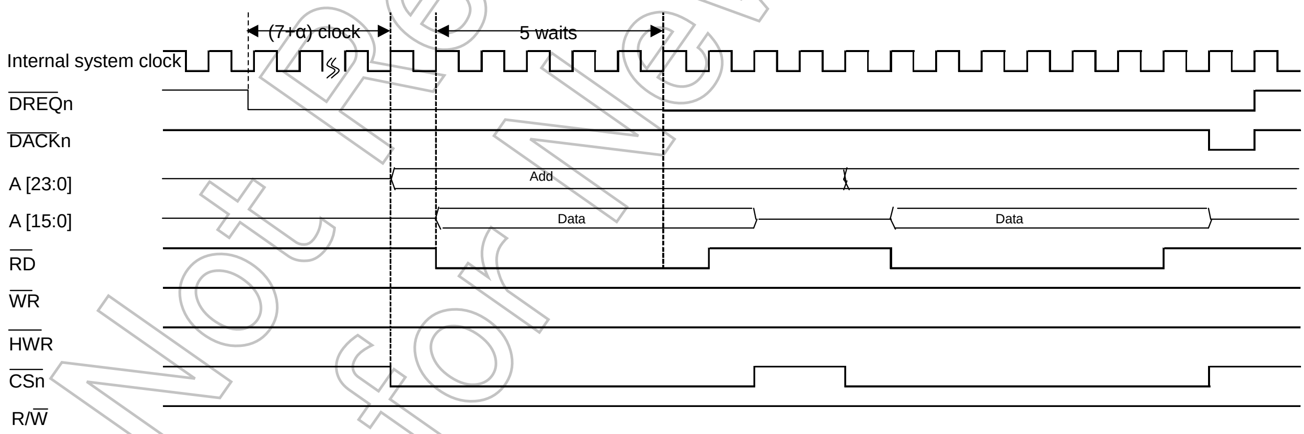


Fig. 10.22 Level Mode (from External Memory to Internal RAM)

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

  Fig. 10.23 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

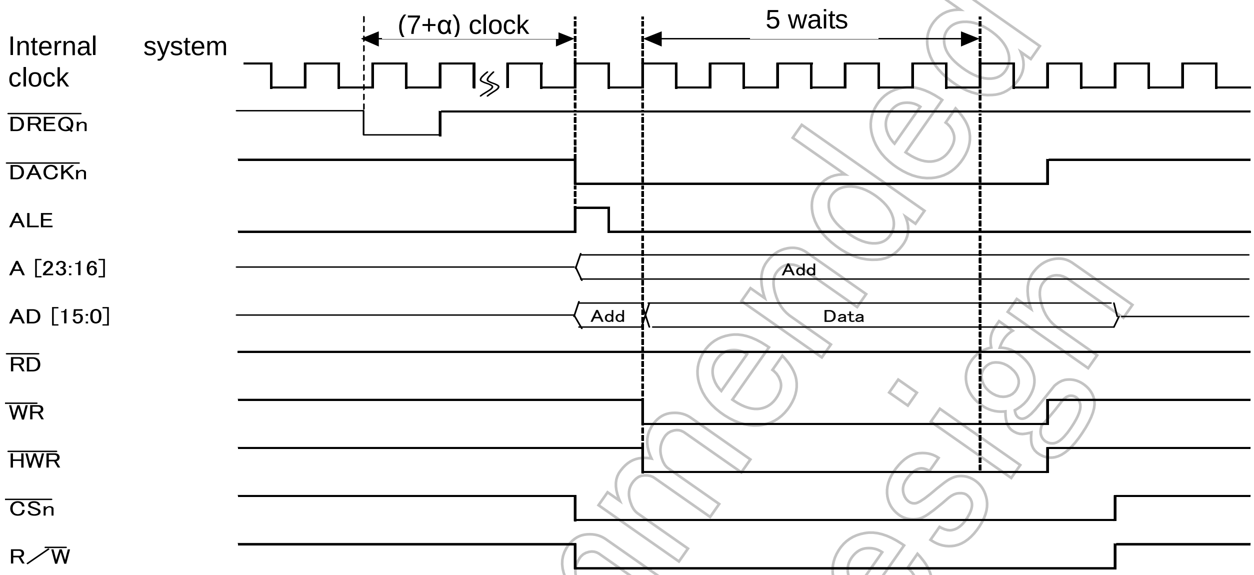


Fig. 10.23 Edge Mode (from Internal RAM to External Memory)

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

  Fig. 10.24 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

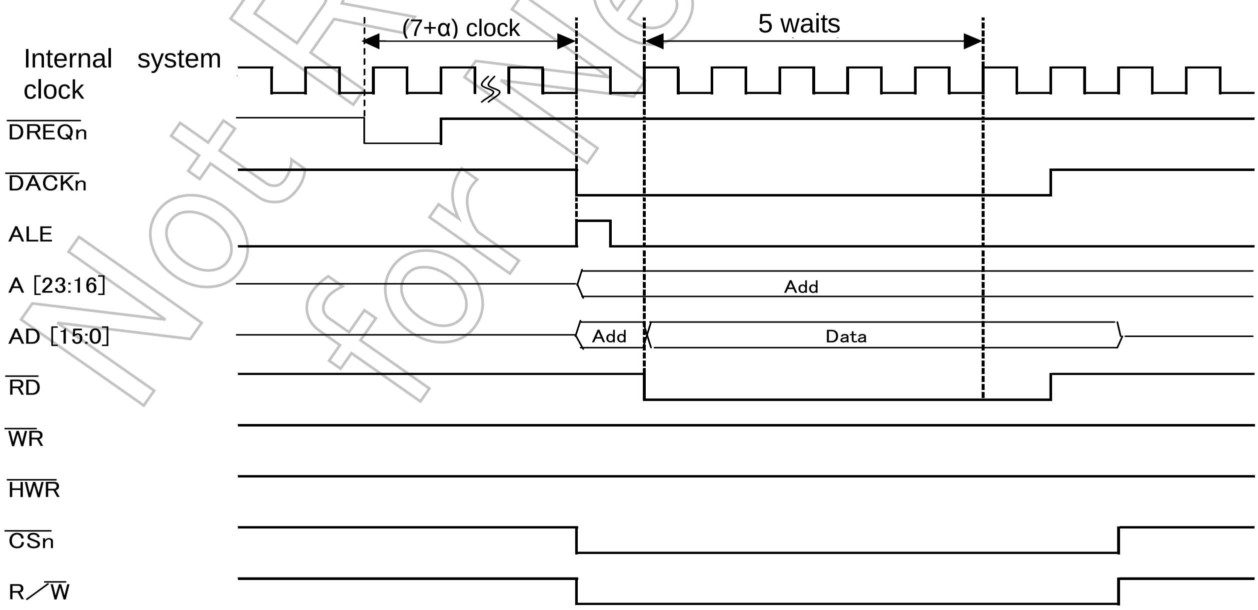


Fig. 10.24 Edge Mode (from External Memory Internal RAM)

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

Fig. 10.25 shows one timing cycle in which 16-bit data is transferred once from internal RAM to external memory (16-bit width).

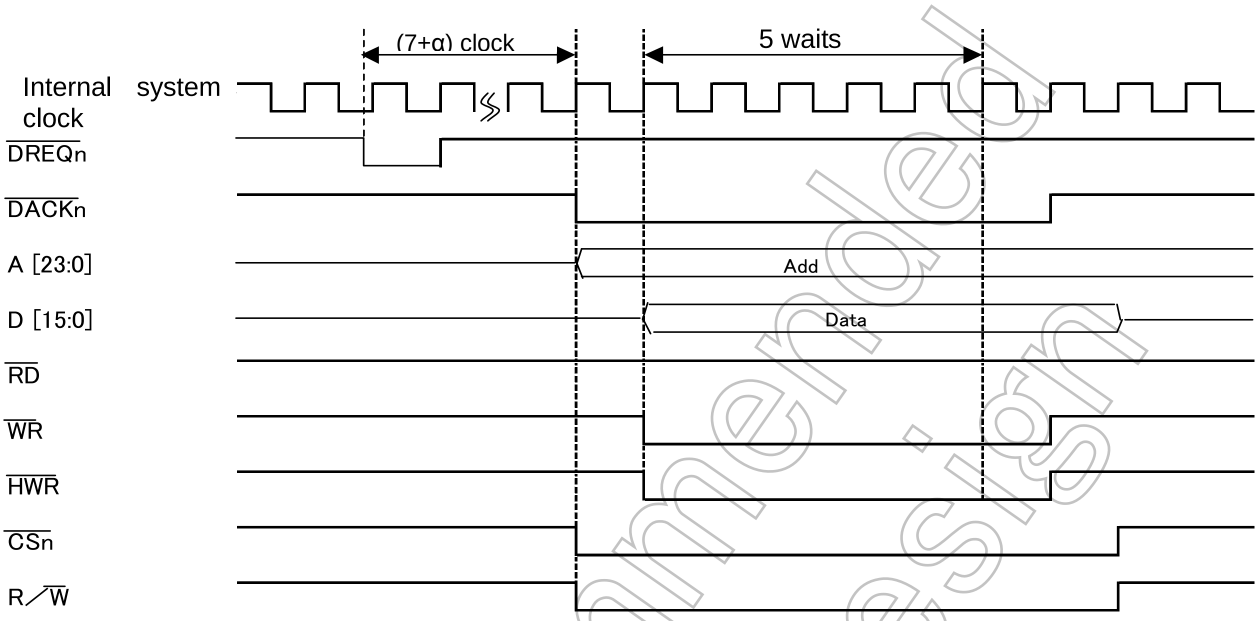


Fig. 10.25 Edge Mode (from Internal RAM to External Memory)

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

Fig. 10.26 shows one timing cycle in which 16-bit data is transferred once from external memory (16-bit width) to internal RAM.

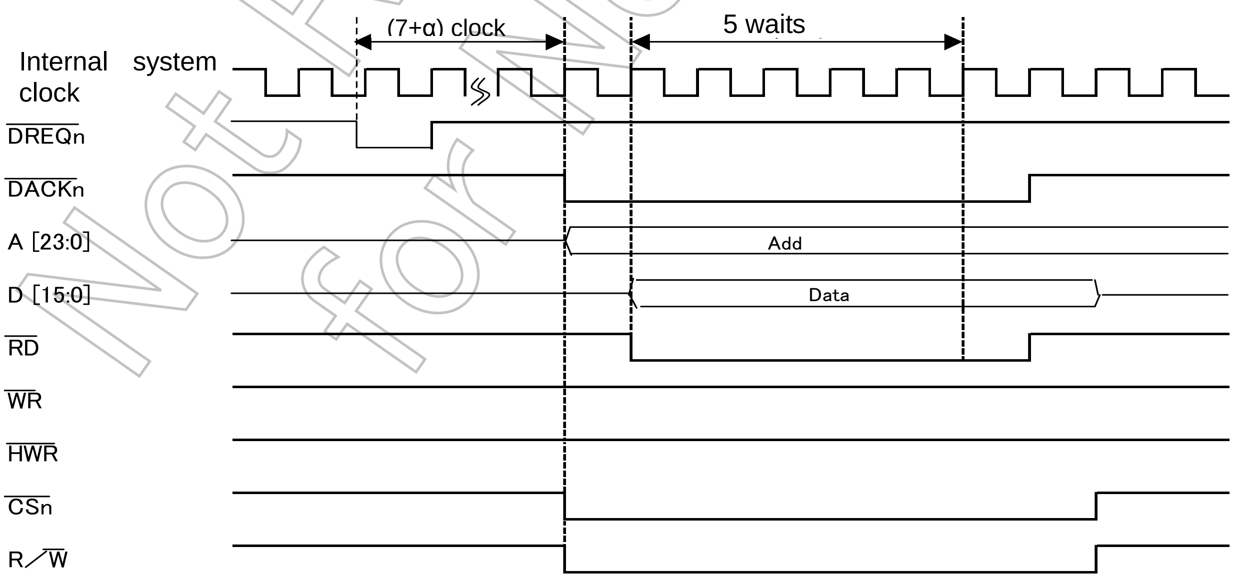


Fig. 10.26 Edge Mode (from External Memory Internal RAM)

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 byte

< Serial channel setting >

- Data length 8 bits: UART
- Serial channel: ch1
- Transfer rate: 9600bps

<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)

31				27				23				19			
1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

15				11				7				3			
0	1	0	1	1	x	1	1	x	0	0	0	1	1	1	1

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

Each of the thirty-six channels (TMRB00 through TMRB23) has a multi-functional 16-bit timer/event counter. TMRBs operate in the following four operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square-wave output (PPG) mode
- Two-phase pulse input counter mode (quad/normal-speed, TMRB0C and TMRB12 only).

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 register.

Each channel (TMRB00~TMRB23) functions independently and the channels operate in the same way, except for the differences in their specifications as shown in Table 11.1 and the two-phase pulse count function. Therefore, the operational descriptions here are only for TMRB14 and for the two-phase pulse count function (TMRB0C, TMRB12).

Table 11.1 Differences in the Specifications of TMRB Modules

Specification \ Channel		TMRB00	TMRB01	TMRB02	TMRB03
External pins	External clock/ capture trigger input pins	-	-	-	-
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers				
Register names (addresses)	Timer RUN register	TB00RUN (0xFFFF_F200)	TB01RUN (0xFFFF_F210)	TB02RUN (0xFFFF_F220)	TB03RUN (0xFFFF_F230)
	Timer control register	TB00CR (0xFFFF_F201)	TB01CR (0xFFFF_F211)	TB02CR (0xFFFF_F221)	TB03CR (0xFFFF_F231)
	Timer mode register	TB00MOD (0xFFFF_F202)	TB01MOD (0xFFFF_F212)	TB02MOD (0xFFFF_F222)	TB03MOD (0xFFFF_F232)
	Timer flip-flop control register	TB00FFCR (0xFFFF_F203)	TB01FFCR (0xFFFF_F213)	TB02FFCR(0xFFFF_F223)	TB03FFCR (0xFFFF_F233)
	Timer status register	TB00ST (0xFFFF_F204)	TB01ST (0xFFFF_F214)	TB02ST (0xFFFF_F224)	TB03ST (0xFFFF_F234)
	Timer up counter register	TB00UCL TB00UCH	TB01UCL TB01UCH	TB02UCL TB02UCH	TB03UCL TB03UCH
	Timer register	TB00RG0L (0xFFFF_F208) TB00RG0H(0xFFFF_F209) TB00RG1L (0xFFFF_F20A) TB00RG1H(0xFFFF_F20B)	TB01RG0L (0xFFFF_F218) TB01RG0H 0xFFFF_F219) TB01RG1L (0xFFFF_F21A) TB01RG1H 0xFFFF_F21B)	TB02RG0L (0xFFFF_F228) TB02RG0H 0xFFFF_F229) TB02RG1L (0xFFFF_F22A) TB02RG1H 0xFFFF_F22B)	TB03RG0L (0xFFFF_F238) TB03RG0H(0xFFFF_F239) TB03RG1L (0xFFFF_F23A) TB03RG1H(0xFFFF_F23B)
	Capture register	TB00CP0L (0xFFFF_F20C) TB00CP0H(0xFFFF_F20D) TB00CP1L (0xFFFF_F20E) TB00CP1H (0xFFFF_F20F)	TB01CP0L (0xFFFF_F21C) TB01CP0H(0xFFFF_F21D) TB01CP1L (0xFFFF_F21E) TB01CP1H (0xFFFF_F21F)	TB02CP0L (0xFFFF_F22C) TB02CP0H 0xFFFF_F22D) TB02CP1L (0xFFFF_F22E) TB02CP1H (0xFFFF_F22F)	TB03CP0L (0xFFFF_F23C) TB03CP0H(0xFFFF_F23D) TB03CP1L (0xFFFF_F23E) TB03CP1H (0xFFFF_F23F)

Specification \ Channel		TMRB04	TMRB05	TMRB06	TMRB07
External pins	External clock/ capture trigger input pins	-	-	-	-
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers				
Register names (addresses)	Timer RUN register	TB04RUN (0xFFFF_F240)	TB05RUN (0xFFFF_F250)	TB06RUN (0xFFFF_F260)	TB07RUN (0xFFFF_F270)
	Timer control register	TB04CR (0xFFFF_F241)	TB05CR (0xFFFF_F251)	TB06CR (0xFFFF_F261)	TB07CR (0xFFFF_F271)
	Timer mode register	TB04MOD (0xFFFF_F242)	TB05MOD (0xFFFF_F252)	TB06MOD (0xFFFF_F262)	TB07MOD (0xFFFF_F272)
	Timer flip-flop control register	TB04FFCR (0xFFFF_F243)	TB05FFCR (0xFFFF_F253)	TB06FFCR(0xFFFF_F263)	TB07FFCR(0xFFFF_F273)
	Timer status register	TB04ST (0xFFFF_F244)	TB05ST (0xFFFF_F254)	TB06ST (0xFFFF_F264)	TB07ST (0xFFFF_F274)
	Timer up counter register	TB04UCL TB04UCH	TB05UCL TB05UCH	TB06UCL TB06UCH	TB07UCL TB07UCH
	Timer register	TB04RG0L (0xFFFF_F248) TB04RG0H(0xFFFF_F249) TB04RG1L (0xFFFF_F24A) TB04RG1H(0xFFFF_F24B)	TB05RG0L (0xFFFF_F258) TB05RG0H(0xFFFF_F259) TB05RG1L (0xFFFF_F25A) TB05RG1H(0xFFFF_F25B)	TB06RG0L (0xFFFF_F268) TB06RG0H(0xFFFF_F269) TB06RG1L(0xFFFF_F26A) TB06RG1H(0xFFFF_F26B	TB07RG0L (0xFFFF_F278) TB07RG0H(0xFFFF_F279) TB70RG1L(0xFFFF_F27A) TB70RG1H(0xFFFF_F27A)
	Capture register	TB04CP0L (0xFFFF_F24C) TB04CP0H(0xFFFF_F24D) TB40CP1L (0xFFFF_F24E) TB40CP1H (0xFFFF_F24F)	TB05CP0L (0xFFFF_F25C) TB05CP0H(0xFFFF_F25D) TB05CP1L (0xFFFF_F25E) TB05CP1H (0xFFFF_F25F)	TB06CP0L(0xFFFF_F26C) TB06CP0H(0xFFFF_F26D) TB06CP1L (0xFFFF_F26E) TB06CP1H(0xFFFF_F26F)	TB07CP0L(0xFFFF_F27C) TB07CP0H(0xFFFF_F27D) TB07CP1L (0xFFFF_F27E) TB07CP1H(0xFFFF_F27F)

Specification \ Channel		TMRB08	TMRB09	TMRB0A	TMRB0B
External pins	External clock/ capture trigger input pins	TB08IN0 (shared with PB0) TB08IN1 (shared with PB1)	TB09IN0 (shared with PB2) TB09IN1 (shared with PB3)	TBAIN0 (shared with PB4) TBAIN1 (shared with PB5)	TBBIN0 (shared with PB6) TBBIN1 (shared with PB7)
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers	**TB1OUT**	**TB1OUT**	**TB1OUT**	**TB1OUT**
Register names (addresses)	Timer RUN register	TB08RUN (0xFFFF_F280)	TB09RUN (0xFFFF_F290)	TB0ARUN (0xFFFF_F2A0)	TB0BRUN (0xFFFF_F2B0)
	Timer control register	TB08CR (0xFFFF_F281)	TB09CR (0xFFFF_F291)	TB0ACR (0xFFFF_F2A1)	TB0BCR (0xFFFF_F2B1)
	Timer mode register	TB08MOD (0xFFFF_F282)	TB09MOD (0xFFFF_F292)	TB0AMOD (0xFFFF_F2A2)	TB0BMOD (0xFFFF_F2B2)
	Timer flip-flop control register	TB08FFCR 0xFFFF_F283)	TB09FFCR(0xFFFF_F293)	TB0AFFCR(0xFFFF_F2A3)	TB0BFFCR (0xFFFF_F2B3)
	Timer status register	TB08ST (0xFFFF_F284)	TB09ST (0xFFFF_F294)	TB0AST (0xFFFF_F2A4)	TB0BST (0xFFFF_F2B4)
	Timer up counter register	TB08UCL TB08UCH	TB09UCL TB09UCH	TB0AUCL TB0AUCH	TB0BUCL TB0BUCH
	Timer register	TB08RG0L(0xFFFF_F288) TB08RG0H(0xFFFF_F289) TB08RG1L(0xFFFF_F28A) TB08RG1H(0xFFFF_F28B)	TB09RG0L(0xFFFF_F298) TB09RG0H(0xFFFF_F299) TB09RG1L(0xFFFF_F29A) TB09RG1H(0xFFFF_F29B)	TB0ARG0L (0xFFFF_F2A8) TB0ARG0H (0xFFFF_F2A9) TB0ARG1L (0xFFFF_F2AA) TB0ARG1H (0xFFFF_F2AB)	TB0BRG0L (0xFFFF_F2B8) TB0BRG0H (0xFFFF_F2B9) TB0BRG1L (0xFFFF_F2BA) TB0BRG1H (0xFFFF_F2BB)
	Capture register	TB08CP0L 0xFFFF_F28C) TB08CP0H(0xFFFF_F28D) TB08CP1L(0xFFFF_F28E) TB08CP1H(0xFFFF_F28F)	TB09CP0L(0xFFFF_F29C) TB09CP0H(0xFFFF_F29D) TB09CP1L(0xFFFF_F29E) TB09CP1H(0xFFFF_F29F)	TB0ACP0L (0xFFFF_F2AC) TB0ACP0H (0xFFFF_F2AD) TB0ACP1L (0xFFFF_F2AE) TB0ACP1H (0xFFFF_F2AF)	TBBCP0L (0xFFFF_F2BC) TBBCP0H (0xFFFF_F2BD) TBBCP1L (0xFFFF_F2BE) TBBCP1H (0xFFFF_F2BF)

Specification \ Channel		TMRB0C	TMRB0D	TMRB0E	TMRB0F
External pins	External clock/ capture trigger input pins	TBCIN0 (shared with PC0) TBCIN1 (shared with PC1)	TBDIN0 (shared with PC2) TBDIN1 (shared with PC3)	TBEIN0 (shared with PC4) TBEIN1 (shared with PC5)	TBFIN0 (shared with PC6) TBFIN1 (shared with PC7)
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers	**TB1OUT**	**TB1OUT**	**TB1OUT**	**TB1OUT**
Register names (addresses)	Timer RUN register	TB0CRUN (0xFFFF_F2C0)	TB0DRUN (0xFFFF_F2D0)	TB0ERUN (0xFFFF_F2E0)	TB0FRUN (0xFFFF_F2F0)
	Timer control register	TB0CCR (0xFFFF_F2C1)	TB0DCR (0xFFFF_F2D1)	TB0ECR (0xFFFF_F2E1)	TB0FCR (0xFFFF_F2F1)
	Timer mode register	TB0CMOD (0xFFFF_F2C2)	TB0DMOD (0xFFFF_F2D2)	TB0EMOD (0xFFFF_F2E2)	TB0FMOD (0xFFFF_F2F2)
	Timer flip-flop control register	TB0CFFCR (0xFFFF_F2C3)	TB0DFFCR (0xFFFF_F2D3)	TB0EFFCR(0xFFFF_F2E3)	TB0FFFCR (0xFFFF_F2F3)
	Timer status register	TB0CST (0xFFFF_F2C4)	TB0DST (0xFFFF_F2D4)	TB0EST (0xFFFF_F2E4)	TB0FST (0xFFFF_F2F4)
	Timer up counter register	TB0CUCL TB0CUCH	TB0DUCL TB0DUCH	TB0EUCL TB0EUCH	TB0FUCL TB0FUCH
	Timer register	TB0CRG0L (0xFFFF_F2C8) TB0CRG0H (0xFFFF_F2C9) TB0CRG1L (0xFFFF_F2CA) TB0CRG1H (0xFFFF_F2CB)	TB0DRG0L (0xFFFF_F2D8) TB0DRG0H (0xFFFF_F2D9) TB0D0RG1L (0xFFFF_F2DA) TB0DRG1H (0xFFFF_F2DB)	TB0ERG0L (0xFFFF_F2E8) TB0ERG0H (0xFFFF_F2E9) TB0ERG1L (0xFFFF_F2EA) TB0ERG1H (0xFFFF_F2EB)	TB0FRG0L (0xFFFF_F2F8) TB0FRG0H (0xFFFF_F2F9) TB0FRG1L (0xFFFF_F2FA) TB0FRG1H (0xFFFF_F2FB)
	Capture register	TB0CCP0L (0xFFFF_F2CC) TB0CCP0H (0xFFFF_F2CD) TB0CCP1L (0xFFFF_F2CE) TB0CCP1H (0xFFFF_F2CF)	TB0DCP0L (0xFFFF_F2DC) TB0DCP0H (0xFFFF_F2DD) TB0DCP1L (0xFFFF_F2DE) TB0DCP1H (0xFFFF_F2DF)	TB0ECP0L (0xFFFF_F2EC) TB0ECP0H (0xFFFF_F2ED) TB0ECP1L (0xFFFF_F2EE) TB0ECP1H (0xFFFF_F2EF)	TB0FCP0L (0xFFFF_F2FC) TB0FCP0H (0xFFFF_F2FD) TB0FCP1L (0xFFFF_F2FE) TB0FCP1H (0xFFFF_F2FF)

Specification \ Channel		TMRB10	TMRB11	TMRB12	TMRB13
External pins	External clock/ capture trigger input pins	TB10IN0 (shared with PD0) TB10IN1 (shared with PD1)	TB11IN0 (shared with PD2) TB11IN1 (shared with PD3)	TB12IN0 (shared with PD4) TB12IN1 (shared with PD5)	-
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers	TB2OUT	TB2OUT	TB2OUT	TB2OUT
Register names (addresses)	Timer RUN register	TB10RUN (0xFFFF_F300)	TB11RUN (0xFFFF_F310)	TB12RUN (0xFFFF_F320)	TB13RUN (0xFFFF_F330)
	Timer control register	TB10CR (0xFFFF_F301)	TB11CR (0xFFFF_F311)	TB12CR (0xFFFF_F321)	TB13CR (0xFFFF_F331)
	Timer mode register	TB10MOD (0xFFFF_F302)	TB11MOD (0xFFFF_F312)	TB12MOD (0xFFFF_F322)	TB13MOD (0xFFFF_F332)
	Timer flip-flop control register	TB10FFCR (0xFFFF_F303)	TB11FFCR (0xFFFF_F313)	TB12FFCR(0xFFFF_F323)	TB13FFCR (0xFFFF_F333)
	Timer status register	TB10ST (0xFFFF_F304)	TB11ST (0xFFFF_F314)	TB12ST (0xFFFF_F324)	TB13ST (0xFFFF_F334)
	Timer up counter register	TB10UCL TB10UCH	TB11UCL TB11UCH	TB12UCL TB12UCH	TB13UCL TB13UCH
	Timer register	TB10RG0L (0xFFFF_F308) TB10RG0H (0xFFFF_F309) TB10RG1L (0xFFFF_F30A) TB10RG1H (0xFFFF_F30B)	TB11RG0L (0xFFFF_F318) TB11RG0H (0xFFFF_F319) TB11RG1L (0xFFFF_F31A) TB11RG1H (0xFFFF_F31B)	TB12RG0L (0xFFFF_F328) TB12RG0H (0xFFFF_F329) TB12RG1L (0xFFFF_F32A) TB12RG1H (0xFFFF_F32B)	TB13RG0L (0xFFFF_F338) TB13RG0H (0xFFFF_F339) TB13RG1L (0xFFFF_F33A) TB13RG1H (0xFFFF_F33B)
	Capture register	TB10CP0L (0xFFFF_F30C) TB10CP0H (0xFFFF_F30D) TB10CP1L (0xFFFF_F30E) TB10CP1H (0xFFFF_F30F)	TB11CP0L (0xFFFF_F31C) TB11CP0H (0xFFFF_F31D) TB11CP1L (0xFFFF_F31E) TB11CP1H (0xFFFF_F31F)	TB12CP0L (0xFFFF_F32C) TB12CP0H (0xFFFF_F32D) TB12CP1L (0xFFFF_F32E) TB12CP1H (0xFFFF_F32F)	TB13CP0L (0xFFFF_F33C) TB13CP0H (0xFFFF_F33D) TB13CP1L (0xFFFF_F33E) TB13CP1H (0xFFFF_F33F)

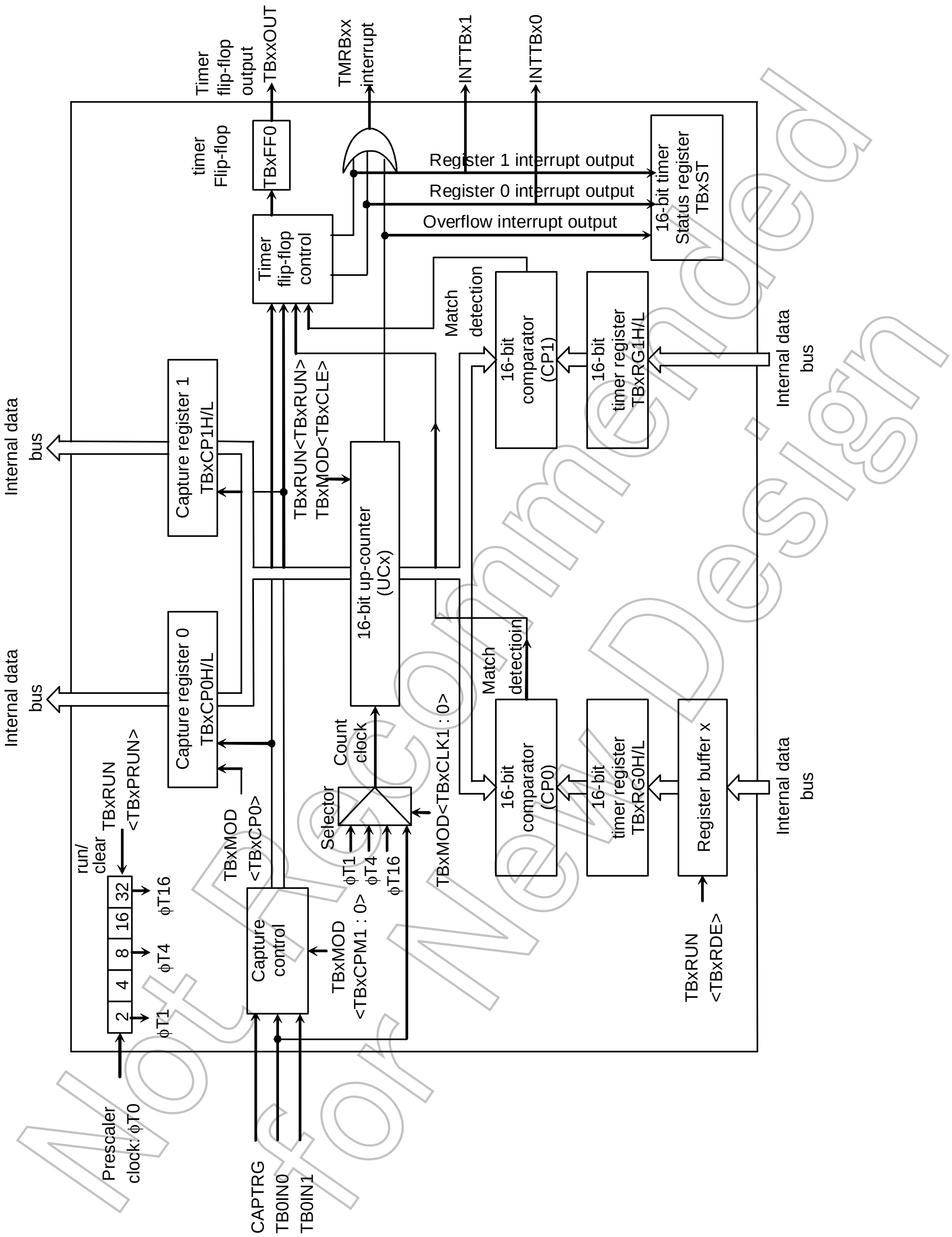
Specification \ Channel		TMRB14	TMRB15	TMRB16	TMRB17
External pins	External clock/ capture trigger input pins	-	-	-	-
	Timer flip-flop output pin	TB14OUT (shared with PD6)	TB15OUT (shared with PD7)	TB16OUT(shared with PE0)	TB17OUT (shared with PE1)
Internal signals	Timer for capture triggers				
Register names (addresses)	Timer RUN register	TB14RUN (0xFFFF_F340)	TB15RUN (0xFFFF_F350)	TB16RUN (0xFFFF_F360)	TB17RUN (0xFFFF_F370)
	Timer control register	TB14CR (0xFFFF_F341)	TB15CR (0xFFFF_F351)	TB16CR (0xFFFF_F361)	TB17CR (0xFFFF_F371)
	Timer mode register	TB14MOD (0xFFFF_F342)	TB15MOD (0xFFFF_F352)	TB16MOD (0xFFFF_F362)	TB17MOD (0xFFFF_F372)
	Timer flip-flop control register	TB14FFCR (0xFFFF_F343)	TB15FFCR (0xFFFF_F353)	TB16FFCR(0xFFFF_F363)	TB17FFCR (0xFFFF_F373)
	Timer status register	TB14ST (0xFFFF_F344)	TB15ST (0xFFFF_F354)	TB16ST (0xFFFF_F364)	TB17ST (0xFFFF_F374)
	Timer up counter register	TB14UCL TB14UCH	TB15UCL TB15UCH	TB16UCL TB16UCH	TB17UCL TB17UCH
	Timer register	TB14RG0L (0xFFFF_F348) TB14RG0H (0xFFFF_F349) TB14RG1L (0xFFFF_F34A) TB14RG1H (0xFFFF_F34B)	TB15RG0L (0xFFFF_F358) TB15RG0H (0xFFFF_F359) TB15RG1L (0xFFFF_F35A) TB15RG1H (0xFFFF_F35B)	TB16RG0L (0xFFFF_F368) TB16RG0H (0xFFFF_F369) TB16RG1L (0xFFFF_F36A) TB16RG1H (0xFFFF_F36B)	TB17RG0L (0xFFFF_F378) TB17RG0H (0xFFFF_F379) TB17RG1L (0xFFFF_F37A) TB17RG1H (0xFFFF_F37B)
	Capture register	TB14CP0L (0xFFFF_F34C) TB14CP0H (0xFFFF_F34D) TB14CP1L (0xFFFF_F34E) TB14CP1H (0xFFFF_F34F)	TB15CP0L (0xFFFF_F35C) TB15CP0H (0xFFFF_F35D) TB15CP1L (0xFFFF_F35E) TB15CP1H (0xFFFF_F35F)	TB16CP0L (0xFFFF_F36C) TB16CP0H (0xFFFF_F36D) TB16CP1L (0xFFFF_F36E) TB16CP1H (0xFFFF_F36F)	TB17CP0L (0xFFFF_F37C) TB17CP0H (0xFFFF_F37D) TB17CP1L (0xFFFF_F37E) TB17CP1H (0xFFFF_F37F)

Specification	Channel	TMRB18	TMRB19	TMRB1A	TMRB1B
External pins	External clock/ capture trigger input pins	-	-	-	-
	Timer flip-flop output pin	TB18OUT (shared with PE2)	TB19OUT (shared with PE3)	TB1AOUT (shared with PE4)	-
Internal signals	Timer for capture triggers				
Register names (addresses)	Timer RUN register	TB18RUN (0xFFFF_F380)	TB19RUN (0xFFFF_F390)	TB1ARUN (0xFFFF_F3A0)	TB1BRUN (0xFFFF_F3B0)
	Timer control register	TB18CR (0xFFFF_F381)	TB19CR (0xFFFF_F391)	TB1ACR (0xFFFF_F3A1)	TB1BCR (0xFFFF_F3B1)
	Timer mode register	TB18MOD (0xFFFF_F382)	TB19MOD (0xFFFF_F392)	TB1AMOD (0xFFFF_F3A2)	TB1BMOD (0xFFFF_F3B2)
	Timer flip-flop control register	TB18FFCR (0xFFFF_F383)	TB19FFCR (0xFFFF_F393)	TB1AFFCR(0xFFFF_F3A3)	TB1BFFCR (0xFFFF_F3B3)
	Timer status register	TB18ST (0xFFFF_F384)	TB19ST (0xFFFF_F394)	TB1AST (0xFFFF_F3A4)	TB1BST (0xFFFF_F3B4)
	Timer up counter register	TB18UCL TB18UCH	TB19UCL TB19UCH	TB1AUCL TB1AUCH	TB1BUCL TB1BUCH
	Timer register	TB18RG0L (0xFFFF_F388) TB18RG0H (0xFFFF_F389) TB18RG1L (0xFFFF_F38A) TB18RG1H (0xFFFF_F38B)	TB19RG0L (0xFFFF_F398) TB19RG0H (0xFFFF_F399) TB19RG1L (0xFFFF_F39A) TB19RG1H (0xFFFF_F39B)	TB1ARG0L (0xFFFF_F3A8) TB1ARG0H (0xFFFF_F3A9) TB1ARG1L (0xFFFF_F3AA) TB1ARG1H (0xFFFF_F3AB)	TB1BRG0L (0xFFFF_F3B8) TB1BRG0H (0xFFFF_F3B9) TB1BRG1L (0xFFFF_F3BA) TB1BRG1H (0xFFFF_F3BB)
	Capture register	TB18CP0L (0xFFFF_F38C) TB18CP0H (0xFFFF_F38D) TB18CP1L (0xFFFF_F38E) TB18CP1H (0xFFFF_F38F)	TB19CP0L (0xFFFF_F39C) TB19CP0H (0xFFFF_F39D) TB19CP1L (0xFFFF_F39E) TB19CP1H (0xFFFF_F39F)	TB1ACP0L (0xFFFF_F3AC) TB1ACP0H (0xFFFF_F3AD) TB1ACP1L (0xFFFF_F3AE) TB1ACP1H (0xFFFF_F3AF)	TB1BCP0L (0xFFFF_F3BC) TB1BCP0H (0xFFFF_F3BD) TB1BCP1L (0xFFFF_F3BE) TB1BCP1H (0xFFFF_F3BF)

Specification	Channel	TMRB1C	TMRB1D	TMRB1E	TMRB1F
External pins	External clock/ capture trigger input pins	-	-	-	-
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers				
Register names (addresses)	Timer RUN register	TB1CRUN (0xFFFF_F3C0)	TB1DRUN (0xFFFF_F3D0)	TB1ERUN (0xFFFF_F3E0)	TB1FRUN (0xFFFF_F3F0)
	Timer control register	TB1CCR (0xFFFF_F3C1)	TB1DCR (0xFFFF_F3D1)	TB1ECR (0xFFFF_F3E1)	TB1FCR (0xFFFF_F3F1)
	Timer mode register	TB1CMOD (0xFFFF_F3C2)	TB1DMOD (0xFFFF_F3D2)	TB1EMOD (0xFFFF_F3E2)	TB1FMOD (0xFFFF_F3F2)
	Timer flip-flop control register	TB1CFFCR (0xFFFF_F3C3)	TB1DFFCR (0xFFFF_F3D3)	TB1EFFCR(0xFFFF_F3E3)	TB1FFFCR (0xFFFF_F3F3)
	Timer status register	TB1CST (0xFFFF_F3C4)	TB1DST (0xFFFF_F3D4)	TB1EST (0xFFFF_F3E4)	TB1FST (0xFFFF_F3F4)
	Timer up counter register	TB1CUCL TB1CUCH	TB1DUCL TB1DUCH	TB1EUCL TB1EUCH	TB1FUCL TB1FUCH
	Timer register	TB1CRG0L (0xFFFF_F3C8) TB1CRG0H (0xFFFF_F3C9) TB1CRG1L (0xFFFF_F3CA) TB1CRG1H (0xFFFF_F3CB)	TB1DRG0L (0xFFFF_F3D8) TB1DRG0H (0xFFFF_F3D9) TB1DRG1L (0xFFFF_F3DA) TB1DRG1H (0xFFFF_F3DB)	TB1ERG0L (0xFFFF_F3E8) TB1ERG0H (0xFFFF_F3E9) TB1ERG1L (0xFFFF_F3EA) TB1ERG1H (0xFFFF_F3EB)	TB1FRG0L (0xFFFF_F3F8) TB1FRG0H (0xFFFF_F3F9) TB1FRG1L (0xFFFF_F3FA) TB1FRG1H (0xFFFF_F3FB)
	Capture register	TB1CCP0L (0xFFFF_F3CC) TB1CCP0H (0xFFFF_F3CD) TB1CCP1L (0xFFFF_F3CE) TB1CCP1H (0xFFFF_F3CF)	TB1DCP0L (0xFFFF_F3DC) TB1DCP0H (0xFFFF_F3DD) TB1DCP1L (0xFFFF_F3DE) TB1DCP1H (0xFFFF_F3DF)	TB1ECP0L (0xFFFF_F3EC) TB1ECP0H (0xFFFF_F3ED) TB1ECP1L (0xFFFF_F3EE) TB1ECP1H (0xFFFF_F3EF)	TB1FCP0L (0xFFFF_F3FC) TB1FCP0H (0xFFFF_F3FD) TB1FCP1L (0xFFFF_F3FE) TB1FCP1H (0xFFFF_F3FF)

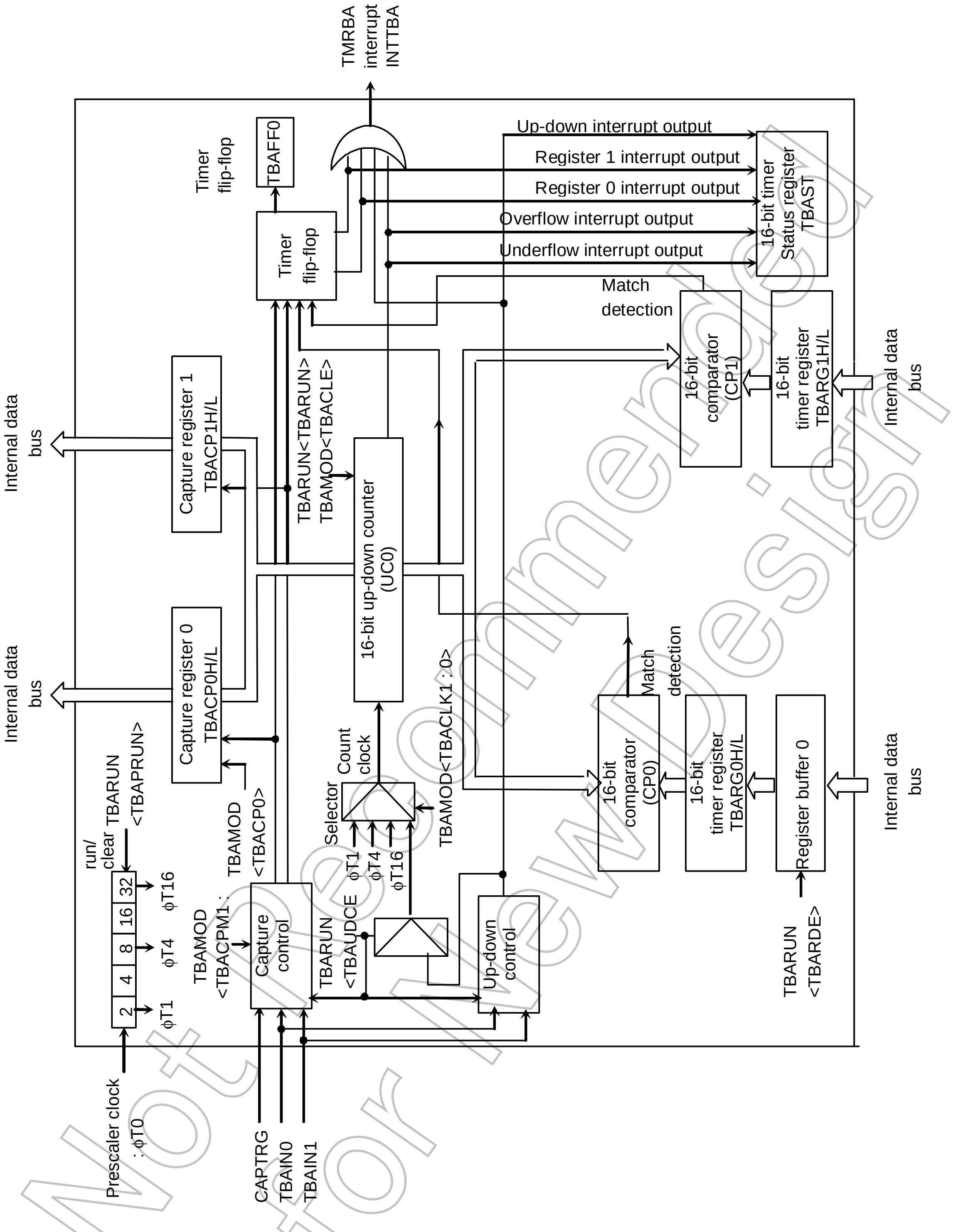
Specification \ Channel		TMRB20	TMRB21	TMRB22	TMRB23
External pins	External clock/ capture trigger input pins	-	-	-	-
	Timer flip-flop output pin	-	-	-	-
Internal signals	Timer for capture triggers				
Register names (addresses)	Timer RUN register	TB20RUN (0xFFFF_F400)	TB21RUN (0xFFFF_F410)	TB22RUN (0xFFFF_F420)	TB23RUN (0xFFFF_F430)
	Timer control register	TB20CR (0xFFFF_F401)	TB21CR (0xFFFF_F411)	TB22R (0xFFFF_F421)	TB23CR (0xFFFF_F431)
	Timer mode register	TB20MOD (0xFFFF_F402)	TB21MOD (0xFFFF_F412)	TB22MOD (0xFFFF_F422)	TB23MOD (0xFFFF_F432)
	Timer flip-flop control register	TB20FFCR (0xFFFF_F403)	TB21FFCR (0xFFFF_F413)	TB22FFCR(0xFFFF_F423)	TB23FFCR (0xFFFF_F433)
	Timer status register	TB20ST (0xFFFF_F404)	TB21ST (0xFFFF_F414)	TB22ST (0xFFFF_F424)	TB23ST (0xFFFF_F434)
	Timer up counter register	TB20UCL TB20UCH	TB21UCL TB21UCH	TB22UCL TB22UCH	TB23UCL TB23UCH
	Timer register	TB20RG0L (0xFFFF_F408) TB20RG0H (0xFFFF_F409) TB20RG1L (0xFFFF_F40A) TB20RG1H (0xFFFF_F40B)	TB21RG0L (0xFFFF_F418) TB21RG0H (0xFFFF_F419) TB21RG1L (0xFFFF_F41A) TB21RG1H (0xFFFF_F41B)	TB22RG0L (0xFFFF_F428) TB22RG0H (0xFFFF_F429) TB22RG1L (0xFFFF_F42A) TB22RG1H (0xFFFF_F42B)	TB23RG0L (0xFFFF_F438) TB23RG0H (0xFFFF_F439) TB23RG1L (0xFFFF_F43A) TB23RG1H (0xFFFF_F43B)
	Capture register	TB20CP0L (0xFFFF_F40C) TB20CP0H (0xFFFF_F40D) TB20CP1L (0xFFFF_F40E) TB20CP1H (0xFFFF_F40F)	TB21CP0L (0xFFFF_F41C) TB21CP0H (0xFFFF_F41D) TB21CP1L (0xFFFF_F41E) TB21CP1H (0xFFFF_F41F)	TB22CP0L (0xFFFF_F42C) TB22CP0H (0xFFFF_F42D) TB22CP1L (0xFFFF_F42E) TB22CP1H (0xFFFF_F42F)	TB23CP0L (0xFFFF_F43C) TB23CP0H (0xFFFF_F43D) TB23CP1L (0xFFFF_F43E) TB23CP1H (0xFFFF_F43F)

11.1 Block Diagram of Each Channel



(Note) TMRB0 through TMRB7 have no input pins for external clock or capture trigger.

Fig. 11.1.1 TMRB14 Block Diagram (Same for Channels 15 through 1A)



(Note) There is no output pin for TMRB00 through 13, 1B through 23.

Fig. 11.1.2 TMRB00 (same for channels 01 through 13 and 1B through 23) Block Diagram

11.2 Description of Operations for Each Circuit

11.2.1 Prescaler

There is a 5-bit prescaler for acquiring the TMRB14 clock source. 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 TB14RUN<TB14PRUN> where writing “1” starts counting and writing “0” clears and stops counting. Table 11.2.1 shows prescaler output clock resolutions.

Table 11.2.1 Prescaler Output Clock Resolutions @fc = 54MHz

Release peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1 : 0>	Prescaler output clock resolutions		
			φT1	φT4	φT16
0 (fgear)	000 (fc)	00(fperiph/16)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{3}(0.15 μs)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	fc/2^{2}(0.07 μs)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)
	100 (fc/2)	00(fperiph/16)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)	fc/2^{10}(18.96 μs)
		01(fperiph/8)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		10(fperiph/4)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
		11(fperiph/2)	fc/2^{3}(0.15 μs)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)
	110 (fc/4)	00(fperiph/16)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)	fc/2^{11}(37.93 μs)
		01(fperiph/8)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)	fc/2^{10}(18.96 μs)
		10(fperiph/4)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		11(fperiph/2)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
	111 (fc/8)	00(fperiph/16)	fc/2^{8}(4.74 μs)	fc/2^{10}(18.96 μs)	fc/2^{12}(75.85 μs)
		01(fperiph/8)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)	fc/2^{11}(37.93 μs)
		10(fperiph/4)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)	fc/2^{10}(18.96 μs)
		11(fperiph/2)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
1 (fc)	000 (fc)	00(fperiph/16)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{3}(0.15 μs)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	fc/2^{2}(0.07 μs)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)
	100 (fc/2)	00(fperiph/16)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{3}(0.15 μs)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	-	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)
	110 (fc/4)	00(fperiph/16)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	-	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	-	fc/2^{4}(0.30 μs)	fc/2^{6}(1.19 μs)
	111 (fc/8)	00(fperiph/16)	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	-	fc/2^{6}(1.19 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	-	fc/2^{5}(0.59 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	-	-	fc/2^{6}(1.19 μs)

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

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

(Note 3) **"-" denotes a setting prohibited.**

11.2.2 Up-counter (UC0) and Up-counter Capture Registers (TBxxUCL,TBxxUCH)

This is the 16-bit binary counter that counts up in response to the input clock specified by TBxxMOD<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 TBxxIN0 pin. For UC0, start, stop and clear are specified by TB0RUN<TB0RUN> and if UC0 matches the TBxxRG1H / L timer register, it is cleared to "0" provided the setting is "clear enable." Clear enable/disable is specified by TBxxMOD<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 TBxxUCL andTBxxUCH 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 INTTBxx overflow interrupt is generated.

TMRB0C has the two-phase pulse input count function. The two-phase pulse count mode is activated by TB0CRUN<TB0CUDCE>. This counter serves as the up-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 reloaded. When the two-phase pulse count mode is not active, the counter counts up only.

11.2.3 Timer Registers (TBxxRG0H/L, TBxxRG1H/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 TBxxRG0H/L and TBxxRG1H/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.

TBxxRG0 of this timer register is paired with register buffer 0 - a double-buffered configuration. TBxxRG0 uses TBxxRUN<TB0RDE> to control the enabling/disabling of double buffering. 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 TBxxRG0 timer register when there is a match between UC0 and TBxxRG1.

The values of TBxxRG0 and TBxxRG1 become undefined after a reset; therefore it is necessary to write data to them beforehand in case of using a 16-bit timer. 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.

TBxxRG0 and the register buffers are assigned to the same address: 0xFFFF_FxxA/0xFFFF_FxxB. If <TB0RDE> = "0," the same value is written to TBxxRG0 and each register buffer; if <TB0RDE> = "1," the value is only written to each register buffer. To write an initial value to the timer register, therefore, the register buffers must be set to "disable."

11.2.4 Capture Registers (TBxxCP0H/L, TBxxCP1H/L)

These are 16-bit registers for latching values from the UC0 up-counter. The data in the capture register must be read out in the order of low-order bits followed by high-order bits by using the 1 byte data transfer instruction twice.

(Do not read out the data while executing 2 bytes transfer instruction.)

11.2.5 Capture

This is a circuit that controls the timing of latching values from the UC0 up-counter into the TBxxCP0 and TBxxCP1 capture registers. The timing with which to latch data is specified by TBxxMOD<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 TBxxCP0 capture register each time “0” is written to TBxxMOD<TB0CP0>. To use this capability, the prescaler must be running (TBxxRUN<TB0PRUN> =“1”).

In the two-phase pulse count mode (for the TMRB0C), 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 do not stop the timer after a read of low-order 8 bits.

11.2.6 Comparators (CP0, CP1)

These are 16-bit comparators for detecting a match by comparing set values of the UC0 up-counter with set values of the TBxxRG0 and TBxxRG1 timer registers. If a match is detected, INTTB0 is generated.

11.2.7 Timer Flip-flop (TBxxFF0)

The timer flip-flop (TBxxFF0) 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 TBxxFFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>.

The value of TBxxFF0 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 TBxxFF0 can be output to the timer output pin, TBxxOUT (shared with a port). To enable timer output, the port related registers PxCR and PxFC1 must be programmed beforehand.

11.3 Register Description

TMRBn RUN register (n=00 ~ 23, except for 0C and 12)

TBnRUN (0xFFFF_F2x0)

	7	6	5	4	3	2	1	0
Bit symbol	TBnWBUF				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 Buffer 0: Disable 1: Enable	Write “0”.	Write “0”.	Write “0”.	IDLE 0: Stop 1: Operation	Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as “0.”		

<TBnRUN>: Controls the TMRBn count operation.

<TBnPRUN>:Controls the TMRBn prescaler operation.

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

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

TMRB0C RUN register

TB0CRUN (0xFFFF_F2C0)

	7	6	5	4	3	2	1	0
Bit symbol	TB0CRDE		UDACK	TB0CUDCE	I2TBA	TB0CPRUN		TB0CRUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0: Disable 1: Enable	Write “0”.	Sampling clock 0: 1: ΦT0/4	Enable/ disable two-phase counter 0: Disable 1: Enable	IDLE 0: Stop 1: Operation	Timer Run/Stop Control 0: Stop & clear 1: Count * The first bit can be read as “0.”		

<TB0CRUN>:Controls the TMRB0C count operation.

<TB0CPRUN>:Controls the TMRB0C prescaler operation.

<I2TBA>:Controls the operation in the IDLE mode.

<TB0CUDCE>: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.

<UD0CCK>:Selects the two-phase pulse input sampling clock.

<TB0CRDE>:Controls enabling/disabling of double buffering.

TMRBn control register（n=00 ~ 23）

TBnCR (0xFFFF_Fxx1)		7	6	5	4	3	2	1	0
	Bit symbol	TBnEN							
	Read/Write	R/W	R/W	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
	Function	TMRBn operation 0: Disable 1: Enable	Write “0”.	This can be read as “0”.	This can be read as “0”.	This can be read as “0”.	This can be read as “0”.	This can be read as “0”.	This can be read as “0”.

＜TBnEN＞: Specifies the TMRB operation. When the operation is disabled, no clock is supplied to the other registers in the TMRB module. This can reduce power consumption. (This disables reading from and writing to the other registers.) To use the TMRB, enable the TMRB operation (set to “1”) before programming each register in the TMRB module. If the TMRB operation is executed and then disabled, the settings will be maintained in each register.

TMRBn mode register（n=00 ~ 23, except for 0C and 12）

TBnMOD (0xFFFF_Fxx2)		7	6	5	4	3	2	1	0
	Bit symbol		TB**n** RSWR	TBnCP0	TBnCPM1	TBnCPM0	TBnCLE	TBnCLK1	TBnCLK0
	Read/Write	R	R/W	W	R/W				
	After reset	0	0	1	0	0	0	0	0
	Function	This can be read as “0”.	Writing to timer registers 0,1 0：Always enabled 1:Enabled simultaneously	Capture control by software 0: Capture by software 1: Don't care	Capture timing 00: Disable 01: TBnIN0↑ TBnIN1↑ 10: TBnIN0↑ TBnIN0↓ 11: CAPTRG↑ CAPTRG↓		Up-counter control 0: Clear/ disable 1: Clear/ enable	Selects source clock 00: TBnIN0 pin input 01: φT1 10: φT4 11: φT16	

＜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 rising of TBnIN0 pin input. Takes count values into capture register 1 (TBnCP1) upon rising of TBnIN1 pin input.

“10” :Takes count values into capture register 0 (TBnCP0) upon rising of TBnIN0 pin input. Takes count values into capture register 1 (TBnCP1) upon falling of TBnIN0 pin input.

“11” :Takes count values into capture register 0 (TBnCP0) upon rising of timer output for capture trigger (CAPTRG) and into capture register 1 (TBnCP1) upon falling of CAPTRG (**CAPTRG for TMRB08 ~ 0F: TB1OUT, for TMRB10 ~ 13: TB2OUT**).

<TBnCP0>:Captures count values by software and takes them into capture register 0 (TBnCP0).

<TBnRSWR>: Controls writing timing to timer registers 0 and 1 when using double buffer.

“0”:Writing to the timer registers 0 and 1 is enabled individually if either of them is ready to be written.

“1”:Writing to the timer registers 0 and 1 is enabled only when both are ready to be written.

(Note) The value read from bit 5 of TBnMOD is “1”.

TMRBn mode register

TBnMOD
(0xFFFF_Fxx2)

	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		1	0	0	0	0	0
Function	This can be read as "0".		Capture control by software 0: Capture by software 1: Don't care	Capture timing 00: Disable 01: TBnIN0 ↑ TBnIN1 ↑ 10: TBnIN0 ↑ TBnIN0 ↓ 11: CAPTRG ↑ CAPTRG ↓		Up-counter control 0: Clear/ disable 1:Clear/ enable	Selects source clock 00: TBnIN0 pin input 01: φT1 10: φT4 11: φT16	

<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 timer output for capture trigger (CAPTRG) and into capture register 1 (TBnCP1) upon the falling of CAPTRG (**CAPTRG for TMRB08 ~ 0F: TB1OUT, for TMRB10 ~ 13: TB2OUT**).

<TBnCP0>: Captures count values by software and takes them into capture register 0 (TBnCP0).

TMRBn flip-flop control register（n=14 ~ 1A）

TBnFFCR
(0xFFFF_Fxx3)

	7	6	5	4	3	2	1	0
Bit symbol			TBnC1T1	TBnC0T1	TBnE1T1	TBnE0T1	TBnFF0C1	TBnFF0C0
Read/Write	R		R/W				W	
After reset	1	1	0	0	0	0	1	1
Function	This is always read as "11."		TBnFF0 reverse trigger 0: Disable trigger 1: Enable trigger				TBnFF0 control 00: Invert 01: Set 10: Clear 11: Don't care * This is always read as "11."	
			When the up-counter value is taken into TBnCP1.	When the up-counter value is taken into TBnCP0.	When the up-counter matches TBnRG1.	When the up-counter matches TBnRG0.		

<TBnFF0C1:0>: Controls the timer flip-flop.

"00" : Reverses the value of TBnFF0 (reverse by using software).

"01" : Sets TBnFF0 to "1."

"10" : Clears TBnFF0 to "0."

"11":Don't care

(Note) This is always read as "11."

<TBnE1:0>: Reverses the timer flip-flop when the up-counter matches the timer register 0,1 (TBnRG0,1).

<TBnC1:0>: Reverses the timer flip-flop when the up-counter value is taken into the capture register 0,1 (TBnCP0,1).

(Note) Do not change the setting of TBnMOD and TBnFFCR registers while timer is in operation (TB0RUN＝"H").

TMRBn status registers (1)

TMRBn status registers (n=00 ~ 23)

TBnST
(0xFFFF_Fxx4)

	7	6	5	4	3	2	1	0
Bit symbol						INTTBOFn	INTTBn1	INTTBn0
Read/Write	R					R		
After reset	0					0	0	0
Function	This can be read as “0”.					0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated	0: Interrupt not generated 1: Interrupt generated

<INTTBn0>:Interrupt generated if there is a match with timer register 0 (TBnRG0)
<INTTBn1>:Interrupt generated if there is a match with timer register 1 (TBnRG1)
<INTTBOFn>:Interrupt generated if an up-counter overflow occurs

（Note）If any interrupt is generated, the flag that corresponds to the interrupt is set to TBnST and the generation of interrupt is notified to INTC. The flag is cleared by reading the TBnST register.

TMRB0C status registers (2)

① When TB0CRUN＜TBAUDCE＞= 0: Normal timer mode

TB0CST
(0xFFFF_F2C4)

	7	6	5	4	3	2	1	0
Bit symbol						INTTBOFC	INTTBC1	INTTBC0
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

＜INTTBC0>: Interrupt generated if there is a match with timer register 0 (TB0CRG0)
<INTTBC1>: Interrupt generated if there is a match with timer register 1 (TB0CRG1)
<INTTBOFC>:Interrupt generated if an up-counter overflow occurs

② When TB0CRUN＜TBAUDCE＞= 1: Two-phase pulse input count mode

TB0CST
(0xFFFF_F2C4)

	7	6	5	4	3	2	1	0
Bit symbol				INTTBUDC	INTTBUDFC	INTTBOUFC		
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”.	

<INTTBOVFC>:Interrupt generated if an up-and-down counter overflow occurs
<INTTBUDFC>:Interrupt generated if an up-and-down counter underflow occurs
<INTTBUDC>: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 TB0CST and the generation of interrupt is notified to INTC. The flag is cleared by reading the TB0CST register.

TBnIM mask registers

TBnIM mask registers（n=00 ~ 23, except for 0C and 12）

		7	6	5	4	3	2	1	0
TBnIM (0xFFFF_Fxx5)	Bit symbol						TBIMOFn	TBIMn1	TBIMn0
	Read/Write	R					W/R	W/R	W/R
	After reset	0					0	0	0
	Function	This can be read as “0”.					1：Mask INTTBOFn	1:Mask INTTBn1	1:Mask INTTBn0

<TBIMOFn>:Masks an over-flow interrupt.
<TBIMn1>:Masks an interrupt if there is a match between timer register 1 and counter value.
<TBIMn0>:Masks an interrupt if there is a match between timer register 0 and counter value.

TBnRG0H/L, TBnRG1H/L timer registers

TBnRG0H/L timer registers（n=00 ~ 23）

		7	6	5	4	3	2	1	0
TBnRG0L (0xFFFF_Fxx8)	Bit symbol	TBnRG0L 7	TBnRG0L 6	TBnRG0L 5	TBnRG0L 4	TBnRG0L 3	TBnRG0L 2	TBnRG0L 1	TBnRG0L 0
	Read/Write	R/W							
	After reset	Undefined							
	Function	Timer capture value, Data of low-order 8 bits							

		7	6	5	4	3	2	1	0
TBnRG0H (0xFFFF_Fxx9)	Bit symbol	TBnRG0H 7	TBnRG0H 6	TBnRG0H 5	TBnRG0H 4	TBnRG0H 3	TBnRG0H 2	TBnRG0H 1	TBnRG0H 0
	Read/Write	R/W							
	After reset	Undefined							
	Function	Timer capture 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=00 ~ 23）

		7	6	5	4	3	2	1	0
TBnRG1L (0xFFFF_F1xA)	Bit symbol	TBnRG1L 7	TBnRG1L 6	TBnRG1L 5	TBnRG1L 4	TBnRG1L 3	TBnRG1L 2	TBnRG1L 1	TBnRG1L 0
	Read/Write	R/W							
	After reset	Undefined							
	Function	Timer capture value, Data of low-order 8 bits							

		7	6	5	4	3	2	1	0
TBnRG1H (0xFFFF_F1xB)	Bit symbol	TBnRG1H 7	TBnRG1H 6	TBnRG1H 5	TBnRG1H 4	TBnRG1H 3	TBnRG1H 2	TBnRG1H 1	TBnRG1H 0
	Read/Write	R/W							
	After reset	Undefined							
	Function	Timer capture 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, TBnCP1H/L capture registers

TBnCP0H/L capture registers (n=00 ~ 23)

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=00 ~ 23)

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 TB00RG1 timer register.

		7	6	5	4	3	2	1	0	
TB00CR		1	0	X	X	X	X	X	X	Starts the TMRB0 module.
TB00RUN	←	0	0	0	0	–	0	X	0	Stops the TMRB0.
IMC5	←	X	1	1	0	X	1	0	0	Enables INTTB0, and sets it to level 4.
		X	–	–	0	X	–	–	–	(Setting of INTTB0 only is shown here. This is a 32-bit
		X	–	–	0	X	–	–	–	register and requires settings of other interrupts as
		X	–	–	0	X	–	–	–	well.)
TB00FFCR	←	X	X	0	0	0	0	–	–	Disables the trigger.
TB00MOD	←	X	X	1	0	0	1	*	*	Designates the prescaler output clock as the input clock,
TB00RG1L	←	*	*	*	*	*	*	*	*	and specifies the time interval.
TB00RG1H		*	*	*	*	*	*	*	*	(16-bit)
TB00RUN	←	0	0	0	0	–	1	X	1	Starts the TMRB0.

X; Don't care –; no change

11.4.2 16-bit Event Counter Mode

<< By using an input clock as an external clock (TBxIN0 pin input), it is possible to make it the event counter.>>

The up-counter counts up on the rising edge of TBxIN0 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	
TBxxCR	←	1	0	X	X	X	X	X	X	Starts the TMRBxx module.
TBxxRUN	←	0	0	0	0	–	0	X	0	Stops the TMRBxx.
PxCR	←	–	–	–	–	–	–	–	0	Sets Px0 to the input mode.
PxFC1	←	–	–	–	–	–	–	–	1	
IMC5	←	X	1	1	0	X	1	0	0	Enables INTTBxx, and sets it to level 4.
		X	–	–	0	X	–	–	–	(Setting of INTTB0 only is shown here. This is a 32-bit
		X	–	–	0	X	–	–	–	register and requires settings of other interrupts as well.)
		X	–	–	0	X	–	–	–	
TBxxFFCR	←	X	X	0	0	0	0	–	–	Disables the trigger.
TBxxMOD	←	X	X	1	0	0	1	0	0	Designates the TBxxIN0 pin input as the input clock.
TBxxRUN	←	0	0	0	0	–	1	X	1	Starts the TMRBxx.
TBxxMOD	←	X	X	0	0	0	1	0	0	Captures a value using software.
TBxxCP0L	←	*	*	*	*	*	*	*	*	Reads the lower 8-bit count value
TBxxCP0H	←	*	*	*	*	*	*	*	*	Reads the higher 8-bit count value.

X; Don't care –; no change

To be used as the event counter, put the prescaler in a "RUN" state (TBxxRUN<TBxxPRUN> = “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 TBxxOUT pin by triggering the timer flip-flop (TBxxFF) to reverse when the set value of the up-counter (UCO) matches the set values of the timer registers (TBxxRG0H/L、TBxxRG1H/L). Note that the set values of TBxxRG0H/L and TBxxRG1H/L must satisfy the following requirement:

(Set value of TBxxRG0H/L) < (Set value of TBxxRG1H/L)

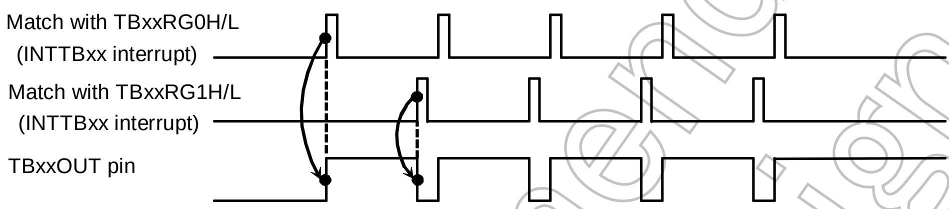


Fig. 11.4.3.1 Example of Output of Programmable Square Wave (PPG)

In this mode, by enabling the double buffering of TBxxRG0H/L, the value of register buffer 0 is shifted into TBxxRG0H/L when the set value of the up-counter matches the set value of TBxxRG1H/L. This facilitates handling of small duties.

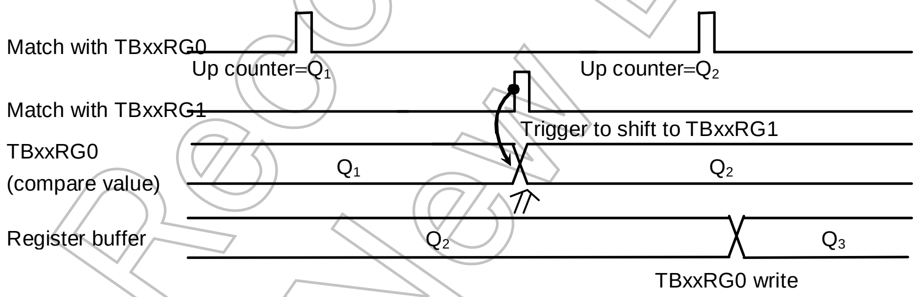


Fig. 11.4.3.2 Register Buffer Operation

The block diagram of this mode is shown below.

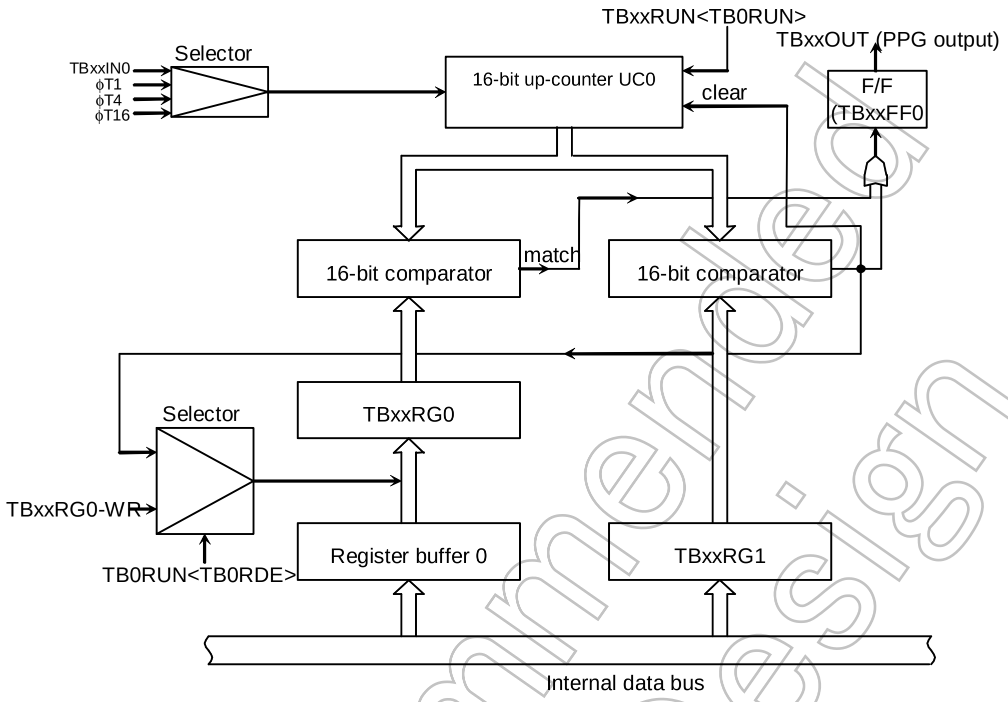


Fig. 11.4.3.3 Block Diagram of 16-bit PPG Mode

<< Each register in the 16-bit PPG output mode must be programmed as listed below. **>>**

		7	6	5	4	3	2	1	0	
TBxxCR	←	1	0	X	X	X	X	X	X	Starts the TMRBxx module.
TBxxRUN	←	0	0	0	0	–	0	X	0	Disables the TBxxRG0 double buffering and stops TMRBxx.
TBxxRG0L	←	*	*	*	*	*	*	*	*	Specifies a duty. (16 bits)
TBxxRG0H	←	*	*	*	*	*	*	*	*	
TBxxRG1L	←	*	*	*	*	*	*	*	*	Specifies a cycle. (16 bits)
TBxxRG1H	←	*	*	*	*	*	*	*	*	
TBxxRUN	←	1	0	0	0	–	0	X	0	Enables the TBxxRG0 double buffering. (Changes the duty/cycle when the INTTBxx interrupt is generated)
TBxxFFCR	←	X	X	0	0	1	1	1	0	Specifies to trigger TBxxFF0 to reverse when a match with TBxxRG0 or TBxxRG1 is detected, and sets the initial value of TBxxFF0 to "0."
TBxxMOD	←	X	X	1	0	0	1	*	*	Designates the prescaler output clock as the input clock, and disables the capture function.
						(** = 01, 10, 11)				
PxCR	←	–	–	–	–	–	1	–	–	Assigns Px2 to TBxxOUT.
PxFC1	←	–	–	–	–	–	1	–	–	
TBxxRUN	←	1	0	0	0	–	1	X	1	Starts TMRBxx.

X; Don't care –; no change

11.4.4 Applications using the Capture Function

The capture function can be used to develop many applications, including those described below:

① One-shot pulse output triggered by an external pulse
② Frequency measurement
③ Pulse width measurement
④ Time difference measurement

① One-shot pulse output triggered by an external pulse

One-shot pulse output triggered by an external pulse is carried out as follows:

The 16-bit up-counter UC0 is made to count up by putting it in a free-running state using the prescaler output clock. An external pulse is input through the TBxxIN0 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 (TBxxCP0H/L).

The INTC must be programmed so that an interrupt INTx is generated at the rising of an external trigger pulse. This interrupt is used to set the timer registers (TBxxRG0H/L) to the sum of the TBxxCP0H/L value (c) and the delay time (d), (c + d), and set the timer registers (TBxxRG1H/L) to the sum of the TBxxRG0H/L values and the pulse width (p) of one-shot pulse, (c + d + p).

In addition, the timer flip-flop control registers (TBxxFFCR<TBxxE1T1, TBxxE0T1>) must be set to "11." This enables triggering the timer flip-flop (TB5FF0) to reverse when UC5 matches TBxxRG0H/L and TB0RG1H/L. This trigger is disabled by the INTTBxx interrupt after a one-shot pulse is output.

Symbols (c), (d) and (p) used in the text correspond to symbols c, d and p in Fig. 11.4.4.1.

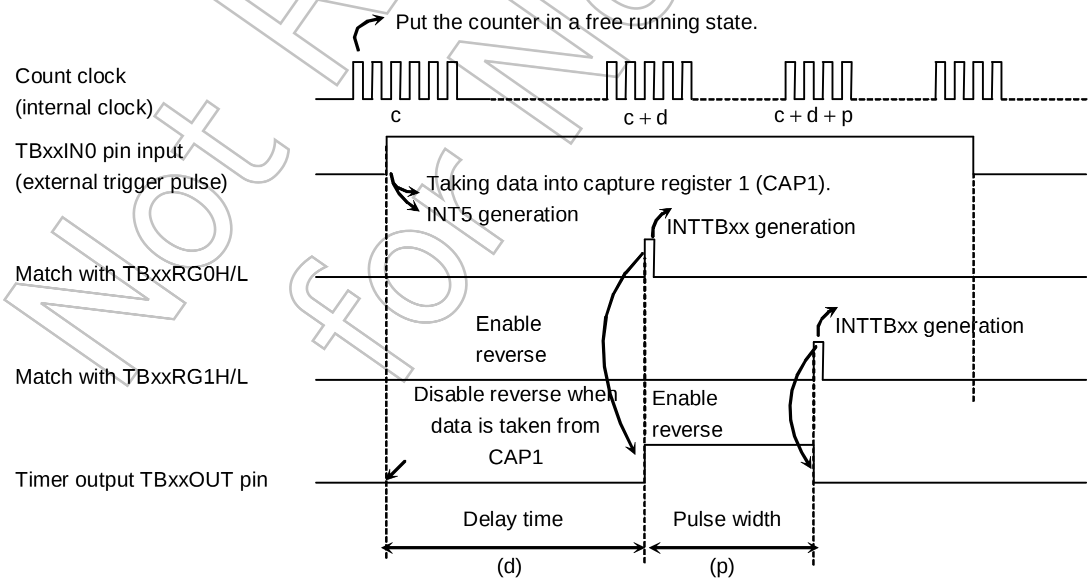


Fig. 11.4.4.1 One-shot Pulse Output (With Delay)

Programming example: Output a 2-ms one-shot pulse triggered by an external pulse from the TBxxIN0 pin with a 3-ms delay

* Clock conditions		
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	
TBxxCR	←	1	0	X	X	X	X	X	X	Start the TMRBxx module.
TBxxMOD	←	X	X	1	0	1	0	0	1	Puts to a free-running state. Uses ϕT1 for counting. Takes data into TBxxCP0 at the rising of TBxxIN0 input
TBxxFFCR	←	X	X	0	0	0	0	1	0	Clears TBxxFF0 to zero. Disables TBxxFF0 to reverse.
PxCR	←	–	–	–	–	–	1	–	–	Assigns Px2 pin to TBxxOUT.
PxFC1	←	–	–	–	–	–	1	–	–	
IMC1	←	X	–	–	0	X	–	–	–	Enables INT5. These are 32-bit registers and must be all processed.
		X	1	1	0	X	1	0	0	
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	
IMC5	←	X	1	1	0	X	0	0	0	Disables INTTBxx. These are 32-bit registers and must be all processed.
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	
TBxxRUN	←	–	0	0	0	–	1	X	1	Starts TMRBxx.

INT0 での設定

TBxxRG0L	←	*	*	*	*	*	*	*	*	TBxxCP0 + 3ms/ϕT1
TBxxRG0H	←	*	*	*	*	*	*	*	*	
TBxxRG1L	←	*	*	*	*	*	*	*	*	TBxxRG0 + 2ms/ϕT1
TBxxRG1H	←	*	*	*	*	*	*	*	*	
TBxxFFCR	←	X	X	–	–	1	1	–	–	Enables TBxxFF0 to reverse when there is a match with TBxxRG0, 1.
IMC5	←	X	1	1	0	X	1	0	0	Enables INTTBxx.
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	

INTTBxx での設定

TBxxFFCR	←	X	X	–	–	0	0	–	–	Disables TBxxFF0 to reverse when there is a match with TBxxRG0, 1
IMC5	←	X	1	1	0	X	0	0	0	Disables INTTBxx.
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	
		X	–	–	0	X	–	–	–	

X; Don't care —;no change

If a delay is not required, TBxxFF0 is reversed when data is taken into TBxxCP0H/L, and TBxxRG1L/H is set to the sum of the TBxxCP0H/L value (c) and the one-shot pulse width (p), (c + p), by generating the INT5 interrupt. TB5FF0 is enabled to reverse when UC0 matches with TBxxRG1L/H, and is disabled by generating the INTTBxx interrupt.

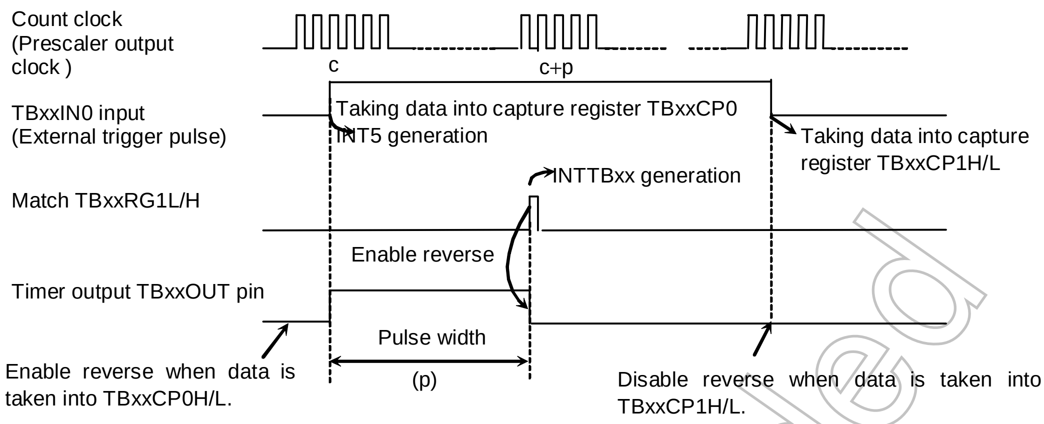


Fig. 11.4.4.2 One-shot Pulse Output Triggered by an External Pulse (Without Delay)

② Frequency measurement

The frequency of an external clock can be measured by using the capture function.

To measure frequency, another 16-bit timer (TMRB01) is used in combination with the 16-bit event counter mode (TMRB01 reverses TB01FFCR to specify the measurement time).

The TBxxIN0 pin input is selected as the TMRBxx count clock to perform the count operation using an external input clock. TBxxMOD<TBxxCPM1 : 0> is set to "11." This setting allows a count value of the 16-bit up-counter UC0 to be taken into the capture register (TBxxCP0) upon rising of a timer flip-flop (TB1FFCR) of the 16-bit timer (TMRB1), and an UC0 counter value to be taken into the capture register (TBxxCP1H/L) upon falling of TB1FF of the 16-bit timer (TMRB01).

A frequency is then obtained from the difference between TBxxCP0H/L and TBxxCP1H/L based on the measurement, by generating the INTTB1 16-bit timer interrupt.

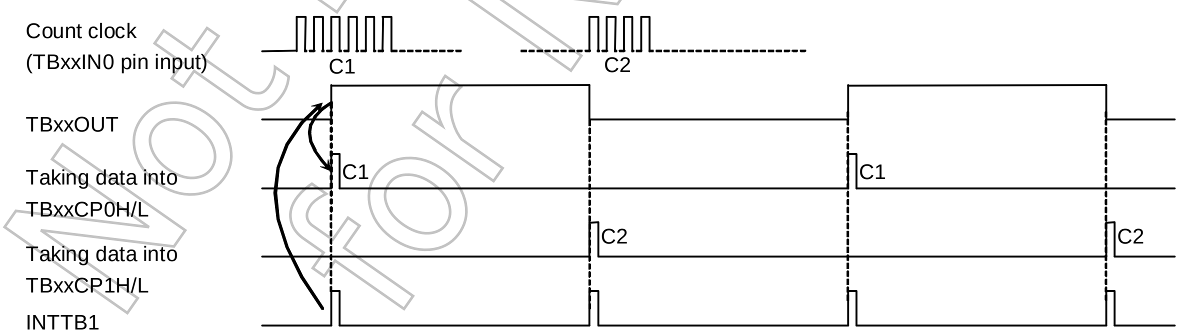


Fig. 11.4.4.3 Frequency Measurement

For example, if the set width of TB1FF level "1" of the 16-bit timer is 0.5 s and if the difference between TBxxCP0H/L and TBxxCP1H/L is 100, the frequency is 100 / 0.5 s = 200 Hz.

③ Pulse width measurement

By using the capture function, the "H" level width of an external pulse can be measured. Specifically, by putting it in a free-running state using the prescaler output clock, an external pulse is input through the TBxxIN1 pin and the up-counter (UC5) is made to count up. 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 (TBxxCP0H/L, TBxxCP1H/L). The INTC must be programmed so that INTCPTxx is generated at the falling edge of the TBxxIN1 pin.

The "H" level pulse width can be calculated by multiplying the difference between TBxxCP0H/L and TBxxCP1H/L by the clock cycle of an internal clock.

For example, if the difference between TBxxCP0H/L and TBxxCP1H/L is 100 and the cycle of the prescaler output clock is 0.5 *u*s, the pulse width is 100×0.5 *u*s $=$ 50 *u*s.

Caution must be exercised when measuring pulse widths exceeding the UC0 maximum count time which is dependant upon the source clock used. The measurement of such pulse widths must be made using software.

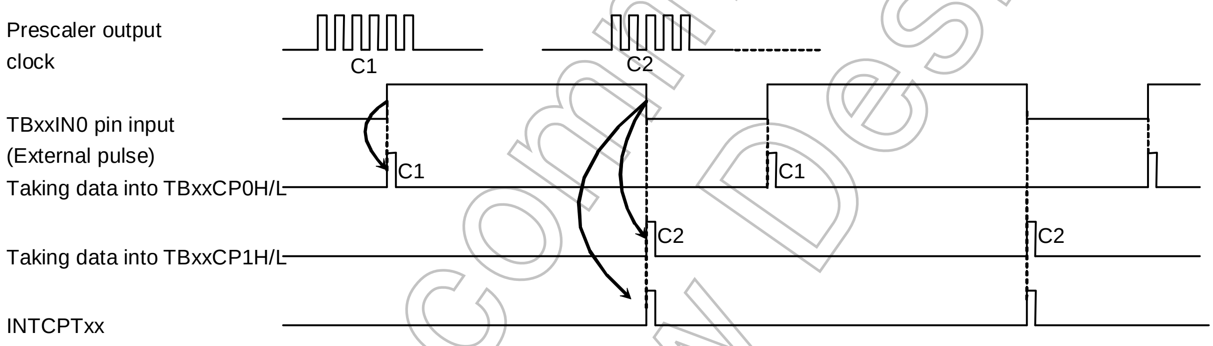


Fig. 11.4.4.4 Pulse Width Measurement

The "L" level width of an external pulse can also be measured. In such cases, the difference between C2 generated the first time and C1 generated the second time is initially obtained by performing the second stage of INTCPT interrupt processing as shown in "Fig. 11.4.4.5 Time Difference Measurement" and this difference is multiplied by the cycle of the prescaler output clock.

(Note) INTCPTxx interruption is generated when the value of the up-counter is taken into the capture register TBxxCP1H/L.

④ Time Difference Measurement

The up-counter (UC0) is made to count up by putting it in a free-running state using the prescaler output clock. The value of UC0 is taken into the capture register (TBxxCP0H/L) at the rising edge of the TBxxIN0 pin input pulse.

The value of UC0 is taken into the capture register TBxxCP1H/L at the rising edge of the TBxxIN1 pin input pulse. The INTC must be programmed to generate INTCPTxx interrupt at this time.

The time difference can be calculated by multiplying the difference between TBxxCP1H/L and TBxxCP0H/L by the clock cycle of an internal clock.

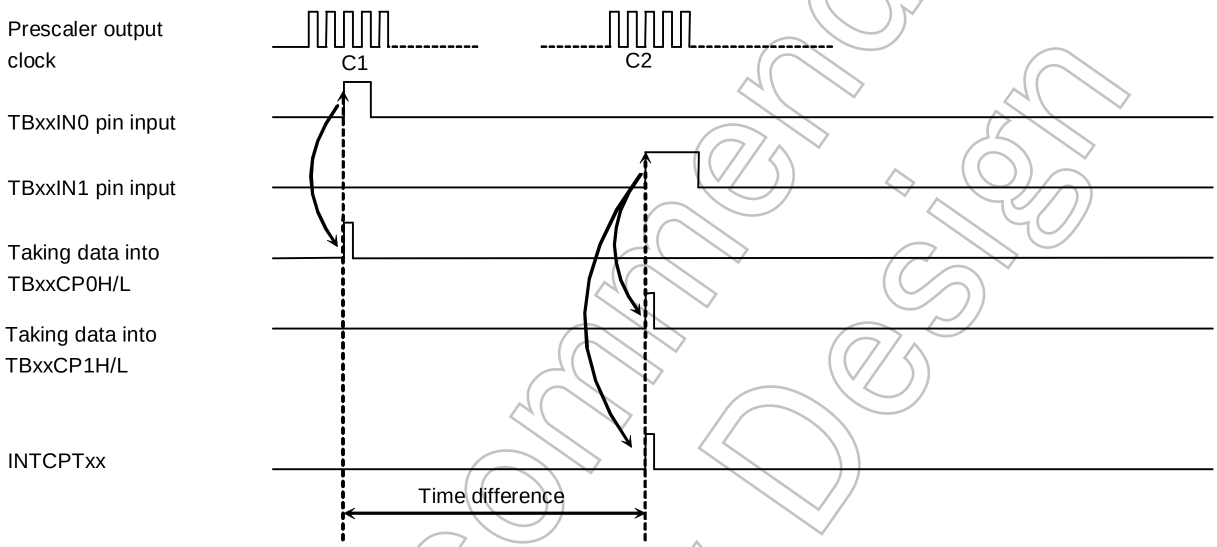


Fig. 11.4.4.5 Time Difference Measurement

(Note) INTCPTxx interruption is generated when the value of UC0 is taken into the capture register TBxxCP1H/L.

11.4.5 Two-phase Pulse Input Count Mode (TMRB0C)

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 TB0CIN0 and TB0CIN1 and has phase difference. Interrupt is output in the ups and downs counter mode by the count operation.

This is a quadruple mode that performs count-up/ down in every modes.

TMRB0C has the same two phase pulse mode as TMRB12. Here we give a detailed description of TMRB0C.

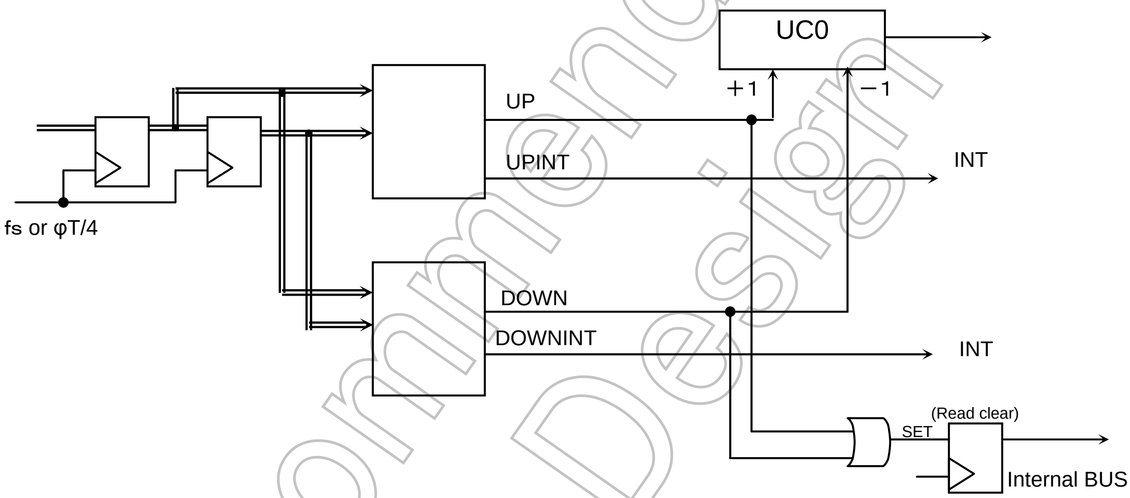
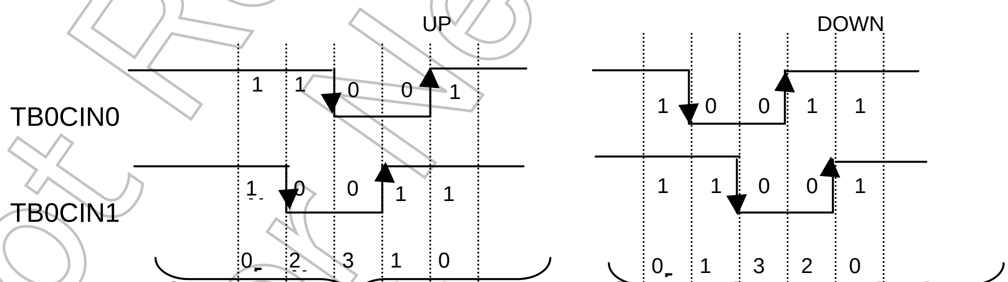


Fig. 11.4.5.1 Count Circuit of Two-phase Counter

11.4.6 Quadruple mode



① Count up at each edge **② Count down at each edge**

	Pin state					
Count condition	UP			DOWN		
TB0CIN0,TB0CIN1	0	→	2	0	→	1
	2		3	1		3
	3		1	3		2
	1		0	2		0

TMR0C RUN register (TB0CRUN)

TB0CRUN (0xFFFF_F2C0)

	7	6	5	4	3	2	1	0
Bit symbol	TB0CRDE		UD0CCK	TB0CUDCE	I2TB0C	TB0CPRUN		TB0CRUN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	Double Buffer 0: Disable 1: Enable	Write “0”.	Sampling clock selection 1:φT0/4	Enable/ disable two-phase counter 0: Disable 1: Enable	IDLE 0: Stop 1: Operation	Timer Run/Stop Control 0: Stop & Clear 1: Run (Count Up)		

Fig. 11.4.6.1 Two-phase Pulse Input Count Mode Setting Register

Set the 5th bit of TB0CRUN register <UDCCK> to “1” as a sampling clock.

② Interrupt

- In the NORMAL mode

The INTTB0C interrupt is enabled using the interrupt controller (INTC). The INTTB0C interrupt is generated by counting up or down. Reading the status register TB0CST during interrupt handling allows simultaneous check for occurrences of an overflow and an underflow. If TB0CST<INTTBOUFC> is “1,” it indicates that an overflow has occurred. If <INTTBUDF0C> 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.

TB0CST (0xFFFF_F1E4)

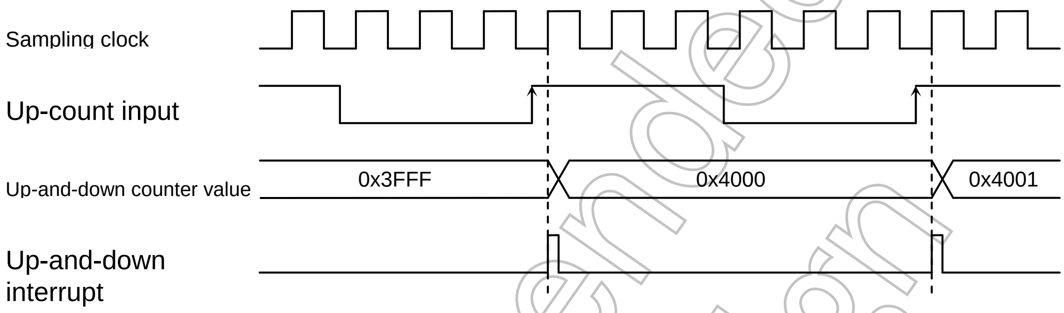
	7	6	5	4	3	2	1	0
Bit symbol				INTTBUD0C	INTTBUDF0C	INTTBOUF0C		
Read/Write	R			R			R	
After reset	0			0	0	0	0	
Function	This can be read as “0”.			Up-and-down count 0: Not occurred 1: Occurred	Underflow 0: Not occurred 1: Occurred	Overflow 0: Not occurred 1: Occurred	This can be read as “0”.	

Fig. 11.4.6.2 TMRB0C status register

(Note) The status is cleared after the register is read.

③ Up-and-down counter

When the two-phase input count mode is selected (TB0CRUN<TB0CUDCE> = “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 TB0CST 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 INTTB0C interrupt handling.

12. 32-bit Input Capture (TMRC)

TMRC consists of two channels (TBTA and TBTB) with a 32-bit time base timer (TBT), two channels (CAPx0~1) each with a 32-bit input capture register, and two channels (CMPx0~1) each with a 32-bit compare register.

TBTA and TBTB operate individually and have the same operational structure. Here we are going to explain the case of TBTA.

Fig. 12.1 shows the TMRC block diagram.

12.1 TMRC Block Diagram

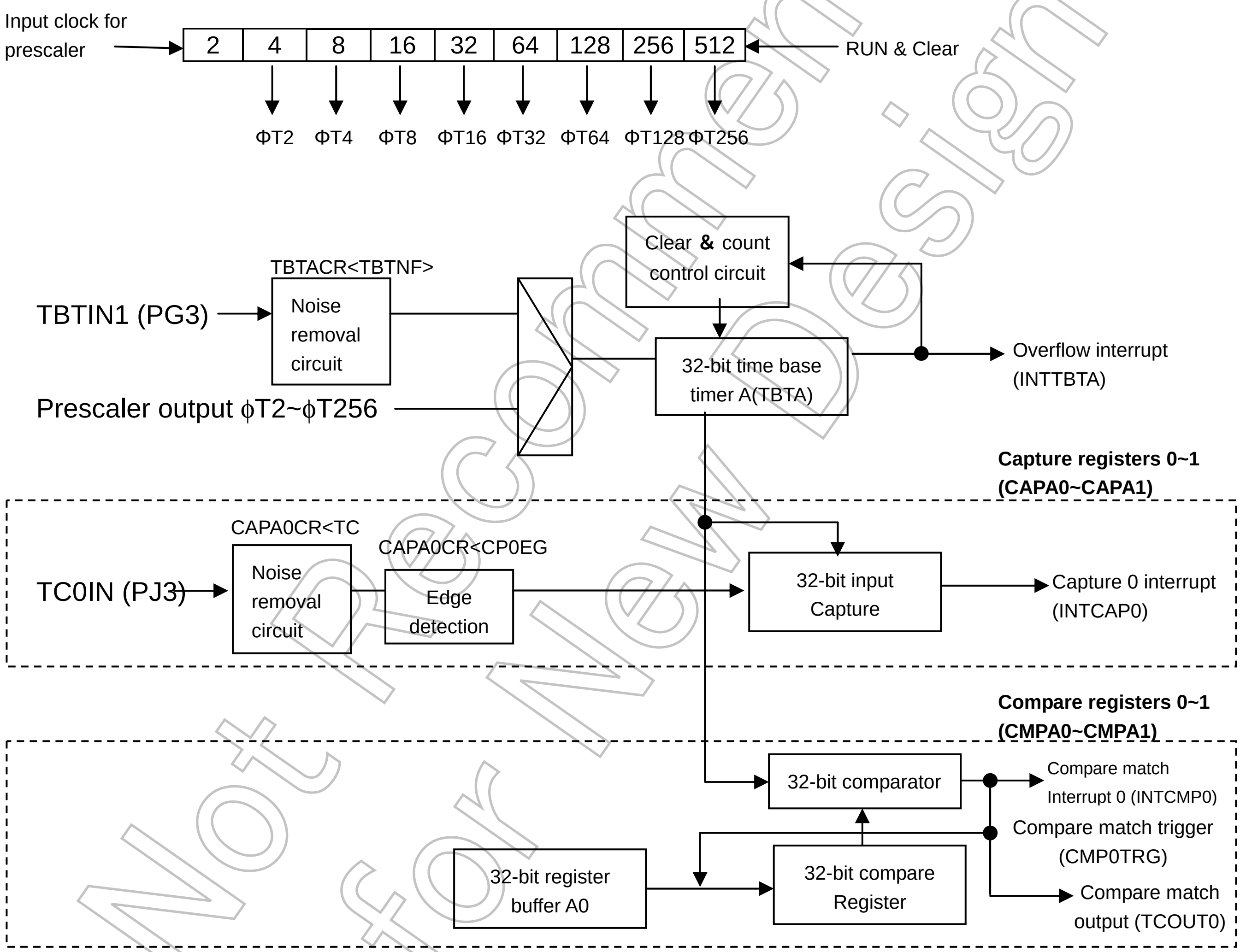


Fig. 12.1 Timer C Block Diagram

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 TBTACR<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 TBTARUN<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

(when high speed clock gear is selected from 1/1, 1/2, 1/4 and 1/8)

@fc = 54MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolution			
			ΦT2	ΦT4	ΦT8	ΦT16
0(fgear)	000(fc)	00(fperiph/16)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	fc/2^{3}(0.15 μs)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)
	100(fc/2)	00(fperiph/16)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
		01(fperiph/8)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		10(fperiph/4)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
		11(fperiph/2)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)
	110(fc/4)	00(fperiph/16)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		01(fperiph/8)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
		10(fperiph/4)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		11(fperiph/2)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
	111(fc/8)	00(fperiph/16)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		01(fperiph/8)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		10(fperiph/4)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
		11(fperiph/2)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
1(fc)	000(fc)	00(fperiph/16)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	fc/2^{3}(0.15 μs)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)
	100(fc/2)	00(fperiph/16)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	fc/2^{3}(0.15 μs)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)
	110(fc/4)	00(fperiph/16)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	–	fc/2^{4}(0.30 μs)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)
	111(fc/8)	00(fperiph/16)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)
		01(fperiph/8)	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)
		10(fperiph/4)	–	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)	fc/2^{7}(2.37 μs)
		11(fperiph/2)	–	–	fc/2^{5}(0.59 μs)	fc/2^{6}(1.19 μs)

@fc = 54MHz

Select peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Select prescaler clock <PRCK1:0>	Prescaler output clock resolution			
			ΦT32	ΦT64	ΦT128	ΦT256
0(fgear)	000(fc)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	100(fc/2)	00(fperiph/16)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)
		01(fperiph/8)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		10(fperiph/4)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		11(fperiph/2)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
	110(fc/4)	00(fperiph/16)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)	fc/2^{15}(606.8 μs)
		01(fperiph/8)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)
		10(fperiph/4)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		11(fperiph/2)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
	111(fc/8)	00(fperiph/16)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)	fc/2^{15}(606.8 μs)	fc/2^{16}(1213.6 μs)
		01(fperiph/8)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)	fc/2^{15}(606.8 μs)
		10(fperiph/4)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)	fc/2^{14}(303.4 μs)
		11(fperiph/2)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
1(fc)	000(fc)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	100(fc/2)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	110(fc/4)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)
	111(fc/8)	00(fperiph/16)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)	fc/2^{13}(151.7 μs)
		01(fperiph/8)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)	fc/2^{12}(75.85 μs)
		10(fperiph/4)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)	fc/2^{11}(37.93 μs)
		11(fperiph/2)	fc/2^{7}(2.37 μs)	fc/2^{8}(4.74 μs)	fc/2^{9}(9.48 μs)	fc/2^{10}(18.96 μs)

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

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

(Note 3) **"-" denotes "setting prohibited."**

12.2.2 Noise Removal Circuit

The noise removal circuit removes noises from an external clock source input (TBTIN) and a capture trigger input (TCnIN) of the time base timer (TBTA). 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 TBTA control register TBTACR.

Based on the TBTCR<TBTCLK3 : 0> setting, an input clock is selected from external clocks supplied through the TBTIN1 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 TBTARUN<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. INTTBTA is controlled by CAPINT and CMPINT that categorized in the same group as INTCAPn described in the part of 32-bit capture register.

This counter can perform a read capture operation. When it is performing a read capture operation, it is possible to read a counter value by accessing the TBTA read capture register (TBTARDCAP) 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 captured" by provisioning the capture control register CAPAnCR<CPnEG1:0>. Fig. 12.2.4.1 shows capture inputs, outputs (capture factor outputs) produced by the edge detection circuit.

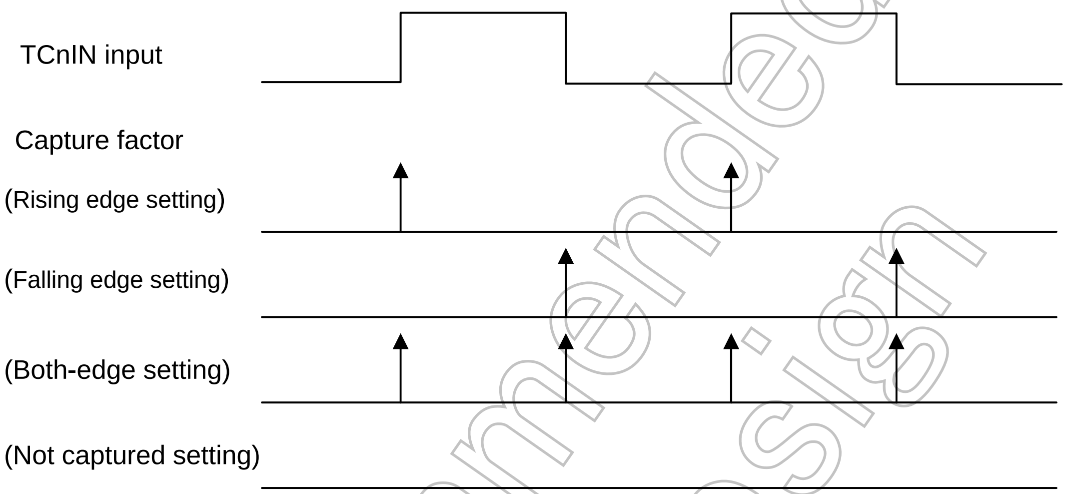


Fig. 12.2.4.1 Capture Inputs and Capture Factor Outputs (Outputs Produced by the Edge Detection Circuit)

12.2.5 32-bit Capture Register

This is a 32-bit register for capturing count values of TBTA by using capture factors as triggers. If a capture operation is performed, the capture interrupt INTCAPn is generated. Two interrupt requests INTCAP0 through INTCAP1 are grouped into one set of interrupt requests which are then notified to the interrupt controller. Which one of interrupt requests must be processed can be identified by reading the status register TCGST during interrupt processing. Additionally, it is possible to mask unnecessary interrupts by setting the interrupt mask register TCGIM to an appropriate bit setting. While a read of the capture register is ongoing, count values cannot be captured even if there are triggers.

12.2.6 32-bit Compare Register

This is a 32-bit register for specifying a compare value. TMRC has two built-in compare registers, CMPA0 and CMPA1. If values set in these compare registers match the value of TBTA, the match detection signal of a comparator becomes active. "Compare enable" or "compare disable" can be specified with the compare control register CMPCTL<CMPEN1:0>.

To set TCCMPn to a specific value, data must be transferred to TCCMPn in the order of lower to higher bits by using a byte data transfer instruction four times.

CMPAn 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 CMPAn takes place when the value of TBTA matches that of CMPAn.

Because CMPAn is indeterminate when a reset is performed, it is necessary to prepare and write data in advance. A reset initializes CMPACTL <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.

CMPAn and the register buffer are assigned to the same address. If < CMPRDEn > is "0," the same value is written to CMPAn 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

TCACR (FFFFF500H)

	7	6	5	4	3	2	1	0
bit Symbol	TCEN	I2TBT						
Read/Write	R/W		R					
After reset	0	0	0	0	0	0	0	0
Function	TMRC operation 0: Disable 1: Enable	IDLE 0: Stop 1: Operation						

<I2TBT>: Controls the operation in idle mode.

<TCEN>: Specifies enabling/disabling of the TMRC operation. If set to "disable," a clock is not supplied to other registers of the TMRC module and, therefore, a reduction in power consumption is possible (a read of or a write to other registers cannot be executed). To use TMRC, the TMRC operation must be set to "enable" ("1") before making individual register settings of TMRC modules. If TMRC is operated and then set to "disable," individual register settings are retained.

(Note) TCCR bits 0~5 are read as "0".

TBTRUN register

TBTARUN (FFFFF501H)

	7	6	5	4	3	2	1	0
bit Symbol						TBTCAP	TBTPRUN	TBTRUN
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function					Make sure to write "0".	TBT counter software capture 0: D'ont Care 1: software capture	Timer Run/Stop Control 0: stop & clear 1: count	

<TBTRUN>:Controls the TBT count operation.

<TBTPRUN>:Controls the TBT prescaler operation.

<TBTCAP>: If this is set to "1," the count value of the time base timer (TBT) is taken into the capture register TBTCAPn.

(Note) TBTRUN bits 4~7 are read as "0".

Fig. 12.3.1 TMRC-related Registers

TBT control register

TBTACR (FFFFF502H)		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	TBTIN0 Input noise removal 0:disable 1:enable	Make sure to write “0”.			TBT source clock 0000: ϕT2 0001: ϕT4 0010: ϕT8 0011: ϕT16 0100: ϕT32 0101: ϕT64 0110: ϕT128 0111: ϕT256 1111: TBTIN pin input			

<TA0CLK3:0>: This is an input clock for TBT. Clocks from "0000" to "0111" are available as prescaler output clocks. A clock "1111" is input through the TBTIN pin.

<TBTNF>: Controls the noise removal for the TBTIN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph＝fc＝54MHz) is accepted as a source clock for TBT, at whichever level the TBTIN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph＝fc＝54MHz) is regarded as noise and removed, at whichever level the TBTIN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

TBTA capture register（TBTACAP）

TBATCAPLL (FFFFF504H)		7	6	5	4	3	2	1	0
	bit Symbol	CAP07	CAP06	CAP05	CAP04	CAP03	CAP02	CAP01	CAP00
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture data (bit 7~0)							

TBTACAPLH (FFFFF505H)		7	6	5	4	3	2	1	0
	bit Symbol	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10	CAP09	CAP08
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture data (bit 15~8)							

TBTACAPHL (FFFFF506H)		7	6	5	4	3	2	1	0
	bit Symbol	CAP23	CAP22	CAP21	CAP20	CAP19	CAP18	CAP17	CAP16
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture data (bit 23~16)							

TBTACAPHH (FFFFF507H)		7	6	5	4	3	2	1	0
	bit Symbol	CAP31	CAP30	CAP29	CAP28	CAP27	CAP26	CAP25	CAP24
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture data (bit 31~24)							

Fig. 12.3.2TMRC-related registers

TBT read capture register（TBTARDCAP）

TBTARDCAPLL(0xFFFF_F508)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP07	RDCAP06	RDCAP05	RDCAP04	RDCAP03	RDCAP02	RDCAP01	RDCAP00
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bit 7~0)							

TBTARDCAPLH(0xFFFF_F509)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP17	RDCAP16	RDCAP15	RDCAP14	RDCAP13	RDCAP12	RDCAP11	RDCAP10
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bit 15~8)							

TBTARDCAPHL(0xFFFF_F50A)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP27	RDCAP26	RDCAP25	RDCAP24	RDCAP23	RDCAP22	RDCAP21	RDCAP20
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bit 23~16)							

TBTARDCAPHH(0xFFFF_F50B)

	7	6	5	4	3	2	1	0
bit Symbol	RDCAP37	RDCAP36	RDCAP35	RDCAP34	RDCAP33	RDCAP32	RDCAP31	RDCAP30
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture data (bit 31~24)							

Fig. 12.3.3 TMRC-related registers

TMRC capture 0 control register

		7	6	5	4	3	2	1	0
CAPA0CR (FFFFF520H)	bit Symbol	TC0NF						CP0EG1	CP0EG0
	Read/Write	R/W	R					R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	TC0IN Input noise removal 0:disable 1:enable						Select effective edge of TC0IN input 00：Not captured 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 (CAAP0). If this is set to "00," the capture operation is disabled.

<TC0NF>: Controls the noise removal for the TC0IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph=fc=54MHz) is accepted as a trigger input for TCCAP0, at whichever level the TC0IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph=fc=54MHz) is regarded as noise and removed, at whichever level the TC0IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) CAPA0CR bits 2~6 are read as “0”.

TMRC capture 0 register（CAPA0）

		7	6	5	4	3	2	1	0
CAPA0LL (FFFFF524H)	bit Symbol	CAP007	CAP006	CAP005	CAP004	CAP003	CAP002	CAP001	CAP000
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture 0 data (bit 7~0)							

		7	6	5	4	3	2	1	0
CAPA0LH (FFFFF525H)	bit Symbol	CAP017	CAP016	CAP015	CAP014	CAP013	CAP012	CAP011	CAP010
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture 0 data (bit 15~8)							

		7	6	5	4	3	2	1	0
CAPA0HL (FFFFF5216H)	bit Symbol	CAP027	CAP026	CAP025	CAP024	CAP023	CAP022	CAP021	CAP020
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture 0 data (bit 23~16)							

		7	6	5	4	3	2	1	0
CAPA0HH (FFFFF527H)	bit Symbol	CAP037	CAP036	CAP035	CAP034	CAP033	CAP032	CAP031	CAP030
	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	Capture 0 data (bit 31~24)							

(Note)Data is not captured during a read of the capture register.

Fig. 12.3.4 TMRC-related register

TMRC capture 1 control reigster

CAPA1CR (FFFFF528H)

	7	6	5	4	3	2	1	0
bit Symbol	TC1NF						CP1EG1	CP1EG0
Read/Write	R/W	R					R/W	
After reset	0	0	0	0	0	0	0	0
Function	TC1IN Input noise removal 0:disable 1:enable						Select effective edge of TC1IN input 00：Not captured 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 TC1IN pin input.
If this is set to "0" (removal disabled), any input of more than 2/fsys (37ns@fperiph＝fc＝54MHz) is accepted as a trigger input for TCCAP1, at whichever level TC1IN pin is, "H" or "L."
If this is set to "1" (removal enabled), any input of less than 6/fsys (111ns@fperiph＝fc＝54MHz) is regarded as noise and removed, at whichever level the TC1IN pin is, "H" or "L." The range of removal changes depending on the selected clock gear and a system clock used.

(Note) CAPA1CR bits 2~6 area read as “0”.

TMRC capture 1 register（CAPA1）

CAPA1LL (FFFFF52CH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP107	CAP106	CAP105	CAP104	CAP103	CAP102	CAP101	CAP100
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 1 data (bit 7~0)							

CAPA1LH (FFFFF52DH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP117	CAP116	CAP115	CAP114	CAP113	CAP112	CAP111	CAP110
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 1 data (bit 15~8)							

CAPA1HL (FFFFF52EH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP127	CAP126	CAP125	CAP124	CAP123	CAP122	CAP121	CAP120
Read/Write	R							
After reset								
Function	Capture 1 data (bit 23~16)							

CAPA1HH (FFFFF52FH)

	7	6	5	4	3	2	1	0
bit Symbol	CAP137	CAP136	CAP135	CAP134	CAP133	CAP132	CAP131	CAP130
Read/Write	R							
After reset	0	0	0	0	0	0	0	0
Function	Capture 1 data (bit 31~24)							

(Note) Data is not captured during a read of the capture register.

Fig. 12.3.5 TMRC-related register

TMRC capture interrupt determination, interrupt mask register

CAPINT
(0xFFFF_E718)

	7	6	5	4	3	2	1	0
bit Symbol			INTCAPB1	INTCAPB0			INTCAPA1	INTCAPA0
Read/Write	R							
After reset			0	0			0	0
Function			Interrupt 0:disable 1:enable	Interrupt 0:disable 1:enable			Interrupt 0:disable 1:enable	Interrupt 0:disable 1:enable

	15	14	13	12	11	10	9	8
bit Symbol			IMINTCAPB1	IMINTCAPB0			IMINTCAPA1	IMINTCAPA0
Read/Write	R/W							
After reset			0	0			0	0
Function			Mask 0:disable 1:enable	Mask 0:disable 1:enable			Mask 0:disable 1:enable	Mask 0:disable 1:enable

TMRC capture interrupt determination, interrupt mask register

CMPINT
(0xFFFF_E71C
)

	7	6	5	4	3	2	1	0
bit Symbol	INTCMPB1	INTCMPB0			INTCMPA1	INTCMPA0		
Read/Write	R	R			R	R		
After reset	0	0			0	0		
Function	Interrupt 0:disable 1:enable	Interrupt 0:disable 1:enable			Interrupt 0:disable 1:enable	Interrupt 0:disable 1:enable		

	15	14	13	12	11	10	9	8
bit Symbol	IMINTCMPB1	IMINTCMPB0			IMINTCMPA1	IMINTCMPA0		
Read/Write	R/W	R/W			R/W	R/W		
After reset	0	0			0	0		
Function	Mask 0:disable 1:enable	Mask 0:disable 1:enable			Mask 0:disable 1:enable	Mask 0:disable 1:enable		

Fig. 12.3.6 TMRC-related register

TMRC compare control register（CMPACTLn）

CMPA0CTL (FFFFF510H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN0	TCFFC01	TCFFC00			CMPRDE0	CMPEN0
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF0 reverse 0:disable 1:enable	TCFF0 control 00:reverse 01:set 10:clear 11:Don't care				Double buffer 0 0:disable 1:enable	Compare 0 enable 0:disable 1:enable

CMPA1CTL (FFFFF518H)

	7	6	5	4	3	2	1	0
bit Symbol		TCFFEN1	TCFFC11	TCFFC10			CMPRDE1	CMPEN1
Read/Write	R	R/W			R		R/W	
After reset	0	0	1	1	0	0	0	0
Function		TCFF0 reverse 0:disable 1:enable	TCFF0 control 00:reverse 01:set 10:clear 11:Don't care				Double buffer 1 0:disable 1:enable	Compare 1 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.

(Note) CMPACTLn bits 7 and 3~2 are read as "0".

Fig. 12.3.7 TMRC-related register

TMRC compare register 0 (CMPA0)

CMPA0LL (FFFFF514H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP007	CMP006	CMP005	CMP004	CMP003	CMP002	CMP001	CMP000
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bit 7~0)							

CMPA0LH (FFFFF515H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP017	CMP016	CMP015	CMP014	CMP013	CMP012	CMP011	CMP010
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bit 15~8)							

CMPA0HL (FFFFF516H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP027	CMP026	CMP025	CMP024	CMP023	CMP022	CMP021	CMP020
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bit 23~16)							

CMPA0HH (FFFFF517H)

	7	6	5	4	3	2	1	0
bit Symbol	CMP037	CMP036	CMP035	CMP034	CMP033	CMP032	CMP031	CMP030
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 0 data (bit 31~24)							

TMRC compare register 1 (CMPA1)

CMPA1LL (FFFFF51CH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP107	CMP106	CMP105	CMP104	CMP103	CMP102	CMP101	CMP100
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bit 7~0)							

CMPA1LH (FFFFF51DH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP117	CMP116	CMP115	CMP114	CMP113	CMP112	CMP111	CMP110
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bit 15~8)							

CMPA1HL (FFFFF51EH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP127	CMP126	CMP125	CMP124	CMP123	CMP122	CMP121	CMP120
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bit 23~16)							

CMPA1HH (FFFFF51FH)

	7	6	5	4	3	2	1	0
bit Symbol	CMP137	CMP136	CMP135	CMP134	CMP133	CMP132	CMP131	CMP130
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Compare register 1 data (bit 31~24)							

Fig. 12.3.8 TMRC-related register

13 Serial Channel (SIO)

13.1 Features

This device has eleven serial I/O channels: SIO0 to SIOA. 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 transmit and receive I/O 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. 13.2.1 shows the block diagram of SIO0.

Each channel consists of a prescaler, a serial clock generation circuit, a receive buffer, its control circuit, a transmit buffer and its control circuit. Each channel functions independently.

As the SIOs 0 to SIOA operate in the same way, only SIO0 is described here.

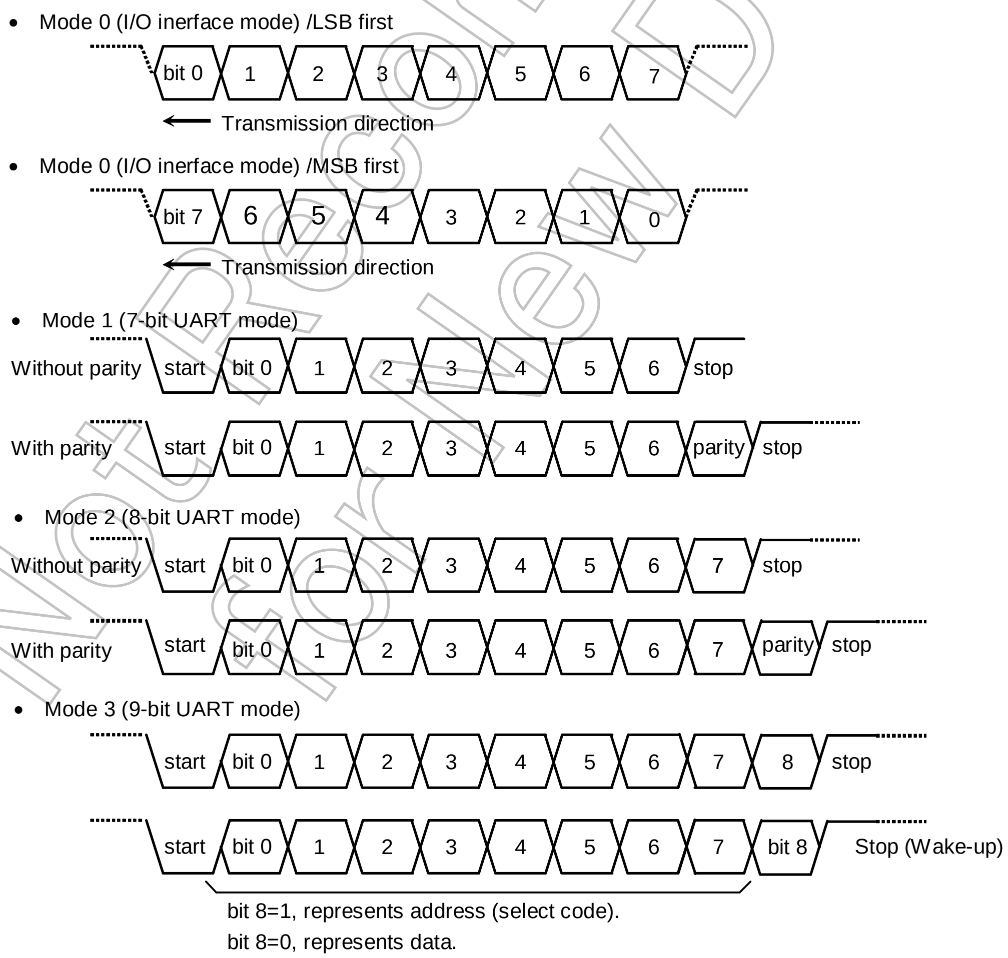


Fig. 13.1 Data format

13.2 Block Diagram (Channel 0)

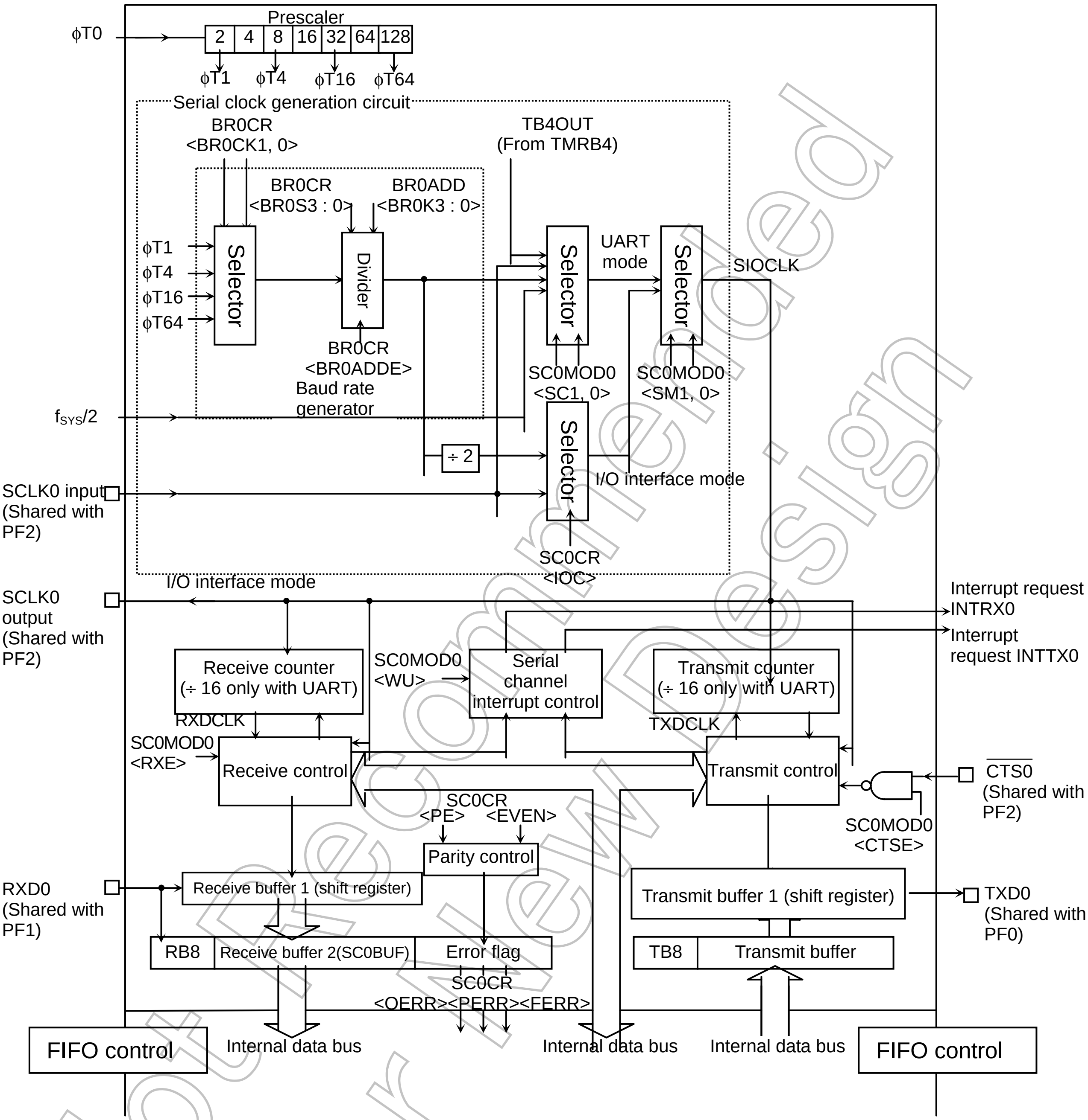


Fig. 13.2.1 SIO0 Block Diagram

13.3 Operation of Each Circuit (Channel 0)

13.3.1 Prescaler

The device includes a 7-bit prescaler to generate necessary clocks to drive SIO0. The input clock φT0 to the prescaler is selected by SYSCR of CG <PRCK1:0> to provide the frequency of fperiph/2, fperiph/4, fperiph/8, or fperiph/16.

The clock frequency fperiph is either the clock “fgear,” to be selected by SYSCR1<FPSEL> of CG, or the clock “fc” before it is divided by the clock gear.

The prescaler becomes active only when the baud rate generator is selected for generating the serial transfer clock. Table 13.3.1 lists the prescaler output clock resolution.

Table 13.3.1 Input Clock Resolution to the Baud Rate Generator @fc = 54MHz

Clear peripheral clock <FPSEL>	Clock gear value <GEAR2:0>	Prescaler clock selection <PRCK1 : 0>	Prescaler output clock resolution			
			φT1	φT4	φT16	φT64
0 (fgear)	000(fc)	00(fperiph/16)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	fc/2^{3}(0.15μs)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)
		11(fperiph/2)	fc/2^{2}(0.07μs)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)
	100(fc/2)	00(fperiph/16)	fc/2^{6}(1.2μs)	Fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)
		01(fperiph/8)	fc/2^{5}(0.6μs)	Fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		10(fperiph/4)	fc/2^{4}(0.3μs)	Fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)
		11(fperiph/2)	fc/2^{3}(0.15μs)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)
	110(fc/4)	00(fperiph/16)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)	fc/2^{13}(152μs)
		01(fperiph/8)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)
		10(fperiph/4)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		11(fperiph/2)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7 μs)	fc/2^{10}(19.0μs)
	111(fc/8)	00(fperiph/16)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)	fc/2^{14}(303μs)
		01(fperiph/8)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)	fc/2^{13}(152μs)
		10(fperiph/4)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)	fc/2^{12}(75.9μs)
		11(fperiph/2)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
1 (fc)	000(fc)	00(fperiph/16)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	fc/2^{3}(0.15μs)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)
		11(fperiph/2)	fc/2^{2}(0.07μs)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)
	100(fc/2)	00(fperiph/16)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	fc/2^{3}(0.15μs)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)
		11(fperiph/2)	-	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)
	110(fc/4)	00(fperiph/16)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	-	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)
		11(fperiph/2)	-	fc/2^{4}(0.3μs)	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)
	111(fc/8)	00(fperiph/16)	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)	fc/2^{11}(37.9μs)
		01(fperiph/8)	-	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)	fc/2^{10}(19.0μs)
		10(fperiph/4)	-	fc/2^{5}(0.6μs)	fc/2^{7}(2.4μs)	fc/2^{9}(9.5μs)
		11(fperiph/2)	-	-	fc/2^{6}(1.2μs)	fc/2^{8}(4.7μs)

(Note 1) The prescaler output clock φTn must be selected so that the relationship “φTn < fsys/2” is satisfied (so that φTn is slower than fsys/2).
(Note 2) Do not change the clock gear while SIO is operating.
(Note 3) The horizontal lines in the above table indicate that the setting is prohibited.

The serial interface baud rate generator uses four different clocks, i.e., φT1, φT4, φT16 and φT64, supplied from the prescaler output clock.

13.3.2 Baud Rate Generator

The baud rate generator generates transmit and receive clocks to determine the serial channel transfer rate.

The baud rate generator uses either the ϕT1, ϕT4, ϕT16 or ϕT64 clock supplied from the 7-bit prescaler. This input clock selection is made by setting the baud rate generator control register, BR0CR <BR0CK1:0>.

The baud rate generator contains built-in dividers for divide by 1, N + m/16 (N=2~15, m=0~15), and 16. The division is performed according to the settings of the baud rate generator 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

   (K=1、2、3 … 15) 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

$$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} \div 16$$

The highest baud rate out of the baud rate generator is 843.75 kbps when ϕT1 is 13.5 MHz

The fsys/2 frequency, which is independent of the baud rate generator, can be used as the serial clock. In this case, the highest baud rate will be 1.68 Mbps when fsys is 54 MHz.

2) I/O interface mode

$$\text{Baud rate} = \frac{\text{Baud rated generator input clock}}{\text{Frequency divided by the divide ratio}} \div 2$$

The highest baud rate will be generated when ϕT1 is 13.5 MHz. The divide ratio can be set to 1 if double buffer is used and the resulting output baud rate will be 6.75 Mbps. (If double buffering is not used, the highest baud rate will be 3.375 Mbps applying the divide ratio of “2”).

- Example baud rate setting

1) Division by an integer (divide by N):

Selecting fc = 54MHz 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:

* Clock condition
- System clock :High-speed (fc)
- High-speed clock gear:1x (fc)
- Prescaler clock :fperiph/16 (fperiph = fsys)

$$\text{Baud rate} = \frac{fc/32}{4} \div 16$$

$$= 54 \times 10^{6} \div 32 \div 4 \div 16 = 26367 \text{ (bps)}$$

(Note) The divide by (N + (16-K)/16) function is inhibited and thus BR0ADD <BR0K3:0> is ignored.

2) For divide by N + (16-K)/16 (only for UART mode):

Selecting fc = 54MHz for fperiph, setting ϕT0 to fperiph/16, using the baud rate generator input clock ϕT1, setting the divide ratio N (BR0CR<BR0S3:0>)=4, setting K (BR0ADD<BR0K3:0>)=14, and selecting BR0CR<BR0ADDE>=1, the resulting baud rate is calculated as follows:

* Clock condition
- System clock :High-speed (fc)
- High-speed clock gear:1x (fc)
- Prescaler clock :fperiph/16 (fperiph = fsys)

$$\text{Baud rate} = \frac{fc/32}{4 + \frac{(16-14)}{16}} \div 16$$

$$= 54 \times 10^{6} \div 32 \div \left(4 + \frac{2}{16}\right) \div 16 = 25568 \text{ (bps)}$$

Also, an external clock input may be used as the serial clock. The resulting baud rate calculation is shown below:

- Baud rate calculation for an external clock input:

1) UART mode

Baud rate = external clock input ÷ 16

In this, the period of the external clock input must be equal to or greater than 4/fsys.

If fsys = 54 MHz, the highest baud rate will be 54÷4÷16=844 (kbps).

2) I/O interface mode

Baud rate = external clock input

When double buffering is used, it is necessary to satisfy the following relationship:

(External clock input period) > 12/fsys

Therefore, when fsys = 54 MHz, the highest baud rate must be set to a rate lower than

54÷12 = 4.5 (Mbps)

When double buffering is not used, it is necessary to satisfy the following relationship:

(External clock input period) > 16/fsys

Therefore, when fsys = 54 MHz, the highest baud rate must be set to a rate lower than 54÷16 = 3.375 (Mbps).

The baud rate examples for the UART mode are shown in Table 13.3.2.1 and Table 13.3.2.2.

Table 13.3.2.1 Selection of UART Baud Rate

(Using 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 13.3.2.2 Selection of UART Baud Rate

(The TMRB7 timer output (internal TB7OUT) is used with the timer input clock set to φT0.)

Unit: (kbps)

fc / TB7RG0H/L	29.4912 MHz	24.576 MHz	24 MHz	19.6608 MHz	16 MHz	12.288 MHz
0001H	230.4	192	187.5	153.6	125	96
0002H	115.2	96	93.75	76.8	62.5	48
0003H	76.8	64	62.5	51.2	41.67	32
0004H	57.6	48	46.88	38.4	31.25	24
0005H	46.08	38.4	37.5	30.72	25	19.2
0006H	38.4	32	31.25	25.6	20.83	16
0008H	28.8	24	23.44	19.2	15.63	12
000AH	23.04	19.2	18.75	15.36	12.5	9.6
0010H	14.4	12	11.72	9.6	7.81	6
0014H	11.52	9.6	9.38	7.68	6.25	4.8

Baud rate calculation to use the TMRB7 timer:

$$\text{Transfer rate} = \frac{\text{Clock frequency selected by SYSCR0<PRCK1:0>}}{\text{TB7REG}\times\underline{2}\times 16}$$

(When input clock to the timer TMRB7 is φT0)

(Note 1) In the I/O interface mode, the TMRB7 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 TMRB4 timer, or the external clock (SCLKO pin) is selected to generate the basic clock, SIOCLK.

13.3.4 Receive Counter

The receive counter is a 4-bit binary counter used in the asynchronous (UART) mode and is up-counted by SIOCLK. Sixteen SIOCLK clock pulses are used in receiving a single data bit while the data symbol is sampled at the seventh, eighth, and ninth pulses. From these three samples, majority logic is applied to decide the received data.

13.3.5 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 SC0MOD1<FDPX1:0>=01), the INTRX0 interrupt is generated at the same time. If the receive FIFO has been enabled (SCNFCNF<CNFG>=1 and SC0MOD1<FDPX1:0>=01/11), an interrupt will be generated according to the SCORFC<RIL2:0> setting.

The CPU will read the data from either the second receive buffer (SC0BUF) or from the receive FIFO (the address is the same as that of the receive buffer). If the receive FIFO has not been enabled, the receive buffer full flag SC0MOD2<RBFLL> is cleared to “0” by the read operation. The next data received can be stored in the first receive buffer even if the CPU has not read the previous data from the second receive buffer (SC0BUF) or the receive FIFO.

If SCLK is set to generate clock output in the I/O interface mode, the double buffer control bit SC0MOD2 <WBUF> can be programmed to enable or disable the operation of the second receive buffer (SCOBUF).

By disabling the second receive buffer (i.e., the double buffer function) and also disabling the receive FIFO (SCOFCNF＜CNFG＞=0 and <FDPX1:0>=01), handshaking with the other side of communication can be enabled and the SCLK output stops each time one frame of data is transferred. In this setting, the CPU reads data from the first receive buffer. By the read operation of CPU, the SCLK output resumes.

If the second receive buffer (i.e., double buffering) is enabled but the receive FIFO is not enabled, the SCLK output is stopped when the first receive data is moved from the first receive buffer to the second receive buffer and the next data is stored in the first buffer filling both buffers with valid data. When the second receive buffer is read, the data of the first receive buffer is moved to the second receive buffer and the SCLK output is resumed upon generation of the received 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:

SCORFC<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).

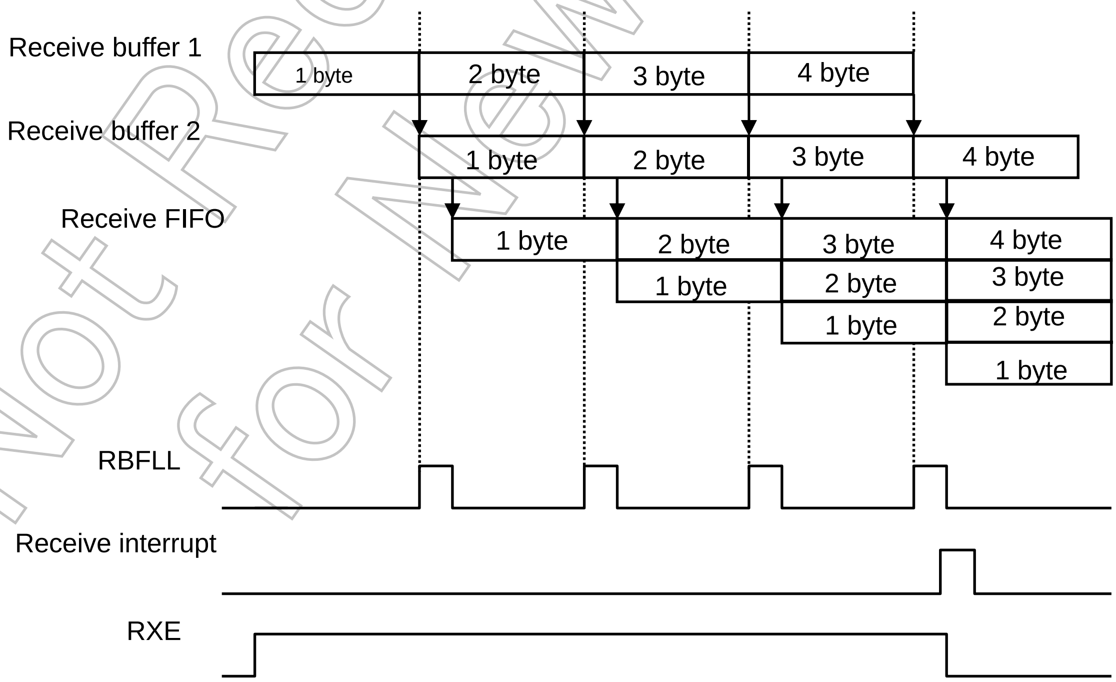


Fig. 13.3.8.1 Receive FIFO Operation

② I/O interface mode with SCLK input:

The following example describes the case a 10-byte data stream is received:

SC0RFC <7:6> = 10: Clears receive FIFO and sets the condition of interrupt generation

SC0RFC <1:0> = 00: Sets the interrupt to be generated at fill level 4.

SC0FCNF <1:0> = 10101: Automatically allows continued reception after reaching the fill level.

The number of bytes to be used in the receive FIFO is the maximum allowable number.

In this condition, 4-byte data reception may be initiated by setting the half duplex transmission mode and writing "1" to the RXE bit. After receiving 4 bytes, receive FIFO interrupt is generated. This setting enables the next data reception as well. The next 4 bytes can be received before all the data is read from FIFO.

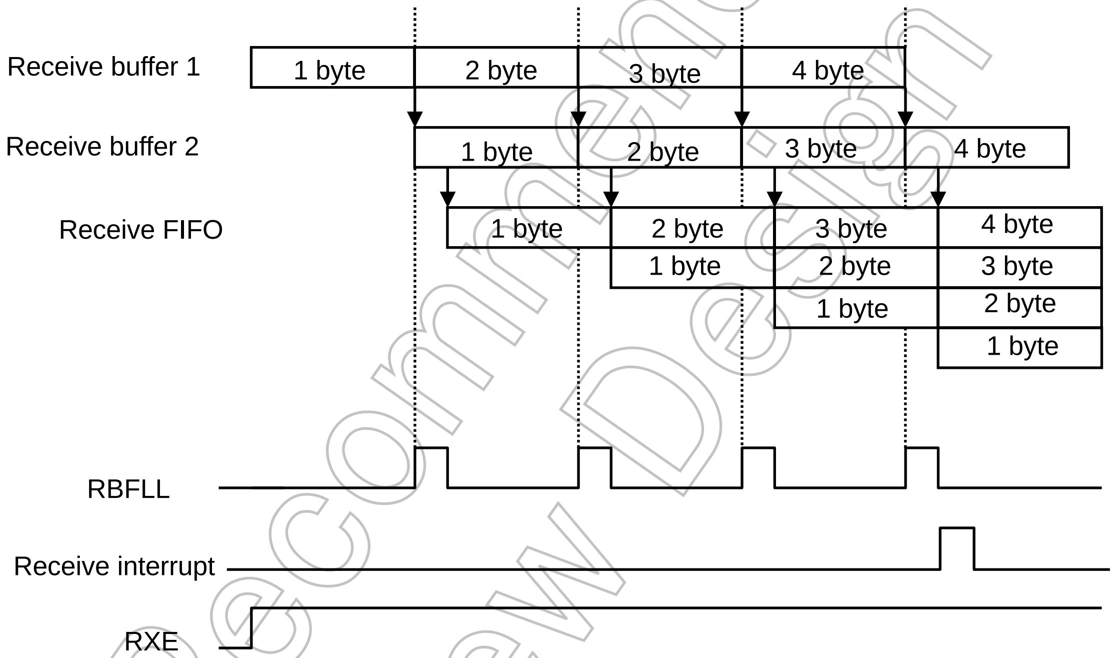


Fig. 13.3.8.2 Receive FIFO Operation

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 received counter and generates a transmit clock (TXDCLK) on every 16th clock pulse.

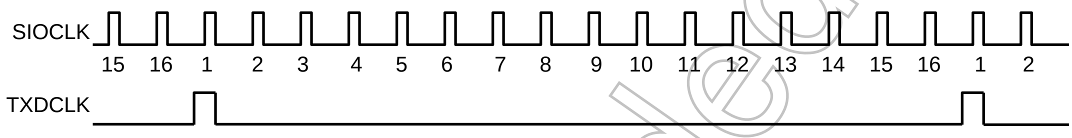


Fig. 13.3.9.1 Transmit Clock Generation

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 transmit 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 transmit 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 transmit buffer, data transmission is initiated on the rising edge of the next TXDCLK and the transmit shift clock (TXDSFT) is also generated.

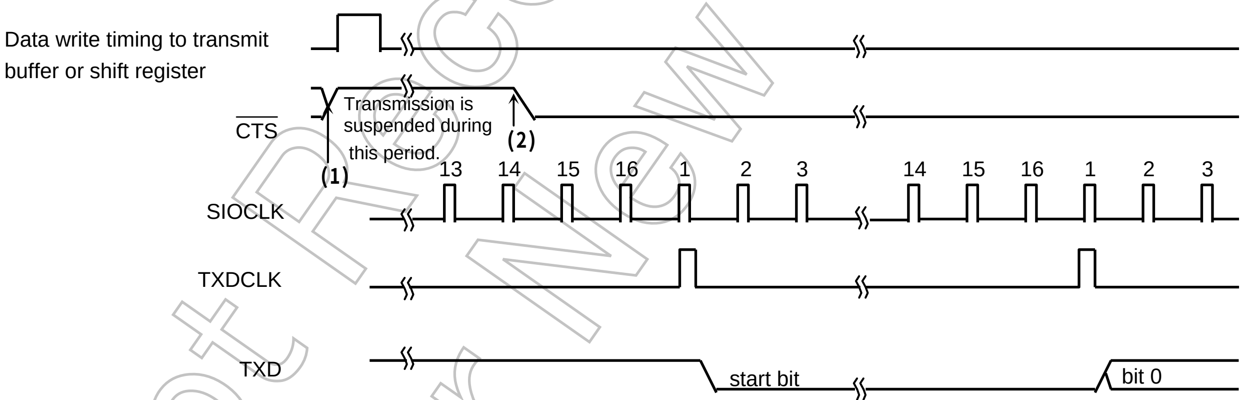
- <u>Handshake function</u>

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 $\overline{\text{CTS}}$ pin is set to the "H" level, the current data transmission can be completed but the next data transmission is suspended until the $\overline{\text{CTS}}$ pin returns to the "L" level. However in this case, the INTTX0 interrupt is generated, the next transmit data is requested to the CPU, data is written to the transmit 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. 13.3.10.1 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. 13.3.10.2 $\overline{\text{CTS}}$ (Clear to Transmit) Signal Timing

13.3.11 Transmit Buffer

The transmit 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 Transmit Buffer 2 (SCOBUF) is moved to Transmit 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 transmit buffer empty flag <TBEMP> of SC0MOD2 is set to "1." This flag indicates that transmit buffer 2 is now empty and that the next transmit data can be written. When the next data is written to transmit 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 transmit buffer 2 and <TBEMP> flag is immediately cleared to "0." The CPU writes data to transmit 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 transmit buffer 2 before the next frame clock input, which occurs upon completion of data transmission from transmit 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 transmit buffer 1 is completed, the transmit buffer 2 data is moved to transmit buffer 1 and any data in transmit FIFO is moved to transmit buffer 2 at the same time.

If the transmit FIFO is disabled in the I/O interface SCLK output mode, when data in transmit buffer 2 is moved to transmit 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 transmit 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 transmit 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 transmit 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 transmit 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 transmit buffer, the transmit FIFO interrupt is generated. When transmission of the last data is completed, the clock is stopped and the transmission sequence is terminated.

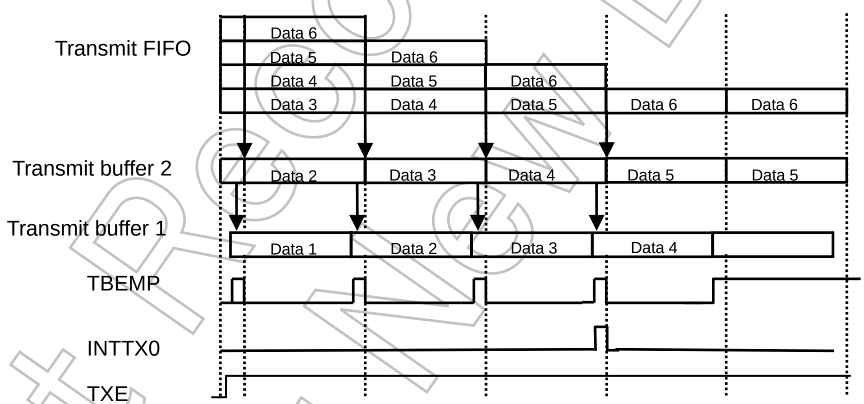


Fig. 13.3.13.1 Transmit FIFO Operation

② I/O interface mode with SCLK input (normal mode):

The following example describes the case a 4-byte data stream is transmitted:

SC0TFC <1:0> = 01: Clears the transmit FIFO and sets the condition of interrupt generation.

SC0TFC <7:2> = 000000: Sets the interrupt to be generated at fill level 0.

SC0FCNF <4:0> = 01001: Allows continued transmission after reaching the fill level.

In this condition, data transmission can be initiated along with the input clock by setting the transfer mode to half duplex, writing 4 bytes of data to the transmit FIFO, and setting the <TXE> bit to "1." When the last transmit data is moved to the transmit buffer, the transmit FIFO interrupt is generated.

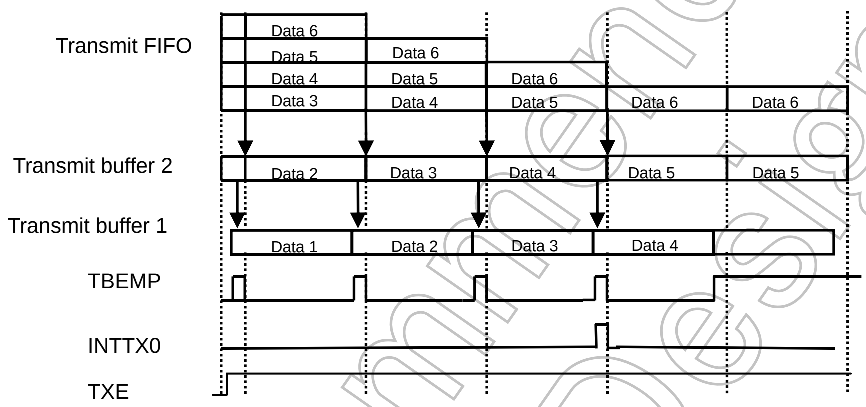


Fig. 13.3.13.2 Transmit FIFO Operation

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 transmit 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 transmit 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 cleared 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 transmit buffer 2 is disabled, the under-run flag <PERR> will not be set. This flag is cleared 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 transmit buffer 2 is empty. When data is moved from transmit buffer 2 to transmit buffer 1 (shift register), this bit is set to "1" indicating that transmit buffer 2 is now empty. When data is set to the transmit 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 Transmit/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 Signal Generation Timing

① UART mode

Receive Side

Mode	9-bit	8-bit + parity	8-bit, 7-bit + parity, 7-bit
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 + parity	8-bit, 7-bit + parity, 7-bit
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 transmit buffer 1 (just before start bit transmission)	Immediately after data is moved to transmit buffer 1 (just before start bit transmission)	Immediately after data is moved to transmit 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	SCLKinput 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 transmit 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 transmit buffer 1
Underrun 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 make any change in control register when data is being sent or received (in a state ready to transmit 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_F702)

	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	Transmit data bit 8	Handshake function control 0: Disables CTS 1: Enables CTS	Receive control 0: disable 1: enable	Wakeup function 0: disable 1: enable	Serial transfer mode 00: I/O interface mode 01: 7-bit length UART mode 10: 8-bit length UART mode 11: 9-bit length UART mode		Serial transfer clock (for UART) 00: Timer TB4OUT 01: Baud rate generator 10: Internal $f_{SYS}/2$ clock 11: External clock (SCLK0 input)	

Note) In the I/O interface mode, the serial control register (SC0CR) is used for clock

Wakeup function

	9-bit UART	Other modes
0	Interrupt when received	don't care
1	Interrupt at RB8 = 1	

Handshake function ($\overline{\text{CTS}}$ pin) enable

0	Disable (transmission is always allowed)
1	Enable

(Note) With <RXE> set to "0," set each mode register (SC0MOD0, SC0MOD1 and SC0MOD2). Then set <RXE> to "1."

Fig. 13.4.1 Serial Mode Control Register 0 (for SIO0, SC0MOD0)

SC0MOD1 (0xFFFF_F705)

	7	6	5	4	3	2	1	0
bit Symbol	I2S0	FDPX1	FDPX0	TXE	SINT2	SINT1	SINT0	
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	IDLE 0: Stop 1: Operation	Transfer mode setting 00: Transfer prohibited 01: Half duplex (RX) 10: Half duplex (TX) 11: Full duplex		Transmit control 0: Disable 1: Enable	Interval time of continuous transmission 000: None 100: 8SCLK 001: 1SCLK 101:16SCLK 010: 2SCLK 110: 32SCLK 011: 4SCLK 111: 64SCLK			Write "0."

Fig. 13.4.2 Serial Mode Control Register 1 (for SIO0, SC0MOD1)

<SINT2:0>: Specifies the interval time of continuous transmission when double buffering or FIFO is enabled in the I/O interface mode. This parameter is invalid for the UART mode or when an external clock is used.

<TXE>: This bit enables transmission and is valid for all the transfer modes. If disabled while transmission is in progress, transmission is inhibited only after the current frame of data is completed for transmission.

<FDPX1:0>: Configures the transfer mode in the I/O interface mode. Also configures the FIFO if it is enabled. In the UART mode, it is used only to specify the FIFO configuration.

<I2S0>: Specifies the Idle mode operation.

SC0MOD2
(0xFFFF_F706)

	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	Transmit 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 bits	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 transmit/receive buffers to transmit (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 transmit buffer contains some data waiting for the next transmission.

<RBFLL>: This is a flag to show whether the received double buffers are full or not. 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 transmit double buffers are empty. When data in the transmit double buffers is moved to the transmit shift register and the double buffers are empty, this bit is set to "1." Writing data again to the double buffers sets this bit to "0."
If double buffering is disabled, this flag is insignificant.

<SBLEN>: This specifies the length of stop bit transmission in the UART mode. On the receive side, the decision is made using only a single bit regardless of the <SBLEN> setting.

(Note)	**While data transmission is in progress, any software reset operation must be executed twice in succession.**

Fig. 13.4.3 Serial Mode Control Register

SC0CR
(0xFFFF_F701)

	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 Overrun	Parity/ under-run	Framing	0: SCLK0 1: SCLK0	0: Baud rate generato r 1: SCLK0 pin input

I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK0 input operation

0	Data transmit/receive at rising edges of SCLK0
1	Data transmit/receive at falling edges of SCLK0

Framing error flag
Parity error/ underrun error flag
Overrun error flag

Cleared to "0" when read.

Add/check even parity

0	Odd parity
1	Even parity

(Note) Any error flag is cleared when read.

Fig. 13.4.4 Serial Control Register (for SIO0, SC0CR)

BR0CR (0xFFFF_F703)

	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_F704)

	7	6	5	4	3	2	1	0
bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write	R				R/W			
After reset	0	0	0	0	0	0	0	0
Function					Specify K for the "N + (16 - K)/16" division			

Select input clock to the baud rate generator

	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0
BR0CR <BR0S3:0> / BR0ADD <BR0K3:0>	0000(N = 16) ≀ 0001(N = 1)	0010(N = 2) ≀ 1111(N = 15)	0001(N = 1) (ONLY UART) ≀ 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001(K = 1) ≀ 1111(K = 15)	Disable	$N + \frac{(16-K)}{16}$ division	Divide by N

(Note 1) In the UART mode, the division ratio "1" of the baud rate generator can be specified only when the "N + (16 - K)/16" division function is not used. In the I/O interface mode, the division ratio "1" of the baud rate generator can be specified only when double buffering is used.

(Note 2) To use the "N + (16 - K)/16" division function, be sure to set BR0CR <BR0ADDE> to "1" after setting the K value (K = 1 to 15) to BR0ADD <BR0K3:0>. However, don't use the "N + (16 - K)/16" division function when BR0CR <BR0S3:0> is set to either "0000" or "0001" (N = 16 or 1).

(Note 3) The "N + (16 - K)/16" division function can only be used in the UART mode. In the I/O interface mode, the "N + (16 - K)/16" division function must be disabled (prohibited) by setting BR0CR <BR0ADDE> to "0."

Fig. 13.4.5 Baud Rate Generator Control (for SIO0, BR0CR, BR0ADD)

Register		7	6	5	4	3	2	1	0
SC0BUF (0xFFFF_F700)	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: Transmit buffer + FIFO RB7 to RB0: Receive buffer + FIFO							

(Note) SC0BUF works as a transmit buffer for WR operation and as a receive buffer for RD operation.

Fig. 13.4.6 SIOO Transmit/ Receive Buffer Register

Register		7	6	5	4	3	2	1	0
SC0FCNF (0xFFFF_F70C)	bit Symbol				RFST	TFIE	RFIE	RXTXCNT	CNFG
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	**Be sure to write "000."**			Bytes used in RX FIFO 0: Maximum 1: Same as Fill level of RX FIFO	TX interrupt for TX FIFO 0: Disable 1: Enable	RX interrupt for RX FIFO 0: Disable 1: Enable	Automatic disable of RXE/TXE 0: None 1: Auto disable	FIFO Enable 0: Disable 1: Enable

<CNFG>: If enabled, the SCOMOD1 <FDPX1:0> setting automatically configures FIFO as follows:
<FDPX1:0>= 01 (Half duplex RX) ---- 4-byte RX FIFO
<FDPX1:0>=10 (Half duplex TX) ---- 4-byte TX FIFO
<FDPX1:0>=11 (Full duplex) ---- 2-Byte RX FIFO + 2-Byte TX FIFO
<RXTXCNT>:0 The function to automatically disable RXE/TXE bits is disabled.
: 1 If enabled, the SCOMOD1 <FDPX1:0> is used to set as follows:
<FDPX1:0>= 01 (Half duplex RX) ------When the RX FIFO is filled up with 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. 13.4.7 FIFO Configuration Register

SC0RFC
(0xFFFF_F708)

	7	6	5	4	3	2	1	0
bit Symbol	RFCS	RFIS					RIL1	RIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0	0	0	0	0	0
Function	Clear RX FIFO 1: Clear Always reads "0."	Select interrupt generation condition					FIFO fill level to generate RX interrupts 00: 4 bytes (2 bytes if full duplex) 01: 1byte 10: 2bytes 11: 3bytes Note: RIL1 is ignored when FDPX1:0 = 11 (full duplex)	

0: An interrupt is generated if FIFO is filled up with the data up to the specified level.
1: An interrupt is generated if FIFO is filled up with the data up to the specified level or more when it is read.

Fig. 13.4.8 Receive FIFO Control Register

SC0TFC
(0xFFFF_F709)

	7	6	5	4	3	2	1	0
bit Symbol	TFCS	TFIS					TIL1	TIL0
Read/Write	W	R/W	R				R/W	
After reset	0	0	0	0	0	0	0	0
Function	Clear TX FIFO 1: Clear Always reads "0."	Select interrupt generation condition					FIFO fill level to generate TX interrupts 00: Empty 01: 1byte 10: 2bytes 11: 3bytes Note: TIL1 is ignored when FDPX1:0 = 11 (full duplex).	

0: An interrupt is generated when FIFO is filled up with the data up to the specified level.
1: An interrupt is generated if FIFO is filled up with the data up to the specified level or more when it is read.

Fig. 13.4.9 Transmit FIFO Configuration Register

SC0RST (0xFFFF_F70A)

	7	6	5	4	3	2	1	0
bit Symbol	ROR					RLVL2	RLVL1	RLVL0
Read/Write	R	R				R		
After reset	0	0	0	0	0	0	0	0
Function	RX FIFO Overrun 1: Generated					Status of RX FIFO fill level 000: Empty 001: 1Byte 010: 2Bytes 011: 3Byte s 100: 4Bytes		

(Note) The <ROR> bit is cleared to "0" when receive data is read from the SC0BUF register.

Fig. 13.4.10 Receive FIFO Status Register

SC0TST (0xFFFF_F70B)

	7	6	5	4	3	2	1	0
bit Symbol	TUR					TLVL2	TLVL1	TLVL0
Read/Write	R	R				R		
After reset	1	0	0	0	0	0	0	0
Function	TX FIFO Under run 1: Generated					Status of TX FIFO fill level 000: Empty 001: 1Byte 010: 2Bytes 011: 3Byte s 100: 4Bytes		

(Note) The <TUR> bit is cleared to "0" when transmit data is written to the SC0BUF register.

Fig. 13.4.11 Transmit FIFO Status Register

SC0EN (0xFFFF_F707)

	7	6	5	4	3	2	1	0
bit Symbol								SIOE
Read/Write	R							R/W
After reset	0	0	0	0	0	0	0	0
Function								SIO operation 0: Disable 1: Enable

<SIOE>: It specifies SIO operation. When SIO operation is disabled, the clock will not be supplied to the SIO module except for the register part and thus power consumption can be reduced (other registers cannot be accessed for read/write operation). When SIO is to be used, be sure to enable SIO by setting "1" to this register before setting any other registers of the SIO module. If SIO is enabled once and then disabled, any register setting is maintained.

Fig. 13.4.12 SIO Enable Register

13.5 Operation in Each Mode

13.5.1 Mode 0 (I/O interface mode)

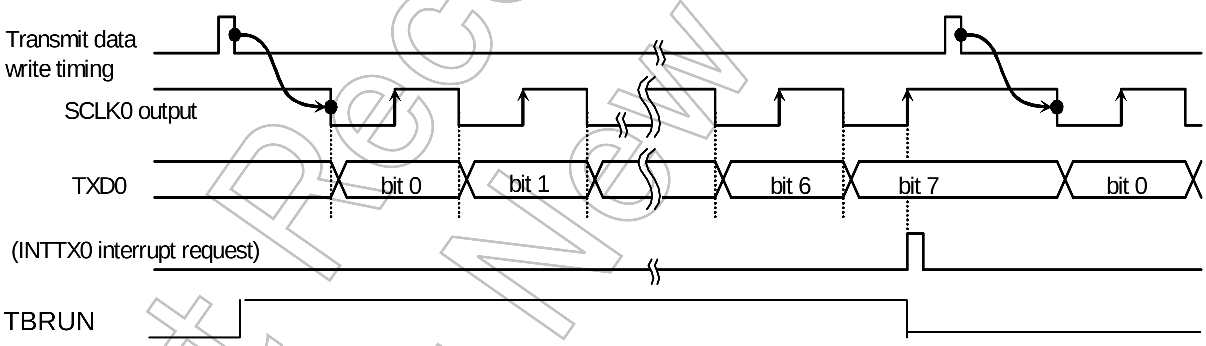
Mode 0 consists of two modes, i.e., the "SCLK output" mode to output synchronous clock and the "SCLK input" mode to accept synchronous clock from an external source. The following operational descriptions are for the case use of FIFO is disabled. For details of FIFO operation, refer to the previous sections describing receive/transmit FIFO functions.

① Sending data

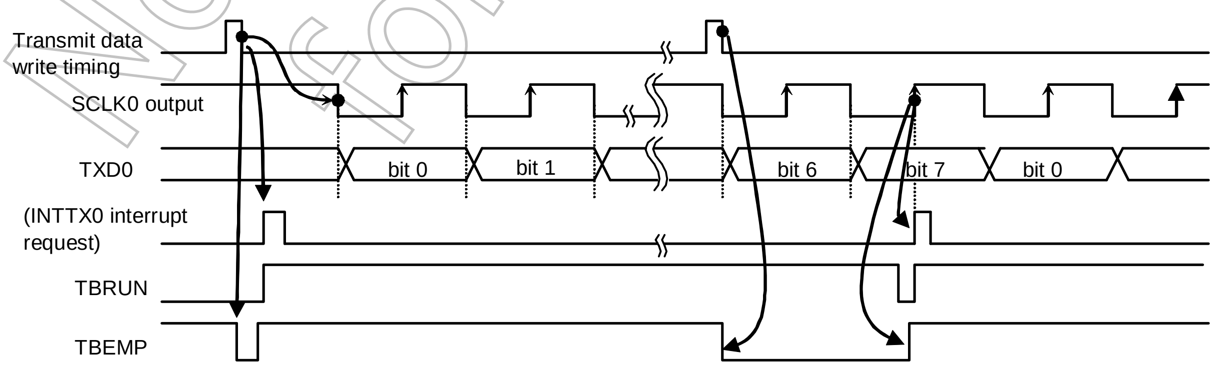
SCLK output mode

In the SCLK output mode, if SC0MOD2<WBUF> is set to "0" and the transmit 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 transmit buffer. When all data is output, the INTTX0 interrupt is generated.

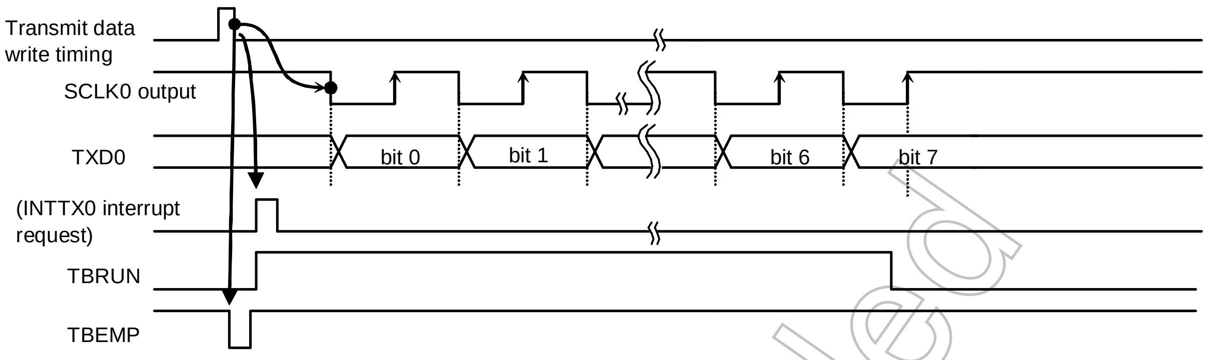
If SC0MOD2 <WBUF> is set to "1" and the transmit double buffers are enabled, data is moved from transmit buffer 2 to transmit buffer 1 when the CPU writes data to transmit buffer 2 while data transmission is halted or when data transmission from transmit buffer 1 (shift register) is completed. When data is moved from transmit buffer 2 to transmit buffer 1, the transmit buffer empty flag SC0MOD2 <TBEMP> is set to "1," and the INTTX0 interrupt is generated. If transmit buffer 2 has no data to be moved to transmit buffer 1, the INTTX0 interrupt is not generated and the SCLK0 output stops.



<WBUF>＝"0" (if double buffering is disabled)



<WBUF>＝"1" (if double buffering is enabled) (if there is data in buffer 2)



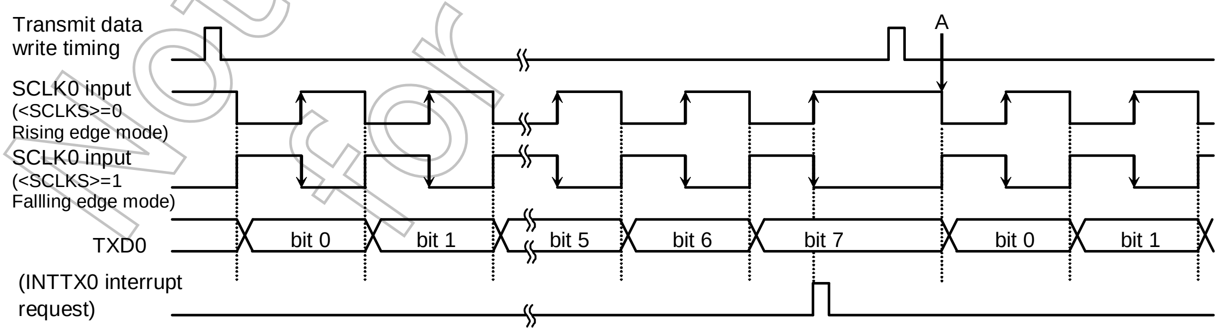
<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 13.5.1.1 Send Operation in the I/O Interface Mode (SCLK0 Output Mode)

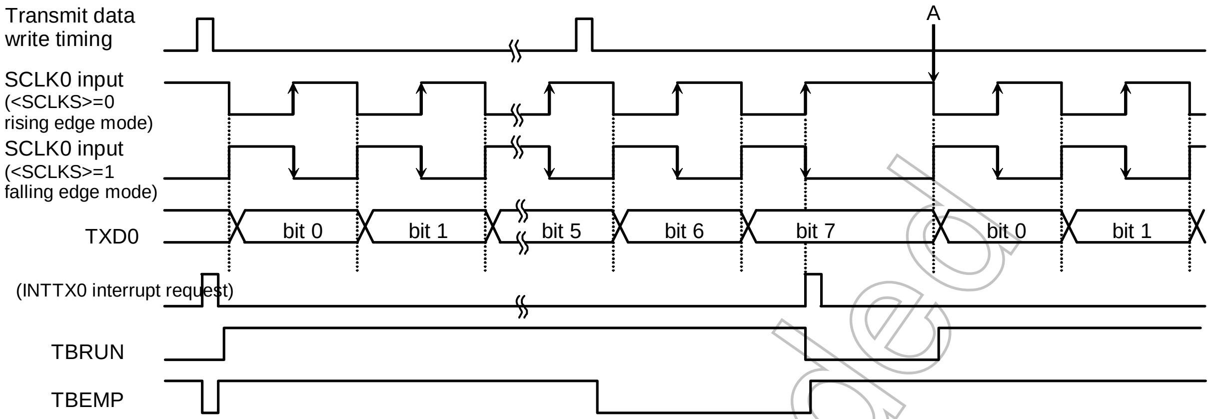
SCLK input mode

In the SCLK input mode, if SC0MOD2 <WBUF> is set to "0" and the transmit double buffers are disabled, 8-bit data that has been written in the transmit buffer is output from the TXD0 pin when the SCLK0 input becomes active. When all 8 bits are transmitted, the INTTX0 interrupt is generated. The next data to be transmitted must be written before the timing point "A" as shown in Fig. 13.5.1.2.

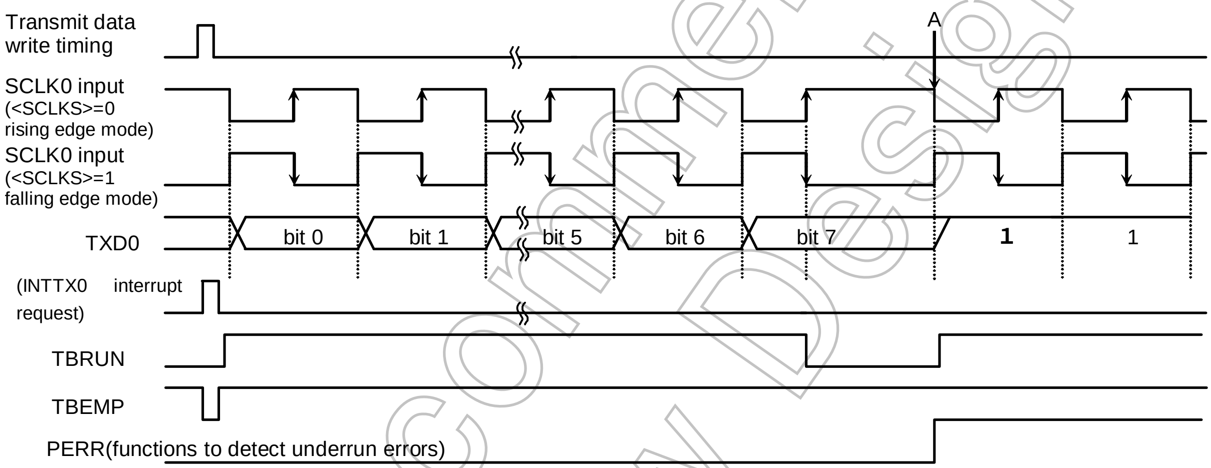
If SC0MOD2 <WBUF> is set to "1" and the transmit double buffers are enabled, data is moved from transmit buffer 2 to transmit buffer 1 when the CPU writes data to transmit buffer 2 before the SCLK0 becomes active or when data transmission from transmit buffer 1 (shift register) is completed. As data is moved from transmit buffer 2 to transmit buffer 1, the transmit 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 transmit buffer 2, although the internal bit counter is started, an under-run error occurs and 8-bit dummy data (FFh) is transmitted.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if there is data in buffer 2)



<WBUF> = "1" (if double buffering is enabled) (if there is no data in buffer 2)

Fig. 13.5.1.2 Send Operation in the I/O Interface Mode (SCLK0 Input Mode)

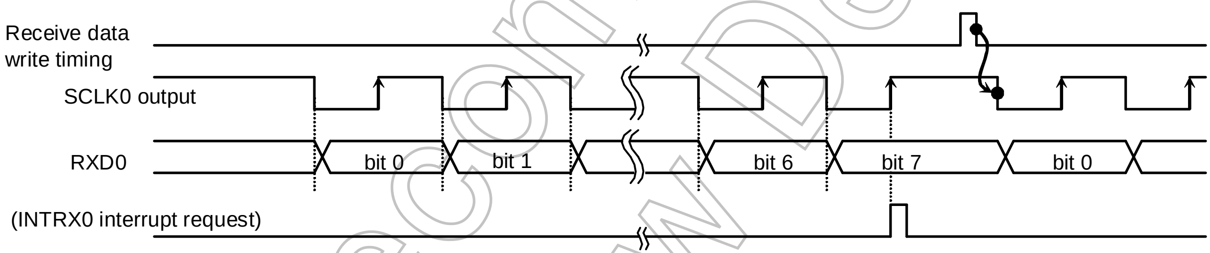
② Receiving data

SCLK output mode

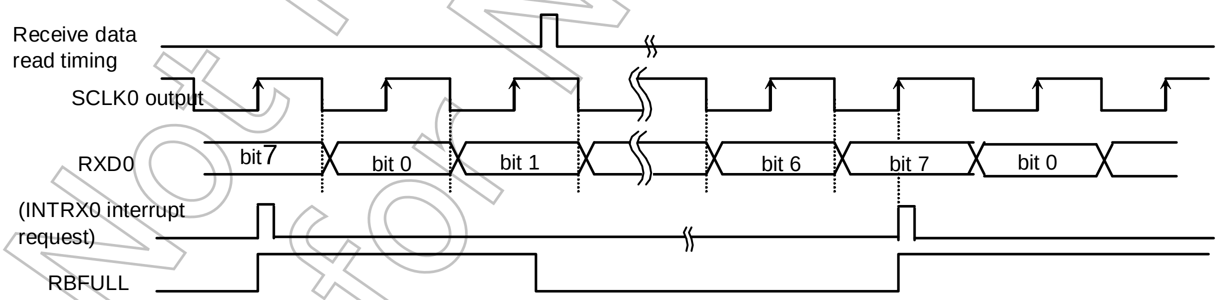
In the SCLK output mode, if SC0MOD2 <WBUF> = "0" and receive double buffering is disabled, a synchronous clock pulse is output from the SCLK0 pin and the next data is shifted into receive buffer 1 each time the CPU reads received data. When all the 8 bits are received, the INTRX0 interrupt is generated.

The first SCLK output can be started by setting the receive enable bit SC0MOD0 <RXE> to "1." If the receive double buffering is enabled with SC0MOD2 <WBUF> set to "1," the first frame received is moved to receive buffer 2 and receive buffer 1 can receive the next frame successively. As data is moved from receive buffer 1 to receive buffer 2, the receive buffer full flag SC0MOD2 <RBFULL> is set to "1" and the INTRX0 interrupt is generated.

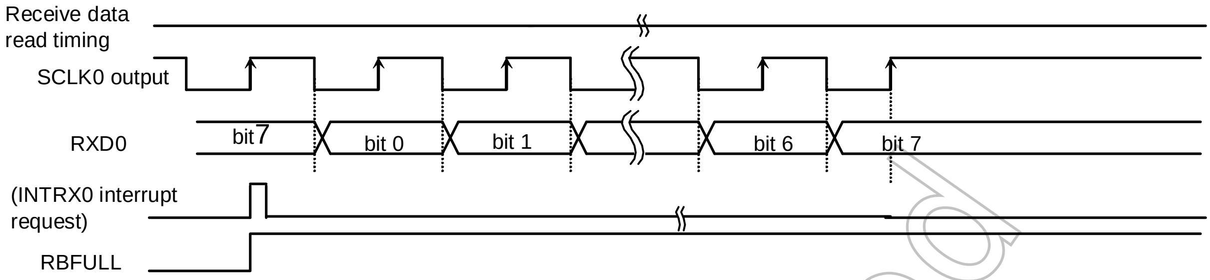
While data is in receive buffer 2, if CPU/DMAC cannot read data from receive buffer 2 in time before completing reception of the next 8 bits, the INTRX0 interrupt is not generated and the SCLK0 clock stops. In this state, reading data from receive buffer 2 allows data in receive buffer 1 to move to receive buffer 2 and thus the INTRX0 interrupt is generated and data reception resumes.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (if data is read from buffer 2)



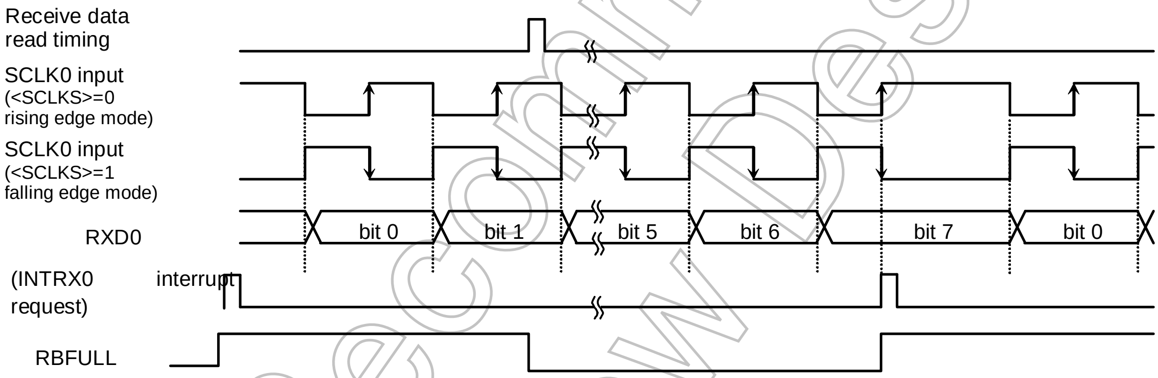
<WBUF> = "1" (if double buffering is enabled) (if data cannot be read from buffer 2)

Fig. 13.5.1.3 Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

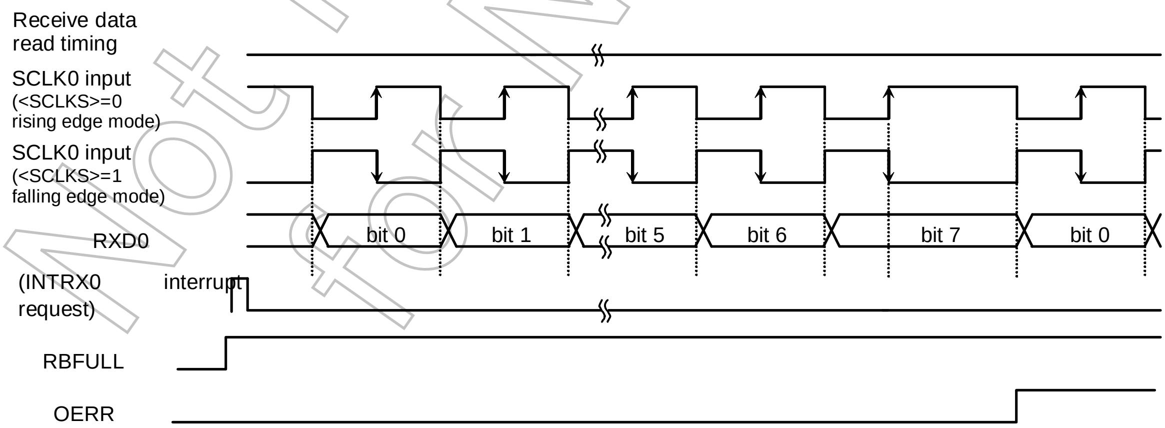
SCLK input mode

In the SCLK input mode, since receive double buffering is always enabled, the received frame can be moved to receive buffer 2 and receive buffer 1 can receive the next frame successively.

The INTRX receive interrupt is generated each time received data is moved to receive buffer 2.



If data is read from buffer 2



If data cannot be read from buffer 2

Fig. 13.5.1.4 Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

(Note) To receive data, SC0MOD <RXE> must always be set to "1" (receive enable) regardless of the SCLK input or output mode.

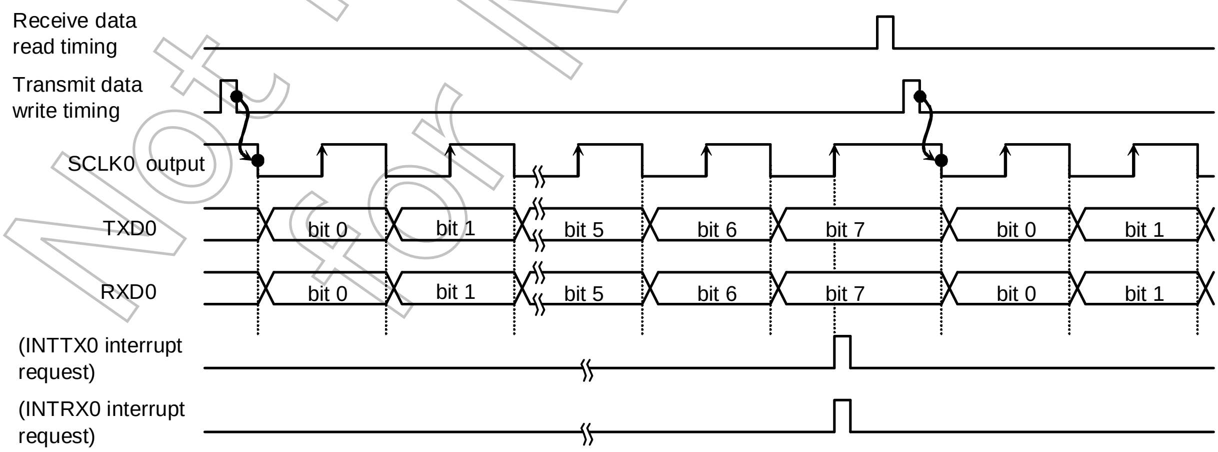
③ Send and receive (full-duplex)

The full-duplex mode is enabled by setting bit 6 <FDPX0> of the serial mode control register 1 (SC0MOD1) to "1."

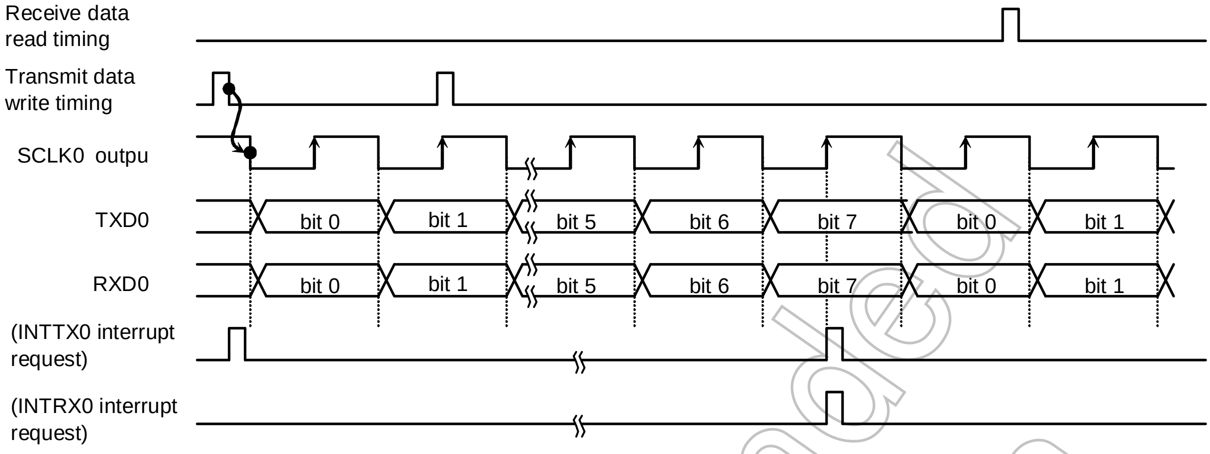
SCLK output mode

In the SCLK output mode, if SC0MOD2 <WBUF> is set to "0" and both the send and receive double buffers are disabled, SCLK is output when the CPU writes data to the transmit 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 transmit 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 transmit buffer by the CPU. The order of reading the receive buffer and writing to the transmit buffer can be freely determined. Data transmission is resumed only when both conditions are satisfied.

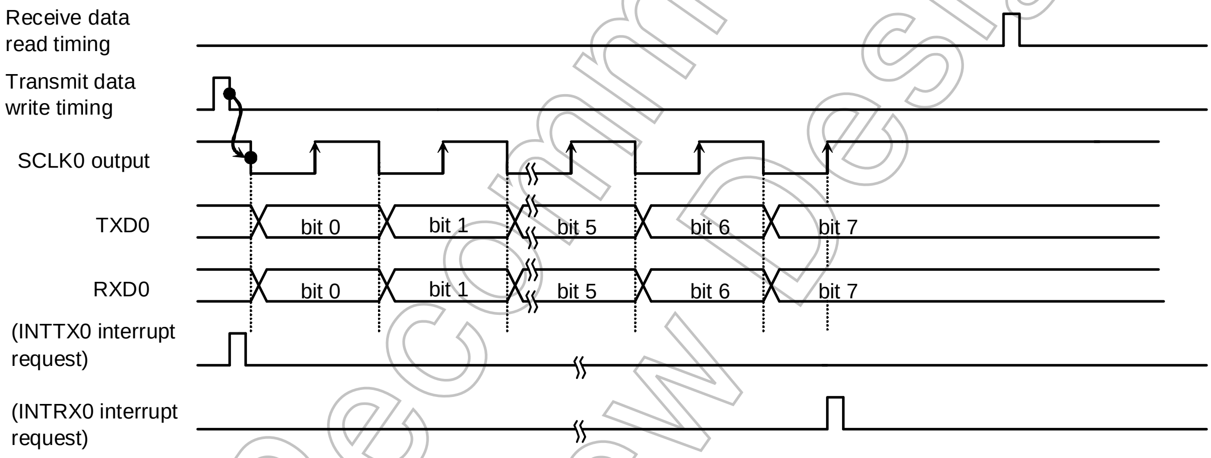
If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, SCLK is output when the CPU writes data to the transmit buffer. Subsequently, 8 bits of data are shifted into receive buffer 1, moved to receive buffer 2, and the INTRX0 interrupt is generated. While 8 bits of 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 transmit buffer 2 to transmit buffer 1. If transmit buffer 2 has no data to be moved to transmit buffer 1 (SC0MOD2 <TBEMP> = 1) or when receive buffer 2 is full (SC0MOD2 <RBFULL> = 1), the SCLK output is stopped. When both conditions are satisfied, i.e., receive data is read and send data is written, the SCLK output is resumed and the next round of data transmission is started.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled)



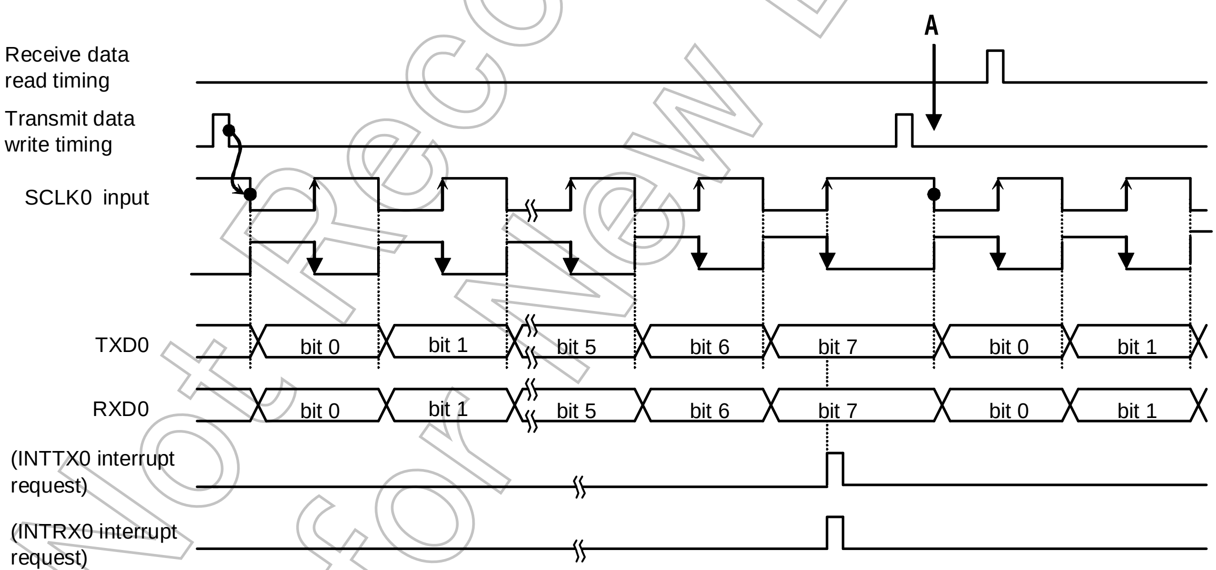
<WBUF> = "1" (if double buffering is enabled)

Fig. 13.5.1.5 Send/Receive Operation in the I/O Interface Mode (SCLK0 Output Mode)

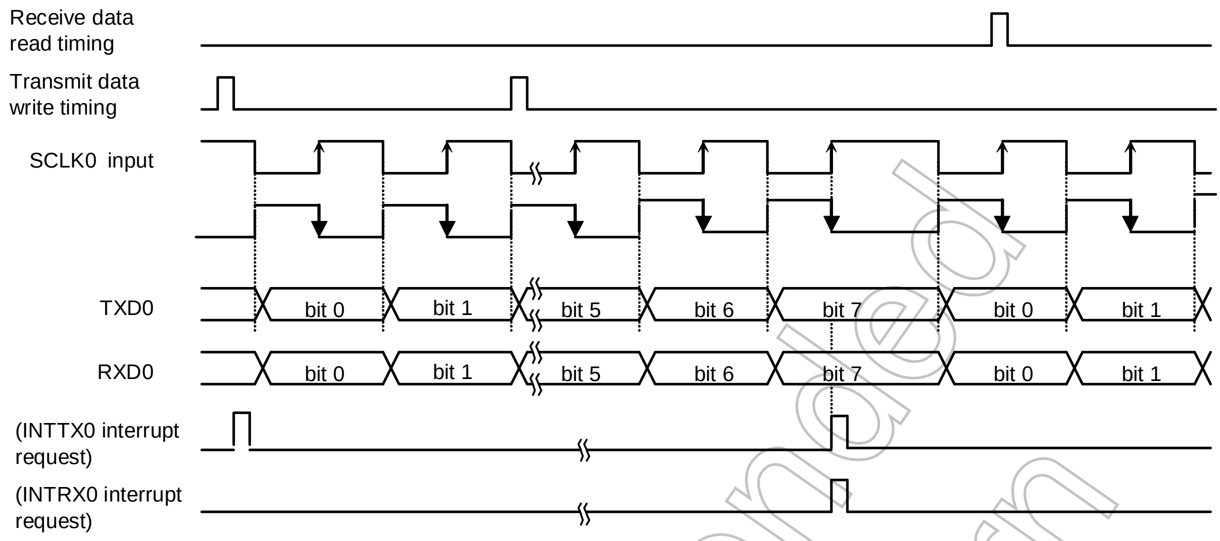
SCLK input mode

In the SCLK input mode with SC0MOD2 <WBUF> set to "0" and the send double buffers are disabled (double buffering is always enabled for the receive side), 8-bit data written in the transmit 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 transmit buffer before the SCLK input for the next frame (data must be written before the point A in Fig. 13.5.1.6). As double buffering is enabled for data reception, data must be read before completing reception of the next frame data.

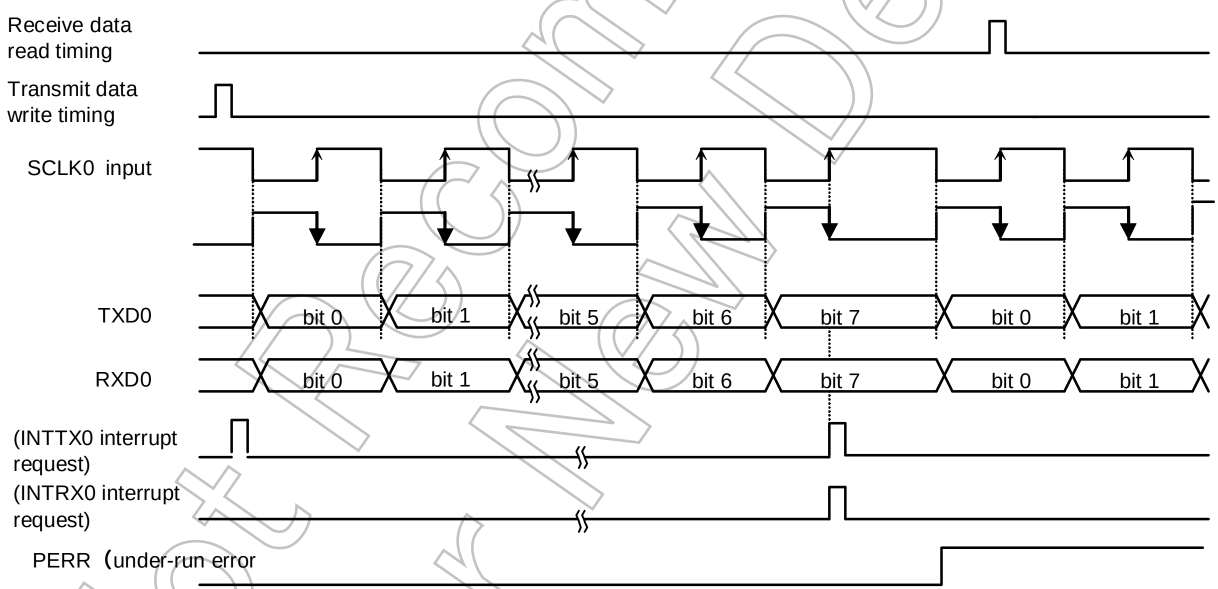
If SC0MOD2 <WBUF> = "1" and double buffering is enabled for both transmission and reception, the interrupt INTRX0 is generated at the timing transmit buffer 2 data is moved to transmit buffer 1 after completing data transmission from transmit buffer 1. At the same time, the 8 bits of data received is 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 transmit buffer 1 (in which data has been moved from transmit 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 in transmit buffer 2 when SCLK for the next frame is input, an under-run error occurs.



<WBUF> = "0" (if double buffering is disabled)



<WBUF> = "1" (if double buffering is enabled) (no errors)



<WBUF> = "1" (if double buffering is enabled) (error generation)

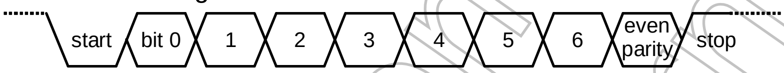
Fig. 13.5.1.6 Send/Receive Operation in the I/O Interface Mode (SCLK0 Input Mode)

13.5.2 Mode 1 (7-bit UART Mode)

The 7-bit UART mode can be selected by setting the serial mode control register (SC0MOD <SM1, 0>) to "01."

In this mode, parity bits can be added to the transmit data stream; the serial mode control register (SC0CR <PE>) controls the parity enable/disable setting. When <PE> is set to "1" (enable), either even or odd parity may be selected using the SC0CR <EVEN> bit. The length of the stop bit can be specified using SC0MOD2<SBLEN>.

Example: The control register settings for transmitting in the following data format are listed in the following table.



← Transmission direction (Transmission rate of 2400 bps @ fc = 24.576 MHz)

* Clock conditions			
	System clock	:	High-speed (fc)
	High-speed clock gear	:	1x (fc)
	Prescaler clock	:	fperiph/4 (fperiph=fsys)

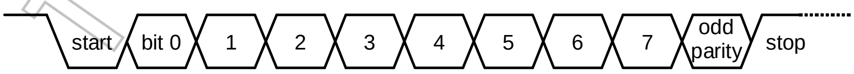
		7	6	5	4	3	2	1	0	
PCCR	←	−	−	−	−	−	−	−	1	Designates PC0 as the TXD0 pin.
PCFC	←	−	−	−	−	−	−	−	1	
SC0MOD	←	X	0	−	X	0	1	0	1	Sets the 7-bit UART mode.
SC0CR	←	X	1	1	X	X	X	0	0	Adds even parity.
BR0CR	←	0	0	1	0	1	0	1	0	Sets the data rate to 2400 bps.
IMC3	←	−	1	1	−	0	1	0	0	Enables the INTTX0 interrupt and sets to level 4 by the <31:24> bits of the 32 bit register.
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 9600 bps @ fc = 24.576 MHz)

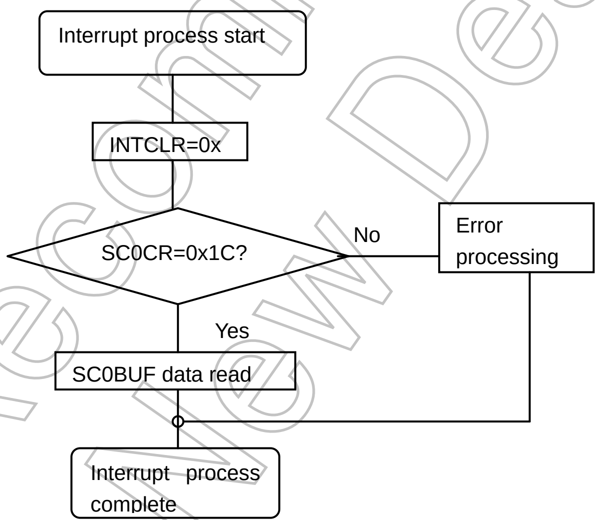
* Clock conditions			
	System clock	:	High-speed (fc)
	High-speed clock gear	:	1x (fc)
	Prescaler clock	:	fperiph/4 (fperiph=fsys)

Main routine settings

		7	6	5	4	3	2	1	0	
PCCR	←	–	–	–	–	–	–	0	–	Designates PC1 as the RXD0 pin.
PCFC	←	–	–	–	–	–	–	1	–	
SC0MOD	←	–	0	0	X	1	0	0	1	Selects the 8-bit UART mode.
SC0CR	←	X	0	1	X	X	X	0	0	Sets odd parity.
BR0CR	←	0	0	0	1	0	1	0	1	Sets the data rate to 9600 bps.
IMC3	←	–	1	1	–	0	1	0	0	Enables the INTRX0 interrupt and sets to level 4 by the <23:16> bits of the 32 bit register.
SC0MOD	←	–	–	1	X	–	–	–	–	Enables reception of data.

An example of interrupt routine process

INTCLR 0 0 0 1 1 1 0 0 0 — Clears the interrupt request. (0x0000_0038)
Reg. ← SC0CR AND 0x1C
if Reg. ≠ 0 then ERROR processing — Performs error check.
Reg. ← SC0BUF — Reads received data.
Interrupt processing is completed.
(Note) X: don't care –: no change



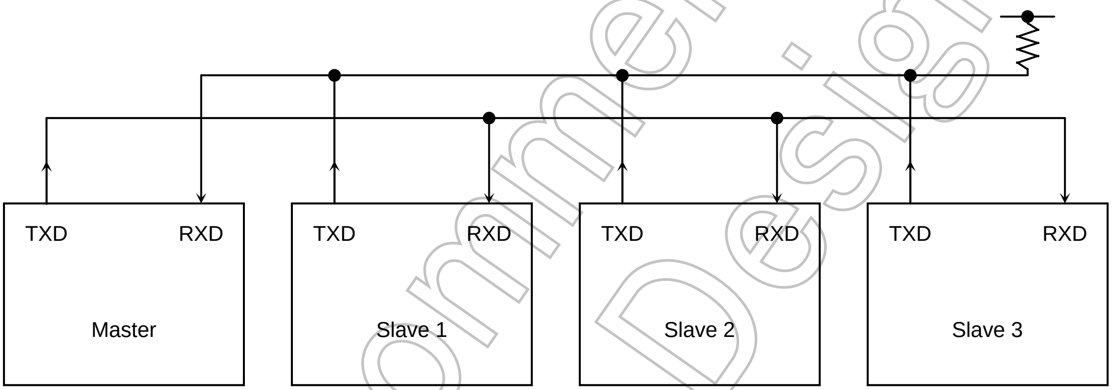
13.5.4 Mode 3 (9-bit UART)

The 9-bit UART mode can be selected by setting SC0MOD0 <SM1:0> to "11." In this mode, parity bits must be disabled (SC0CR <PE> = "0").

The most significant bit (9th bit) is written to bit 7 <TB8> of the serial mode control register 0 (SC0MOD0) for transmit data and it is stored in bit 7 <RB8> of the serial control register SC0CR upon receiving data. When writing or reading data to/from the buffers, the most significant bit must be written or read first before writing or reading to/from SC0BUF. The stop bit length can be specified using SC0MOD2 <SBLEN>.

Wakeup function

In the 9-bit UART mode, slave controllers can be operated in the wake-up mode by setting the wake-up function control bit SC0MOD0 <WU> to "1." In this case, the interrupt INTRX0 will be generated only when SC0CR <RB8> is set to "1."



(Note) The TXD pin of the slave controller must be set to the open drain output mode using the ODE register.

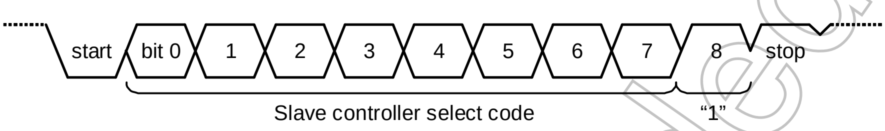
Fig. 13.5.4.1 Serial Links to Use Wake-up Function

Protocol

① Select the 9-bit UART mode for the master and slave controllers.

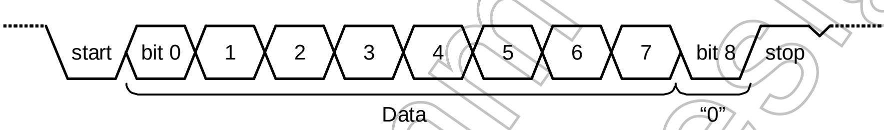
② Set SC0MOD <WU> to "1" for the slave controllers to make them ready to receive data.

③ The master controller is to send a single frame of data that includes the slave controller select code (8 bits). In this, the most significant bit (bit 8) <TB8> must be set to "1."



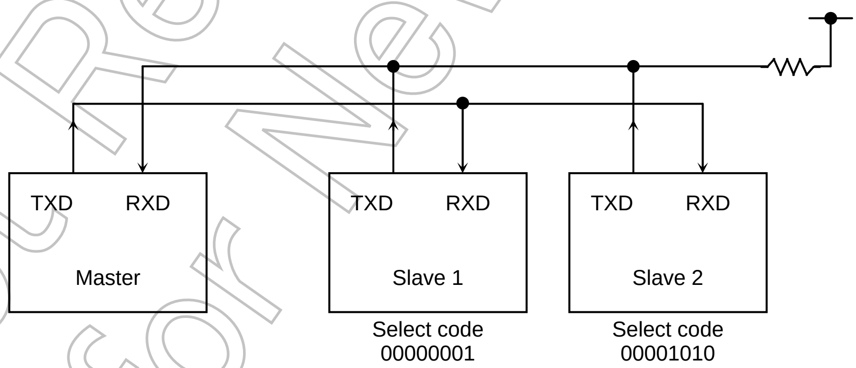
④ Every slave controller receives the above data frame; if the code received matches with the controller's own select code, it clears the WU bit to "0."

⑤ The master controller transmits data to the designated slave controller (the controller of which SC0MOD <WU> bit is cleared to "0"). In this, the most significant bit (bit 8) <TB8> must be set to "0."



⑥ The slave controllers with the <WU> bit set to "1" ignore the receive data because the most significant bit (bit 8) <RB8> is cleared to "0" and thus no interrupt (INTRX0) is generated. Also, the slave controller with the <WU> bit cleared 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

PCCR	← – – – – – – 0 1	Designates PC0/PC1 as the TXD0/RXD0 pins, respectively.
PCFC	← – – – – – – 1 1	
IMC3	← – 1 1 – 0 1 0 1	Enables the INTRX0 interrupt and sets to level 5 by the <23:16> bits of the 32 bit register.
	← – 1 1 – 0 1 0 0	Enables the INTTX0 interrupt and sets to level 4 by the <31:24> bits of the 32 bit register.
SC0MOD0	← 1 0 1 0 1 1 1 0	Sets the 9-bit UART mode and 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	0 0 0 1 1 1 1 0 0	Clears the interrupt request. (0x0000_003C).
SC0MOD0	← 0 – – – – – – –	Sets TB8 to "0."
SC0BUF	← * * * * * * * *	Sets the data to be sent.
interrupt processing is completed.		

④ Slave controller setting

Main routine

PCCR	← – – – – – – 0 1	Designates PC0 as TXD (open drain output) and P61 as RXD.
PCFC	← – – – – – – 1 1	
PCODE	← – – – – – – – 1	
IMC3	← – 1 1 – 0 1 1 0	Enables INTTX0 and INTRX0...
	← – 1 1 – 0 1 0 1	
SC0MOD0	← 0 0 1 1 1 1 1 0	Sets the 9-bit UART mode and f_{SYS}/2 transfer clock and sets <WU> to "1."

Interrupt routine (INTRX0)

INTCLR	0 0 0 1 1 1 0 0 0	Clears the interrupt request.
Reg.	← SC0BUF	
if Reg. = Select code		
Then		
SC0MOD0	← – – – 0 – – – –	Clears <WU> to "0."

14. Serial Bus Interface (SBI)

The TMP19A63 contains two Serial Bus Interface (SBI) channels; CH0 and CH1 that operate identically (only CH0 is described here). The Serial Bus Interfaces have the following two operation modes.

- I²C bus mode (with multi-master capability)
- Clock-synchronous 8-bit SIO mode

In the I²C bus mode, the SBI is connected to external devices via PE5 (SDA) and PE6 (SCL). In the clock-synchronous 8-bit SIO mode, the SBI is connected to external devices via PE7 (SCK), PE5 (SO) and PE6 (SI).

The following table shows the programming required to put the SBI in each operating mode.

	PFODE <PFODE1:0>		
I²C bus mode	11	X11	011
Clock-synchronous 8- bit SIO mode	XX	101 (clock output) 001 (clock input)	111

X: Don't care

14.1 Configuration

The configuration is shown in Fig. 14.1.

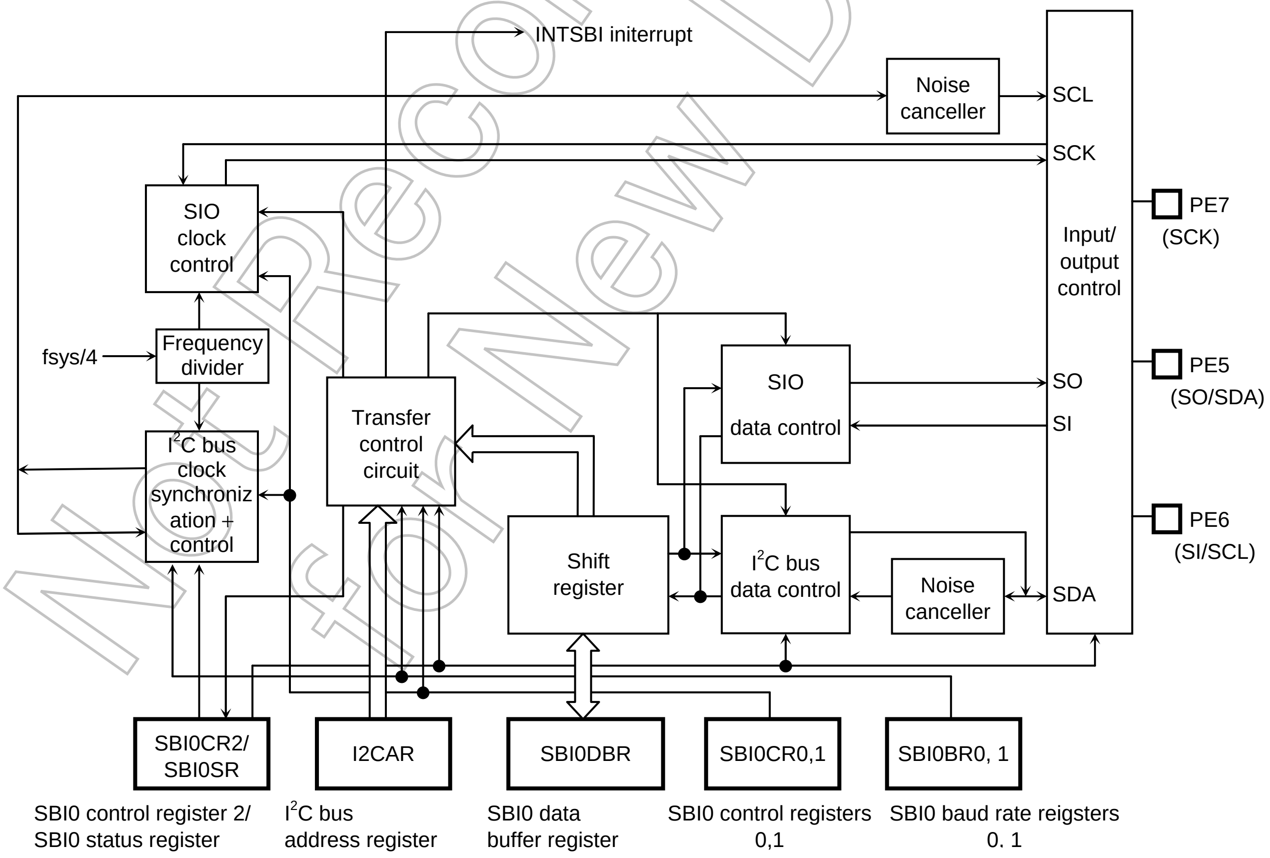


Fig. 14.1 SBI Block Diagram

14.2 Control

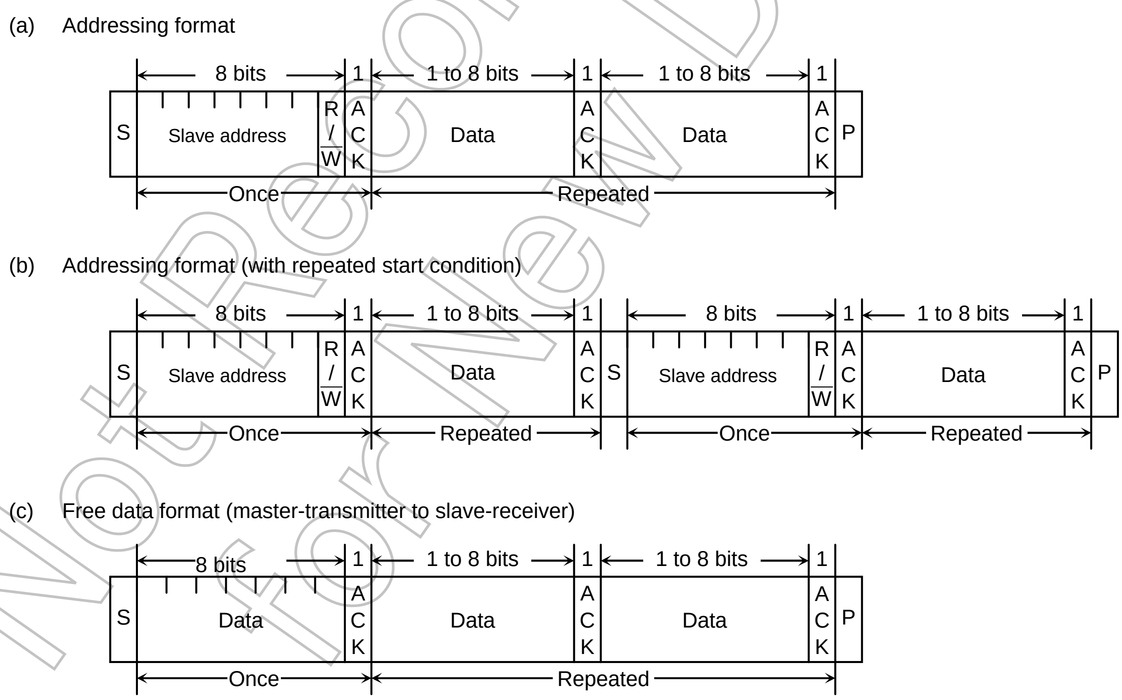
The following registers control the serial bus interface and provide its status information for monitoring.

- Serial bus interface control register 0 (SBI0CR0)
- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface buffer register (SBI0DBR)
- I^{2}C bus address register (I2CAR)
- Serial bus interface status register (SBI0SR)
- Serial bus interface baud rate register 0 (SBI0BR0)

The functions of these registers vary, depending on the mode in which the SBI is operating. For a detailed description of the registers, refer to "14.5 Control in the I^{2}C Bus Mode" and "14.7 Control in the Clock-synchronous 8-bit SIO Mode."

14.3 I^{2}C Bus Mode Data Formats

Fig. 14.3 shows the data formats used in the I^{2}C bus mode.



Note) S: Start condition
R/$\overline{W}$: Direction bit
ACK: Acknowledge bit
P: Stop condition

Fig. 14.3 I^{2}C Bus Mode Data Formats

14.4 Control Registers in the I^2C Bus Mode

The following registers control the serial bus interface (SBI) in the I^2C bus mode and provide its status information for monitoring.

Serial bus interface control register 0

SBI0CR0
(0xFFFF_F607

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0	0	0	0	0	0	0
Function	SBI operation 0:disable 1:enable							

<SBIEN>: To use the SBI, enable the SBI operation ("1") before setting each register in the SBI module.

(Note) SBICR0 bits 0 to 6 are read as “0”.

Fig. 14.4.1 I^2C Bus Mode Register

Serial bus interface control register 1

SBI0CR1 (0xFFFF_F600)

	7	6	5	4	3	2	1	0
bit Symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	R/W			R/W	R	R/W		R/W
After reset	0	0	0	0	1	0	0	1
Function	Select the number of bits per transfer (Note 1)			Acknowledgment clock 0: Not generate 1: Generate		Select internal SCL output clock frequency (Note 2) and reset monitor		

On writing <SCK2:0>: Select internal SCL output clock frequency

000	n=5	265 kHz
001	n=6	201 kHz
010	n=7	136 kHz
011	n=8	83 kHz
100	n=9	46 kHz
101	n=10	25 kHz
110	n=11	13 kHz
111		reserved

System clock: fsys (=54 MHz)

Clock gear : fc/1

Frequency = $\frac{fsys/2}{2^n + 70}$ [Hz]

On reading <SWRMON> Software reset condition 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>	<ACK> = 0		<ACK> = 1	
	Number of clock cycles	Data length	Number of clock cycles	Data length
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

(Note 1) **Clear <BC2:0> to "000" before switching the operation mode to the clock-synchronous 8-bit SIO mode.**

(Note 2) **For details on the SCL line clock frequency, refer to "14.5.3 Serial Clock."**

(Note 3) **After a reset, the <SCK0/SWRMON> bit is read as "1." However, if the SIO mode is selected at the SBICR2 register, the initial value of the <SCK0> bit is "0."**

Fig. 14.4.2 I^2C Bus Mode Register

Serial bus interface control register 2

SBI0CR2 (0xFFFF_F603)

	7	6	5	4	3	2	1	0
bit Symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
Read/Write	W				W		W	
After reset	0	0	0	1	0	0	0	0
Function	Select master/slave 0: Slave 1: Master	Select transmit/ receive 0: Receive 1: Transmit	Start/stop condition generation 0: Stop condition generated 1: Start condition generated	Clear INTSBI interrupt request 0: – 1: Clear interrupt request	Select serial bus interface operating mode (Note 2) 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (Reserved)		Software reset generation Write "10" followed by "01" to generate a reset.	

Select serial bus interface operating mode (Note 2)

00	Port mode (Serial bus interface output disabled)
01	Clock-synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

(Note 1) **Reading this register causes it to function as the SBISR register.**

(Note 2) **Ensure that the bus is free before switching the operating mode to the port mode. Ensure that the port is at the "H" level before switching the operating mode from the port mode to the I^2C bus or clock-synchronous 8-bit SIO mode.**

Fig. 14.4.3 I^2C Bus Mode Register

Table 14.4.4 Base Clock Resolution

@fsys = 54 MHz

Clock gear value <GEAR2:0>	Base clock resolution
000 (fc)	fsys/2^2 (0.07μs)
100 (fc/2)	fsys/2^3 (0.14μs)
110 (fc/4)	fsys/2^4 (0.28μs)
111 (fc/8)	fsys/2^5 (0.58μs)

Serial bus interface status register

SBI0SR (0xFFFF_F603)		7	6	5	4	3	2	1	0
	bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
	Read/Write	R							
	After reset	0	0	0	1	0	0	0	0
	Function	Master/ slave selection monitor 0: Slave 1: Master	Transmit/ receive selection monitor 0: Receive 1: Transmit	I^2C bus state monitor 0: Free 1: Busy	INTSBI interrupt request monitor 0: Interrupt request generated 1: Interrupt request cleared	Arbitration lost detection 0: – 1: Detected	Slave address match detection 0: – 1: Detected	General call detection 0: – 1: Detected	Last received bit monitor 0: "0" 1: "1"

Last received bit monitor

0	The last bit received was "0."
1	The last bit received was "1."

Slave address match detection

0	–
1	Addressed as slave match or general call detected

Arbitration lost detection

0	–
1	Arbitration lost is detected.

(Note) Writing to this register causes it to function as SBICR2.

Fig. 14.4.5 I^2C Bus Mode Register

Serial bus interface baud rate register 0

SBI0BR0 (0xFFFF_F604)

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI						
Read/Write	R	R/W	R					R/W
After reset	1	0	1	1	1	1	1	0
Function		IDLE 0: Stop 1: Operate						Be sure to write "0".

Operation in the IDLE mode

0	Stop
1	Operate

Serial bus interface data buffer register

SBI0DBR (0xFFFF_F601)

	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

SBI0I2CAR (0xFFFF_F602)

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set the slave address when the SBI acts as a slave device.							Specify address recognition mode

Specify address recognition mode

0	Recognizes the slave address.
1	Does not recognize slave address.

Fig 14.4.6 I^2C Bus Mode Register

14.5 Control in the I²C Bus Mode

14.5.1 Setting the Acknowledgement Mode

Setting SBI0CR1<ACK> to "1" selects the acknowledge mode. When operating as a master, the SBI adds one clock for acknowledgment signals. As a transmitter, the SBI releases the SDA pin during this clock cycle to receive acknowledgment signals from the receiver. As a receiver, the SBI pulls the SDA pin to the "L" level during this clock cycle and generates acknowledgment signals.

Setting <ACK> to "0" selects the non-acknowledgment mode. When operating as a master, the SBI does not generate clock for acknowledgement signals.

14.5.2 Setting the Number of Bits per Transfer

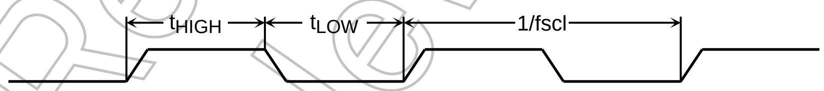
SBI0CR1 <BC2:0> specifies the number of bits of the next data to be transmitted or received.

Under the start condition, <BC2:0> is set to "000," causing a slave address and the direction bit to be transferred in a packet of eight bits. At other times, <BC2:0> keeps a previously programmed value.

14.5.3 Serial Clock

① Clock source

SBI0CR1 <SCK2:0> specifies the maximum frequency of the serial clock to be output from the SCL pin in the master mode.



$$t_{LOW} = 2^{n-1}/(fsys/2) + 58/(fsys/2)$$
$$t_{HIGH} = 2^{n-1}/(fsys/2) + 12/(fsys/2)$$
$$fscl = 1/(t_{Low} + t_{HIGH}) = \frac{fsys/2}{2^n + 70}$$

SBI0CR1 <SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Fig. 14.5.3.1 Clock Source

The highest speeds in the standard and high-speed modes are specified to 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 I²C bus is driven by using the wired-AND connection due to its pin structure. The first master that pulls its clock line to the "L" level overrides other masters producing the "H" level on their clock lines. This must be detected and responded by the masters producing the "H" level.

Clock synchronization assures correct data transfer on a bus that has two or more masters.

For example, the clock synchronization procedure for a bus with two masters is shown below.

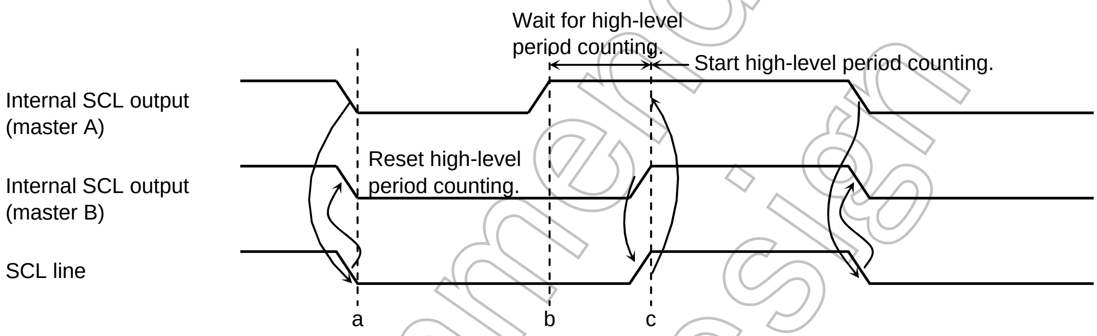


Fig. 14.5.3.2 Example of Clock Synchronization

At point a, Master A pulls its internal SCL output to the "L" level, bringing the SCL bus line to the "L" level. Master B detects this transition, resets its "H" level period counter, and pulls its internal SCL output level to the "L" level.

Master A completes counting of its "L" level period at point b, and brings its internal SCL output to the "H" level. However, Master B still keeps the SCL bus line at the "L" level, and Master A stops counting of its "H" level period counting. After Master A detects that Master B brings its internal SCL output to the "H" level and brings the SCL bus line to the "H" level at point c, it starts counting of its "H" level period.

This way, the clock on the bus is determined by the master with the shortest "H" level period and the master with the longest "L" level period among those connected to the bus.

14.5.4 Slave Addressing and Address Recognition Mode

When the SBI is configured to operate as a slave device, the slave address <SA6:0> and <ALS> must be set at I2CAR. Setting <ALS> to "0" selects the address recognition mode.

14.5.5 Configuring the SBI as a Master or a Slave

Setting SBI0CR2<MST> to "1" configures the SBI to operate as a master device.

Setting <MST> to "0" configures the SBI as a slave device. <MST> is cleared to "0" by the hardware when the stop condition has been detected on the bus or when arbitration has been lost.

14.5.6 Configuring the SBI as a Transmitter or a Receiver

Setting SBI0CR2 <TRX> to "1" configures the SBI as a transmitter. Setting <TRX> to "0" configures the SBI as a receiver.

In the slave mode, the SBI receives the direction bit ($R/\overline{W}$) from the master device on the following occasions:

- when data is transmitted in the addressing format
- when the received slave address matches the value specified at I2CCR
- when a general-call address is received; i.e., the eight bits following the start condition are all zeros

If the value of the direction bit ($R/\overline{W}$) is "1," <TRX> is set to "1" by the hardware. If the bit is "0," <TRX> is set to "0."

As a master device, the SBI receives acknowledgement from a slave device. If the direction bit of "1" is transmitted, <TRX> is set to "0" by the hardware. If the direction bit is "0," <TRX> changes to "1." If the SBI does not receive acknowledgement, <TRX> retains the previous value.

<TRX> is cleared to "0" by the hardware when the stop condition has been detected on the bus or when arbitration has been lost.

14.5.7 Generating Start and Stop Conditions

When SBI0SR<BB> is "0," writing "1" to SBI0CR2 <MST, TRX, BB, PIN> causes the SBI to generate the start condition on the bus and output 8-bit data. <ACK> must be set to "1" in advance.

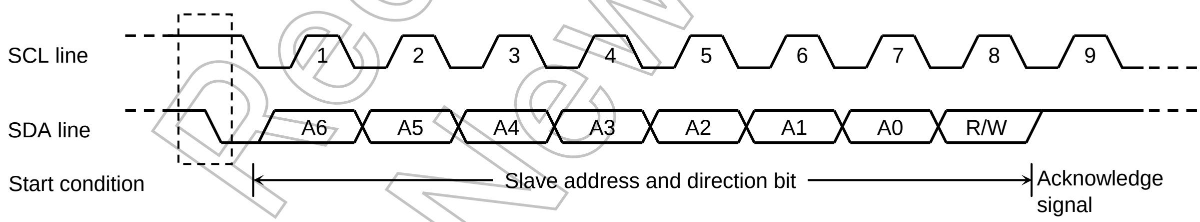


Fig. 14.5.7.1 Generating the Start Condition and a Slave Address

When <BB> is "1," writing "1" to <MST, TRX, PIN> and "0" to <BB> causes the SBI to start a sequence for generating the stop condition on the bus. The contents of <MST, TRX, BB, PIN> should not be altered until the stop condition appears on the bus.

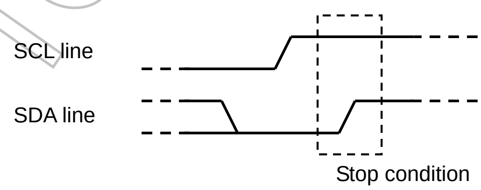


Fig. 14.5.7.2 Generating the Stop Condition

SBI0SR<BB> can be read to check the bus state. <BB> is set to "1" when the start condition is detected on the bus (the bus is busy), and set to "0" when the stop condition is detected (the bus is free).

14.5.8 Interrupt Service Request and Release

When a serial bus interface interrupt request (INTSBI) is generated, SBI0CR2 <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 SBI0DBR. 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 SBI0CR2<PIN>, it is set to "1." However, writing "0" does clear this bit to "0."

14.5.9 Serial Bus Interface Operating Modes

SBI0CR2 <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^{2}C bus mode. Make sure that the bus is free before switching the operating mode to the port mode.

14.5.10 Lost-arbitration Detection Monitor

The I^{2}C 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^{2}C-bus arbitration takes place on the SDA line.

The arbitration procedure for two masters on a bus is shown below. Up until point a, Master A and Master B output the same data. At point a, Master A outputs the "L" level and Master B outputs the "H" level. Then Master A pulls the SDA bus line to the "L" level because the line has the wired-AND connection. When the SCL line goes high at point b, the slave device reads the SDA line data, i.e., data transmitted by Master A. At this time, data transmitted by Master B becomes invalid. In other words, Master B loses arbitration. Master B releases its SDA pin, so that it does not affect the data transfer initiated by another master. If two or more masters have transmitted exactly the same first data word, the arbitration procedure continues with the second data word.

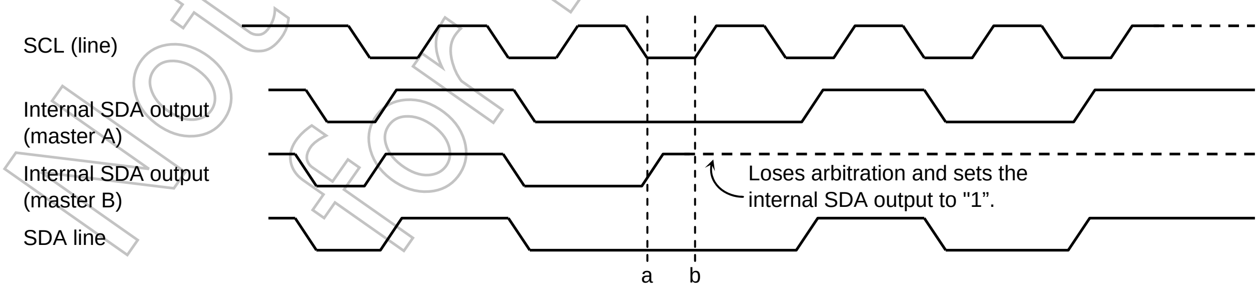


Fig. 14.5.10.1 Lost Arbitration

A master compares the SDA bus line level and the internal SDA output level at the rising of the SCL line. If there is a difference between these two values, the master loses arbitration and sets SBI0SR <AL> to "1."

When <AL> is set to "1," SBI0SR <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 SBI0DBR or data is written to SBI0CR2.

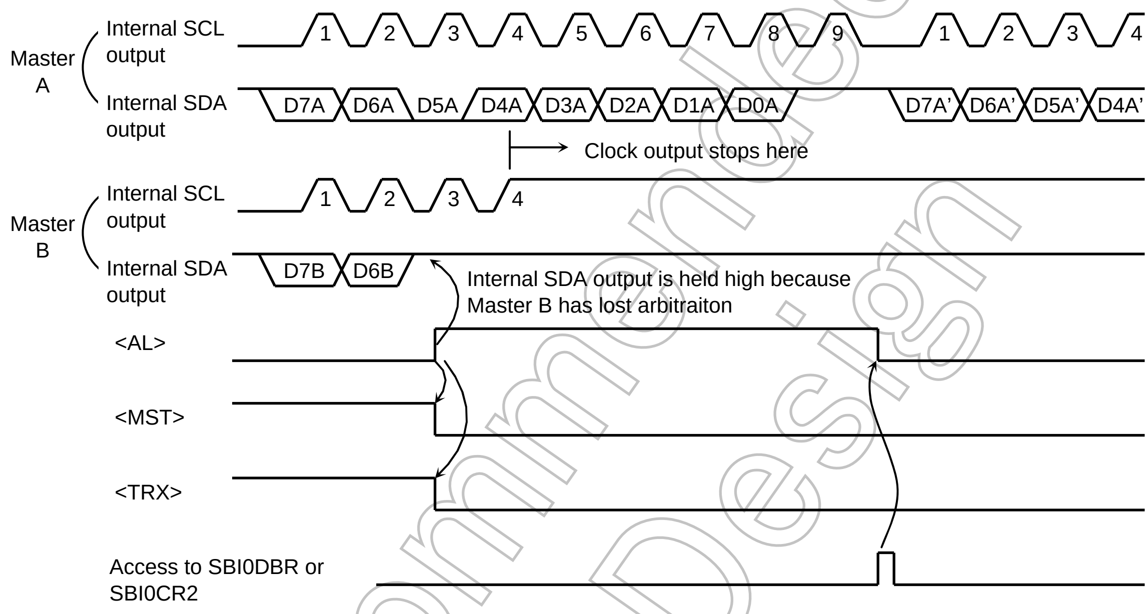


Fig. 14.5.10.2 Example of Master B Losing Arbitration (D7A = D7B, D6A = D6B)

14.5.11 Slave Address Match Detection Monitor

When the SBI operates as a slave device in the address recognition mode (I2CCR <ALS> = "0"), SBI0SR <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 SBI0DBR.

14.5.12 General-call Detection Monitor

When the SBI operates as a slave device, SBI0SR <AD0> is set to "1" when it receives the general-call address; i.e., the eight bits following the start condition are all zeros. <AD0> is cleared to "0" when the start or stop condition is detected on the bus.

14.5.13 Last Received Bit Monitor

SBI0SR <LRB> is set to the SDA line value that was read at the rising of the SCL line. In the acknowledgment mode, reading SBISR <LRB> immediately after generation of the INTSBI interrupt request causes ACK signal to be read.

14.5.14 Software Reset

If the serial bus interface circuit locks up due to external noise, it can be initialized by using a software reset.

Writing "10" followed by "01" to SBI0CR2 <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 I²C mode to the synchronous communication mode.

14.5.15 Serial Bus Interface Data Buffer Register (SBI0DBR)

Reading or writing SBI0DBR initiates reading received data or writing transmitted data. When the SBI is acting as a master, setting a slave address and a direction bit to this register generates the start condition.

14.5.16 I²C Bus Address Register (I2CAR)

When the SBI is configured as a slave device, the I2CAR<SA6:0> bit is used to specify a slave address. If I2C0AR <ALS> is set to "0," the SBI recognizes a slave address transmitted by the master device and receives data in the addressing format. If <ALS> is set to "1," the SBI does not recognize a slave address and receives data in the free data format.

14.5.17 IDLE Setting Register (SBI0BR0)

The SBI0BR0<I2SBI> register determines if the SBI operates or not when it enters the IDLE mode. This register must be programmed before executing an instruction to switch to the standby mode.

14.6 Data Transfer Procedure in the I²C Bus Mode

14.6.1 Device Initialization

First, program SBI0CR1<ACK, SCK2:0> by writing "0" to bits 7 to 5 and bit 3 in SBI0CR1.

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 SBI0CR2 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	
SBI0CR1	←	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.
SBI0CR2	←	0	0	0	1	1	0	0	0	Configures the SBI as a slave receiver.

(Note) X: Don’t care

14.6.2 Generating the Start Condition and a Slave Address

① Master mode

In the master mode, the following steps are required to generate the start condition and a slave address.

First, ensure that the bus is free (<BB> = "0"). Then, write "1" to SBI0CR1 <ACK> to select the acknowledgment mode. Write to SBI0DBR a slave address and a direction bit to be transmitted.

When <BB> = "0," writing "1111" to SBI0CR2 <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 SBI0DBR with the first eight clocks, and releases the SDA line in the ninth clock to receive an acknowledgment signal from the slave device.

The INTSBI interrupt request is generated on the falling of the ninth clock, and <PIN> is cleared to "0." In the master mode, the SBI holds the SCL line at the "L" level while <PIN> is "0." <TRX> changes its value according to the transmitted direction bit at generation of the INTSBI interrupt request, provided that an acknowledgment signal has been returned from the slave device.

Settings in main routine

		7	6	5	4	3	2	1	0	
Reg.	← SBISR									
Reg.	← Reg. e 0x20									
if Reg.	≠ 0x00									Ensures that the bus is free.
Then										
SBI0CR1	←	X	X	X	1	0	X	X	X	Selects the acknowledgement mode.
SBI0DR1	←	X	X	X	X	X	X	X	X	Specifies the desired slave address and direction.
SBI0CR2	←	1	1	1	1	1	0	0	0	Generates the start condition.

Example of INTSBI interrupt routine

INTCLR ← 0X50	Clears the interrupt request.
Processing	
End of interrupt	

② Slave mode

In the slave mode, the SBI receives the start condition and a slave address.

After receiving the start condition from the master device, the SBI receives a slave address and a direction bit from the master device during the first eight clocks on the SCL line. If the received address matches its slave address specified at I2CAR or is equal to the general-call address, the SBI pulls the SDA line to the "L" level during the ninth clock and outputs an acknowledgment signal.

The 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.**

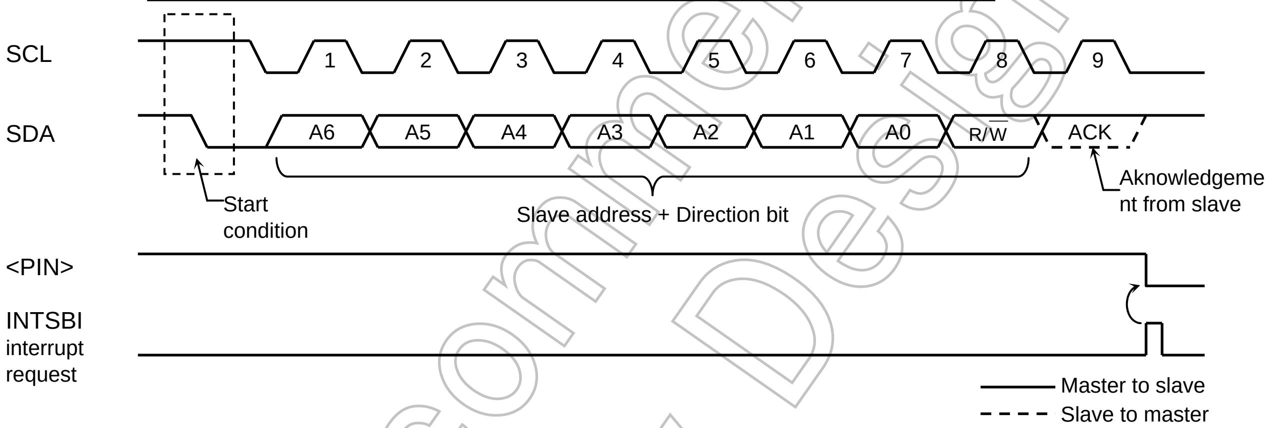


Fig. 14.6.2.1 Generation of the Start Condition and a Slave Address

14.6.3 Transferring a Data Word

At the end of a data word transfer, the INTSBI interrupt is generated to test <MST> to determine whether the SBI is in the master or slave mode.

① Master mode (<MST> = "1")

Test <TRX> to determine whether the SBI is configured as a transmitter or a receiver.

Transmitter mode (<TRX> = "1")

Test <LRB>. If <LRB> is "1," that means the receiver requires no further data. The master then generates the stop condition as described later to stop transmission.

If <LRB> is "0," that means the receiver requires further data. If the next data to be transmitted has eight bits, the data is written into SBI0DBR. If the data has different length, <BC2:0> and <ACK> are programmed and the transmit data is written into SBI0DBR. Writing the data makes <PIN> to"1," causing the SCL pin to generate a serial clock for transfer of a next data word, and the SDA pin to transfer the data word. After the transfer is completed, the INTSBI interrupt request is generated, <PIN> is set to "0," and the SCL pin is pulled to the "L" level... To transmit more data words, test <LRB> again and repeat the above procedure.

INTSBI interrupt

if MST = 0
Then go to the slave-mode processing
if TRX = 0
Then go to the receiver-mode processing
if LRB = 0
Then go to processing for generating the stop condition

SBI0CR1	← X X X X 0 X X X	Specifies the number of bits to be transmitted and specify whether ACK is required.
SBI0DBR	← X X X X X X X X	Writes the transmit data.

End of interrupt processing

(Note) X: Don’t care

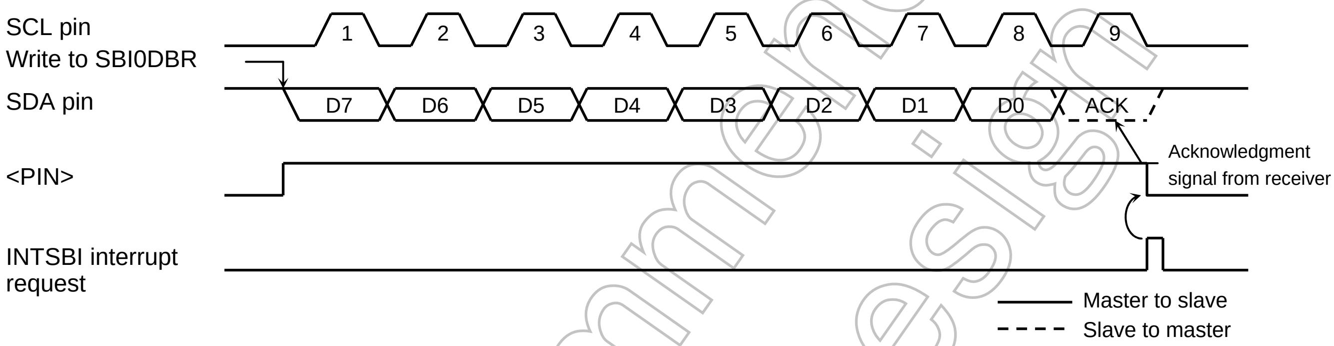


Fig. 14.6.3.1 <BC2:0> = "000" and <ACK> = "1" (Transmitter Mode)

Receiver mode (<TRX> = "0")

If the next data to be transmitted has eight bits, the transmit data is written into SBI0DBR. If the data has different length, <BC2:0> and <ACK> are programmed and the received data is read from SBI0DBR to release the SCL line. (The data read immediately after transmission of a slave address is undefined.) On reading the data, <PIN> is set to "1 and the serial clock is output to the SCL pin to transfer the next data word. In the last bit, when the acknowledgment signal becomes the "L" level, "0" is output to the SDA pin.

After that, the INTSBI interrupt request is generated, and <PIN> is cleared to "0," pulling the SCL pin to the "L" level. Each time the received data is read from SBI0DBR, one-word transfer clock and an acknowledgement signal are output.

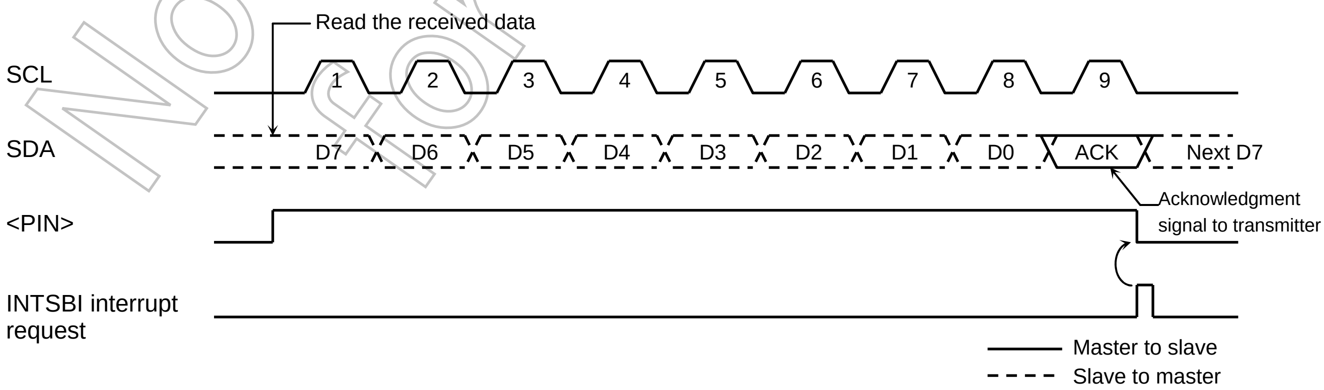


Fig. 14.6.3.2 <BC2:0> = "000" and <ACK> = "1" (Receiver Mode)

To terminate the data transmission from the transmitter, <ACK> must be set to "0" immediately before reading the second to last data word. This disables generation of an acknowledgment clock for the last data word. When the transfer is completed, an interrupt request is generated. After the interrupt processing, <BC2:0> must be set to "001" and the data must be read so that a clock is generated for 1-bit transfer. At this time, the master receiver holds the SDA bus line at the "H" level, which signals the end of transfer to the transmitter as an acknowledgment signal.

In the interrupt processing for terminating the reception of 1-bit data, the stop condition is generated to terminate the data transfer.

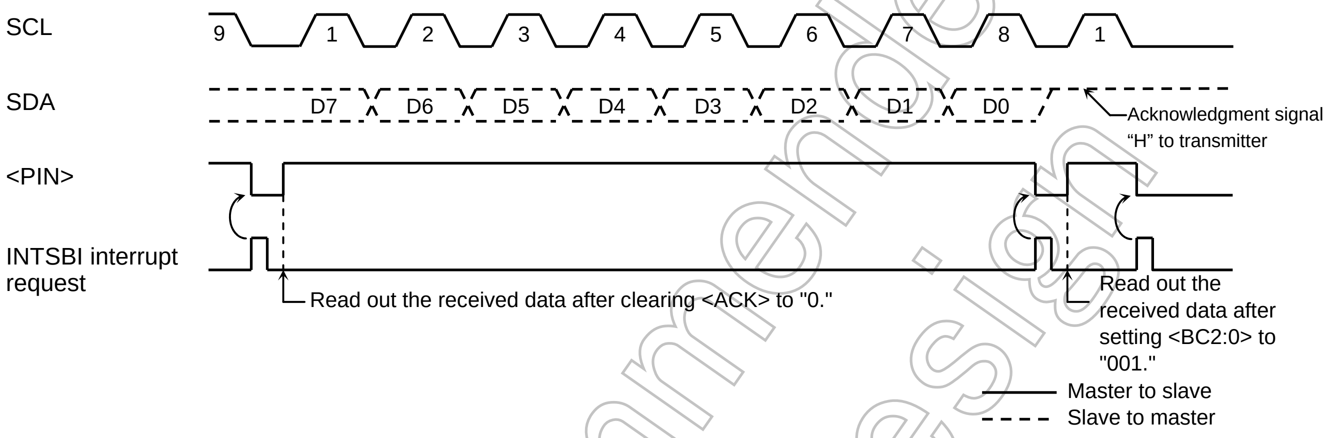


Fig. 14.6.3.3 Terminating Data Transmission in the Master Receiver Mode

Example: When receiving N data words

INTSBI interrupt (after data transmission)

	7 6 5 4 3 2 1 0	
SBI0CR1	← X X X X 0 X X X	Sets the number of bits of data to be received and specify whether ACK is required.
Reg.	← SBI0CBR	Reads dummy data.
End of interrupt		

INTSBI interrupt (first to (N-2)th data reception)

	7 6 5 4 3 2 1 0	
Reg.	← SBIDBR	Reads the first to (N-2)th data words.
End of interrupt		

INTSBI interrupt ((N-1)th data reception)

	7 6 5 4 3 2 1 0	
SBI0CR1	← X X X 0 0 X X X	Disables generation of acknowledgement clock.
Reg.	← SBIDBR	Reads the (N-1)th data word.
End of interrupt		

INTSBI interrupt (Nth data reception)

	7 6 5 4 3 2 1 0	
SBI0CR1	← 0 0 1 0 0 X X X	Generates a clock for 1-bit transfer.
Reg.	← SBIDBR	Reads the Nth data word.
End of interrupt		

INTSBI interrupt (after completing data reception)

Processing to generate the stop condition	Terminates the data transmission.
End of interrupt	

(Note) X: Don't care

② Slave mode (<MST> = "0")

In the slave mode, the SBI generates the INTSBI interrupt request on four occasions: 1) when the SBI has received any slave address from the master, 2) when the SBI has received a general-call address, 3) when the received slave address matches its own address, and 4) when a data transfer has been completed in response to a general-call. Also, if the SBI loses arbitration in the master mode, it switches to the slave mode. Upon the completion of data word transfer in which arbitration is lost, the INTSBI interrupt request is generated, <PIN> is cleared to "0," and the SCL pin is pulled to the "L" level. When data is written to or read from SBI0DBR or when <PIN> is set to "1," the SCL pin is released after a period of t_{LOW}.

In the slave mode, the normal slave mode processing or the processing as a result of lost arbitration is carried out.

SBISR <AL>, <TRX>, <AAS> and <AD0> are tested to determine the processing required. Table 14.6.3.4 shows the slave mode states and required processing.

Example: When the received slave address matches the SBI's own address and the direction bit is "1" in the slave receiver mode

INTSBI interrupt

```
if TRX = 0
Then go to other processing
if AL = 1
Then go to other processing
if AAS = 0
Then go to other processing
SBI0CR1  ←  X X X 1 0 X X X        Sets the number of bits to be transmitted.
SBI0DBR  ←  X X X X 0 X X X        Sets the transmit data.

(Note)                X: Don't care
```

Table 14.6.3.4 Processing in Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	State	Processing
1	1	1	0	Arbitration was lost while the slave address was being transmitted, and the SBI received a slave address with the direction bit "1" transmitted by another master.	Set the number of bits in a data word to <BC2:0> and write the transmit data into SBI0DBR.
	0	1	0	In the slave receiver mode, the SBI received a slave address with the direction bit "1" transmitted by the master.	
		0	0	In the slave transmitter mode, the SBI has completed a transmission of one data word.	Test LRB. If it has been set to "1," that means the receiver does not require further data. Set <PIN> to 1 and reset <TRX> to 0 to release the bus. If <LRB> has been reset to "0," that means the receiver requires further data. Set the number of bits in the data word to <BC2:0> and write the transmit data to the SBIDBR.
0	1	1	1/0	Arbitration was lost while a slave address was being transmitted, and the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by another master.	Read the SBIDBR (a dummy read) to set <PIN> to 1, or write "1" to <PIN>.
		0	0	Arbitration was lost while a slave address or a data word was being transmitted, and the transfer terminated.	
	0	1	1/0	In the slave receiver mode, the SBI received either a slave address with the direction bit "0" or a general-call address transmitted by the master.	
		0	1/0	In the slave receiver mode, the SBI has completed a reception of a data word.	Set the number of bits in the data word to <BC2:0> and read the received data from SBIDBR.

14.6.4 Generating the Stop Condition

When SBI0SR <BB> is "1," writing "1" to SBI0CR2 <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	
SBI0CR2	←	1	1	0	1	1	0	0	0	Generates the stop condition.

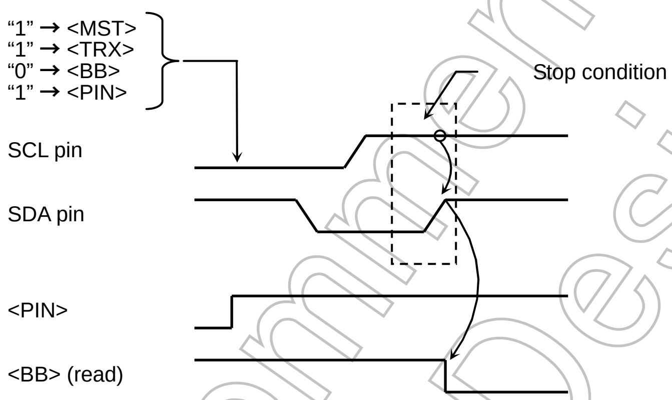


Fig. 14.6.4.1 Generating the Stop Condition

14.6.5 Repeated Start Procedure

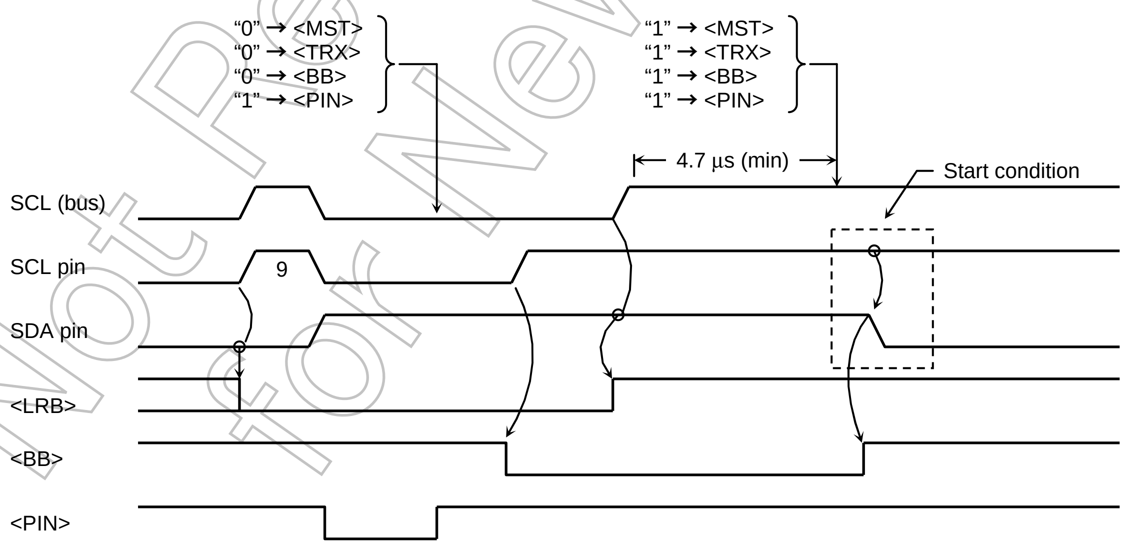
Repeated start is used when a master device changes the data transfer direction without terminating the transfer to a slave device. The procedure of generating a repeated start in the master mode is described below.

First, set SBI0CR2 <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 SBI0SR <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
SBI0CR2          ←      0 0 0 1 1 0 0 0     Releases the bus.
if SBI0SR<BB> ≠ 0                           Checks that the SCL pin is released.
Then
if SBI0SR<LRB> ≠ 1                          Checks that no other device is pulling the SCL pin to the
                                            "L" level.
Then
4.7 µs Wait
SBI0CR1          ←      X X X 1 0 X X X     Selects the acknowledgment mode.
SBI0DBR          ←      X X X X X X X X     Sets the desired slave address and direction.
SBI0CR2          ←      1 1 1 1 1 0 0 0     Generates the start condition.

(Note)                  X: Don't care
```



(Note) Do not write <MST> to "0" when it is "0." (Repeated start cannot be done.)

Fig. 14.6.5.1 Timing Chart of Generating a Repeated Start

14.7 Control in the Clock-synchronous 8-bit SIO Mode

The following registers control the serial bus interface in the clock-synchronous 8-bit SIO mode and provide its status information for monitoring.

Serial bus interface control register 0

SBI0CR0 (0xFFFF_F607)

	7	6	5	4	3	2	1	0
bit Symbol	SBIEN							
Read/Write	R/W	R						
After reset	0	0	0	0	0	0	0	0
Function	SBI operation 0：Disable 1：Enable							

<SBIEN>： To use the SBI, enable the SBI operation ("1") before setting each register of SBI module.

(Note) SBI0CR0 bits 0 through 6 are read as “0”..

Serial bus interface control register 1

SBI0CR1 (0xFFFF_F600)

	7	6	5	4	3	2	1	0
bit Symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
Read/Write	R/W				R	R/W		R/W
After reset	0	0	0	0	1	0	0	1
Function	Start transfer 0: Stop 1: Start	Abort transfer 0: Continue 1: Abort	Select transfer mode 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode			Select serial clock frequency		

On writing <SCK2:0>: Select serial clock frequency

000	n = 4	1.69 MHz
001	n = 5	844 kHz
010	n = 6	422 kHz
011	n = 7	211 kHz
100	n = 8	105 kHz
101	n = 9	53 kHz
110	n =10	26 kHz
111	—	External clock

System clock : fsys (=54 MHz)
Clock gear : fc/1
Frequency $= \frac{fsys/2}{2^n}$ [Hz]

(Note 1) **Set <SIOS> to "0" and <SIOINH> to "1" before programming the transfer mode and the serial clock.**

(Note 2) **<SCK0> bit is read as “0” after reset and its default value is “0”if SIO mode is selected in SBICR2 register.**

Serial bus interface data buffer register

SBI0DBR (0xFFFF_F601)

	7	6	5	4	3	2	1	0
bit Symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (Receive)/W (Transmit)							
After reset	0							

Fig. 14.7.1.1 SIO Mode Registers

Serial bus interface control register 2

SBI0CR2 (0xFFFF_F603)

	7	6	5	4	3	2	1	0
bit Symbol					SBIM1	SBIM0		
Read/Write	R				W		R	
After reset	1	1	1	1	0	0	1	1
Function					Select serial bus interface operating mode 00: Port mode 01: Clock-synchronous 8-bit SIO mode 10: I²C bus mode 11: (Reserved)			

Serial bus interface register

SBI0SR (0xFFFF_F603)

	7	6	5	4	3	2	1	0
bit Symbol					SIOF	SEF		
Read/Write	R				R		R	
After reset	1	1	1	1	0	0	1	1
Function					Serial transfer status monitor 0: Terminated 1: In progress	Shift operation status monitor 0: Terminated 1: In progress		

Serial bus interface baud rate register 0

SBI0BR0 (0xFFFF_F604)

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI						
Read/Write	R	R/W	R					R/W
After reset	1	0	1	1	1	1	1	0
Function		IDLE 0: Stop 1: Operate						Be sure to write “0”.

Fig. 14.7.1.2 SIO Mode Registers

14.7.1 Serial Clock

① Clock source

Internal or external clocks can be selected by programming SBI0CR1 <SCK2:0>.

Internal clock

In the internal clock mode, one of the seven frequencies can be selected as a serial clock, which is output to the outside through the SCK pin. At the beginning of a transfer, the SCK pin output becomes the "H" level.

If the program cannot keep up with this serial clock rate in writing the transmit data or reading the received data, the SBI automatically enters a wait period. During this period, the serial clock is stopped automatically and the next shift operation is suspended until the processing is completed.

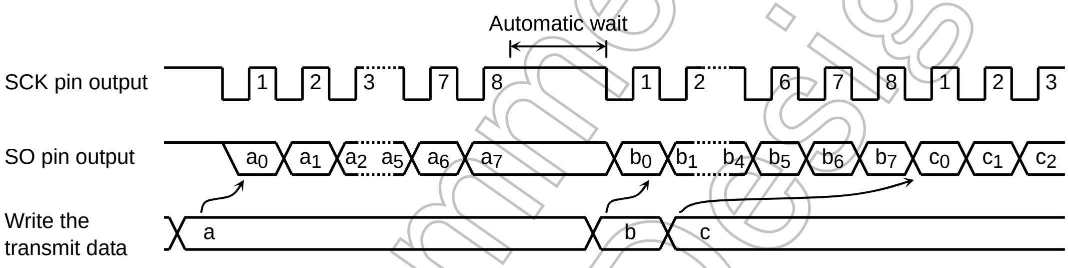


Fig. 14.7.1.3 Automatic Wait

External clock (<SCK2:0> ="111")

The SBI uses an external clock supplied from the outside to the SCK pin as a serial clock. For proper shift operations, the serial clock at the "H" and "L" levels must have the pulse widths as shown below.

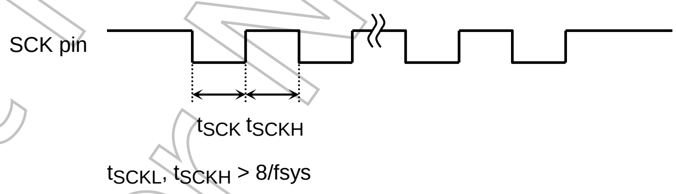


Fig. 14.7.1.4 Maximum Transfer Frequency of External Clock Input

② Shift Edge

Leading-edge shift is used in transmission. Trailing-edge shift is used in reception.

<u>Leading-edge shift</u>

Data is shifted at the leading edge of the serial clock (or the falling edge of the SCK pin input/output).

<u>Trailing-edge shift</u>

Data is shifted at the trailing edge of the serial clock (or the rising edge of the SCK pin input/output).

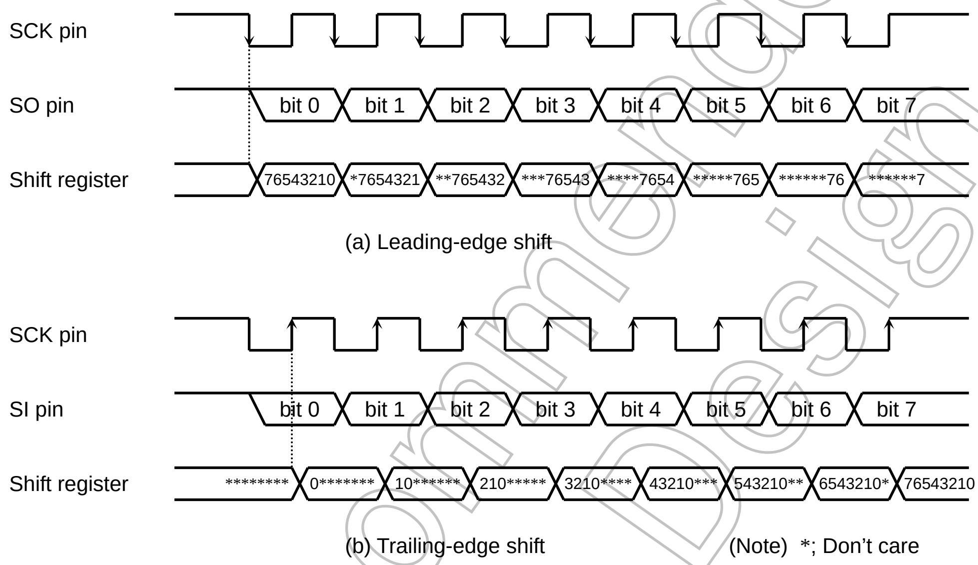


Fig. 14.7.1.5 Shift Edge

14.7.2 Transfer Modes

The transmit mode, the receive mode or the transmit/receive mode can be selected by programming SBI0CR1 <SIOM1:0>.

① 8-bit transmit mode

Set the control register to the transmit mode and write the transmit data to SBI0DBR.

After writing the transmit data, writing "1" to SBI0CR1 <SIOS> starts the transmission. The transmit data is moved from SBI0DBR 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, SBI0DBR becomes empty, and the INTSBI (buffer-empty) interrupt is generated, requesting the next transmit data.

In the internal clock mode, the serial clock will be stopped and automatically enter the wait state, if next data is not loaded after the 8-bit data has been fully transmitted. The wait state will be cleared when SBI0DBR is loaded with the next transmit data.

In the external clock mode, SBI0DBR 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 SBI0DBR 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 SBI0SR <SIOF> to "1" to the falling edge of SCK.

Transmission can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, remaining data is output before transmission ends. The program checks SBI0SR <SIOF> to determine whether transmission has come to an end. <SIOF> is cleared to "0" at the end of transmission. If <SIOINH> is set to "1," the transmission is aborted immediately and <SIOF> is cleared to "0."

In the external clock mode, <SIOS> must be set to "0" before the next transmit data shift operation is started. Otherwise, operation will stop after dummy data is transmitted.

```
                  7 6 5 4 3 2 1 0
SBI0CR1   ←       0 1 0 0 0 X X X      Selects the transmit mode.

SBI0DBR   ←       X X X X X X X X      Writes the transmit data.
SBI0CR1   ←       1 0 0 0 0 X X X      Starts transmission.
```

INTSBI interrupt

```
SBI0DBR   ←       X X X X X X X X      Writes the transmit data.
```

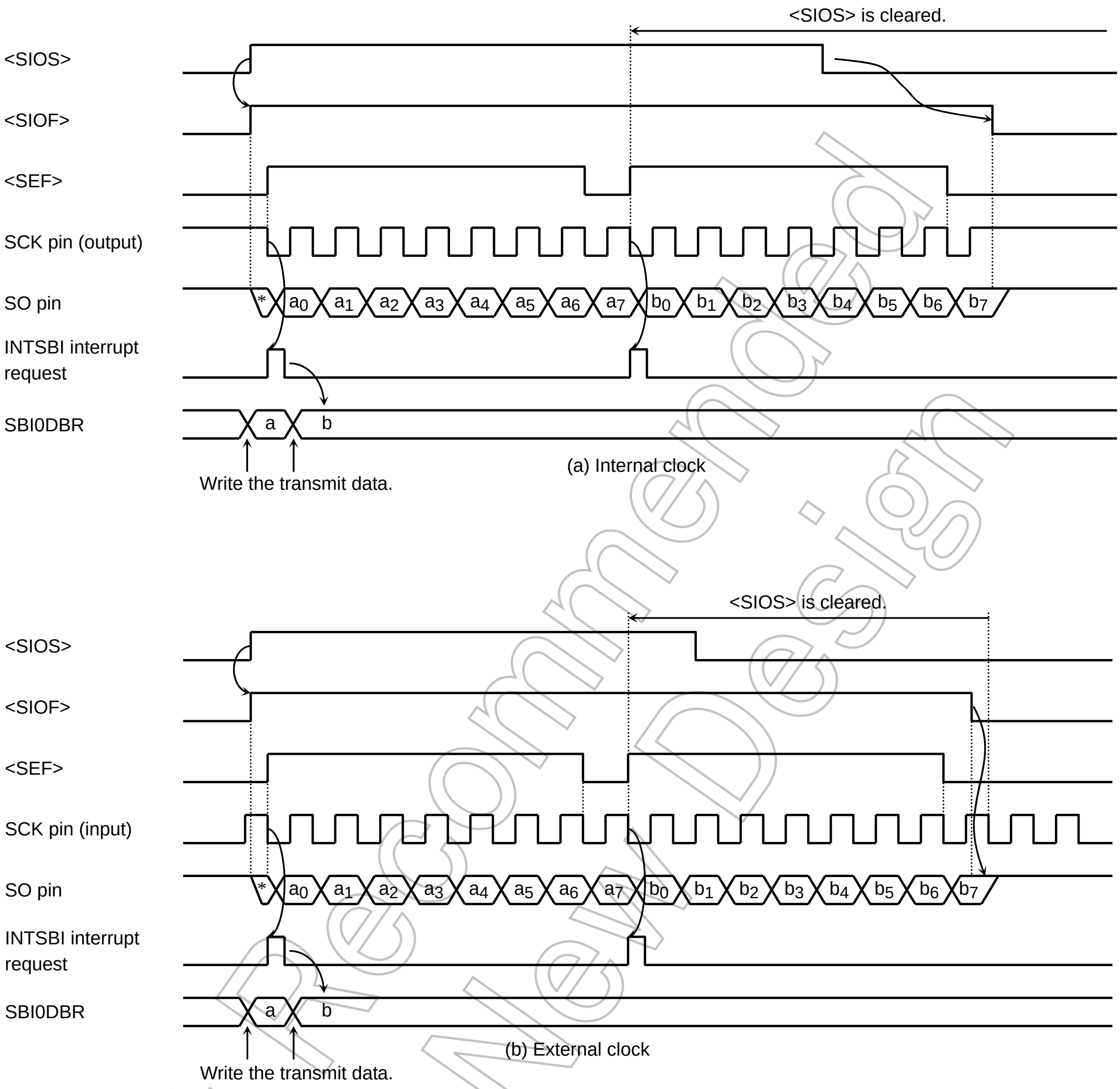


Fig. 14.7.2.1 Transmit Mode

Example of programming (MIPS16) to terminate transmission by <SIO> (external clock)

```
                ADDIU   r3,   r0,   0x04
STEST1    :     LB      r2, (SBI0SR)          ; If SBISR<SEF> = 1 then loop
                AND     r2,   r3
                BNEZ    r2,   STEST1
                ADDIU   r3,   r0,   0x20
STEST2    :     LB      r2, (Px)              ; If SCK = 0 then loop
                AND     r2,   r3
                BEQZ    r2,   STEST2
                ADDIU   r3,   r0,   0y00000111
                STB     r3, (SBI0CR1)         ; <SIOS> ← 0
```

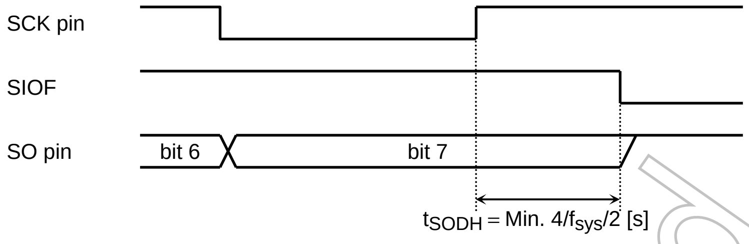


Fig. 14.7.2.2 Transmit Data Retention Time at the End of Transmission

② 8-bit receive mode

Set the control register to the receive mode. Then writing "1" to SBI0CR1 <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 SBI0DBR and the INTSBI (buffer-full) interrupt request is generated to request reading the received data. The interrupt service program then reads the received data from SBI0DBR.

In the internal clock mode, the serial clock will be stopped and automatically be in the wait state until the received data is read from SBIDBR.

In the external clock mode, shift operations are executed in synchronization with the external clock. The maximum data transfer rate varies, depending on the maximum latency between generating the interrupt request and reading the received data.

Reception can be terminated by clearing <SIOS> to "0" or setting <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, reception continues until all the bits of received data are written to SBI0DBR. The program checks SBI0SR <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 SBI0DBR 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	
SBI0CR1	←	0	1	1	1	0	X	X	X	Selects the receive mode.
SBI0CR1	←	1	0	1	1	0	0	0	0	Starts reception.

INTSBI interrupt

Reg.	← SBI0DBR	Reads the received data.

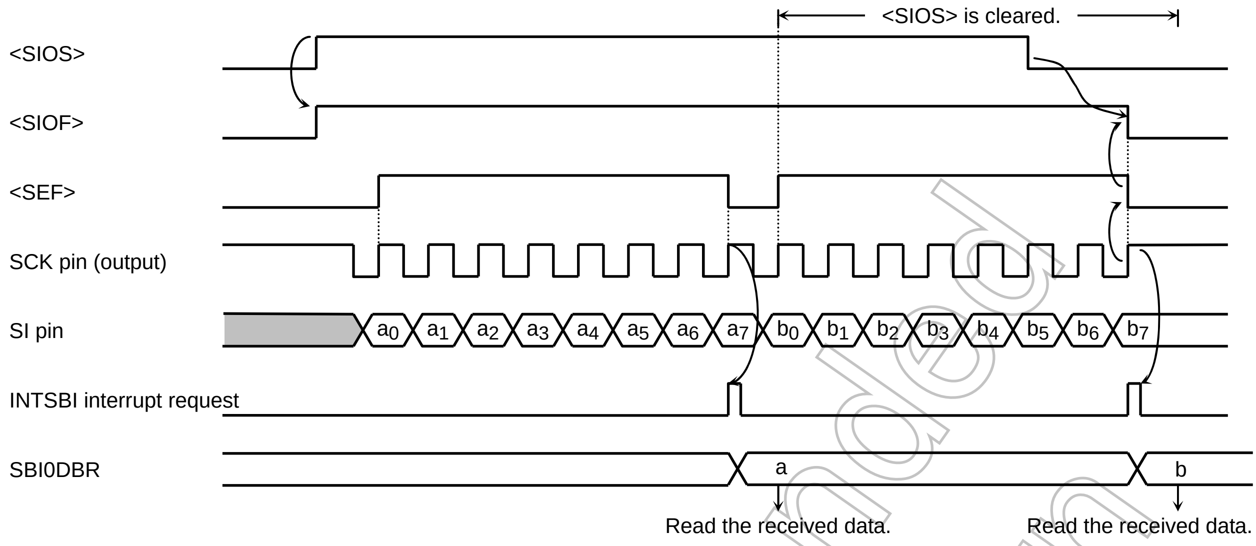


Fig. 14.7.2.3 Receive Mode (Example: Internal Clock)

③ 8-bit transmit/receive mode

Set the control register to the transfer/receive mode. Then writing the transmit data to SBI0DBR and setting SBI0CR1 <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 SBI0DBR and the INTSBI interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the next transmit data. Because SBI0DBR 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 SBI0CR1 <SIOINH> to "1" in the INTSBI interrupt service program. If <SIOS> is cleared, transmission and reception continue until the received data is fully transferred to SBI0DBR. The program checks SBI0SR <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 SBI0DBR will not be retained after the transfer mode is changed. The ongoing transmission and reception must be completed by clearing <SIOS> to "0" and the last received data must be read before the transfer mode is changed.

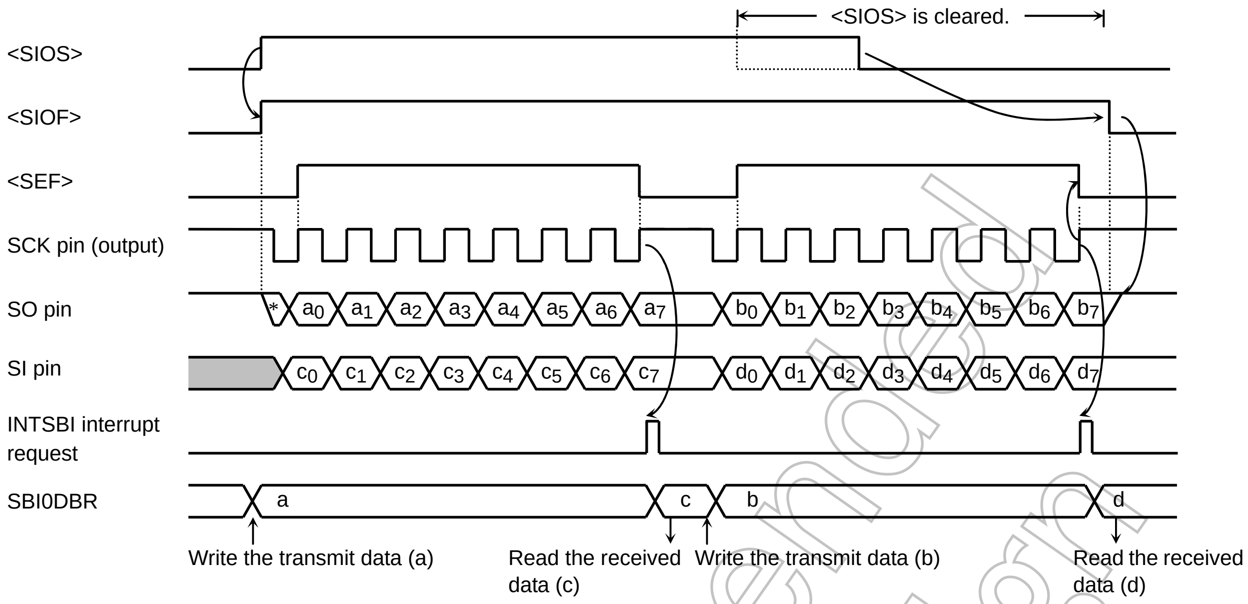


Fig. 14.7.2.4 Transmit/Receive Mode (Example: Internal Clock)

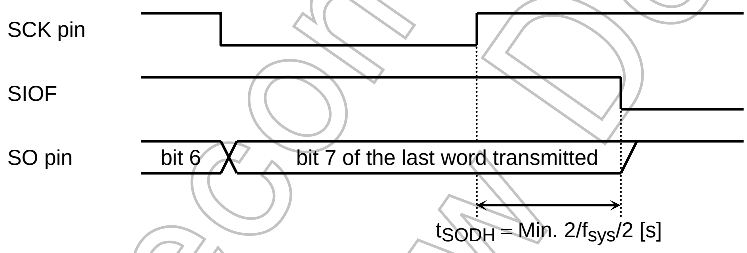


Fig. 14.7.2.5 Transmit Data Retention Time at the End of Transmission/Reception
(In the Transmit/Receive Mode)

```
                7 6 5 4 3 2 1 0
SBI0CR1   ←    0 1 1 0 0 X X X       Selects the transmit mode.

SBI0DBR   ←    X X X X X X X X       Writes the transmit data.
SBI0CR1   ←    1 0 1 0 0 X X X       Starts reception/transmission.
```

INTSBI interrupt

```
Reg.      ← SBIODBR                  Reads the received data.
SBI0DBR   ←    X X X X X X X X       Writes the transmit data.
```

15. Analog/Digital Converter

Two 10-bit, sequential-conversion analog/digital converters (A/D converter) are built into the TMP19A63. These A/D converters are equipped with 16 analog input channels. These units operate independently and offer identical performance, so only unit A is described here.

Fig. 15.1 shows the block diagram of this A/D converter.

These 16 analog input channels (pins ANA0 through AN15) are also used as input ports.

(Note)	**If it is necessary to reduce a power current by operating the TMP19A63 in IDLE 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 TMP19A63 into standby mode:**
1)	**The TMP19A63 must be put into IDLE mode when ADMOD1<I2AD> is "0."**
2)	**The TMP19A63 must be put into STOP mode.**

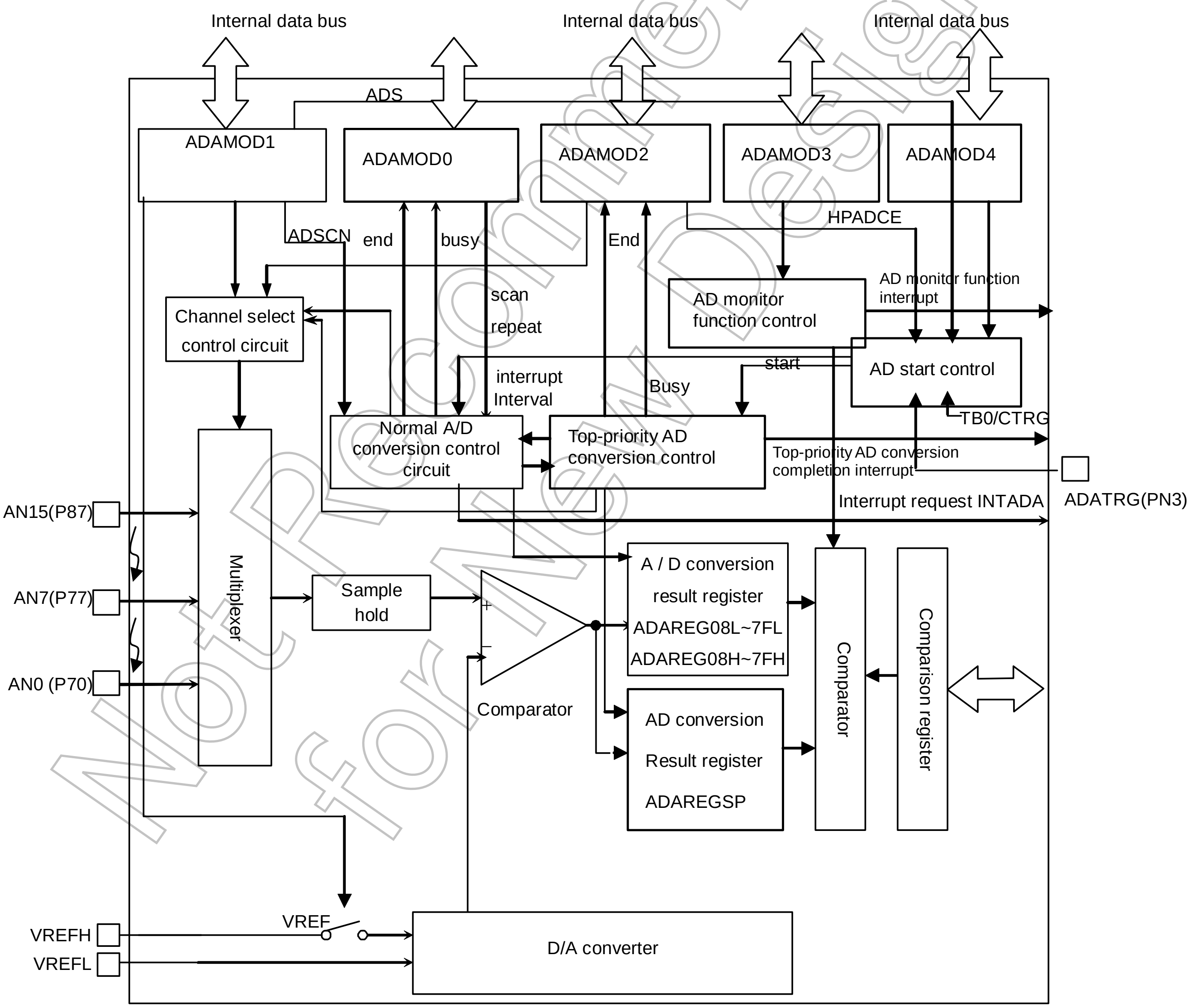


Fig. 15.1 A/D Converter Block Diagram

15.1 Control Register

The A/D converter is controlled by A/D mode control registers (ADAMOD0, ADAMOD1, ADAMOD2, ADAMOD3 and ADAMOD4). Results of A/D conversion are stored in 16 upper and lower A/D conversion result registers ADAREG08H/L through ADAREG7FH/L. Results of top-priority conversion are stored in ADAREGSPH/L.

Fig. 15.2 shows the registers related to the A/D converter.

A/D Mode Control Register 0

ADAMOD0 (0xFFFF_F814)

	7	6	5	4	3	2	1	0
bit Symbol	EOCFN	ADBFN		ITM1	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R	R/W				
After reset	0	0	0	0	0	0	0	0
Function	Normal A/D conversion completion flag 0: Before or during conversion 1: Completion	Normal A/D conversion BUSY flag 0: Conversion stop 1: During conversion	"0" is read.	Specify interrupt in fixed channel repeat conversion mode	Specify interrupt in fixed channel repeat conversion mode	Specify repeat mode 0: Single conversion mode 1: Repeat conversion mode	Specify scan mode 0: Fixed channel mode 1: Channel scan mode	Start A/D conversion 0: Don't care 1: Start conversion "0" is always read.

Specify A/D conversion interrupt in fixed channel repeat conversion mode

	Fixed channel repeat conversion mode <SCAN> = "0", <REPEAT> = "1"
00	Generate interrupt once every single conversion
01	Generate interrupt once every 4 conversions
10	Generate interrupt once every 8 conversions
11	Setting prohibited

Fig.15.2 Registers related to the A/D Converter

Note: The following must be set before the A/D conversion to guarantee accuracy in the conversion.
0xFFFF_F819＝0x58

ADACBAS (0xFFFF_F819)

	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	Be sure to write "0".	Be sure to write "0".	Be sure to write "0".	Be sure to write "0".	Be sure to write "0".	Be sure to write "0".	Be sure to write "0".	Be sure to write "0".

A/D Mode Control Register 1

ADAMOD1
(0xFFFF_ F815)

	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 : Active	Specify operation mode for channel scanning 0: 4ch scan 1: 8ch scan	Write “0”.	Select analog input channel.			

Selection of analog input channel

<SCAN> / <ADCH3.2, 1, 0>	0 Fixed channel	1 Channel scanning (ADSCN=0)	1 Channel scanning (ADSCN=1)
0000	ANA0	ANA0	ANA0
0001	ANA1	ANA0~ANA1	ANA0~ANA1
0010	ANA2	ANA0~ANA2	ANA0~ANA2
0011	ANA3	ANA0~ANA3	ANA0~ANA3
0100	ANA4	ANA4	ANA0~ANA4
0101	ANA5	ANA4~ANA5	ANA0~ANA5
0110	ANA6	ANA4~ANA6	ANA0~ANA6
0111	ANA7	ANA4~ANA7	ANA0~ANA7
1000	ANA8	ANA8	ANA8
1001	ANA9	ANA8~ANA9	ANA8~ANA9
1010	ANA10	ANA8~ANA10	ANA8~ANA10
1011	ANA11	ANA8~ANA11	ANA8~ANA11
1100	ANA12	ANA12	ANA8~ANA12
1101	ANA13	ANA12~ANA13	ANA8~ANA13
1110	ANA14	ANA12~ANA14	ANA8~ANA14
1111	ANA15	ANA12~ANA15	ANA8~ANA15

(Note 1) **Before starting A/D 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 ADMAOD0<ADS> bit.**

(Note 2) **To go into standby mode upon completion of A/D conversion, set <VREFON> to "0."**

Fig.15.3 Registers related to the A/D Converter

A/D Mode Control Register 2

ADAMOD2
(0xFFFF_ F816)

	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	ANA0
0001	ANA1
0010	ANA2
0011	ANA3
0100	ANA4
0101	ANA5
0110	ANA6
0111	ANA7
1000	ANA8
1001	ANA9
1010	ANA10
1011	ANA11
1100	ANA12
1101	ANA13
1110	ANA14
1111	ANA15

A/D Mode Control Register 3

ADAMOD3 (0xFFFF _F817)

	7	6	5	4	3	2	1	0
bit Symbol			ADOBIC	REGS3	REGS2	REGS1	REGS0	ADOBSV
Read/Write	R/W	R	R/W					
After reset	0	0	0	0	0	0	0	0
Function	Write "0".	"0" is read.	Make AD monitor function interrupt setting 0: Smaller than comparison Register 1: Larger than comparison Register	BIT for selecting the AD conversion result storage Register that is to be compared with the comparison Register if the AD monitor function is enabled				AD monitoring function 0: Disable 1: Enable

<REGS.2, 1, 0>	AD conversion result storage Register to be compared
0000	ADAREG08
0001	ADAREG19
0010	ADAREG2A
0011	ADAREG3B
0100	ADAREG4C
0101	ADAREG5D
0110	ADAREG6E

A/D Mode Control Register 4

ADAMOD4 (0xFFFF_F818)

	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: Match with TB9RG0	HW for activating top-priority A/D conversion 0: Disable 1: Enable	HW source for activating normal A/D conversion 0: External TRG 1: Match with TB1RG0	HW for activating normal A/D conversion 0: Disable 1: Enable	"0" is read.		Overwriting 10 with 01 allows ADC to be software reset.	

(Note 1) If A/D 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 A/D conversion: <ADHTG>, <HADHTG>

③ Start the timer.

(Note 2) Do not make a top-priority A/D conversion setting and a normal A/D conversion setting simultaneously.

(Note 3) The external trigger cannot be specified as HW source for activating normal A/D conversion when it is specified as HW source for activating top-priority A/D conversion.

(Note 4) Software reset initializes all the bits of the mode registers. Therefore, resetting these values is required.

Lower A/D Conversion Result Register 08

		7	6	5	4	3	2	1	0
ADAREG08L (0xFFFF_F800)	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 generated 1: Generated	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 08

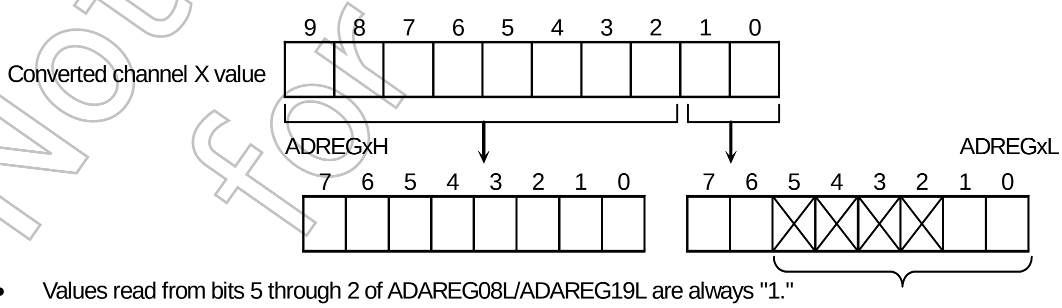
		7	6	5	4	3	2	1	0
ADAREG08H (0xFFFF_F801)	bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	0							
	Function	Store upper 8 bits of A/D conversion result.							

Lower A/D Conversion Result Register 19

		7	6	5	4	3	2	1	0
ADAREG19L (0xFFFF_F802)	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 generated 1: Generated	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 19

		7	6	5	4	3	2	1	0
ADAREG19H (0xFFFF_F803)	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 of ADAREG08L/ADAREG19L are always "1."
- Bit 0 of ADAREG08L/ADAREG19L 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 (ADAREGxL) will set this bit to "0."
- Bit 1 of ADAREG08L/ADAREG19L is the over RUN flag <OVRx>. This bit is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADAREGxH,ADAREGxL) are read. A read of a flag will clear this bit to "0."

Fig.15.4 Registers related to the A/D Converter

Lower A/D Conversion Result Register 2A

ADAREG2AL (0xFFFF_F804)

	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 generated. 1: Generated.	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 2A

ADAREG2AH (0xFFFF_F805)

	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

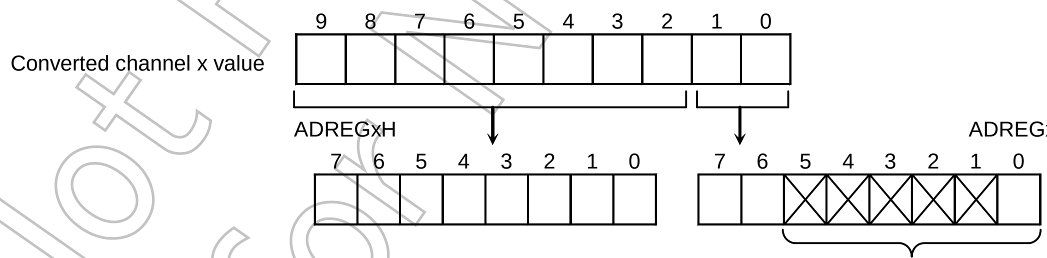
ADAREG3BL (0xFFFF_F806)

	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 generated 1: Generated	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 3B

ADAREG3BH (0xFFFF_F807)

	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 of ADAREG2AL/ADAREG3BL are always "1."
- Bit 0 of ADAREG2AL/ADAREG3BL 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 (ADAREGxL) will set this bit to "0."
- Bit 1 of ADAREG2AL/ADAREG3BL is the over RUN flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADAREGxH,ADAREGxL) 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.15.5 Registers related to the A/D Converter

Lower A/D Conversion Result Register 4C

ADAREG4CL (0xFFFF_F808)

	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 generated 1: Generated	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 4C

ADAREG4CH (0xFFFF_F809)

	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

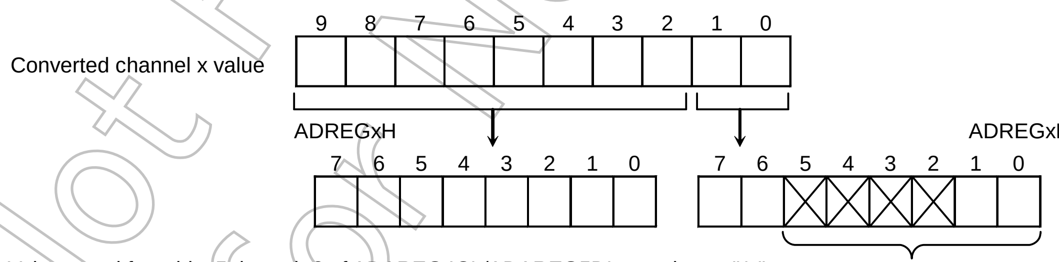
ADAREG5DL (0xFFFF_F80A)

	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 generated 1: Generated	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 5D

ADAREG5DH (0xFFFF_F80B)

	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 of ADAREG4CL/ADAREG5DL are always "1."
- Bit 0 of ADAREG4CL/ADAREG5DL 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 of ADAREG4CL/ADAREG5DL is the over Run flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADAREGxH,ADAREGxL) 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

ADAREG6EL (0xFFFF_F80C)

	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

ADAREG6EH (0xFFFF_F80D)

	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

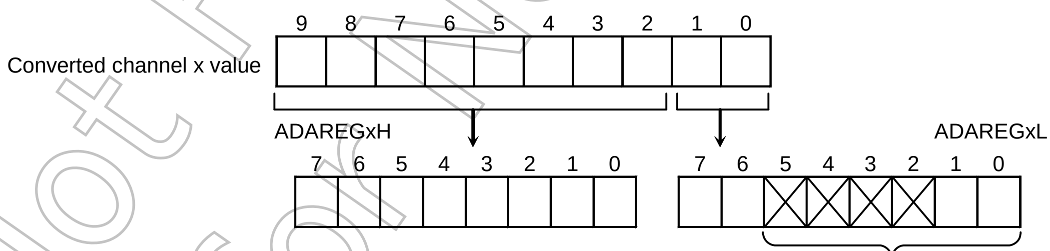
ADAREG7FL (0xFFFF_F80E)

	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 RUN flag 0: Not generate 1: Generate	A/D conversion result storage flag 1: Presence of conversion result

Upper A/D Conversion Result Register 7F

ADAREG7FH (0xFFFF_F80F)

	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							



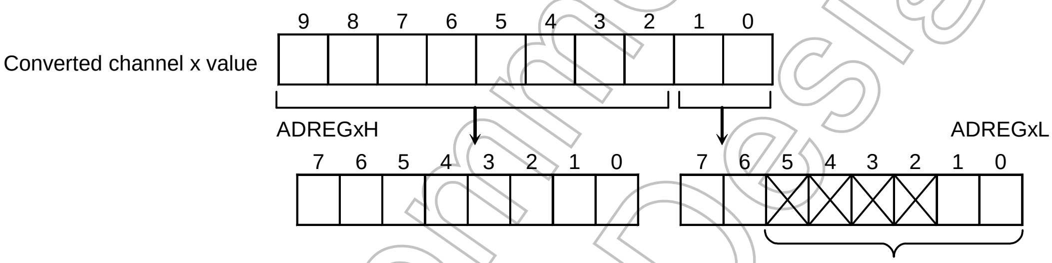
- Values read from bits 5 through 2 of ADAREG6EL/ADAREG7FL are always "1."
- Bit 0 of ADAREG6EL/ADAREG7FL is the A/D conversion result storage flag <ADRxRF>. It is set to "1" if an A/D converted value is stored. A read of a lower register (ADAREGxL) will set this bit to "0."
- Bit 1 of ADAREG6EL/ADAREG7FL is the over Run flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADAREGxH,ADAREGxL) 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

ADAREGSPL (0xFFFF_F810)		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

ADAREGSPH (0xFFFF_F811)		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 of ADAREGSPL are always "1."
- Bit 0 of ADAREGSPL 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 (ADAREGSPL) will set this bit to "0."
- Bit 1 of ADAREGSPL is the over RUN flag <OVRx>. It is set to "1" if a conversion result is overwritten before both conversion result storage registers (ADAREGxH,ADAREGxL) 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

ADACOMREGL (0xFFFF_F812)

	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

ADACOMREGH (0xFFFF_F813)

	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 (ADAMOD3<ADOBSV>＝"0").**

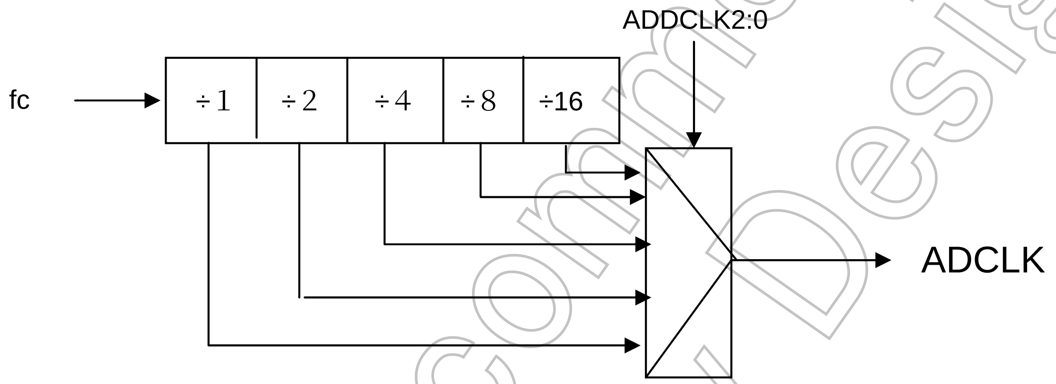
15.2 Conversion Clock

- The conversion time is calculated by the 46 conversion clock at the minimum (conversion clock ≤ 40MHz).

A/D Conversion Clock Setting Register

ADACLK (0xFFFF_F81C)

	7	6	5	4	3	2	1	0
bit Symbol	TSH3	tSH2	tSH1	tSH0		ADCLK2	ADCLK1	ADCLK0
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset	1	0	0	0	0	0	0	0
Function	Select the A/D sample hold time 1000: 8 conversion clock 1001: 16 conversion clock 1010: 24 conversion clock 1011: 32 conversion clock 0011: 64 conversion clock 1100: 128 conversion clock 1101: 512 conversion clock Other than the above: reserved				"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	prescaler	Tconv. (Conversion time)
40MHz	1	1.15μs
	1/2	2.3μs
	1/4	4.6μs

Variable S/H time

Conversion clock	S/H time	Tconv. (Conversion time)
40MHz	Conversion clk*8(0.2us)	1.15μs
	Conversion clk*16(0.4us)	1.35μs
	Conversion clk*24(0.6us)	1.55μs
	Conversion clk*32(0.8us)	1.75μs
	Conversion clk*64(1.6us)	2.55μs
	Conversion clk*128 (3.2us)	4.15μs
	Conversion clk*512 (12.8us)	13.75μs

Example: If fsys = fc = 54 MHz (46 conversion clock at the minimum)

fc	prescaler	tconv.(Conversion time)
54MHz	1	**No setting available (Note)**
	1/2	1.7us
	1/4	3.4us

Note) The maximum conversion clock is 40MHz.

Variable S/H time

Conversion clock	S/H time	tconv.(conversion time)
27MHz (fc=54)	Conversion clk*8 (0.3us)	1.7us
	Conversion clk*16 (0.6us)	2.0us
	Conversion clk*24 (0.9us)	2.3us
	Conversion clk*32 (1.2us)	2.6us
	Conversion clk*64 (2.4us)	3.8us
	Conversion clk*128 (4.7us)	6.1us
	Conversion clk*512 (19.0us)	20.4us

ADCLK & conversion time per typical oscillators

fosc (MHz)	**fsys (MHz)**		**ADCLK (MHz)**	**tconv (conversion time) (46ADCLK)(uSec)**	**Note**
13.5	54.0		**xxxx**	**xxxx**	**No setting for ADC is available**
10.0	40.0		40.0	1.15	Gear 1/1
8.0	32.0		32.0	1.44	Gear 1/1
13.5	27.0		27.0	1.70	Clock gear 1/2
10.0	20.0		20.0	2.30	Clock gear 1/2
8.0	16.0		16.0	2.88	Clock gear1/2
13.5	13.5		13.5	3.41	Clock gear 1/4
10.0	10.0		10.0	4.60	Clock gear 1/4
8.0	8.0		8.0	5.75	Clock gear 1/4

* We specify 8.0 MHz, 10.0 MHz and 13.5 MHz as the typical oscillators.

"Please do not change the analog to digital conversion clock setting in the analog to digital translation.

15.3 Description of Operations

15.3.1 Analog Reference Voltage

The "H" level of the analog reference voltage shall be applied to the VREFH pin, and the "L" level shall be applied to the VREFL pin. By writing "0" to the ADAMOD1<VREFON> bit, a switched-on state of VREFH－VREFL can be turned into a switched-off state. To start A/D conversion, make sure that you first write "1" to the <VREFON> bit, wait for 3 μs during which time the internal reference voltage should stabilize, and then write "1" to the ADMOD0<ADS> bit.

15.3.2 Selecting the Analog Input Channel

How the analog input channel is selected is different depending on A/D converter operation mode used.

(1) Normal A/D conversion mode

- If the analog input channel is used in a fixed state (ADAMOD0<SCAN>="0"):

One channel is selected from analog input pins AINA0 through AINA15 by setting ADAMOD1<ADCH3 to 0> to an appropriate setting.

- If the analog input channel is used in a scan state (ADAMOD0<SCAN>="1"):

  One scan mode is selected from 16 scan modes by setting ADAMOD1 <ADCH3 to 0> and ADSCN to appropriate settings.

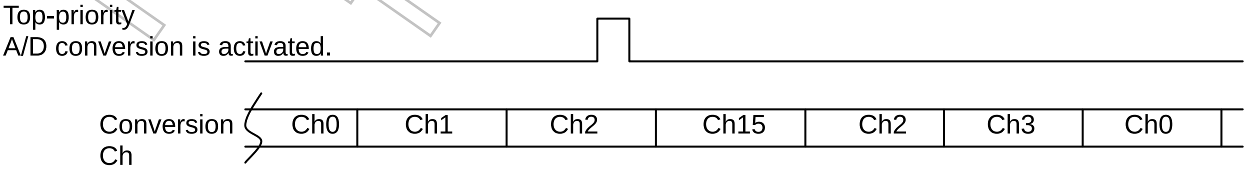
(2) Top-priority A/D conversion mode

One channel is selected from analog input pins AINA0 through AINA15 by setting ADAMOD2<HPADCH3 to 0> to an appropriate setting.

After a reset, ADMOD0<SCAN> is initialized to "0" and ADAMOD1<ADCH3:0> is initialized to "0000." This initialization works as a trigger to select a fixed channel input through the ANA0 pin. The pins that are not used as analog input channels can be used as ordinary input ports.

If top-priority A/D conversion is activated during normal A/D conversion, normal A/D conversion is discontinued, top-priority A/D conversion is executed and completed, and then normal A/D conversion is resumed.

Example: A case in which repeat-scan conversion is ongoing at channels AINA0 through AINA3 with ADAMOD0<REPEAT：SCAN> set to "11" and ADAMOD1<ADCH3:0> set to 0011, and top-priority A/D conversion has been activated at AINA15 with ADAMOD2<HPADCH3:0>=1111:



15.3.3 Starting A/D Conversion

Two types of A/D conversion are supported: normal A/D conversion and top-priority A/D conversion. Normal A/D conversion is software activated by setting ADAMOD0<ADS> to "1." Top-priority A/D conversion is software activated by setting ADAMOD2<HPADCE> to "1." 4 operation modes are made available to normal A/D conversion. In performing normal A/D conversion, one of these operation modes must be selected by setting ADAMOD0<2:1> to an appropriate setting. For top-priority A/D conversion, only one operation mode can be used: fixed channel single conversion mode. Normal A/D conversion can be activated using the HW activation source selected by ADAMOD4<ADHS>, and top-priority A/D conversion can be activated using the HW activation source selected by ADAMOD4<HADHS>. If this bit is "0," normal and top-priority A/D conversions are activated in response to the input of a falling edge through the $\overline{\text{ADTRG}}$ pin. If this bit is "1," normal A/D conversion is activated in response to TB1RG0 generated by the 16-bit timer 1, and top-priority A/D conversion is activated in response to TB9RG0 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 A/D conversion, an external trigger cannot usually be set for HW activation of A/D conversion.

When normal A/D conversion starts, the A/D conversion Busy flag (ADAMOD0<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 (ADAMOD2<ADBFHP>) showing that A/D conversion is under way is set to "1." At that time, the Busy flag for normal A/D conversion retains the value that had been set before top-priority conversion started. 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 reactivated when top-priority A/D conversion is under way. Otherwise, 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 (ADAMOD4<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 ADAMOD2<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 ADAREGSP, 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 ADAMOD2<3:0>. After the result of this top-priority A/D conversion is stored in the storage register ADAREGSP, normal A/D conversion is resumed.

15.3.4 A/D Conversion Modes and A/D Conversion Completion Interrupts

For A/D conversion, the following four operation modes are supported. For normal A/D conversion, an operation mode can be selected by setting ADAMOD0<2：1> to an appropriate setting. For top-priority A/D conversion, the fixed channel single conversion mode is automatically selected, irrespective of the ADAMOD0<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, ADAMOD0<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 ADAMOD0 <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, ADAMOD0<EOCF> is set to "1," ADAMOD0<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 ADAMOD0 <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, ADAMOD0<EOCF> is set to "1," ADAMOD0<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 ADAMOD0<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, ADAMOD <EOCF> is set to "1." ADAMOD0 <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 ADAMOD0 <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 ADAREG08. 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 ADAREG08 through ADAREG3B. After the conversion results are stored in ADAREG3B, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADAREG08. <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 ADAREG08 through ADAREG7F. After the conversion results are stored in ADAREG7F, <EOCF> is set to "1," and the storage of subsequent conversion results starts from ADAREG08.

④ Channel scan repeat conversion mode

If ADAMOD0 <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, ADAMOD0 <EOCF> is set to "1," and the interrupt request INTAD is generated. ADAMOD0 <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 ADAMOD0<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 ADAMOD2<HPADCH3:0>. After the A/D conversion is completed, the top-priority A/D conversion completion interrupt is generated, ADAMOD 2<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)	ADAMOD0		
				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	Each time one scan conversion is completed	1	——	1	1

(Note) EOCF is cleared upon read.

15.3.5 High-priority Conversion Mode

By interrupting ongoing normal A/D conversion, top-priority A/D conversion can be performed. Top-priority A/D conversion can be software activated by setting ADAMOD2<HPADCE> to "1" or it can be activated using the HW resource by setting ADAMOD4<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 ADAMOD2<3:0>. The result of single conversion is stored in ADAREGSP, 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 ANA0 through ANA8 and if <HPADCE> is set to "1" during ANA3 conversion, ANA3 conversion is suspended, and conversion is performed for a channel designated by <HPADC3:0>. After the result of conversion is stored in ADAREGSP, channel repeat conversion is resumed, starting from ANA3.

15.3.6 A/D Monitor Function

If ADAMOD3<ADOBSV> is set to "1," the A/D monitor function is enabled. If the value of the conversion result storage register specified by REGS<3:0> becomes larger or smaller ("larger" or "smaller" to be designated by ADOBIC) than the value of a comparison register, the A/D monitor function interrupt is generated. This comparison operation is performed each time a result is stored in a corresponding conversion result storage register, and the interrupt is generated if the conditions are met. Because storage registers assigned to perform the A/D monitor function are usually not read by software, overrun flag <OVRn> is always set and the conversion result storage flag <ADRnRF> is also set. To use the A/D monitor function, therefore, a flag of a corresponding conversion result storage register must not be used.

15.3.7 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 (ADAREG08H/L through ADARG7FH/L).

In fixed channel repeat conversion mode, A/D conversion results are sequentially stored in ADAREG08H/L through ADAREG7FH/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 ADAREG08H/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 ADAREG08H/L through ADAREG3BH/L.

Table 15.1 shows analog input channels and related A/D conversion result registers.

Table 15.1 Analog Input Channels and Related A/D Conversion Result Registers

<table>
<tr><th rowspan="2">Analog input channel (port A)</th><th colspan="4">A/D conversion result register</th></tr>
<tr><th>Conversion modes other than shown to the right</th><th>Fixed channel repeat conversion mode (every one conversion)</th><th>Fixed channel repeat conversion mode (every four conversions)</th><th>Fixed channel repeat conversion mode (every eight conversions)</th></tr>
<tr><td>ANA0</td><td>ADAREG08H/L</td><td rowspan="16">ADAREG08H/L fixed</td><td rowspan="16">ADAREG08H/
↓
↓
ADAREG3BH/</td><td rowspan="16">ADAREG08H/
↓
↓
↓
ADAREG7FH/</td></tr>
<tr><td>ANA1</td><td>ADAREG19H/L</td></tr>
<tr><td>ANA2</td><td>ADAREG2AH/L</td></tr>
<tr><td>ANA3</td><td>ADAREG3BH/L</td></tr>
<tr><td>ANA4</td><td>ADAREG4CH/L</td></tr>
<tr><td>ANA5</td><td>ADAREG5DH/L</td></tr>
<tr><td>ANA6</td><td>ADAREG6EH/L</td></tr>
<tr><td>ANA7</td><td>ADAREG7FH/L</td></tr>
<tr><td>ANA8</td><td>ADAREG08H/L</td></tr>
<tr><td>ANA9</td><td>ADAREG19H/L</td></tr>
<tr><td>ANA10</td><td>ADAREG2AH/L</td></tr>
<tr><td>ANA11</td><td>ADAREG3BH/L</td></tr>
<tr><td>ANA12</td><td>ADAREG4CH/L</td></tr>
<tr><td>ANA13</td><td>ADAREG5DH/L</td></tr>
<tr><td>ANA14</td><td>ADAREG6EH/L</td></tr>
<tr><td>ANA15</td><td>ADAREG7FH/L</td></tr>
</table>

15.3.8 Data Polling

To process A/D conversion results without using interrupts, ADAMOD0<EOCF> must be polled. If this flag is set, conversion results are stored in a specified A/D conversion result register. After confirming that this flag is set, read that conversion result storage register. In reading the register, make sure that you first read upper bits and then lower bits to detect an overrun. If OVRn is "0" and ADRnRF is "1" in lower bits, a correct conversion result has been obtained.

16. Watchdog Timer (Runaway Detection Timer)

The TMP19A63 has a built-in watchdog timer for detecting runaways.

The watchdog timer (WDT) is for detecting malfunctions (runaways) of the CPU caused by noises or other disturbances and remedying them to return the CPU to normal operation. If the timer detects a runaway, it generates a non-maskable interrupt to notify the CPU.

By connecting the output of the watchdog timer to a reset pin (inside the chip), it is possible to force the watchdog timer to reset itself.

16.1 Configuration

Fig. 16.1 shows the block diagram of the watchdog timer.

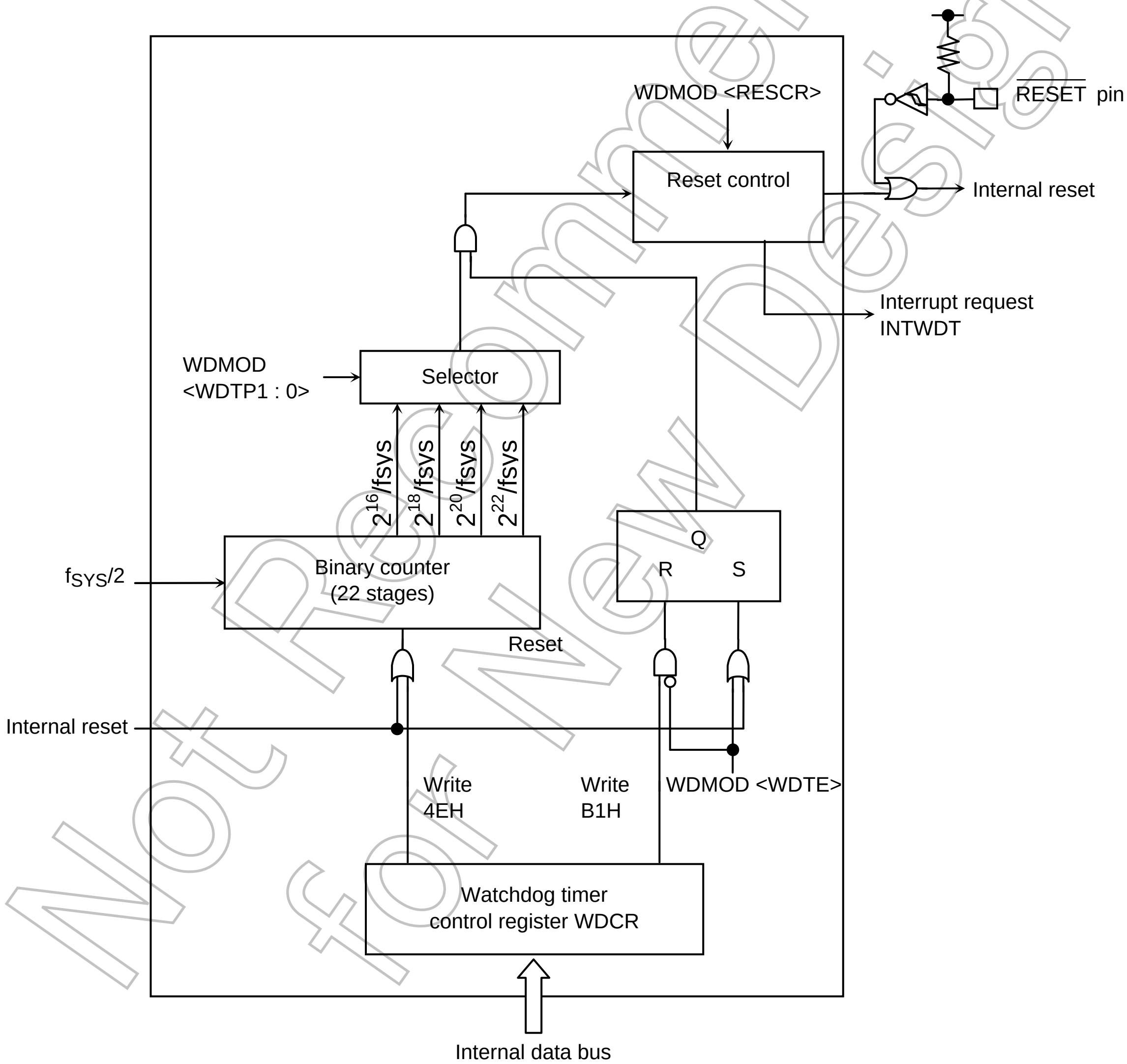


Fig. 16.1 Block Diagram of the Watchdog Timer

16.2 Watchdog Timer Interrupt

The watchdog timer consists of the binary counters that are arranged in 22 stages and work using the $f_{SYS/2}$ system clock as an input clock. The outputs produced by these binary counters are 2^{15}, 2^{17}, 2^{19} and 2^{21}. By selecting one of these outputs with WDMOD <WDTP1:0>, a watchdog timer interrupt can be generated when an overflow occurs, as shown in Fig. 16.2.1.

Because the watchdog timer interrupt is a non-maskable interrupt factor, NMIFLG <WDT> at the INTC performs a task of identifying it.

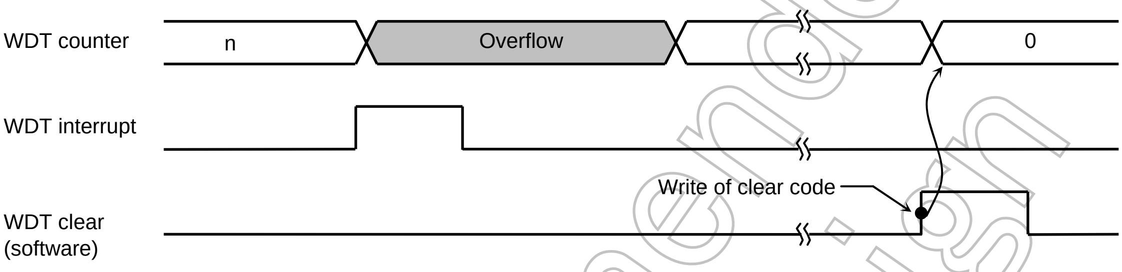


Fig. 16.2.1 Normal Mode

When an overflow occurs, resetting the chip itself is an option to choose. If the chip is reset, a reset is effected for a 32-state time, as shown in Fig. 16.2.2. If this reset is effected, the clock f_{SYS} that the clock gear generates by dividing the clock f_C of the high-speed oscillator by 8 is used as an input clock $f_{SYS/2}$.

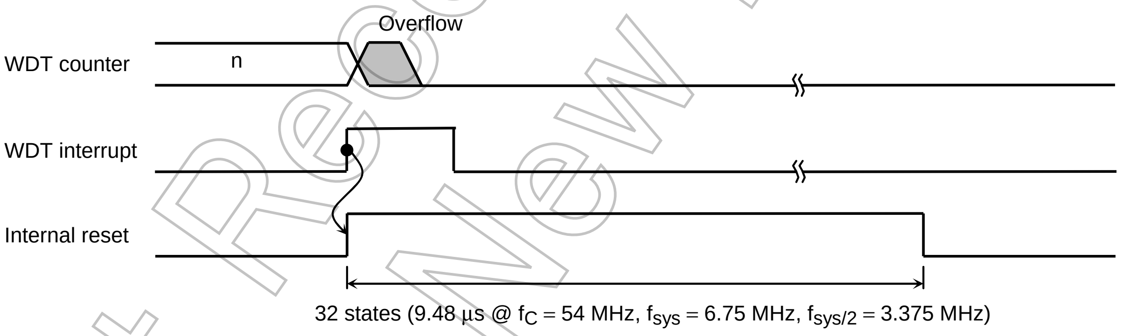


Fig. 16.2.2 Reset Mode

16.3 Control Registers

The watchdog timer (WDT) is controlled by two control registers WDMOD and WDCR.

16.3.1 Watchdog Timer Mode Register (WDMOD)

① Specifying the detection time of the watchdog timer <WDTP1: 0>

This is a 2-bit register for specifying the watchdog timer interrupt time for runaway detection. When a reset is effected, this register is initialized to WDMOD <WDTP1, 0> = "00." 16.3.1.1 shows the detection time of the watchdog timer.

② Enabling/disabling the watchdog timer <WDTE>

When reset, WDMOD <WDTE> is initialized to "1" and the watchdog timer is enabled.

To disable the watchdog timer, this bit must be set to "0" and, at the same time, the disable code (B1H) must be written to the WDCR register. This dual setting is intended to minimize the probability that the watchdog timer may inadvertently be disabled if a runaway occurs.

To change the status of the watchdog timer from "disable" to "enable," set the <WDTE> bit to "1."

③ Watchdog timer out reset connection <RESCR>

This is a register for specifying the connection of non-maskable interrupt (INTWDT) or an internal reset after a runaway is detected. As a reset initializes this setting to WDMOD <RESCR>="0" and non-maskable interrupt is specified. Refer to the part of NMIFLG register in Chapter 6 "Interrupt".

WDMOD (0xFFFF_FA00)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	—
Read/Write	R/W	R/W		R	R	R/W		R/W
After reset	1	0	0	0	0	0	0	0
Function	WDT control 0: Disable 1: Enable	Selects WDT detection time 00: $2^{16}/f_{SYS}$ 01: $2^{18}/f_{SYS}$ 10: $2^{20}/f_{SYS}$ 11: $2^{22}/f_{SYS}$				IDLE 0: Stop 1: Start	Selects internal RESET 0:NMI connection 1: Internal reset connection	Write “0”.

Watchdog timer out control

0	NMI interrupt
1	Connects WDT out to reset

Detection time of watchdog timer @ fc=54 MHz

SYSCR1 clock gear value <GEAR2:0>	Watch Dog Timer detection time			
	WDMOD<WDTP1, 0>			
	00	01	10	11
000 (fc)	1.2 ms	4.9 ms	19.4 ms	77.7 ms
100 (fc/2)	2.4 ms	9.7 ms	38.8 ms	155.3 ms
110 (fc/4)	4.9 ms	19.4 ms	77.7 ms	310.7 ms
111 (fc/8)	9.7 ms	38.8 ms	155.3 ms	621.4 ms

Enable/disable control of the watchdog timer

0	Disable
1	Enable

Fig. 16.3.1.1 Watchdog Timer Mode Register

16.3.2 Watchdog Timer Control Register (WDCR)

This is a register for disabling the watchdog timer function and controlling the clearing function of the binary counter.

- Disabling control

By writing the disable code (B1H) to this WDCR register after setting WDMOD <WDTE> to "0," the watchdog timer can be disabled.

WDMOD	←	0 − − − − − − −	Clears WDTE to "0".
WDCR	←	1 0 1 1 0 0 0 1	Writes the disable code (B1H).

- Enable 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 clear code (4EH).

(Note) Writing the disable code (BIH) clears the binary counter.

WDCR (0xFFFF_FA01)

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	—							
Function	B1H : WDT disable code 4EH : WDT clear code Others: Invalid This register is exclusively for writing. Each bit is read as "0".							

→ Disable & clear of WDT

B1H	Disable code
4EH	Clear code
Others	—

Fig. 16.3.2.1 Watchdog Timer Control Register

16.4 Operation Description

The watchdog timer generates the 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).

17. ROM correction function

This chapter describes the ROM correction function built into the TMP19A63.

17.1 Features

- Using this function, eight-word data per one register can be replaced for 12 registers.
- If an address (lower 5 bit is "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.

17.2 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>. Internally the data replacement is executed after the match detection of the ROMCS signal showing a ROM area and 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	ADDREGn and RAM area	Number of words
ADDREG0	0xFFFF_DE80 - 0xFFFF_DE9F	8
ADDREG1	0xFFFF_DEA0 - 0xFFFF_DEBF	8
ADDREG2	0xFFFF_DEC0 - 0xFFFF_DEDF	8
ADDREG3	0xFFFF_DEE0 - 0xFFFF_DEFF	8
ADDREG4	0xFFFF_DF00 - 0xFFFF_DF1F	8
ADDREG5	0xFFFF_DF20 - 0xFFFF_DF3F	8
ADDREG6	0xFFFF_DF40 - 0xFFFF_DF5F	8
ADDREG7	0xFFFF_DF60 - 0xFFFF_DF7F	8
ADDREG8	0xFFFF_DF80 - 0xFFFF_DF9F	8
ADDREG9	0xFFFF_DFA0 - 0xFFFF_DFBF	8
ADDREGA	0xFFFF_DFC0 - 0xFFFF_DFDF	8
ADDREGB	0xFFFF_DFE0 - 0xFFFF_DFFF	8

(Note 1): The instruction affected by ROM correction under ROM protection is activated by RAM. Therefore, the instruction corrected by ROM can specify neither conditions of the ROM reading nor the DMAC. To enable all the instructions, the ROM must be unprotected.

(Note 2) ROM correction to ROM area ignores the upper address specified by an address register and decodes the address [19:5].

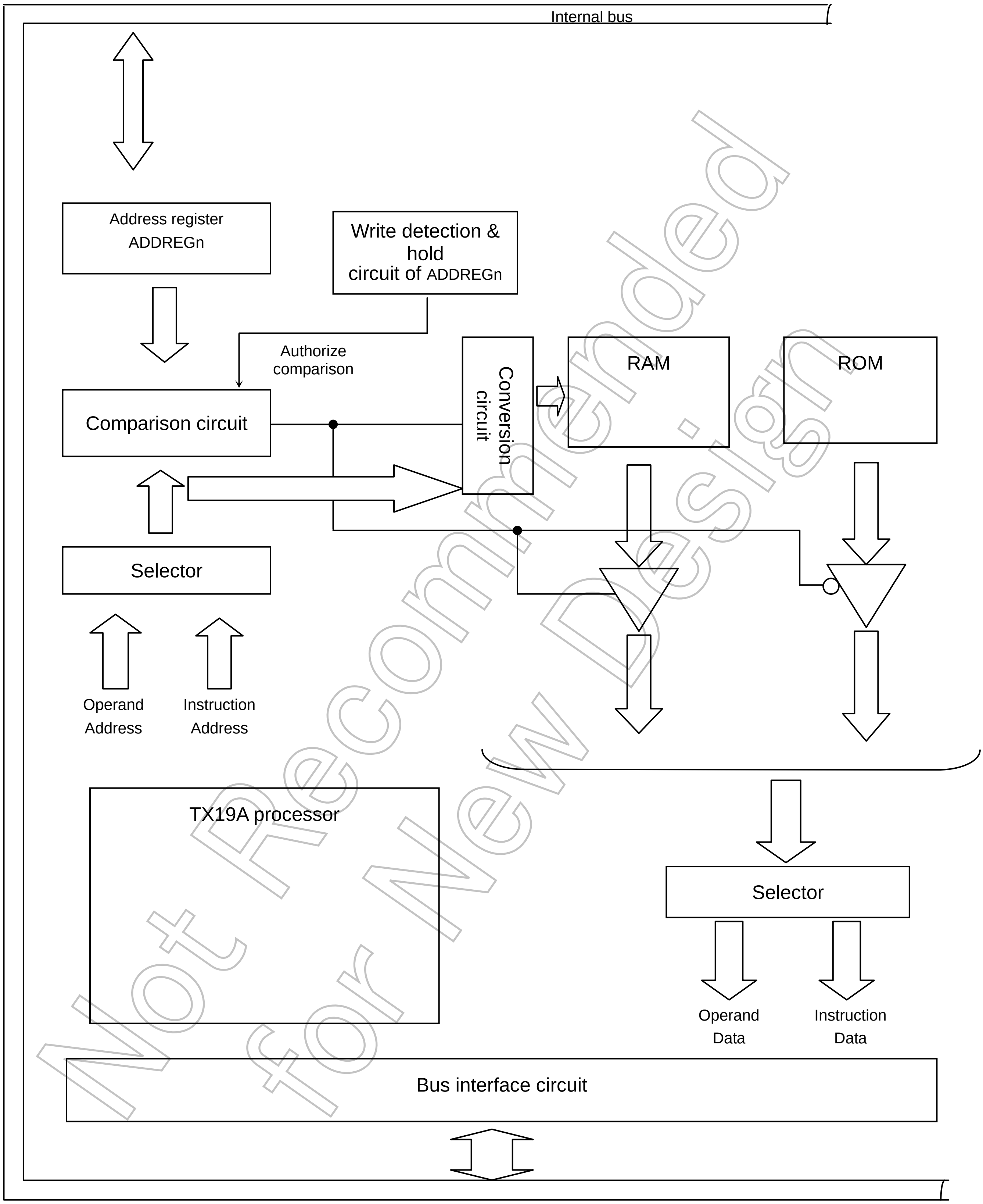


Fig. 17.1 ROM correction system diagram

17.3 Registers

(1) Address registers

ADDREG0 (0xFFFF_E540)

	7	6	5	4	3	2	1	0
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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
Symbol								
Read/Write	R/W			R				
After reset	0	0	0	1	1	1	1	1
	15	14	13	12	11	10	9	8
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16
Symbol								
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
	31	30	29	28	27	26	25	24
Symbol								
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.**

18. Key-on Wake Up

18.1 Outline

- The TMP19A63 has 8 key inputs, KEY0 to KEY7, 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 8 inputs. Each key input can be configured to be used or not, by programming (KWUPSTn).
- The active state of each input can be configured to the rising edge, the falling edge, the high level or the low level, by programming (KWUPSTn).
- An interrupt request is cleared by programming the key interrupt status register KWUPST in the interrupt processing.
- The key input pins have pull-up functions, which can enable/disable the pull-up function by programming the key pull-up control register PKPUP. This programming is needed for each of 32 inputs.

18.2 Key-on Wakeup Operation

The TMP19A63 has 8 key input pins, KEY0 to KEY7. Program the IMCGD<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, KEY0 to KEY7, 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<KEYn1: KEYn0> to define the active state of each key input pin to be used. Detection of key inputs is carried out in the KWUP block, and the detection results are notified to the IMCGD register in the CG as the active high level. Therefore, program IMCGD<EMCGC1:C0> to "01" to determine the detection level to the high level. The results of detection in the CG are also notified to the interrupt controller INTC as the active high level. Therefore, program the INTC to "01" to define the corresponding interrupt as the high level. Setting IMCGD<KWUPEN> to 0 (default) configures all the input pins, KEY0 to KEY7 to the normal interrupts. In this case, you don't have to make settings at the CG, but just specify the INTC detection level to the high level. Program KWUPSTn in the same way to enable or disable each key input and define their active states. Reading KWUPST during interrupt processing clears all the key interrupt requests.

(Note) If two or more key inputs are generated, all the key input requests will be cleared by clearing the interrupt request that corresponds to the first key input. The interrupt request generated after the interrupt to be cleared produces another key interrupt.

18.3 Pull-up function

Each key input has the pull-up function. By setting PKPUP<KEYPUP0:7> to "1", each bit of key input KEY0 through KEY7 can be pulled up.

Cautions on Use of Key Inputs With Pull-up Enabled

A) When you make the first setting after turning the power ON

1) Make a setting of PKPUP (<PKnUP>="1")
2) Set KWUPSTn<KEYnEN> to "1" for the key input to be used.
3) Wait until the pull-up operation is completed.
4) Set KWUPSTn that corresponds to KEYn input to define the active state of the key input to be used.
5) Read KWUPST to clear interrupt requests.
6) Set CG and INTC (see chapter 6 for the procedure).

B) To change the active state of a key input during operation

1) Disable key interrupts by setting IMC3<ILD2:D0> to "000" at the INTC.
2) Change the active state by setting KWUPSTn that corresponds to KEYn input to be changed.
3) Clear interrupt requests by reading KWUPST.
4) Enable the key interrupt at the INTC. Set IMC3<ILD2:D0> to a desired level.

C) To enable a key input during operation

1) Disable key interrupts by setting IMC3<ILD2:D0> to "000" at the INTC.
2) Set KWUPSTn<KEYnEN> to "1" for the KEYn input to be used.
3) Wait until the pull-up operation is completed.
4) Set the active state by setting KWUPSTn that corresponds to KEYn input to be used.
5) Read KWUPST to clear interrupt requests.
6) Enable key interrupts at the INTC. (Set IMC3<ILD2:D0> to a desired level).

Cautions on Use of Key Inputs With Pull-up Disabled

A) When you make the first setting after turning the power ON

1) Set PKPUP (<PKnUP>="1").
2) Set the active state by setting KWUPSTn that corresponds to KEYn input to be used.
3) Clear interrupt requests by reading KWUPST.
4) Set KWUPSTn<KEYnEN> to "1" for the KEYn input to be used.
5) Set CG and INTC (see chapter 6 for the procedure).

B) To change the active state of a key input during operation

1) Disable key interrupts at the INTC (IMC3<ILD2:D0>=000).
2) Change the active state by setting KWUPSTn that corresponds to KEYn input to be changed.
3) Read KWUPST to clear interrupt requests.
4) Enable key interrupts at the INTC. (Set IMC3<ILD2:D0> to a desired level).

C) To enable a key input during operation

1) Disable key interrupts at the INTC (IMC3<ILD2:D0>=000).
2) Set the active state by setting KWUPSTn that corresponds to KEYn input to be used.
3) Read KWUPST to clear interrupt requests.
4) Set KWUPSTn<KEYnEN> to "1" for the KEYn input to be used.
5) Enable key interrupts at the INTC (IMC3<ILD2:D0> to a desired level).

Port K Pull-up Control Register PKPUP

		7	6	5	4	3	2	1	0
PKPUP (0xFFFF_F14B)	Bit Symbol	PK7UP	PK6UP	PK5UP	PK4UP	PK3UP	PK2UP	PK1UP	PK0UP
	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

18.4 KEY input detection

1) <KWUPSTn> Active State Definition

   You can choose one of the active state of each KEYn input from H/L level or rising/ falling edge with KWUPSTn<KEYn1:0>. KEYn input active state detection is always in operation.

KWUPST0 (0xFFFF_F900)

	7	6	5	4	3	2	1	0
bit Symbol			KEY01	KEY00				KEY0EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function			Define the KEY0 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY0 interrupt input 0: Disable 1: Enable

KWUPST1 (0xFFFF_F901)

	7	6	5	4	3	2	1	0
bit Symbol			KEY11	KEY10				KEY1EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function			Define the KEY1 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY1 interrupt input 0: Disable 1: Enable

KWUPST2 (0xFFFF_F902)

	7	6	5	4	3	2	1	0
bit Symbol			KEY21	KEY20				KEY2EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function			Define the KEY2 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY2 interrupt input 0: Disable 1: Enable

KWUPST3 (0xFFFF_F903)

	7	6	5	4	3	2	1	0
bit Symbol			KEY31	KEY30				KEY3EN
Read/Write	R		R/W		R			R/W
After reset	0	0	1	0	0	0	0	0
Function			Define the KEY3 active state 00: "L" level 01: "H" level 10: Falling edge 11: Rising edge					KEY3 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST4	bit Symbol			KEY41	KEY40				KEY4EN
(0xFFFF_F904)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY0 active state 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge					KEY4 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST5	bit Symbol			KEY51	KEY50				KEY5EN
(0xFFFF_F905)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY5 active state 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge					KEY5 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST6	bit Symbol			KEY61	KEY60				KEY6EN
(0xFFFF_F906)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY6 active state 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge					KEY6 interrupt input 0: Disable 1: Enable

		7	6	5	4	3	2	1	0
KWUPST7	bit Symbol			KEY71	KEY70				KEY7EN
(0xFFFF_F907)	Read/Write	R		R/W		R			R/W
	After reset	0	0	1	0	0	0	0	0
	Function			Define the KEY7 active state 00: “L” level 01: “H” level 10: Falling edge 11: Rising edge					KEY7 interrupt input 0: Disable 1: Enable

18.5 Detection of Key Input Interrupts and Clearance of Requests

When KEYnEN is set to 1 and an active signal is input to KEYn, the KEYINTn channel that corresponds to KWUPST is set to "1," indicating that an interrupt is generated. The KWUPST is the read-only register. Reading this register clears the corresponding bit that has been set to "1".

If the active state is set to the high or low level, the corresponding bit of the KWUPINTn register remains "1" after it is read, unless the external input is withdrawn.

KEY Interrupt Status Register: KWUPST

		7	6	5	4	3	2	1	0
KWUPST	bit Symbol	KEYINT7	KEYINT6	KEYINT5	KEYINT4	KEYINT3	KEYINT2	KEYINT1	KEYINT0
(0xFFFF_F910)	Read/Write	R							
	After reset	0	0	0	0	0	0	0	0
	Function	KEY7 Interrupt 0::Not generated 1:Generated	KEY6 Interrupt 0::Not generated 1:Generated	KEY5 Interrupt 0::Not generated 1:Generated	KEY4 Interrupt 0::Not generated 1:Generated	KEY3 Interrupt 0::Not generated 1:Generated	KEY2 Interrupt 0::Not generated 1:Generated	KEY1 Interrupt 0::Not generated 1:Generated	KEY0 Interrupt 0::Not generated 1:Generated

19. 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] 32-bit timer（TMRC）

[4] I²CBUS/ serial channel (SBI)

[5] UART/ serial channel (SIO/UART)

[6] 10-bit A/D converter(ADC)

[7] Key On Wake-up (KWUP)

[8] Watchdog timer (WDT)

[9] Interrupt controller(INTC)

[10] DMA controller (DMAC)

[11] Chip select/ wait controller

[12] FLASH control

[13] ROM correction

[14] INTUNIT

[15] Clock generator (CG)

(Note 1) **The endian setting has no effect on registers that mapped to the addresses from 0xFFFF_F000 to 0xFFFF_FFFF. The register addresses from 0xFFFF_E000 to 0xFFFF_EFFF are changed by the endian setting.**

(Note 2) **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.**

1. Little endian

[1] PORT registers

ADR	Register name
FFFFF000H	P0
1H	P0CR
2H	P0FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF010H	P1
1H	P1CR
2H	P1FC1
3H	P1FC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF020H	P2
1H	P2CR
2H	P2FC1
3H	P2FC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P2IE
FH	

ADR	Register name
FFFFF030H	P3
1H	P3CR
2H	P3FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	P3PUP
CH	
DH	
EH	P3IE
FH	

ADR	Register name
FFFFF040H	P4
1H	P4CR
2H	P4FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	P4PUP
CH	
DH	
EH	P4IE
FH	

ADR	Register name
FFFFF050H	P5
1H	P5CR
2H	P5FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P5IE
FH	

ADR	Register name
FFFFF060H	P6
1H	P6CR
2H	P6FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P6IE
FH	

ADR	Register name
FFFFF070H	P7
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P7IE
FH	

ADR	Register name
FFFFF080H	P8
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P8IE
FH	

ADR	Register name
FFFFF090H	P9
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P9IE
FH	

ADR	Register name
FFFFF0A0H	PA
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PAIE
FH	

ADR	Register name
FFFFF0B0H	PB
1H	PBCR
2H	PBFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PBIE
FH	

ADR	Register name
FFFFF0C0H	PC
1H	PCCR
2H	PCFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PCIE
FH	

ADR	Register name
FFFFF0D0H	PD
1H	PDCR
2H	PDFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PDIE
FH	

ADR	Register name
FFFFF0E0H	PE
1H	PECR
2H	PEFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	PEOD
BH	
CH	
DH	PESEL
EH	PEIE
FH	

ADR	Register name
FFFFF0F0H	PF
1H	PFCR
2H	PFFC1
3H	PFFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PFOD
BH	
CH	
DH	PFSEL
EH	PFIE
FH	

ADR	Register name
FFFFF100H	PG
1H	PGCR
2H	PGFC1
3H	PGFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PGOD
BH	
CH	
DH	PGSEL
EH	PGIE
FH	

ADR	Register name
FFFFF110H	PH
1H	PHCR
2H	PHFC1
3H	PHFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PHOD
BH	
CH	
DH	PHSEL
EH	PHIE
FH	

ADR	Register name
FFFF120H	PI
1H	PICR
2H	PIFC1
3H	PIFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PIOD
BH	
CH	
DH	PISEL
EH	PIIE
FH	

ADR	Register name
FFFFF130H	PJ
1H	PJCR
2H	PJFC1
3H	PJFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PJOD
BH	
CH	
DH	PJSEL
EH	PJIE
FH	

ADR	Register name
FFFFF140H	PK
1H	PKCR
2H	PKFC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	PKPUP
CH	
DH	
EH	PKIE
FH	

ADR	Register name
FFFFF150H	PL
1H	PLCR
2H	PLFC1
3H	PLFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PLOD
BH	
CH	
DH	PLSEL
EH	PLIE
FH	

ADR	Register name
FFFFF160H	PM
1H	PMCR
2H	PMFC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PMIE
FH	

ADR	Register name
FFFFF170H	PN
1H	PNCR
2H	PNFC1
3H	PNFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PNOD
BH	
CH	
DH	PNSEL
EH	PNIE
FH	

ADR	Register name
FFFFF180H	PO
1H	POCR
2H	***reserved***
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	***reserved***
EH	POIE
FH	

ADR	Register name
FFFFF190H	PP
1H	PPCR
2H	***reserved***
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	***reserved***
EH	PPIE
FH	

ADR	Register name
FFFFF1A0H	PQ
1H	PQCR
2H	PQFC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PQIE
FH	

[2] 16-bit timer

ADR	Register name
FFFFF200H	TB00RUN
1H	TB00CR
2H	TB00MOD
3H	TB00FFCR
4H	TB00ST
5H	TB00IM
6H	TB00UCL
7H	TB00UCH
8H	TB00RG0L
9H	TB00RG0H
AH	TB00RG1L
BH	TB00RG1H
CH	TB00CP0L
DH	TB00CP0H
EH	TB00CP1L
FH	TB00CP1H

ADR	Register name
FFFFF210H	TB01RUN
1H	TB01CR
2H	TB01MOD
3H	TB01FFCR
4H	TB01ST
5H	TB01IM
6H	TB01UCL
7H	TB01UCH
8H	TB01RG0L
9H	TB01RG0H
AH	TB01RG1L
BH	TB01RG1H
CH	TB01CP0L
DH	TB01CP0H
EH	TB01CP1L
FH	TB01CP1H

ADR	Register name
FFFFF220H	TB02RUN
1H	TB02CR
2H	TB02MOD
3H	TB02FFCR
4H	TB02ST
5H	TB02IM
6H	TB02UCL
7H	TB02UCH
8H	TB02RG0L
9H	TB02RG0H
AH	TB02RG1L
BH	TB02RG1H
CH	TB02CP0L
DH	TB02CP0H
EH	TB02CP1L
FH	TB02CP1H

ADR	Register name
FFFFF230H	TB03RUN
1H	TB03CR
2H	TB03MOD
3H	TB03FFCR
4H	TB03ST
5H	TB03IM
6H	TB03UCL
7H	TB03UCH
8H	TB03RG0L
9H	TB03RG0H
AH	TB03RG1L
BH	TB03RG1H
CH	TB03CP0L
DH	TB03CP0H
EH	TB03CP1L
FH	TB03CP1H

ADR	Register name
FFFFF240H	TB04RUN
1H	TB04CR
2H	TB04MOD
3H	TB04FFCR
4H	TB04ST
5H	TB04IM
6H	TB04UCL
7H	TB04UCH
8H	TB04RG0L
9H	TB04RG0H
AH	TB04RG1L
BH	TB04RG1H
CH	TB04CP0L
DH	TB04CP0H
EH	TB04CP1L
FH	TB04CP1H

ADR	Register name
FFFFF250H	TB05RUN
1H	TB05CR
2H	TB05MOD
3H	TB05FFCR
4H	TB05ST
5H	TB05IM
6H	TB05UCL
7H	TB05UCH
8H	TB05RG0L
9H	TB05RG0H
AH	TB05RG1L
BH	TB05RG1H
CH	TB05CP0L
DH	TB05CP0H
EH	TB05CP1L
FH	TB05CP1H

ADR	Register name
FFFFF260H	TB06RUN
1H	TB06CR
2H	TB06MOD
3H	TB06FFCR
4H	TB06ST
5H	TB06IM
6H	TB06UCL
7H	TB06UCH
8H	TB06RG0L
9H	TB06RG0H
AH	TB06RG1L
BH	TB06RG1H
CH	TB06CP0L
DH	TB06CP0H
EH	TB06CP1L
FH	TB06CP1H

ADR	Register name
FFFFF270H	TB07RUN
1H	TB07CR
2H	TB07MOD
3H	TB07FFCR
4H	TB07ST
5H	TB07IM
6H	TB07UCL
7H	TB07UCH
8H	TB07RG0L
9H	TB07RG0H
AH	TB07RG1L
BH	TB07RG1H
CH	TB07CP0L
DH	TB07CP0H
EH	TB07CP1L
FH	TB07CP1H

ADR	Register name
FFFFF280H	TB08RUN
1H	TB08CR
2H	TB08MOD
3H	TB08FFCR
4H	TB08ST
5H	TB08IM
6H	TB08UCL
7H	TB08UCH
8H	TB08RG0L
9H	TB08RG0H
AH	TB08RG1L
BH	TB08RG1H
CH	TB08CP0L
DH	TB08CP0H
EH	TB08CP1L
FH	TB08CP1H

ADR	Register name
FFFFF290H	TB09RUN
1H	TB09CR
2H	TB09MOD
3H	TB09FFCR
4H	TB09ST
5H	TB09IM
6H	TB09UCL
7H	TB09UCH
8H	TB09RG0L
9H	TB09RG0H
AH	TB09RG1L
BH	TB09RG1H
CH	TB09CP0L
DH	TB09CP0H
EH	TB09CP1L
FH	TB09CP1H

ADR	Register name
FFFFF2A0H	TB0ARUN
1H	TB0ACR
2H	TB0AMOD
3H	TB0AFFCR
4H	TB0AST
5H	TB0AIM
6H	TB0AUCL
7H	TB0AUCH
8H	TB0ARG0L
9H	TB0ARG0H
AH	TB0ARG1L
BH	TB0ARG1H
CH	TB0ACP0L
DH	TB0ACP0H
EH	TB0ACP1L
FH	TB0ACP1H

ADR	Register name
FFFFF2B0H	TB0BRUN
1H	TB0BCR
2H	TB0BMOD
3H	TB0BFFCR
4H	TB0BST
5H	TB0BIM
6H	TB0BUCL
7H	TB0BUCH
8H	TB0BRG0L
9H	TB0BRG0H
AH	TB0BRG1L
BH	TB0BRG1H
CH	TB0BCP0L
DH	TB0BCP0H
EH	TB0BCP1L
FH	TB0BCP1H

ADR	Register name
FFFFF2C0H	TB0CRUN
1H	TB0CCR
2H	TB0CMOD
3H	TB0CFFCR
4H	TB0CST
5H	
6H	TB0CUCL
7H	TB0CUCH
8H	TB0CRG0L
9H	TB0CRG0H
AH	TB0CRG1L
BH	TB0CRG1H
CH	TB0CCP0L
DH	TB0CCP0H
EH	TB0CCP1L
FH	TB0CCP1H

ADR	Register name
FFFFF2D0H	TB0DRUN
1H	TB0DCR
2H	TB0MOD
3H	TB0DFFCR
4H	TB0DST
5H	TB0DIM
6H	TB0DUCL
7H	TB0DUCH
8H	TB0DRG0L
9H	TB0DRG0H
AH	TB0DRG1L
BH	TB0DRG1H
CH	TB0DCP0L
DH	TB0DCP0H
EH	TB0DCP1L
FH	TB0DCP1H

ADR	Register name
FFFFF2E0H	TB0ERUN
1H	TB0ECR
2H	TB0EMOD
3H	TB0EFFCR
4H	TB0EST
5H	TB0EIM
6H	TB0EUCL
7H	TB0EUCH
8H	TB0ERG0L
9H	TB0ERG0H
AH	TB0ERG1L
BH	TB0ERG1H
CH	TB0ECP0L
DH	TB0ECP0H
EH	TB0ECP1L
FH	TB0ECP1H

ADR	Register name
FFFFF2F0H	TB0FRUN
1H	TB0FCR
2H	TB0FMOD
3H	TB0FFFCR
4H	TB0FST
5H	TB0FIM
6H	TB0FUCL
7H	TB0FUCH
8H	TB0FRG0L
9H	TB0FRG0H
AH	TB0FRG1L
BH	TB0FRG1H
CH	TB0FCP0L
DH	TB0FCP0H
EH	TB0FCP1L
FH	TB0FCP1H

ADR	Register name
FFFFF300H	TB10RUN
1H	TB10CR
2H	TB10MOD
3H	TB10FFCR
4H	TB10ST
5H	TB10IM
6H	TB10UCL
7H	TB10UCH
8H	TB10RG0L
9H	TB10RG0H
AH	TB10RG1L
BH	TB10RG1H
CH	TB10CP0L
DH	TB10CP0H
EH	TB10CP1L
FH	TB10CP1H

ADR	Register name
FFFFF310H	TB11RUN
1H	TB11CR
2H	TB11MOD
3H	TB11FFCR
4H	TB11ST
5H	TB11IM
6H	TB11UCL
7H	TB11UCH
8H	TB11RG0L
9H	TB11RG0H
AH	TB11RG1L
BH	TB11RG1H
CH	TB11CP0L
DH	TB11CP0H
EH	TB11CP1L
FH	TB11CP1H

ADR	Register name
FFFFF320H	TB12RUN
1H	TB12CR
2H	TB12MOD
3H	TB0AFFCR
4H	TB12ST
5H	
6H	TB12UCL
7H	TB12UCH
8H	TB12RG0L
9H	TB12RG0H
AH	TB12RG1L
BH	TB12RG1H
CH	TB12CP0L
DH	TB12CP0H
EH	TB12CP1L
FH	TB12CP1H

ADR	Register name
FFFFF330H	TB13RUN
1H	TB13CR
2H	TB13MOD
3H	TB13FFCR
4H	TB13ST
5H	TB13IM
6H	TB13UCL
7H	TB13UCH
8H	TB13RG0L
9H	TB13RG0H
AH	TB13RG1L
BH	TB13RG1H
CH	TB13CP0L
DH	TB13CP0H
EH	TB13CP1L
FH	TB13CP1H

ADR	Register name
FFFFF340H	TB14RUN
1H	TB14CR
2H	TB14MOD
3H	TB14FFCR
4H	TB14ST
5H	TB14IM
6H	TB14UCL
7H	TB14UCH
8H	TB14RG0L
9H	TB14RG0H
AH	TB14RG1L
BH	TB14RG1H
CH	TB14CP0L
DH	TB14CP0H
EH	TB14CP1L
FH	TB14CP1H

ADR	Register name
FFFFF350H	TB15RUN
1H	TB15CR
2H	TB15MOD
3H	TB15FFCR
4H	TB15ST
5H	TB15IM
6H	TB15UCL
7H	TB15UCH
8H	TB15RG0L
9H	TB15RG0H
AH	TB15RG1L
BH	TB15RG1H
CH	TB15CP0L
DH	TB15CP0H
EH	TB15CP1L
FH	TB15CP1H

ADR	Register name
FFFFF360H	TB16RUN
1H	TB16CR
2H	TB16MOD
3H	TB16FFCR
4H	TB16ST
5H	TB16IM
6H	TB16UCL
7H	TB16UCH
8H	TB16RG0L
9H	TB16RG0H
AH	TB16RG1L
BH	TB16RG1H
CH	TB16CP0L
DH	TB16CP0H
EH	TB16CP1L
FH	TB16CP1H

ADR	Register name
FFFFF370H	TB17RUN
1H	TB17CR
2H	TB17MOD
3H	TB17FFCR
4H	TB17ST
5H	TB17IM
6H	TB17UCL
7H	TB17UCH
8H	TB17RG0L
9H	TB17RG0H
AH	TB17RG1L
BH	TB17RG1H
CH	TB17CP0L
DH	TB17CP0H
EH	TB17CP1L
FH	TB17CP1H

ADR	Register name
FFFFF380H	TB18RUN
1H	TB18CR
2H	TB18MOD
3H	TB18FFCR
4H	TB18ST
5H	TB18IM
6H	TB18UCL
7H	TB18UCH
8H	TB18RG0L
9H	TB18RG0H
AH	TB18RG1L
BH	TB18RG1H
CH	TB18CP0L
DH	TB18CP0H
EH	TB18CP1L
FH	TB18CP1H

ADR	Register name
FFFFF390H	TB19RUN
1H	TB19CR
2H	TB19MOD
3H	TB19FFCR
4H	TB19ST
5H	TB19IM
6H	TB19UCL
7H	TB19UCH
8H	TB18RG0L
9H	TB19RG0H
AH	TB19RG1L
BH	TB19RG1H
CH	TB19CP0L
DH	TB19CP0H
EH	TB19CP1L
FH	TB19CP1H

ADR	Register name
FFFFF3A0H	TB1ARUN
1H	TB1ACR
2H	TB1AMOD
3H	TB1AFFCR
4H	TB1AST
5H	TB1AIM
6H	TB1AUCL
7H	TB1AUCH
8H	TB1ARG0L
9H	TB1ARG0H
AH	TB1ARG1L
BH	TB1ARG1H
CH	TB1ACP0L
DH	TB1ACP0H
EH	TB1ACP1L
FH	TB1ACP1H

ADR	Register name
FFFFF3B0H	TB1BRUN
1H	TB1BCR
2H	TB1BMOD
3H	TB1BFFCR
4H	TB1BST
5H	TB1BIM
6H	TB1BUCL
7H	TB1BUCH
8H	TB1BRG0L
9H	TB1BRG0H
AH	TB1BRG1L
BH	TB1BRG1H
CH	TB1BCP0L
DH	TB1BCP0H
EH	TB1BCP1L
FH	TB1BCP1H

ADR	Register name
FFFFF3C0H	TB1CRUN
1H	TB1CCR
2H	TB1CMOD
3H	TB1CFFCR
4H	TB1CST
5H	TB1CIM
6H	TB1CUCL
7H	TB1CUCH
8H	TB1CRG0L
9H	TB1CRG0H
AH	TB1CRG1L
BH	TB1CRG1H
CH	TB1CCP0L
DH	TB1CCP0H
EH	TB1CCP1L
FH	TB1CCP1H

ADR	Register name
FFFFF3D0H	TB1DRUN
1H	TB1DCR
2H	TB1DMOD
3H	TB1DFFCR
4H	TB1DST
5H	TB1DIM
6H	TB1DUCL
7H	TB1DUCH
8H	TB1DRG0L
9H	TB1DRG0H
AH	TB1DRG1L
BH	TB1DRG1H
CH	TB1DCP0L
DH	TB1DCP0H
EH	TB1DCP1L
FH	TB1DCP1H

ADR	Register name
FFFFF3E0H	TB1ERUN
1H	TB1ECR
2H	TB1EMOD
3H	TB1EFFCR
4H	TB1EST
5H	TB1EIM
6H	TB1EUCL
7H	TB1EUCH
8H	TB1ERG0L
9H	TB1ERG0H
AH	TB1ERG1L
BH	TB1ERG1H
CH	TB1ECP0L
DH	TB1ECP0H
EH	TB1ECP1L
FH	TB1ECP1H

ADR	Register name
FFFFF3F0H	TB1FRUN
1H	TB1FCR
2H	TB1FMOD
3H	TB1FFFCR
4H	TB1FST
5H	TB1FIM
6H	TB1FUCL
7H	TB1FUCH
8H	TB1FRG0L
9H	TB1FRG0H
AH	TB1FRG1L
BH	TB1FRG1H
CH	TB1FCP0L
DH	TB1FCP0H
EH	TB1FCP1L
FH	TB1FCP1H

ADR	Register name
FFFFF400H	TB20RUN
1H	TB20CR
2H	TB20MOD
3H	TB20FFCR
4H	TB20ST
5H	TB20IM
6H	TB20UCL
7H	TB20UCH
8H	TB20RG0L
9H	TB20RG0H
AH	TB20RG1L
BH	TB20RG1H
CH	TB20CP0L
DH	TB20CP0H
EH	TB20CP1L
FH	TB20CP1H

ADR	Register name
FFFFF410H	TB21RUN
1H	TB21CR
2H	TB21MOD
3H	TB21FFCR
4H	TB21ST
5H	TB21IM
6H	TB21UCL
7H	TB21UCH
8H	TB21RG0L
9H	TB21RG0H
AH	TB21RG1L
BH	TB21RG1H
CH	TB21CP0L
DH	TB21CP0H
EH	TB21CP1L
FH	TB21CP1H

ADR	Register name
FFFFF420H	TB22RUN
1H	TB22CR
2H	TB22MOD
3H	TB22FFCR
4H	TB22ST
5H	TB22IM
6H	TB22UCL
7H	TB22UCH
8H	TB22RG0L
9H	TB22RG0H
AH	TB22RG1L
BH	TB22RG1H
CH	TB22CP0L
DH	TB22CP0H
EH	TB22CP1L
FH	TB22CP1H

ADR	Register name
FFFFF430H	TB23RUN
1H	TB23CR
2H	TB23MOD
3H	TB23FFCR
4H	TB23ST
5H	TB23IM
6H	TB23UCL
7H	TB23UCH
8H	TB23RG0L
9H	TB23RG0H
AH	TB23RG1L
BH	TB23RG1H
CH	TB23CP0L
DH	TB23CP0H
EH	TB23CP1L
FH	TB23CP1H

[3] TMRC

ADR	Register name
FFFFF500H	TCACR
1H	TBTARUN
2H	TBTACR
3H	
4H	TBTACAPLL
5H	TBTACAPLH
6H	TBTACAPHL
7H	TBTACAPHH
8H	TBTARDCAPLL
9H	TBTARDCAPLH
AH	TBTARDCAPHL
BH	TBTARDCAPHH
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF510H	CMPA0CTL
1H	
2H	
3H	
4H	CMPA0LL
5H	CMPA0LH
6H	CMPA0HL
7H	CMPA0HH
8H	CMPA1CTL
9H	
AH	
BH	
CH	CMPA1LL
DH	CMPA1LH
EH	CMPA1HL
FH	CMPA1HH

ADR	Register name
FFFFF520H	CAPA0CR
1H	
2H	
3H	
4H	CAPA0LL
5H	CAPA0LH
6H	CAPA0HL
7H	CAPA0HH
8H	CAPA1CR
9H	
AH	
BH	
CH	CAPA1LL
DH	CAPA1LH
EH	CAPA1HL
FH	CAPA1HH

ADR	Register name
FFFFF530H	TCBCR
1H	TBTBRUN
2H	TBTBCR
3H	
4H	TBTBCAPLL
5H	TBTBCAPLH
6H	TBTBCAPHL
7H	TBTBCAPHH
8H	TBTBRDCAPLL
9H	TBTBRDCAPLH
AH	TBTBRDCAPHL
BH	TBTBRDCAPHH
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF540H	CMPB0CTL
1H	
2H	
3H	
4H	CMPB0LL
5H	CMPB0LH
6H	CMPB0HL
7H	CMPB0HH
8H	CMPB1CTL
9H	
AH	
BH	
CH	CMPB1LL
DH	CMPB1LH
EH	CMPB1HL
FH	CMPB1HH

ADR	Register name
FFFFF550H	CAPB0CR
1H	
2H	
3H	
4H	CAPB0LL
5H	CAPB0LH
6H	CAPB0HL
7H	CAPB0HH
8H	CAPB1CR
9H	
AH	
BH	
CH	CAPB1LL
DH	CAPB1LH
EH	CAPB1HL
FH	CAPB1HH

[4] SBI

ADR	Register name
FFFFF600H	SBI0CR1
1H	SBI0DBR
2H	SBI0I2CAR
3H	SBI0CR2/SR
4H	SBI0BR0
5H	
6H	
7H	SBI0CR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF610H	SBI1CR1
1H	SBI1DBR
2H	SBI1I2CAR
3H	SBI1CR2/SR
4H	SBI1BR0
5H	
6H	
7H	SBI1CR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] SIO/UART

ADR	Register name
FFFFF700H	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
FFFFF710H	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
FFFFF720H	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
FFFFF730H	SC3BUF
1H	SC3CR
2H	SC3MOD0
3H	BR3CR
4H	BR3ADD
5H	SC3MOD1
6H	SC3MOD2
7H	SC3EN
8H	SC3RFC
9H	SC3TFC
AH	SC3RST
BH	SC3TST
CH	SC3FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF740H	SC4BUF
1H	SC4CR
2H	SC4MOD0
3H	BR4CR
4H	BR4ADD
5H	SC4MOD1
6H	SC4MOD2
7H	SC4EN
8H	SC4RFC
9H	SC4TFC
AH	SC4RST
BH	SC4TST
CH	SC4FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF750H	SC5BUF
1H	SC5CR
2H	SC5MOD0
3H	BR5CR
4H	BR5ADD
5H	SC5MOD1
6H	SC5MOD2
7H	SC5EN
8H	SC5RFC
9H	SC5TFC
AH	SC5RST
BH	SC5TST
CH	SC5FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF760H	SC6BUF
1H	SC6CR
2H	SC6MOD0
3H	BR6CR
4H	BR6ADD
5H	SC6MOD1
6H	SC6MOD2
7H	SC6EN
8H	SC6RFC
9H	SC6TFC
AH	SC6RST
BH	SC6TST
CH	SC6FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF770H	SC7BUF
1H	SC7CR
2H	SC7MOD0
3H	BR7CR
4H	BR7ADD
5H	SC7MOD1
6H	SC7MOD2
7H	SC7EN
8H	SC7RFC
9H	SC7TFC
AH	SC7RST
BH	SC7TST
CH	SC7FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF780H	SC8BUF
1H	SC8CR
2H	SC8MOD0
3H	BR8CR
4H	BR8ADD
5H	SC8MOD1
6H	SC8MOD2
7H	SC8EN
8H	SC8RFC
9H	SC8TFC
AH	SC8RST
BH	SC8TST
CH	SC8FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF790H	SC9BUF
1H	SC9CR
2H	SC9MOD0
3H	BR9CR
4H	BR9ADD
5H	SC9MOD1
6H	SC9MOD2
7H	SC9EN
8H	SC9RFC
9H	SC9TFC
AH	SC9RST
BH	SC9TST
CH	SC9FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF7A0H	SCABUF
1H	SCACR
2H	SCAMOD0
3H	BRACR
4H	BRAADD
5H	SCAMOD1
6H	SCAMOD2
7H	SCAEN
8H	SCARFC
9H	SCATFC
AH	SCARST
BH	SCATST
CH	SCAFCNF
DH	
EH	
FH	

[6] ADC

ADR	Register name
FFFFF800H	ADAREG08L
1H	ADAREG08H
2H	ADAREG19L
3H	ADAREG19H
4H	ADAREG2AL
5H	ADAREG2AH
6H	ADAREG3BL
7H	ADAREG3BH
8H	ADAREG4CL
9H	ADAREG4CH
AH	ADAREG5DL
BH	ADAREG5DH
CH	ADAREG6EL
DH	ADAREG6EH
EH	ADAREG7FL
FH	ADAREG7FH

ADR	Register name
FFFFF810H	ADAREGSPL
1H	ADAREGSPH
2H	ADACOMREGL
3H	ADACOMREGH
4H	ADAMOD0
5H	ADAMOD1
6H	ADAMOD2
7H	ADAMOD3
8H	ADAMOD4
9H	ADACBAS0
AH	*reserved*
BH	*reserved*
CH	ADACLK
DH	*reserved*
EH	*reserved*
FH	

ADR	Register name
FFFFF820H	ADBREG08L
1H	ADBREG08H
2H	ADBREG19L
3H	ADBREG19H
4H	ADBREG2AL
5H	ADBREG2AH
6H	ADBREG3BL
7H	ADBREG3BH
8H	ADBREG4CL
9H	ADBREG4CH
AH	ADBREG5DL
BH	ADBREG5DH
CH	ADBREG6EL
DH	ADBREG6EH
EH	ADBREG7FL
FH	ADBREG7FH

ADR	Register name
FFFFF830H	ADBREGSPL
1H	ADBREGSPH
2H	ADBCOMREGL
3H	ADBCOMREGH
4H	ADBMOD0
5H	ADBMOD1
6H	ADBMOD2
7H	ADBMOD3
8H	ADBMOD4
9H	**ADBCBAS0**
AH	*reserved*
BH	*reserved*
CH	ADBCLK
DH	*reserved*
EH	*reserved*
FH	

[7]KWUP

ADR	Register name
FFFFF900H	KWUPST0
1H	KWUPST1
2H	KWUPST2
3H	KWUPST3
4H	KWUPST4
5H	KWUPST5
6H	KWUPST6
7H	KWUPST7
8H	
9H	
AH	
BH	
CH	
DH	
EH	KWUPST
FH	

[8] WDT

ADR	Register name
FFFFFA00H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] INTC

ADR	Register name
FFFFE000H	IMC0
1H	″
2H	″
3H	″
4H	IMC1
5H	″
6H	″
7H	″
8H	IMC2
9H	″
AH	″
BH	″
CH	IMC3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE010H	IMC4
1H	″
2H	″
3H	″
4H	IMC5
5H	″
6H	″
7H	″
8H	IMC6
9H	″
AH	″
BH	″
CH	IMC7
DH	″
EH	″
FH	″

ADR	Register name
FFFFE020H	IMC8
1H	″
2H	″
3H	″
4H	IMC9
5H	″
6H	″
7H	″
8H	IMCA
9H	″
AH	″
BH	″
CH	IMCB
DH	″
EH	″
FH	″

ADR	Register name
FFFFE030H	IMCC
1H	″
2H	″
3H	″
4H	IMCD
5H	″
6H	″
7H	″
8H	IMCE
9H	″
AH	″
BH	″
CH	IMCF
DH	″
EH	″
FH	″

ADR	Register name
FFFFE040H	IVR
1H	″
2H	″
3H	″
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	″
2H	″
3H	″
4H	***reserved***
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	***reserved***
1H	″
2H	″
3H	″
4H	***reserved***
5H	″
6H	″
7H	″
8H	***reserved***
9H	″
AH	″
BH	″
CH	ILEV
DH	″
EH	″
FH	″

[10] DMAC

ADR	Register name
FFFFE200H	CCR0
1H	″
2H	″
3H	″
4H	CSR0
5H	″
6H	″
7H	″
8H	SAR0
9H	″
AH	″
BH	″
CH	DAR0
DH	″
EH	″
FH	″

ADR	Register name
FFFFE210H	BCR0
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR0
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE220H	CCR1
1H	″
2H	″
3H	″
4H	CSR1
5H	″
6H	″
7H	″
8H	SAR1
9H	″
AH	″
BH	″
CH	DAR1
DH	″
EH	″
FH	″

ADR	Register name
FFFFE230H	BCR1
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR1
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE240H	CCR2
1H	″
2H	″
3H	″
4H	CSR2
5H	″
6H	″
7H	″
8H	SAR2
9H	″
AH	″
BH	″
CH	DAR2
DH	″
EH	″
FH	″

ADR	Register name
FFFFE250H	BCR2
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR2
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE260H	CCR3
1H	″
2H	″
3H	″
4H	CSR3
5H	″
6H	″
7H	″
8H	SAR3
9H	″
AH	″
BH	″
CH	DAR3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE270H	BCR3
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR3
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE280H	CCR4
1H	″
2H	″
3H	″
4H	CSR4
5H	″
6H	″
7H	″
8H	SAR4
9H	″
AH	″
BH	″
CH	DAR4
DH	″
EH	″
FH	″

ADR	Register name
FFFFE290H	BCR4
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR4
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2A0H	CCR5
1H	″
2H	″
3H	″
4H	CSR5
5H	″
6H	″
7H	″
8H	SAR5
9H	″
AH	″
BH	″
CH	DAR5
DH	″
EH	″
FH	″

ADR	Register name
FFFFE2B0H	BCR5
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR5
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2C0H	CCR6
1H	〃
2H	〃
3H	〃
4H	CSR6
5H	〃
6H	〃
7H	〃
8H	SAR6
9H	〃
AH	〃
BH	〃
CH	DAR6
DH	〃
EH	〃
FH	〃

ADR	Register name
FFFFE2D0H	BCR6
1H	〃
2H	〃
3H	〃
4H	
5H	
6H	
7H	
8H	DTCR6
9H	〃
AH	〃
BH	〃
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2E0H	CCR7
1H	〃
2H	〃
3H	〃
4H	CSR7
5H	〃
6H	〃
7H	〃
8H	SAR7
9H	〃
AH	〃
BH	〃
CH	DAR7
DH	〃
EH	〃
FH	〃

ADR	Register name
FFFFE2F0H	BCR7
1H	〃
2H	〃
3H	〃
4H	
5H	
6H	
7H	
8H	DTCR7
9H	〃
AH	〃
BH	〃
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE300H	DCR
1H	〃
2H	〃
3H	〃
4H	RSR
5H	〃
6H	〃
7H	〃
8H	
9H	
AH	
BH	
CH	DHR
DH	〃
EH	〃
FH	〃

[11] CS/WAIT controller

ADR	Register name
FFFFE400H	BMA0
1H	″
2H	″
3H	″
4H	BMA1
5H	″
6H	″
7H	″
8H	BMA2
9H	″
AH	″
BH	″
CH	BMA3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE410H	BMA4
1H	″
2H	″
3H	″
4H	BMA5
5H	″
6H	″
7H	″
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE480H	B01CS
1H	″
2H	″
3H	″
4H	B23CS
5H	″
6H	″
7H	″
8H	B45CS
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE500H	
1H	
2H	
3H	
4H	***reserved***
5H	″
6H	″
7H	″
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] FLASH control

ADR	Register name
FFFFE510H	SEQMOD
1H	″
2H	″
3H	″
4H	SEQCNT
5H	″
6H	″
7H	″
8H	ROMSEC1
9H	
AH	
BH	
CH	ROMSEC2
DH	
EH	
FH	

ADR	Register name
FFFFE520H	FLCS
1H	″
2H	″
3H	″
4H	***reserved***
5H	″
6H	″
7H	″
8H	***reserved***
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

[13] ROM correction

ADR	Register name
FFFFE540H	ADDREG0
1H	″
2H	″
3H	″
4H	ADDREG1
5H	″
6H	″
7H	″
8H	ADDREG2
9H	″
AH	″
BH	″
CH	ADDREG3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE550H	ADDREG4
1H	″
2H	″
3H	″
4H	ADDREG5
5H	″
6H	″
7H	″
8H	ADDREG6
9H	″
AH	″
BH	″
CH	ADDREG7
DH	″
EH	″
FH	″

ADR	Register name
FFFFE560H	ADDREG8
1H	″
2H	″
3H	″
4H	ADDREG9
5H	″
6H	″
7H	″
8H	ADDREGA
9H	″
AH	″
BH	″
CH	ADDREGB
DH	″
EH	″
FH	″

[14] INTUNIT

ADR	Register name
FFFFE700H	ADCINT
1H	
2H	
3H	
4H	TMRBINTA
5H	
6H	
7H	
8H	TMRBINTB
9H	
AH	
BH	
CH	TMRBINTC
DH	
EH	
FH	

ADR	Register name
FFFFE710H	TMRBINTD
1H	
2H	
3H	
4H	TMRBINTE
5H	
6H	
7H	
8H	CAPINT
9H	
AH	
BH	
CH	CMPINT
DH	
EH	
FH	

ADR	Register name
FFFFE720H	TBTINT
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE740H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] 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	″
2H	″
3H	″
4H	IMCGB
5H	″
6H	″
7H	″
8H	IMCGC
9H	″
AH	″
BH	″
CH	IMCGD
DH	″
EH	″
FH	″

ADR	Register name
FFFFEE20H	EICRCG
1H	″
2H	″
3H	″
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	

The endian setting has effect on registers that mapped to the addresses from 0xFFFF_E000 to 0xFFFF_EFFF.

2. Big endian

[1] PORT registers

ADR	Register name
FFFFF000H	P0
1H	P0CR
2H	P0FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF010H	P1
1H	P1CR
2H	P1FC1
3H	P1FC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF020H	P2
1H	P2CR
2H	P2FC1
3H	P2FC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P2IE
FH	

ADR	Register name
FFFFF030H	P3
1H	P3CR
2H	P3FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	P3PUP
CH	
DH	
EH	P3IE
FH	

ADR	Register name
FFFFF040H	P4
1H	P4CR
2H	P4FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	P4PUP
CH	
DH	
EH	P4IE
FH	

ADR	Register name
FFFFF050H	P5
1H	P5CR
2H	P5FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P5IE
FH	

ADR	Register name
FFFFF060H	P6
1H	P6CR
2H	P6FC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P6IE
FH	

ADR	Register name
FFFFF070H	P7
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P7IE
FH	

ADR	Register name
FFFFF080H	P8
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P8IE
FH	

ADR	Register name
FFFFF090H	P9
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	P9IE
FH	

ADR	Register name
FFFFF0A0H	PA
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PAIE
FH	

ADR	Register name
FFFFF0B0H	PB
1H	PBCR
2H	PBFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PBIE
FH	

ADR	Register name
FFFFF0C0H	PC
1H	PCCR
2H	PCFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PCIE
FH	

ADR	Register name
FFFFF0D0H	PD
1H	PDCR
2H	PDFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PDIE
FH	

ADR	Register name
FFFFF0E0H	PE
1H	PECR
2H	PEFC1
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	PEOD
BH	
CH	
DH	PESEL
EH	PEIE
FH	

ADR	Register name
FFFFF0F0H	PF
1H	PFCR
2H	PFFC1
3H	PFFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PFOD
BH	
CH	
DH	PFSEL
EH	PFIE
FH	

ADR	Register name
FFFFF100H	PG
1H	PGCR
2H	PGFC1
3H	PGFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PGOD
BH	
CH	
DH	PGSEL
EH	PCIE
FH	

ADR	Register name
FFFFF110H	PH
1H	PHCR
2H	PHFC1
3H	PHFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PHOD
BH	
CH	
DH	PHSEL
EH	PHIE
FH	

ADR	Register name
FFFF120H	PI
1H	PICR
2H	PIFC1
3H	PIFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PIOD
BH	
CH	
DH	PISEL
EH	PIIE
FH	

ADR	Register name
FFFFF130H	PJ
1H	PJCR
2H	PJFC1
3H	PJFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PJOD
BH	
CH	
DH	PJSEL
EH	PJIE
FH	

ADR	Register name
FFFFF140H	PK
1H	PKCR
2H	PKFC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	PKPUP
CH	
DH	
EH	PKIE
FH	

ADR	Register name
FFFFF150H	PL
1H	PLCR
2H	PLFC1
3H	PLFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PLOD
BH	
CH	
DH	PLSEL
EH	PLIE
FH	

ADR	Register name
FFFFF160H	PM
1H	PMCR
2H	PMFC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PMIE
FH	

ADR	Register name
FFFFF170H	PN
1H	PNCR
2H	PNFC1
3H	PNFC2
4H	
5H	
6H	
7H	
8H	
9H	
AH	PNOD
BH	
CH	
DH	PNSEL
EH	PNIE
FH	

ADR	Register name
FFFFF180H	PO
1H	POCR
2H	***reserved***
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	***reserved***
EH	POIE
FH	

ADR	Register name
FFFFF190H	PP
1H	PPCR
2H	***reserved***
3H	***reserved***
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	***reserved***
EH	PPIE
FH	

ADR	Register name
FFFFF1A0H	PQ
1H	PQCR
2H	PQFC1
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	PQIE
FH	

[2] 16-bit timer

ADR	Register name
FFFFF200H	TB00RUN
1H	TB00CR
2H	TB00MOD
3H	TB00FFCR
4H	TB00ST
5H	TB00IM
6H	TB00UCL
7H	TB00UCH
8H	TB00RG0L
9H	TB00RG0H
AH	TB00RG1L
BH	TB00RG1H
CH	TB00CP0L
DH	TB00CP0H
EH	TB00CP1L
FH	TB00CP1H

ADR	Register name
FFFFF210H	TB01RUN
1H	TB01CR
2H	TB01MOD
3H	TB01FFCR
4H	TB01ST
5H	TB01IM
6H	TB01UCL
7H	TB01UCH
8H	TB01RG0L
9H	TB01RG0H
AH	TB01RG1L
BH	TB01RG1H
CH	TB01CP0L
DH	TB01CP0H
EH	TB01CP1L
FH	TB01CP1H

ADR	Register name
FFFFF220H	TB02RUN
1H	TB02CR
2H	TB02MOD
3H	TB02FFCR
4H	TB02ST
5H	TB02IM
6H	TB02UCL
7H	TB02UCH
8H	TB02RG0L
9H	TB02RG0H
AH	TB02RG1L
BH	TB02RG1H
CH	TB02CP0L
DH	TB02CP0H
EH	TB02CP1L
FH	TB02CP1H

ADR	Register name
FFFFF230H	TB03RUN
1H	TB03CR
2H	TB03MOD
3H	TB03FFCR
4H	TB03ST
5H	TB03IM
6H	TB03UCL
7H	TB03UCH
8H	TB03RG0L
9H	TB03RG0H
AH	TB03RG1L
BH	TB03RG1H
CH	TB03CP0L
DH	TB03CP0H
EH	TB03CP1L
FH	TB03CP1H

ADR	Register name
FFFFF240H	TB04RUN
1H	TB04CR
2H	TB04MOD
3H	TB04FFCR
4H	TB04ST
5H	TB04IM
6H	TB04UCL
7H	TB04UCH
8H	TB04RG0L
9H	TB04RG0H
AH	TB04RG1L
BH	TB04RG1H
CH	TB04CP0L
DH	TB04CP0H
EH	TB04CP1L
FH	TB04CP1H

ADR	Register name
FFFFF250H	TB05RUN
1H	TB05CR
2H	TB05MOD
3H	TB05FFCR
4H	TB05ST
5H	TB05IM
6H	TB05UCL
7H	TB05UCH
8H	TB05RG0L
9H	TB05RG0H
AH	TB05RG1L
BH	TB05RG1H
CH	TB05CP0L
DH	TB05CP0H
EH	TB05CP1L
FH	TB05CP1H

ADR	Register name
FFFFF260H	TB06RUN
1H	TB06CR
2H	TB06MOD
3H	TB06FFCR
4H	TB06ST
5H	TB06IM
6H	TB06UCL
7H	TB06UCH
8H	TB06RG0L
9H	TB06RG0H
AH	TB06RG1L
BH	TB06RG1H
CH	TB06CP0L
DH	TB06CP0H
EH	TB06CP1L
FH	TB06CP1H

ADR	Register name
FFFFF270H	TB07RUN
1H	TB07CR
2H	TB07MOD
3H	TB07FFCR
4H	TB07ST
5H	TB07IM
6H	TB07UCL
7H	TB07UCH
8H	TB07RG0L
9H	TB07RG0H
AH	TB07RG1L
BH	TB07RG1H
CH	TB07CP0L
DH	TB07CP0H
EH	TB07CP1L
FH	TB07CP1H

ADR	Register name
FFFFF280H	TB08RUN
1H	TB08CR
2H	TB08MOD
3H	TB08FFCR
4H	TB08ST
5H	TB08IM
6H	TB08UCL
7H	TB08UCH
8H	TB08RG0L
9H	TB08RG0H
AH	TB08RG1L
BH	TB08RG1H
CH	TB08CP0L
DH	TB08CP0H
EH	TB08CP1L
FH	TB08CP1H

ADR	Register name
FFFFF290H	TB09RUN
1H	TB09CR
2H	TB09MOD
3H	TB09FFCR
4H	TB09ST
5H	TB09IM
6H	TB09UCL
7H	TB09UCH
8H	TB09RG0L
9H	TB09RG0H
AH	TB09RG1L
BH	TB09RG1H
CH	TB09CP0L
DH	TB09CP0H
EH	TB09CP1L
FH	TB09CP1H

ADR	Register name
FFFFF2A0H	TB0ARUN
1H	TB0ACR
2H	TB0AMOD
3H	TB0AFFCR
4H	TB0AST
5H	TB0AIM
6H	TB0AUCL
7H	TB0AUCH
8H	TB0ARG0L
9H	TB0ARG0H
AH	TB0ARG1L
BH	TB0ARG1H
CH	TB0ACP0L
DH	TB0ACP0H
EH	TB0ACP1L
FH	TB0ACP1H

ADR	Register name
FFFFF2B0H	TB0BRUN
1H	TB0BCR
2H	TB0BMOD
3H	TB0BFFCR
4H	TB0BST
5H	TB0BIM
6H	TB0BUCL
7H	TB0BUCH
8H	TB0BRG0L
9H	TB0BRG0H
AH	TB0BRG1L
BH	TB0BRG1H
CH	TB0BCP0L
DH	TB0BCP0H
EH	TB0BCP1L
FH	TB0BCP1H

ADR	Register name
FFFFF2C0H	TB0CRUN
1H	TB0CCR
2H	TB0CMOD
3H	TB0CFFCR
4H	TB0CST
5H	
6H	TB0CUCL
7H	TB0CUCH
8H	TB0CRG0L
9H	TB0CRG0H
AH	TB0CRG1L
BH	TB0CRG1H
CH	TB0CCP0L
DH	TB0CCP0H
EH	TB0CCP1L
FH	TB0CCP1H

ADR	Register name
FFFFF2D0H	TB0DRUN
1H	TB0DCR
2H	TB0MOD
3H	TB0DFFCR
4H	TB0DST
5H	TB0DIM
6H	TB0DUCL
7H	TB0DUCH
8H	TB0DRG0L
9H	TB0DRG0H
AH	TB0DRG1L
BH	TB0DRG1H
CH	TB0DCP0L
DH	TB0DCP0H
EH	TB0DCP1L
FH	TB0DCP1H

ADR	Register name
FFFFF2E0H	TB0ERUN
1H	TB0ECR
2H	TB0EMOD
3H	TB0EFFCR
4H	TB0EST
5H	TB0EIM
6H	TB0EUCL
7H	TB0EUCH
8H	TB0ERG0L
9H	TB0ERG0H
AH	TB0ERG1L
BH	TB0ERG1H
CH	TB0ECP0L
DH	TB0ECP0H
EH	TB0ECP1L
FH	TB0ECP1H

ADR	Register name
FFFFF2F0H	TB0FRUN
1H	TB0FCR
2H	TB0FMOD
3H	TB0FFFCR
4H	TB0FST
5H	TB0FIM
6H	TB0FUCL
7H	TB0FUCH
8H	TB0FRG0L
9H	TB0FRG0H
AH	TB0FRG1L
BH	TB0FRG1H
CH	TB0FCP0L
DH	TB0FCP0H
EH	TB0FCP1L
FH	TB0FCP1H

ADR	Register name
FFFFF300H	TB10RUN
1H	TB10CR
2H	TB10MOD
3H	TB10FFCR
4H	TB10ST
5H	TB10IM
6H	TB10UCL
7H	TB10UCH
8H	TB10RG0L
9H	TB10RG0H
AH	TB10RG1L
BH	TB10RG1H
CH	TB10CP0L
DH	TB10CP0H
EH	TB10CP1L
FH	TB10CP1H

ADR	Register name
FFFFF310H	TB11RUN
1H	TB11CR
2H	TB11MOD
3H	TB11FFCR
4H	TB11ST
5H	TB11IM
6H	TB11UCL
7H	TB11UCH
8H	TB11RG0L
9H	TB11RG0H
AH	TB11RG1L
BH	TB11RG1H
CH	TB11CP0L
DH	TB11CP0H
EH	TB11CP1L
FH	TB11CP1H

ADR	Register name
FFFFF320H	TB12RUN
1H	TB12CR
2H	TB12MOD
3H	TB0AFFCR
4H	TB12ST
5H	
6H	TB12UCL
7H	TB12UCH
8H	TB12RG0L
9H	TB12RG0H
AH	TB12RG1L
BH	TB12RG1H
CH	TB12CP0L
DH	TB12CP0H
EH	TB12CP1L
FH	TB12CP1H

ADR	Register name
FFFFF330H	TB13RUN
1H	TB13CR
2H	TB13MOD
3H	TB13FFCR
4H	TB13ST
5H	TB13IM
6H	TB13UCL
7H	TB13UCH
8H	TB13RG0L
9H	TB13RG0H
AH	TB13RG1L
BH	TB13RG1H
CH	TB13CP0L
DH	TB13CP0H
EH	TB13CP1L
FH	TB13CP1H

ADR	Register name
FFFFF340H	TB14RUN
1H	TB14CR
2H	TB14MOD
3H	TB14FFCR
4H	TB14ST
5H	TB14IM
6H	TB14UCL
7H	TB14UCH
8H	TB14RG0L
9H	TB14RG0H
AH	TB14RG1L
BH	TB14RG1H
CH	TB14CP0L
DH	TB14CP0H
EH	TB14CP1L
FH	TB14CP1H

ADR	Register name
FFFFF350H	TB15RUN
1H	TB15CR
2H	TB15MOD
3H	TB15FFCR
4H	TB15ST
5H	TB15IM
6H	TB15UCL
7H	TB15UCH
8H	TB15RG0L
9H	TB15RG0H
AH	TB15RG1L
BH	TB15RG1H
CH	TB15CP0L
DH	TB15CP0H
EH	TB15CP1L
FH	TB15CP1H

ADR	Register name
FFFFF360H	TB16RUN
1H	TB16CR
2H	TB16MOD
3H	TB16FFCR
4H	TB16ST
5H	TB16IM
6H	TB16UCL
7H	TB16UCH
8H	TB16RG0L
9H	TB16RG0H
AH	TB16RG1L
BH	TB16RG1H
CH	TB16CP0L
DH	TB16CP0H
EH	TB16CP1L
FH	TB16CP1H

ADR	Register name
FFFFF370H	TB17RUN
1H	TB17CR
2H	TB17MOD
3H	TB17FFCR
4H	TB17ST
5H	TB17IM
6H	TB17UCL
7H	TB17UCH
8H	TB17RG0L
9H	TB17RG0H
AH	TB17RG1L
BH	TB17RG1H
CH	TB17CP0L
DH	TB17CP0H
EH	TB17CP1L
FH	TB17CP1H

ADR	Register name
FFFFF380H	TB18RUN
1H	TB18CR
2H	TB18MOD
3H	TB18FFCR
4H	TB18ST
5H	TB18IM
6H	TB18UCL
7H	TB18UCH
8H	TB18RG0L
9H	TB18RG0H
AH	TB18RG1L
BH	TB18RG1H
CH	TB18CP0L
DH	TB18CP0H
EH	TB18CP1L
FH	TB18CP1H

ADR	Register name
FFFFF390H	TB19RUN
1H	TB19CR
2H	TB19MOD
3H	TB19FFCR
4H	TB19ST
5H	TB19IM
6H	TB19UCL
7H	TB19UCH
8H	TB18RG0L
9H	TB19RG0H
AH	TB19RG1L
BH	TB19RG1H
CH	TB19CP0L
DH	TB19CP0H
EH	TB19CP1L
FH	TB19CP1H

ADR	Register name
FFFFF3A0H	TB1ARUN
1H	TB1ACR
2H	TB1AMOD
3H	TB1AFFCR
4H	TB1AST
5H	TB1AIM
6H	TB1AUCL
7H	TB1AUCH
8H	TB1ARG0L
9H	TB1ARG0H
AH	TB1ARG1L
BH	TB1ARG1H
CH	TB1ACP0L
DH	TB1ACP0H
EH	TB1ACP1L
FH	TB1ACP1H

ADR	Register name
FFFFF3B0H	TB1BRUN
1H	TB1BCR
2H	TB1BMOD
3H	TB1BFFCR
4H	TB1BST
5H	TB1BIM
6H	TB1BUCL
7H	TB1BUCH
8H	TB1BRG0L
9H	TB1BRG0H
AH	TB1BRG1L
BH	TB1BRG1H
CH	TB1BCP0L
DH	TB1BCP0H
EH	TB1BCP1L
FH	TB1BCP1H

ADR	Register name
FFFFF3C0H	TB1CRUN
1H	TB1CCR
2H	TB1CMOD
3H	TB1CFFCR
4H	TB1CST
5H	TB1CIM
6H	TB1CUCL
7H	TB1CUCH
8H	TB1CRG0L
9H	TB1CRG0H
AH	TB1CRG1L
BH	TB1CRG1H
CH	TB1CCP0L
DH	TB1CCP0H
EH	TB1CCP1L
FH	TB1CCP1H

ADR	Register name
FFFFF3D0H	TB1DRUN
1H	TB1DCR
2H	TB1DMOD
3H	TB1DFFCR
4H	TB1DST
5H	TB1DIM
6H	TB1DUCL
7H	TB1DUCH
8H	TB1DRG0L
9H	TB1DRG0H
AH	TB1DRG1L
BH	TB1DRG1H
CH	TB1DCP0L
DH	TB1DCP0H
EH	TB1DCP1L
FH	TB1DCP1H

ADR	Register name
FFFFF3E0H	TB1ERUN
1H	TB1ECR
2H	TB1EMOD
3H	TB1EFFCR
4H	TB1EST
5H	TB1EIM
6H	TB1EUCL
7H	TB1EUCH
8H	TB1ERG0L
9H	TB1ERG0H
AH	TB1ERG1L
BH	TB1ERG1H
CH	TB1ECP0L
DH	TB1ECP0H
EH	TB1ECP1L
FH	TB1ECP1H

ADR	Register name
FFFFF3F0H	TB1FRUN
1H	TB1FCR
2H	TB1FMOD
3H	TB1FFFCR
4H	TB1FST
5H	TB1FIM
6H	TB1FUCL
7H	TB1FUCH
8H	TB1FRG0L
9H	TB1FRG0H
AH	TB1FRG1L
BH	TB1FRG1H
CH	TB1FCP0L
DH	TB1FCP0H
EH	TB1FCP1L
FH	TB1FCP1H

ADR	Register name
FFFFF400H	TB20RUN
1H	TB20CR
2H	TB20MOD
3H	TB20FFCR
4H	TB20ST
5H	TB20IM
6H	TB20UCL
7H	TB20UCH
8H	TB20RG0L
9H	TB20RG0H
AH	TB20RG1L
BH	TB20RG1H
CH	TB20CP0L
DH	TB20CP0H
EH	TB20CP1L
FH	TB20CP1H

ADR	Register name
FFFFF410H	TB21RUN
1H	TB21CR
2H	TB21MOD
3H	TB21FFCR
4H	TB21ST
5H	TB21IM
6H	TB21UCL
7H	TB21UCH
8H	TB21RG0L
9H	TB21RG0H
AH	TB21RG1L
BH	TB21RG1H
CH	TB21CP0L
DH	TB21CP0H
EH	TB21CP1L
FH	TB21CP1H

ADR	Register name
FFFFF420H	TB22RUN
1H	TB22CR
2H	TB22MOD
3H	TB22FFCR
4H	TB22ST
5H	TB22IM
6H	TB22UCL
7H	TB22UCH
8H	TB22RG0L
9H	TB22RG0H
AH	TB22RG1L
BH	TB22RG1H
CH	TB22CP0L
DH	TB22CP0H
EH	TB22CP1L
FH	TB22CP1H

ADR	Register name
FFFFF430H	TB23RUN
1H	TB23CR
2H	TB23MOD
3H	TB23FFCR
4H	TB23ST
5H	TB23IM
6H	TB23UCL
7H	TB23UCH
8H	TB23RG0L
9H	TB23RG0H
AH	TB23RG1L
BH	TB23RG1H
CH	TB23CP0L
DH	TB23CP0H
EH	TB23CP1L
FH	TB23CP1H

[3] TMRC

ADR	Register name
FFFFF500H	TCACR
1H	TBTARUN
2H	TBTACR
3H	
4H	TBTACAPLL
5H	TBTACAPLH
6H	TBTACAPHL
7H	TBTACAPHH
8H	TBTARDCAPLL
9H	TBTARDCAPLH
AH	TBTARDCAPHL
BH	TBTARDCAPHH
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF510H	CMPA0CTL
1H	
2H	
3H	
4H	CMPA0LL
5H	CMPA0LH
6H	CMPA0HL
7H	CMPA0HH
8H	CMPA1CTL
9H	
AH	
BH	
CH	CMPA1LL
DH	CMPA1LH
EH	CMPA1HL
FH	CMPA1HH

ADR	Register name
FFFFF520H	CAPA0CR
1H	
2H	
3H	
4H	CAPA0LL
5H	CAPA0LH
6H	CAPA0HL
7H	CAPA0HH
8H	CAPA1CR
9H	
AH	
BH	
CH	CAPA1LL
DH	CAPA1LH
EH	CAPA1HL
FH	CAPA1HH

ADR	Register name
FFFFF530H	TCBCR
1H	TBTBRUN
2H	TBTBCR
3H	
4H	TBTBCAPLL
5H	TBTBCAPLH
6H	TBTBCAPHL
7H	TBTBCAPHH
8H	TBTBRDCAPLL
9H	TBTBRDCAPLH
AH	TBTBRDCAPHL
BH	TBTBRDCAPHH
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF540H	CMPB0CTL
1H	
2H	
3H	
4H	CMPB0LL
5H	CMPB0LH
6H	CMPB0HL
7H	CMPB0HH
8H	CMPB1CTL
9H	
AH	
BH	
CH	CMPB1LL
DH	CMPB1LH
EH	CMPB1HL
FH	CMPB1HH

ADR	Register name
FFFFF550H	CAPB0CR
1H	
2H	
3H	
4H	CAPB0LL
5H	CAPB0LH
6H	CAPB0HL
7H	CAPB0HH
8H	CAPB1CR
9H	
AH	
BH	
CH	CAPB1LL
DH	CAPB1LH
EH	CAPB1HL
FH	CAPB1HH

[4] SBI

ADR	Register name
FFFFF600H	SBI0CR1
1H	SBI0DBR
2H	SBI0I2CAR
3H	SBI0CR2/SR
4H	SBI0BR0
5H	
6H	
7H	SBI0CR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFF610H	SBI1CR1
1H	SBI1DBR
2H	SBI1I2CAR
3H	SBI1CR2/SR
4H	SBI1BR0
5H	
6H	
7H	SBI1CR0
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] SIO/UART

ADR	Register name
FFFFF700H	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
FFFFF710H	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
FFFFF720H	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
FFFFF730H	SC3BUF
1H	SC3CR
2H	SC3MOD0
3H	BR3CR
4H	BR3ADD
5H	SC3MOD1
6H	SC3MOD2
7H	SC3EN
8H	SC3RFC
9H	SC3TFC
AH	SC3RST
BH	SC3TST
CH	SC3FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF740H	SC4BUF
1H	SC4CR
2H	SC4MOD0
3H	BR4CR
4H	BR4ADD
5H	SC4MOD1
6H	SC4MOD2
7H	SC4EN
8H	SC4RFC
9H	SC4TFC
AH	SC4RST
BH	SC4TST
CH	SC4FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF750H	SC5BUF
1H	SC5CR
2H	SC5MOD0
3H	BR5CR
4H	BR5ADD
5H	SC5MOD1
6H	SC5MOD2
7H	SC5EN
8H	SC5RFC
9H	SC5TFC
AH	SC5RST
BH	SC5TST
CH	SC5FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF760H	SC6BUF
1H	SC6CR
2H	SC6MOD0
3H	BR6CR
4H	BR6ADD
5H	SC6MOD1
6H	SC6MOD2
7H	SC6EN
8H	SC6RFC
9H	SC6TFC
AH	SC6RST
BH	SC6TST
CH	SC6FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF770H	SC7BUF
1H	SC7CR
2H	SC7MOD0
3H	BR7CR
4H	BR7ADD
5H	SC7MOD1
6H	SC7MOD2
7H	SC7EN
8H	SC7RFC
9H	SC7TFC
AH	SC7RST
BH	SC7TST
CH	SC7FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF780H	SC8BUF
1H	SC8CR
2H	SC8MOD0
3H	BR8CR
4H	BR8ADD
5H	SC8MOD1
6H	SC8MOD2
7H	SC8EN
8H	SC8RFC
9H	SC8TFC
AH	SC8RST
BH	SC8TST
CH	SC8FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF790H	SC9BUF
1H	SC9CR
2H	SC9MOD0
3H	BR9CR
4H	BR9ADD
5H	SC9MOD1
6H	SC9MOD2
7H	SC9EN
8H	SC9RFC
9H	SC9TFC
AH	SC9RST
BH	SC9TST
CH	SC9FCNF
DH	
EH	
FH	

ADR	Register name
FFFFF7A0H	SCABUF
1H	SCACR
2H	SCAMOD0
3H	BRACR
4H	BRAADD
5H	SCAMOD1
6H	SCAMOD2
7H	SCAEN
8H	SCARFC
9H	SCATFC
AH	SCARST
BH	SCATST
CH	SCAFCNF
DH	
EH	
FH	

[6] ADC

ADR	Register name
FFFFF800H	ADAREG08L
1H	ADAREG08H
2H	ADAREG19L
3H	ADAREG19H
4H	ADAREG2AL
5H	ADAREG2AH
6H	ADAREG3BL
7H	ADAREG3BH
8H	ADAREG4CL
9H	ADAREG4CH
AH	ADAREG5DL
BH	ADAREG5DH
CH	ADAREG6EL
DH	ADAREG6EH
EH	ADAREG7FL
FH	ADAREG7FH

ADR	Register name
FFFFF810H	ADAREGSPL
1H	ADAREGSPH
2H	ADACOMREGL
3H	ADACOMREGH
4H	ADAMOD0
5H	ADAMOD1
6H	ADAMOD2
7H	ADAMOD3
8H	ADAMOD4
9H	ADACBAS0
AH	*reserved*
BH	*reserved*
CH	ADACLK
DH	*reserved*
EH	*reserved*
FH	

ADR	Register name
FFFFF820H	ADBREG08L
1H	ADBREG08H
2H	ADBREG19L
3H	ADBREG19H
4H	ADBREG2AL
5H	ADBREG2AH
6H	ADBREG3BL
7H	ADBREG3BH
8H	ADBREG4CL
9H	ADBREG4CH
AH	ADBREG5DL
BH	ADBREG5DH
CH	ADBREG6EL
DH	ADBREG6EH
EH	ADBREG7FL
FH	ADBREG7FH

ADR	Register name
FFFFF830H	ADBREGSPL
1H	ADBREGSPH
2H	ADBCOMREGL
3H	ADBCOMREGH
4H	ADBMOD0
5H	ADBMOD1
6H	ADBMOD2
7H	ADBMOD3
8H	ADBMOD4
9H	**ADBCBAS0**
AH	*reserved*
BH	*reserved*
CH	ADBCLK
DH	*reserved*
EH	*reserved*
FH	

[7]KWUP

ADR	Register name
FFFFF900H	KWUPST0
1H	KWUPST1
2H	KWUPST2
3H	KWUPST3
4H	KWUPST4
5H	KWUPST5
6H	KWUPST6
7H	KWUPST7
8H	
9H	
AH	
BH	
CH	
DH	
EH	KWUPST
FH	

[8] WDT

ADR	Register name
FFFFFA00H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] INTC

ADR	Register name
FFFFE000H	IMC0
1H	″
2H	″
3H	″
4H	IMC1
5H	″
6H	″
7H	″
8H	IMC2
9H	″
AH	″
BH	″
CH	IMC3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE010H	IMC4
1H	″
2H	″
3H	″
4H	IMC5
5H	″
6H	″
7H	″
8H	IMC6
9H	″
AH	″
BH	″
CH	IMC7
DH	″
EH	″
FH	″

ADR	Register name
FFFFE020H	IMC8
1H	″
2H	″
3H	″
4H	IMC9
5H	″
6H	″
7H	″
8H	IMCA
9H	″
AH	″
BH	″
CH	IMCB
DH	″
EH	″
FH	″

ADR	Register name
FFFFE030H	IMCC
1H	″
2H	″
3H	″
4H	IMCD
5H	″
6H	″
7H	″
8H	IMCE
9H	″
AH	″
BH	″
CH	IMCF
DH	″
EH	″
FH	″

ADR	Register name
FFFFE040H	IVR
1H	″
2H	″
3H	″
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	″
2H	″
3H	″
4H	**DREQFRG**
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	***reserved***
1H	″
2H	″
3H	″
4H	***reserved***
5H	″
6H	″
7H	″
8H	***reserved***
9H	″
AH	″
BH	″
CH	ILEV
DH	″
EH	″
FH	″

[10] DMAC

ADR	Register name
FFFFE200H	CCR0
1H	″
2H	″
3H	″
4H	CSR0
5H	″
6H	″
7H	″
8H	SAR0
9H	″
AH	″
BH	″
CH	DAR0
DH	″
EH	″
FH	″

ADR	Register name
FFFFE210H	BCR0
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR0
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE220H	CCR1
1H	″
2H	″
3H	″
4H	CSR1
5H	″
6H	″
7H	″
8H	SAR1
9H	″
AH	″
BH	″
CH	DAR1
DH	″
EH	″
FH	″

ADR	Register name
FFFFE230H	BCR1
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR1
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE240H	CCR2
1H	″
2H	″
3H	″
4H	CSR2
5H	″
6H	″
7H	″
8H	SAR2
9H	″
AH	″
BH	″
CH	DAR2
DH	″
EH	″
FH	″

ADR	Register name
FFFFE250H	BCR2
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR2
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE260H	CCR3
1H	″
2H	″
3H	″
4H	CSR3
5H	″
6H	″
7H	″
8H	SAR3
9H	″
AH	″
BH	″
CH	DAR3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE270H	BCR3
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR3
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE280H	CCR4
1H	″
2H	″
3H	″
4H	CSR4
5H	″
6H	″
7H	″
8H	SAR4
9H	″
AH	″
BH	″
CH	DAR4
DH	″
EH	″
FH	″

ADR	Register name
FFFFE290H	BCR4
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR4
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2A0H	CCR5
1H	″
2H	″
3H	″
4H	CSR5
5H	″
6H	″
7H	″
8H	SAR5
9H	″
AH	″
BH	″
CH	DAR5
DH	″
EH	″
FH	″

ADR	Register name
FFFFE2B0H	BCR5
1H	″
2H	″
3H	″
4H	
5H	
6H	
7H	
8H	DTCR5
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2C0H	CCR6
1H	〃
2H	〃
3H	〃
4H	CSR6
5H	〃
6H	〃
7H	〃
8H	SAR6
9H	〃
AH	〃
BH	〃
CH	DAR6
DH	〃
EH	〃
FH	〃

ADR	Register name
FFFFE2D0H	BCR6
1H	〃
2H	〃
3H	〃
4H	
5H	
6H	
7H	
8H	DTCR6
9H	〃
AH	〃
BH	〃
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE2E0H	CCR7
1H	〃
2H	〃
3H	〃
4H	CSR7
5H	〃
6H	〃
7H	〃
8H	SAR7
9H	〃
AH	〃
BH	〃
CH	DAR7
DH	〃
EH	〃
FH	〃

ADR	Register name
FFFFE2F0H	BCR7
1H	〃
2H	〃
3H	〃
4H	
5H	
6H	
7H	
8H	DTCR7
9H	〃
AH	〃
BH	〃
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE300H	DCR
1H	〃
2H	〃
3H	〃
4H	RSR
5H	〃
6H	〃
7H	〃
8H	
9H	
AH	
BH	
CH	DHR
DH	〃
EH	〃
FH	〃

[11] CS/WAIT controller

ADR	Register name
FFFFE400H	BMA0
1H	″
2H	″
3H	″
4H	BMA1
5H	″
6H	″
7H	″
8H	BMA2
9H	″
AH	″
BH	″
CH	BMA3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE410H	BMA4
1H	″
2H	″
3H	″
4H	BMA5
5H	″
6H	″
7H	″
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE480H	B01CS
1H	″
2H	″
3H	″
4H	B23CS
5H	″
6H	″
7H	″
8H	B45CS
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE500H	
1H	
2H	
3H	
4H	***reserved***
5H	″
6H	″
7H	″
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] FLASH control

ADR	Register name
FFFFE510H	SEQMOD
1H	″
2H	″
3H	″
4H	SEQCNT
5H	″
6H	″
7H	″
8H	ROMSEC1
9H	
AH	
BH	
CH	ROMSEC2
DH	
EH	
FH	

ADR	Register name
FFFFE520H	FLCS
1H	″
2H	″
3H	″
4H	FLPGEND
5H	″
6H	″
7H	″
8H	***reserved***
9H	″
AH	″
BH	″
CH	
DH	
EH	
FH	

[13] ROM correction

ADR	Register name
FFFFE540H	ADDREG0
1H	″
2H	″
3H	″
4H	ADDREG1
5H	″
6H	″
7H	″
8H	ADDREG2
9H	″
AH	″
BH	″
CH	ADDREG3
DH	″
EH	″
FH	″

ADR	Register name
FFFFE550H	ADDREG4
1H	″
2H	″
3H	″
4H	ADDREG5
5H	″
6H	″
7H	″
8H	ADDREG6
9H	″
AH	″
BH	″
CH	ADDREG7
DH	″
EH	″
FH	″

ADR	Register name
FFFFE560H	ADDREG8
1H	″
2H	″
3H	″
4H	ADDREG9
5H	″
6H	″
7H	″
8H	ADDREGA
9H	″
AH	″
BH	″
CH	ADDREGB
DH	″
EH	″
FH	″

[14] INTUNIT

ADR	Register name
FFFFE700H	ADCINT
1H	
2H	
3H	
4H	TMRBINTA
5H	
6H	
7H	
8H	TMRBINTB
9H	
AH	
BH	
CH	TMRBINTC
DH	
EH	
FH	

ADR	Register name
FFFFE710H	TMRBINTD
1H	
2H	
3H	
4H	TMRBINTE
5H	
6H	
7H	
8H	CAPINT
9H	
AH	
BH	
CH	CMPINT
DH	
EH	
FH	

ADR	Register name
FFFFE720H	TBTINT
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

ADR	Register name
FFFFE740H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] 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	〃
2H	〃
3H	〃
4H	IMCGB
5H	〃
6H	〃
7H	〃
8H	IMCGC
9H	〃
AH	〃
BH	〃
CH	IMCGD
DH	〃
EH	〃
FH	〃

ADR	Register name
FFFFEE20H	EICRCG
1H	〃
2H	〃
3H	〃
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	

20. JTAG Interface

The TMP19A63 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).

20.1 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. 20.1). 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.

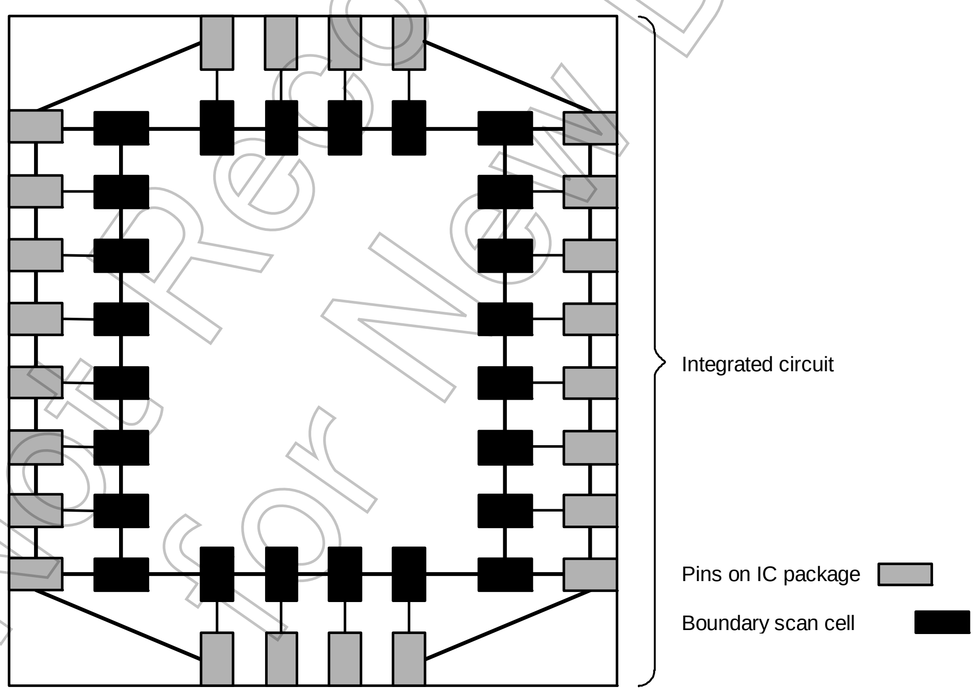


Fig.20.1 JTAG Boundary Scan Cells

(Note) The optional instructions IDCODE, USERCODE, INTEST and RUNBIST are not implemented in the TMP19A63.

20.2 JTAG Interface Signals

JTAG interface signals are as follows (see Fig. 20.2):

- 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

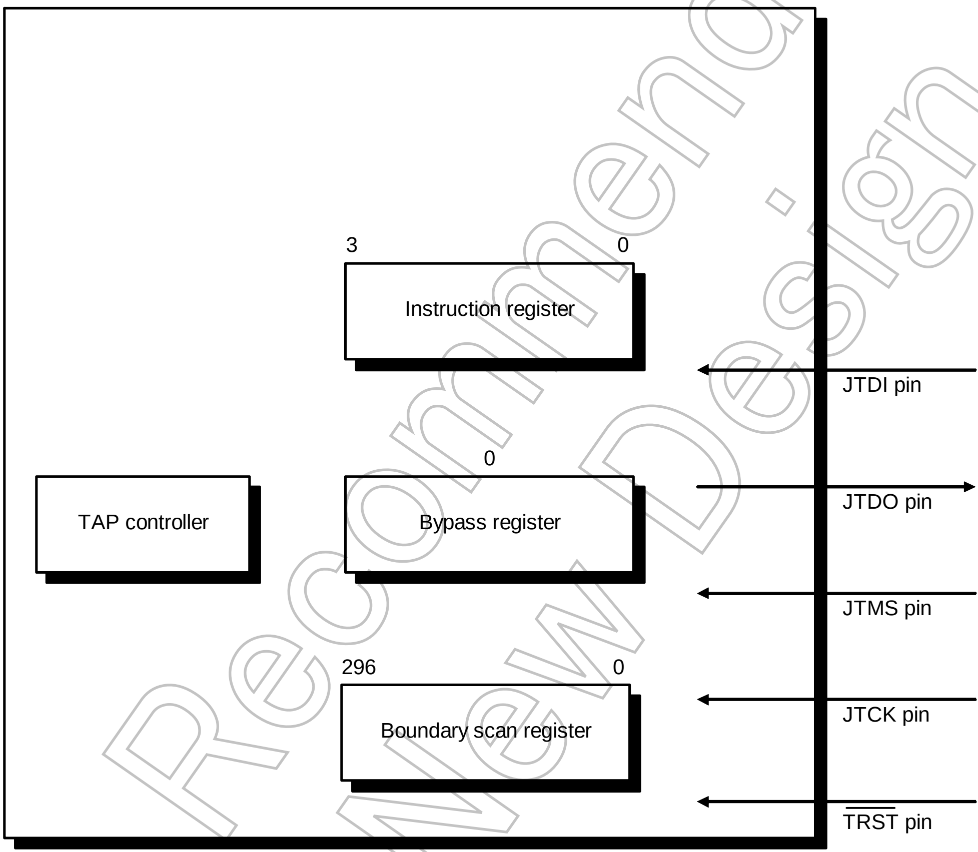


Fig. 20.2 Interface Signals and Registers

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.

20.3 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 20.1. 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.

20.3.1 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 20.1.

Table 20.1 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 ~1110	Reserved	Reserved
1111	BYPASS	Bypass register

Fig. 20.3 shows the format of the instruction register.

3	2	1	0
MSB			LSB

Fig. 20.3 Instruction register

The instruction code is shifted from the least significant bit to the instruction register.

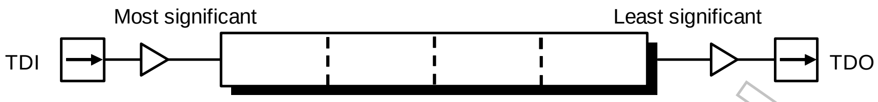


Fig. 20.4 Direction of a Shift of the Instruction Code to the Instruction Register

20.3.2 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. 20.5 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.

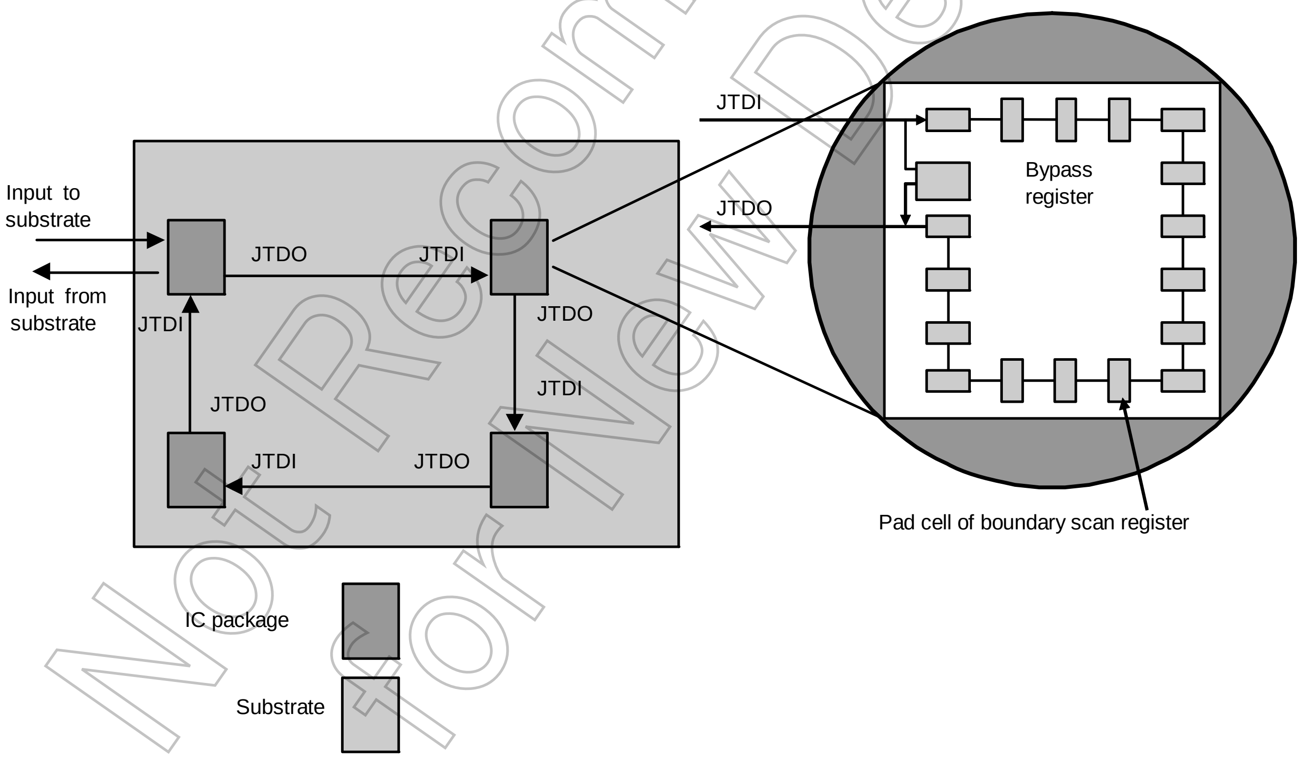


Fig. 20.5 Function of the Bypass Register

20.3.3 Boundary Scan Register

The boundary scan register has all the inputs and outputs for TMP19A63 except for some analog output signals and the control signals. Pins of the TMP19A63 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 TMP19A63.

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.

20.3.4 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. 20.6.

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.

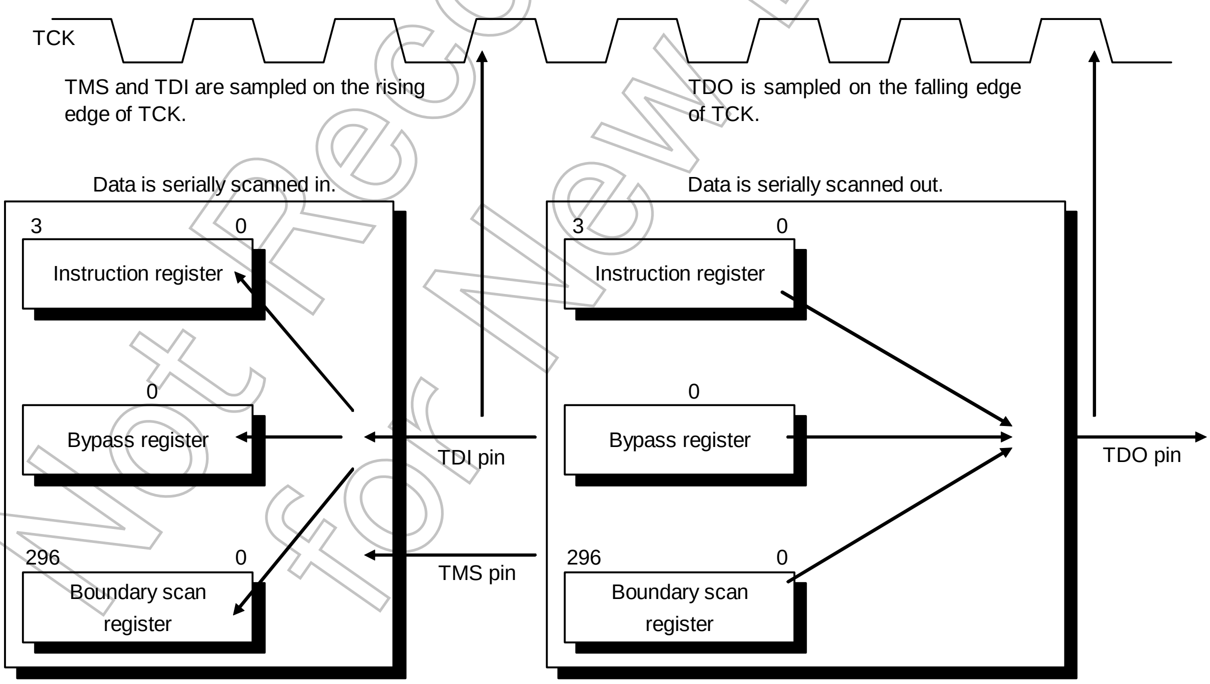


Fig. 20.6 JTAG Test Access Port

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.

20.3.5 TAP Controller

In the processor, a 16-state TAP controller specified in the IEEE JTAG standard is implemented.

20.3.6 Controller Reset

To reset the state machine of the TAP controller,

- assert the $\overline{\mathrm{TRST}}$ 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.

20.3.7 State Transitions of the TAP Controller

Fig. 20.7 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.

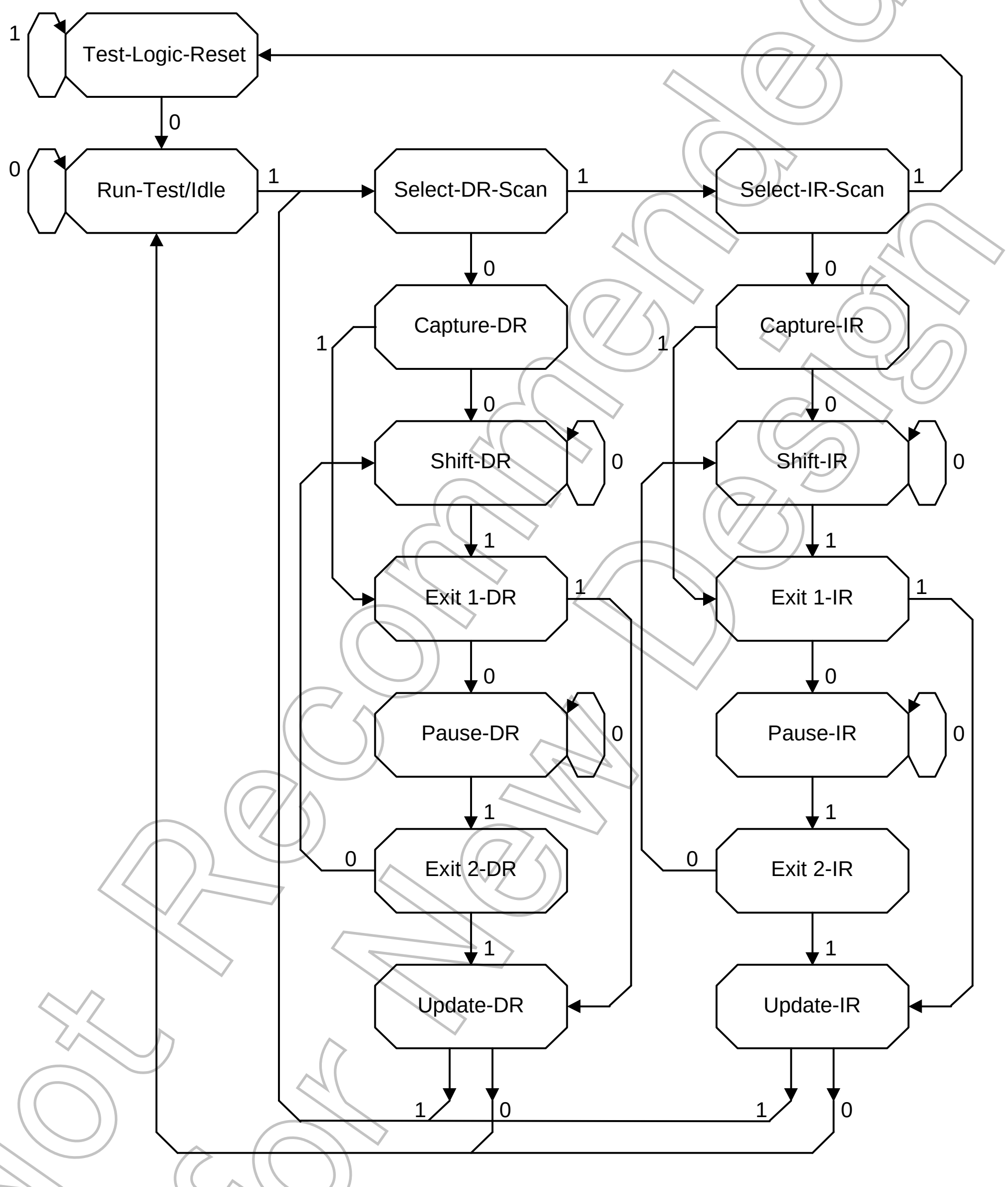


Fig. 20.7 State Transition Diagram of the TAP Controller

The TAP controller operates in each state described below. In Fig. 20.7, 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 20.2 shows the boundary scan sequence relative to processor signals.

Table 20.2 JTAG Scan Sequence Relative to the TMP19A63 Processor Pins

1 TOVR	2 PQ3	3 PQ2	4 PQ1	5 PQ0	6 PP7	7 PP6
8 PP5	9 PP4	10 PP3	11 PP2	12 PP0	13 PO7	14 PO6
15 PO5	16 PO4	17 PO3	18 PO2	19 PO1	20 PO0	21 PN7
22 PN6	23 PN5	24 PN4	25 PN3	26 PN2	27 PN1	28 PN0
29 PM7	30 PM6	31 PM5	32 PM4	33 PM3	34 PM2	35 PM1
36 PM0	37 PLLSEL	38 PL7	39 PL6	40 PL5	41 PL4	42 PL3
43 PL2	44 PL1	45 PL0	46 PK7	47 PK6	48 PK5	49 PK4
50 PK3	51 PK2	52 PK1	53 PK0	54 PJ7	55 PJ6	56 PJ5
57 PJ4	58 PJ3	59 PJ2	60 PJ1	61 PJ0	62 PI7	63 PI6
64 PI5	65 PI4	66 PI3	67 PI2	68 PI1	69 PI0	70 PH7
71 PH6	72 PH5	73 PH4	74 PH3	75 PH2	76 PH1	77 PH0
78 PG7	79 PG6	80 PG5	81 PG4	82 PG3	83 PG2	84 PG1
85 PG0	86 PF7	87 PF6	88 PF5	89 PF4	90 PF3	91 PF2
92 PF1	93 PF0	94 PE7	95 PE6	96 PE5	97 PE4	98 PE3
99 PE2	100 PE1	101 PE0	102 PD7	103 PD6	104 PD5	105 PD4
106 PD3	107 PD2	108 PD1	109 PD0	110 PCST4	111 PCST3	112 PCST2
113 PCST1	114 PCST0	115 PC7	116 PC6	117 PC5	118 PC4	119 PC3
120 PC2	121 PC1	122 PC0	123 PB7	124 PB6	125 PB5	126 PB4
127 PB3	128 PB2	129 PB1	130 PB0	131 PA7	132 PA6	133 PA5
134 PA4	135 PA3	136 PA2	137 PA1	138 PA0	139 P97	140 P96
141 P95	142 P94	143 P93	144 P92	145 P91	146 P90	147 P87
148 P86	149 P85	150 P84	151 P83	152 P82	153 P81	154 P80
155 P77	156 P76	157 P75	158 P74	159 P73	160 P72	161 P71
162 P70	163 P67	164 P66	165 P65	166 P64	167 P63	168 P62
169 P61	170 P60	171 P57	172 P56	173 P55	174 P54	175 P53
176 P52	177 P51	178 P50	179 P47	180 P46	181 P45	182 P44
183 P43	184 P42	185 P41	186 P40	187 P37	188 P36	189 P35
190 P34	191 P33	192 P32	193 P31	194 P30	195 P27	196 P26
197 P25	198 P24	199 P23	200 P22	201 P21	202 P20	203 P17
204 P16	205 P15	206 P14	207 P13	208 P12	209 P11	210 P10
211 P07	212 P06	213 P05	214 P04	215 P03	216 P02	217 P01
218 P00	219 NMI	220 ENDIAN	221 DINT	222 DCLK	223 BW1	224 BW0
225 BUSMD						

Note: The pins shown above are JTAG scan available.

20.4 Instructions Supported by the JTAG Controller Cells

This section describes the instructions supported by the JTAG controller cells of the TMP19A63.

20.4.1 EXTEST instruction

The EXTEST instruction is used for external interconnect test. If this instruction is issued, the BSR cells at output pins output test patterns in the Update-DR state, and the BSR cells at input pins capture test results in the Capture-DR state.

Before the EXTEST instruction is selected, the boundary scan register is usually initialized using the SAMPLE/PRELOAD instruction. If the boundary scan register has not been initialized, there is the possibility that indeterminate data will be transmitted in the Update-DR state and bus conflicts may occur between ICs. Fig. 20.8 shows the flow of data while the EXTEST instruction is selected.

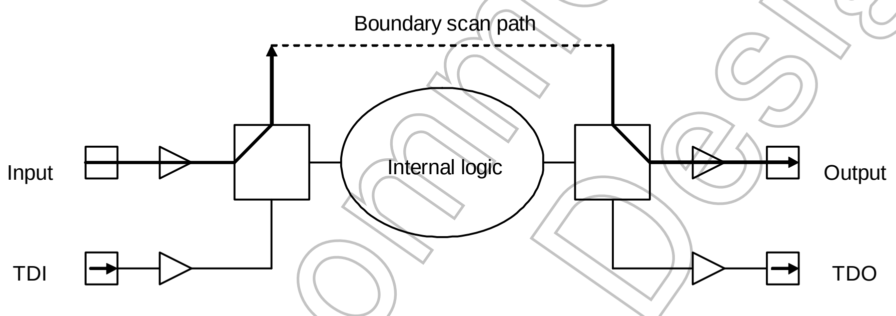


Fig.20.8 Flow of Data While the EXTEST Instruction Is Selected

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 the test pattern that was shifted into the boundary scan register for output to the output pin.

Repeat steps 6 through 8 for each test pattern.

(Note) When using EXTEST instruction, please note that malfunction may occur depending on the data input from terminal pin on ground that CPU is in operating state and make sure to execute the test after the system reset is released

20.4.2 SAMPLE/PRELOAD Instructions

The SAMPLE and PRELOAD instructions are used to connect TDI and TDO by way of the boundary scan register. This instruction has two functions.

- 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. 20-9 shows the flow of data while the SAMPLE instruction is selected.

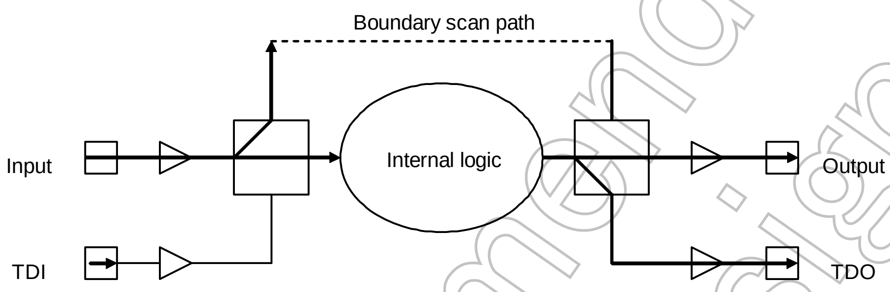


Fig. 20.9 Flow of Data While SAMPLE Is Selected

- 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. 20.10 shows the flow of data while the PRELOAD instruction is selected.

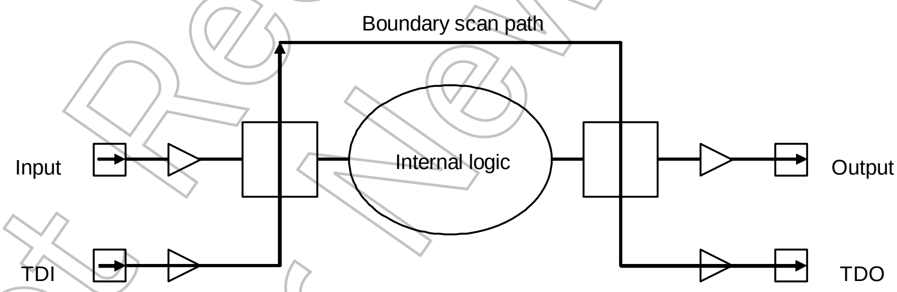


Fig. 20.10 Flow of Test Data While PRELOAD Is Selected

When using the SAMPLE instruction, complete the instruction update during the system reset. After the reset is released, do not switch the TAP instruction.

20.4.3 BYPASS instruction

When conducting the type of test in which an IC does not need to be controlled or monitored, the BYPASS instruction is used to form the shortest serial path bypassing an IC by connecting the bypass register between JTDI and JTDO. The BYPASS instruction does not affect the normal operation of the system logic implemented on a chip. Data passes through the bypass register while the BYPASS instruction is selected, as shown in Fig. 20.11.

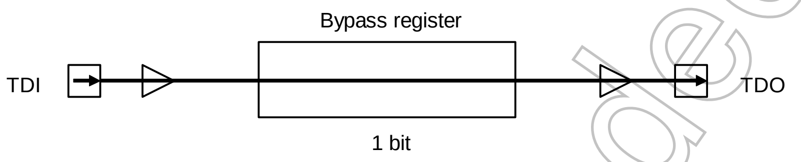


Fig. 20.11 Flow of Data While the Bypass Register Is Selected

20.5 Points to Note

This section describes the points to note regarding JTAG boundary scan operations implemented in this processor.

- The X2 and X1 signal pads do not comply with JTAG.

- To reset the JTAG circuit, execute either of the following:
  ① Initialize the JTAG circuit by asserting $\overline{\text{TRST}}$, and then negate $\overline{\text{TRST}}$.
  ② Set the TMS pin to "1," and supply TCK with more than 5 clocks.

21. Flash Memory Operation

This section describes the hardware configuration and operation of the flash memory.

Flash Memory

Features

1) Memory capacity
The TMP19A63 F10XBG device contains 8M bits (1024kB) of flash memory capacity. The memory area consists of 8 independent memory blocks (128 kB × 8) to enable independent write access to each block. When the CPU is to access the internal flash memory, 32-bit data bus width is used.

2) Flash memory access
Interleave access is used in this device.

3) Write/erase time
Write time: 1sec/Chip (Typ) 0.5sec/128Kbyte (Typ.)
Erase: 0.2sec/Chip (Typ) 100msec/128Kbyte(Typ.)
(Note) The above values are theoretical values not including data transfer time.
The write time per chip depends on the write method to be used by the user.

Programming method
The onboard programming mode is available for the user to program (rewrite) the device while it is mounted on the user's board.

4-1) User boot mode
The user's original rewriting method can be supported.

4-2) Single boot mode
The rewriting method to use serial data transfer (Toshiba's unique method) can be supported.

Rewriting method
The flash memory included in this device is generally compliant with the applicable JEDEC standards except for some specific functions. Therefore, if the user is currently using an external flash memory device, it is easy to implement the functions into this device. Furthermore, the user is not required to build his/her own programs to realize complicated write and erase functions because such functions are automatically performed using the circuits already built-in the flash memory chip.
This device is also implemented with a read-protect function to inhibit reading flash memory data from any external writer device. On the other hand, rewrite protection is available only through command-based software programming; any hardware setting method to apply +12VDC is not supported. The above described protection function is automatically enabled when all the two area are configured for protection. When the user removes protection, the internal data is automatically erased before the protection is actually removed.

JEDEC compliant functions	Modified, added, or deleted functions
• Automatic programming • Automatic chip erase • Automatic block erase • Data polling/toggle bit	<Modified> Block protect (only software protection is supported) <Deleted> Erase resume - suspend function <Added>Automatic multiple block erase (supported to the chip level)

Block Diagram of the Flash Memory Section

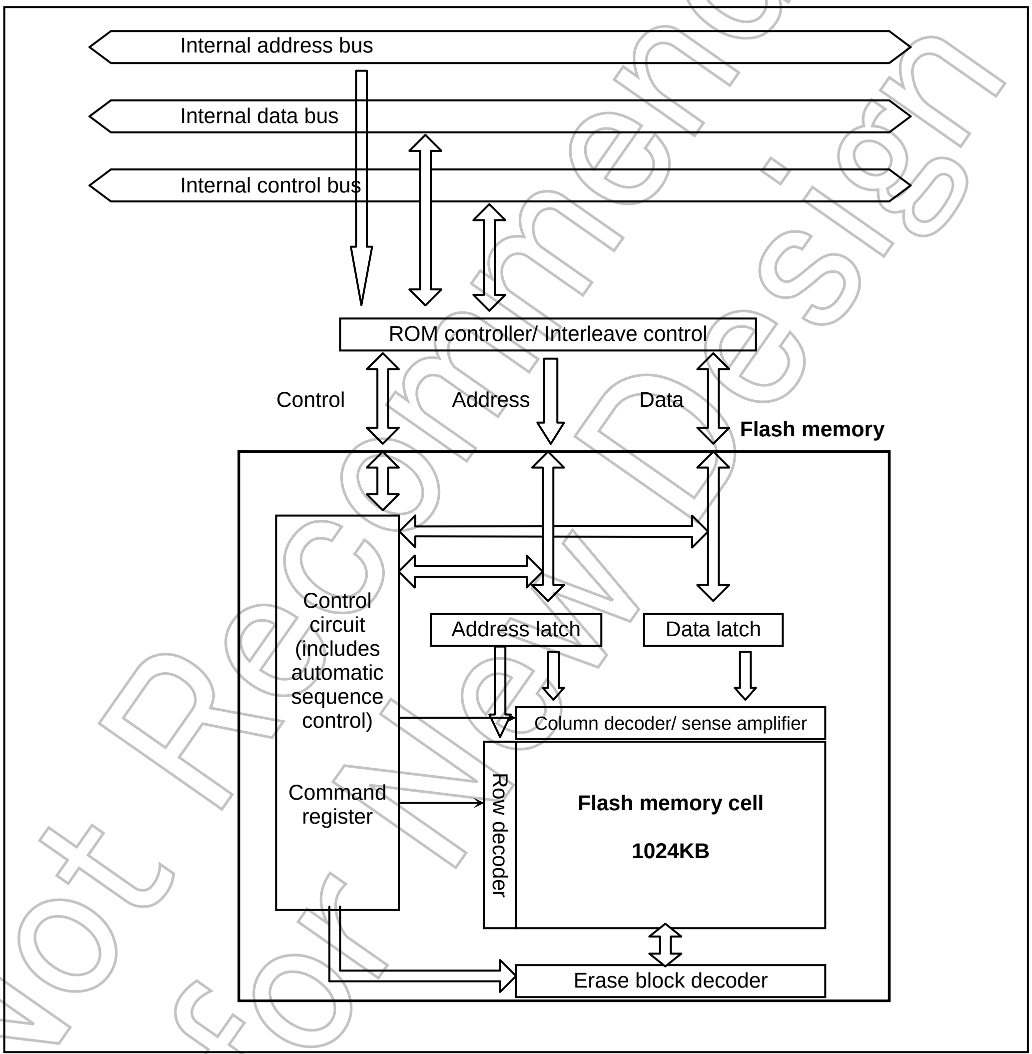


Fig. 21.1 Block Diagram of the Flash Memory Section

Operation Mode

This device has two operation modes including the mode not to use the internal flash memory.

Table21.1 Operation Modes

Operation mode	Operation details
Single chip mode	After reset is cleared, it starts up from the internal flash memory.
Normal mode	In this operation mode, two different modes, i.e., the mode to execute user application programs and the mode to rewrite the flash memory onboard the user's card, are defined. The former is referred to as "normal mode" and the latter "user boot mode."
User boot mode	The user can uniquely configure the system to switch between these two modes. For example, the user can freely design the system such that the normal mode is selected when the port "00" is set to "1" and the user boot mode is selected when it is set to "0." The user should prepare a routine as part of the application program to make the decision on the selection of the modes.

Among the flash memory operation modes listed in the above table, the User Boot mode is the programmable modes. This mode is referred to as "Onboard Programming" modes where onboard rewriting of internal flash memory can be made on the user's card.

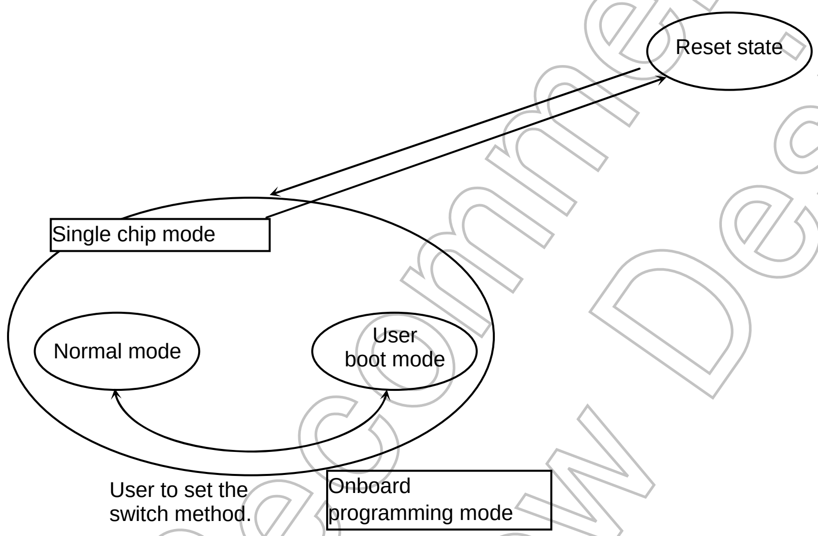
Either the Single Chip or Single Boot operation mode can be selected by externally setting the level of the $\overline{\text{BOOT}}$ input pin while the device is in reset status.

After the level is set, the CPU starts operation in the selected operation mode when the reset condition is removed. Regarding the BOOT pin, be sure not to change the levels during operation once the mode is selected.
The mode setting method and the mode transition diagram are shown below:

Table 21.2 Operation Mode Setting

Operation mode	Input pin
	$\overline{\text{RESET}}$
Single chip mode	0 → 1

Fig. 21.2 Mode Transition Diagram



21.2.1 Reset Operation

To reset the device, ensure that the power supply voltage is within the operating voltage range, that the internal oscillator has been stabilized, and that the $\overline{\text{RESET}}$ input is held at "0" for a minimum duration of 12 system clocks (1.8 μs with 54MHz operation; the "1/8" clock gear mode is applied after reset).

(Note 1) Regarding power-on reset of devices with internal flash memory;
For devices with internal flash memory, it is necessary to apply "0" to the $\overline{\text{RESET}}$ inputs upon power on for a minimum duration of 500 microseconds regardless of the operating frequency.

(Note 2) While flash programming or deletion is in progress, at least 0.5 microseconds of reset period is required regardless of the system clock frequency.

21.2.2 DSU (EJTAG) - PROBE Interface

This interface is used when the DSU probe is used in debugging. This is the dedicated interface for connection to the DSU probe. Please refer to the operation manual for the DSU probe you are going to use for details of debugging procedures to use the DSU probe. Here, the function to enable/disable the DSU probe in the DSU (EJTAG) mode is described.

1) Protect function

This device allows use of on-board DSU probes for debugging. To facilitate this, the device is implemented with a protection function to prevent easy reading of the internal flash memory by a third party other than the authorized user. By enabling the protection function, it becomes impossible to read the internal flash memory from a DSU probe. Use this function together with the protection function of the internal flash memory itself as described later.

2) DSU probe enable/disable function

This device allows use of on-board DSU probes for debugging operations. To facilitate this, the device is implemented with the "DSU probe inhibit" function (hereafter referred to as the **"DSU inhibit" function**) to prevent easy reading of the internal flash memory by a third party other than the authorized user. By enabling the DSU inhibit function, use of any DSU probe becomes impossible.

3) DSU enable (Enables use of DSU probes for debugging)

In order to prevent the DSU inhibit function from being accidentally removed by system runaway, etc., the method to cancel the DSU inhibit function is in double protection structure so it is necessary to set SEQMOD<DSUOFF> to "0" and also write the protect code "0x0000_00C5" to the DSU protect control register SEQCNT to cancel the function. Then, debugging to use a DSU probe can be allowed. While power to the device is still applied, setting SEQMOD <SEQON> to "1" and writing "0x0000_00C5" to the SEQCNT register will enable the protection function again.

Table 21.3 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	Always reads "0."							1: DSU disable 0: DSU disable
	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								
	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 32-bit accessed.
(Note 2) This register is initialized only by power-on reset. It is not initialized by a normal reset.

Table 21.4 DSU Protect Control Register

SEQCNT (0xFFFF_E514)

	7	6	5	4	3	2	1	0
Bit Symbol								
Read/Write	W							
After reset								
Function	Write “0x0000_00C5”.							
	15	14	13	12	11	10	9	8
Bit Symbol								
Read/Write	W							
After reset								
Function	Write “0x0000_00C5”.							
	23	22	21	20	19	18	17	16
Bit Symbol								
Read/Write	W							
After reset								
Function	Write “0x0000_00C5”.							
	31	30	29	28	27	26	25	24
Bit Symbol								
Read/Write	W							
After reset								
Function	Write “0x0000_00C5”.							

(Note 1) This register must be 32-bit accessed.

4) Example use by the user

An example to use a DSU probe together with this function is shown as follows:

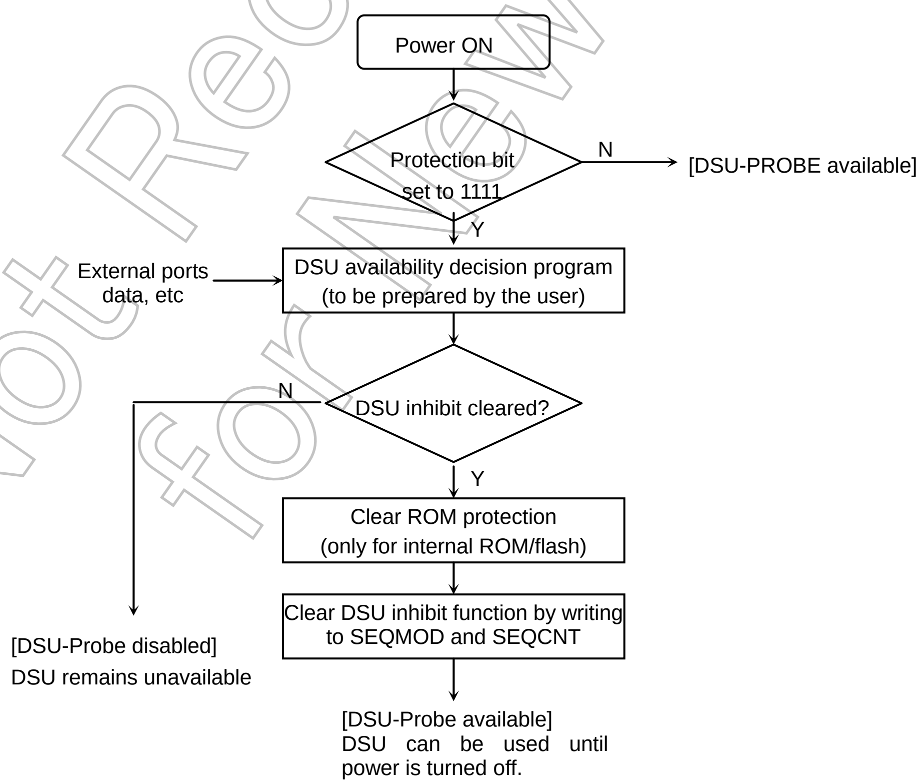


Fig. 21.3 Example Use of DSU Inhibit Function

21.2.3 User Boot Mode (Single chip mode)

User Boot mode is to use flash memory programming routine defined by users. It is used when the data transfer buses for flash memory program code on the old application and for serial I/O are different. It operates at the single chip mode; therefore, a switch from normal mode in which user application is activated at the single chip mode to User Boot Mode for programming flash is required. Specifically, add a mode judgment routine to a reset program in the old application.

The condition to switch the modes needs to be set by using the I/O of 19A63 in conformity with the user's system setup condition. Also, flash memory programming routine that the user uniquely makes up needs to be set in the new application. This routine is used for programming after being switched to User Boot Mode. The execution of the programming routine must take place while it is stored in the area other than the flash memory since the data in the internal flash memory cannot be read out during delete/ writing mode. Once re-programming is complete, it is recommended to protect relevant flash blocks from accidental corruption during subsequent Single-Chip (Normal mode) operations. All the interruption including a non-maskable are inhibited at User Boot Mode.

(1-A) and (1-B) are the examples of programming with routines in the internal flash memory and in the external memory. For a detailed description of the erase and program sequence, refer to 21.3 On-board Programming of Flash Memory (Rewrite/Erase).

User Boot Mode

(1-A) Storing a Programming Routine in the Flash Memory

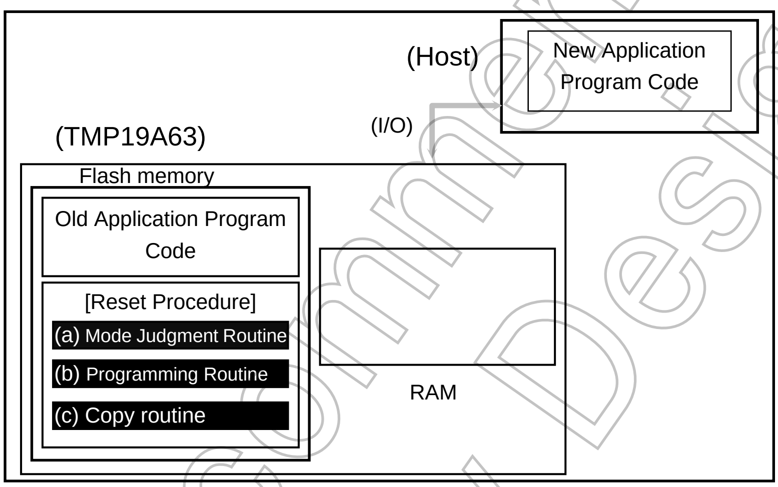
(Step-1)

Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP19A63 on a printed circuit board, write the following three program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

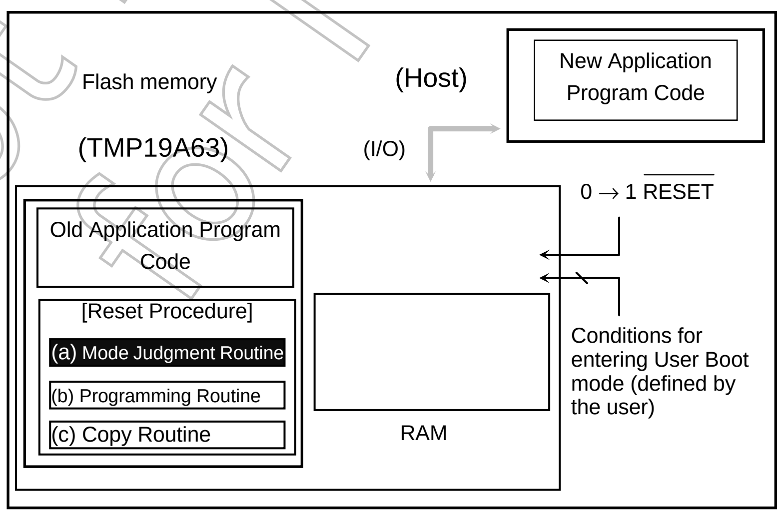
(b) Programming routine: Code to download new program code from a host controller and re-program the flash memory

(c) Copy routine: Code to copy the flash programming routine from the TMP19A63 flash memory to either the TMP19A63 on-chip RAM or external memory device.



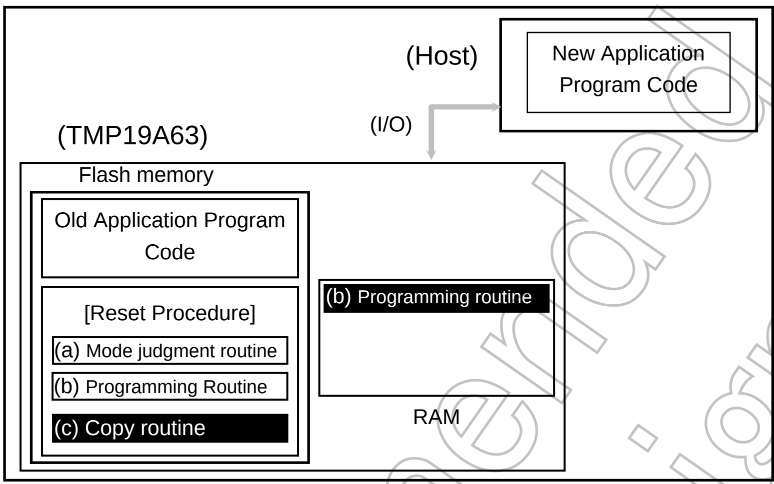
(Step-2)

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP19A63 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode.)



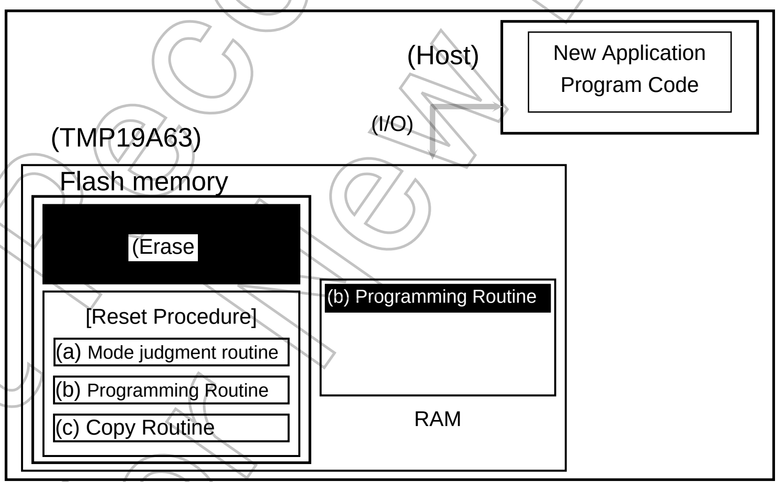
(Step-3)

Once transition to User Boot mode is occurred, execute the copy routine (c) to copy the flash programming routine (b) to either the TMP19A63 on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used).



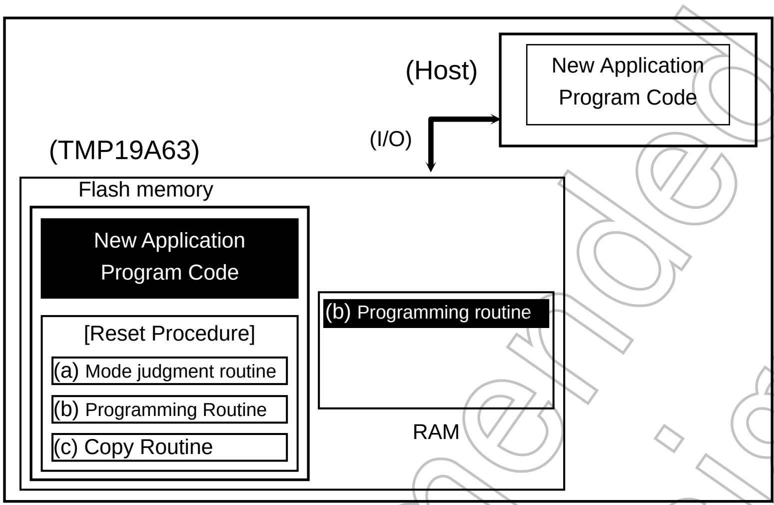
(Step-4)

Jump to the flash programming routine in the on-chip RAM. Cancel the protection for overwriting to erase a flash block containing the old application program code.



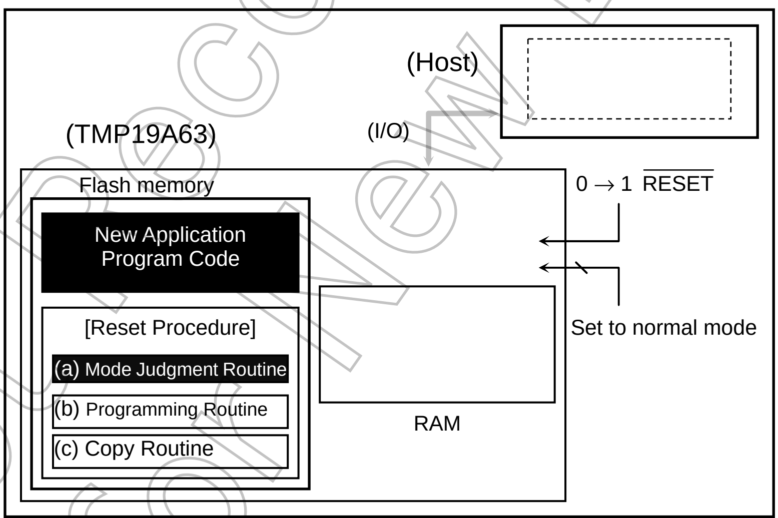
(Step-5)

Continue executing the flash programming routine to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(Step-6)

Set $\overline{\text{RESET}}$ to “0” to reset the TMP19A63. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



(1-B) Transferring a Programming Routine from an External Host

(Step-1)

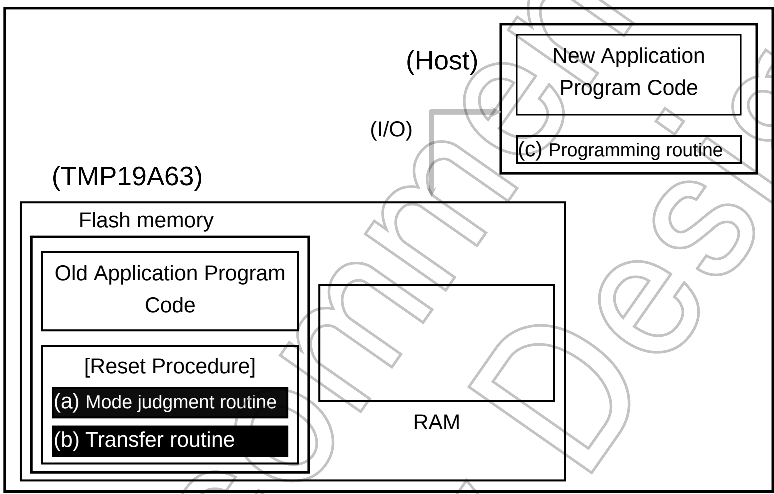
Determine the conditions (e.g., pin states) required for the flash memory to enter User Boot mode and the I/O bus to be used to transfer new program code. Create hardware and software accordingly. Before installing the TMP19A63 on a printed circuit board, write the following two program routines into an arbitrary flash block using programming equipment.

(a) Mode judgment routine: Code to determine whether or not to switch to User Boot mode

(b) Transfer routine: Code to download new program code from a host controller

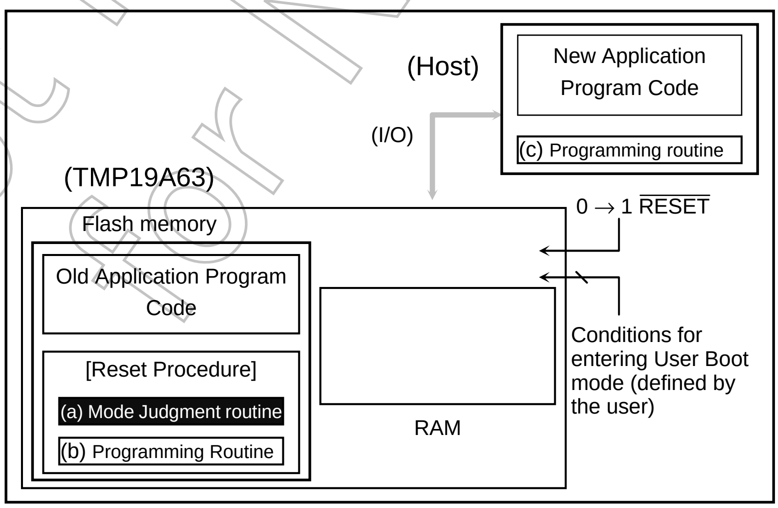
Also, prepare a programming routine shown below on the host controller:

(c) Programming routine: Code to download new program code from an external host controller and re-program the flash memory



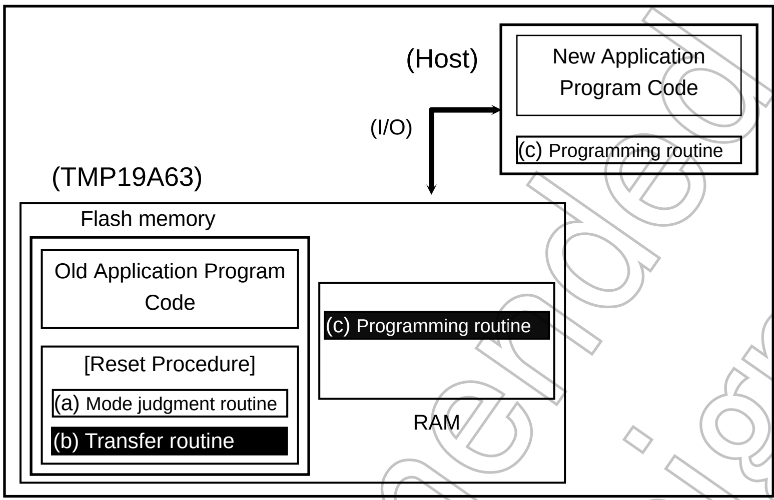
(Step-2)

After $\overline{\text{RESET}}$ is released, the reset procedure determines whether to put the TMP19A63 flash memory in User Boot mode. If mode switching conditions are met, the flash memory enters User Boot mode. (All interrupts including NMI must be disabled while in User Boot mode).



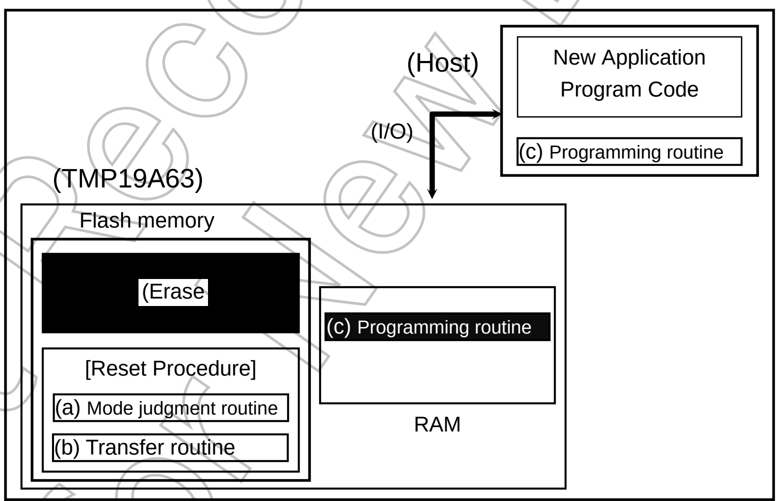
(Step-3)

Once User Boot mode is entered, execute the transfer routine (b) to download the flash programming routine (c) from the host controller to either the TMP19A63 on-chip RAM or an external memory device. (In the following figure, the on-chip RAM is used).



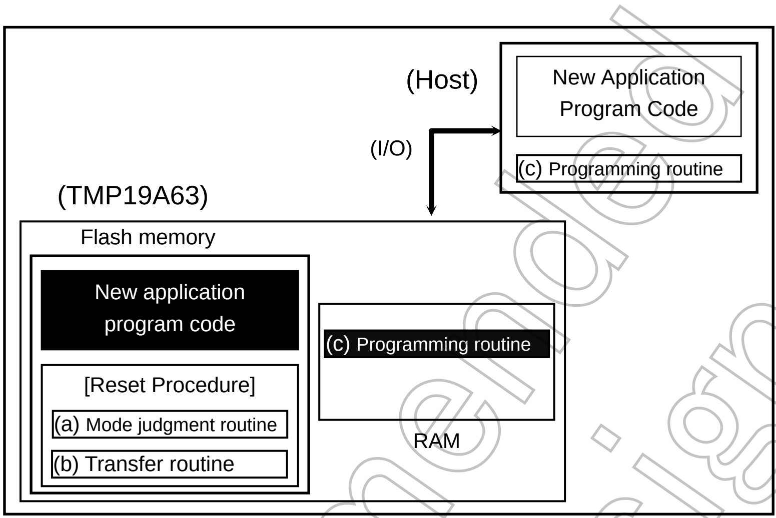
(Step-4)

Jump to the flash programming routine in the on-chip RAM. Cancel the protection for overwriting to erase a flash block containing the old application program code.



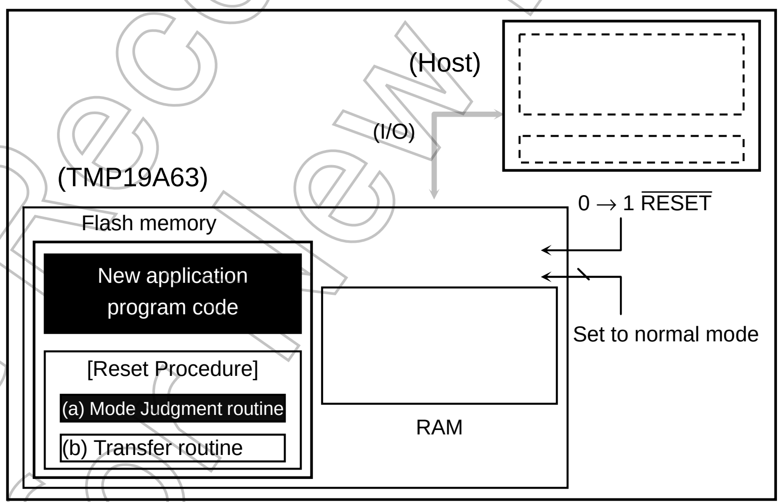
(Step-5)

Continue executing the flash programming routine (c) to download new program code from the host controller and program it into the erased flash block. Once programming is complete, turn on the protection of that flash block.



(Step-6)

Set $\overline{\text{RESET}}$ to “0” to reset the TMP19A63. Upon reset, the on-chip flash memory is put in Normal mode. After $\overline{\text{RESET}}$ is released, the CPU will start executing the new application program code.



21.3 On-board Programming of Flash Memory (Rewrite/Erase)

In on-board programming, the CPU is to execute software commands for rewriting or erasing the flash memory. The rewrite/erase control program should be prepared by the user beforehand. Because the flash memory content cannot be read while it is being written or erased, it is necessary to run the rewrite/erase program from the internal RAM or from an external memory device after shifting to the user boot mode. In this section, flash memory addresses are represented in virtual addresses unless otherwise noted.

21.3.1 Flash Memory

Except for some functions, writing and erasing flash memory data are in accordance with the standard JEDEC commands. In writing or erasing, use the SW command of the CPU to enter commands to the flash memory. Once the command is entered, the actual write or erase operation is automatically performed internally.

Table 21.5 Flash Memory Functions

Major functions	Description
Automatic page program	Writes data automatically.
Automatic chip erase	Erases the entire area of the flash memory automatically.
Automatic block erase	Erases a selected block automatically. (128 kB at a time)
Write protect	The write or erase function can be individually inhibited for each block (of 128 kB). When all blocks are set for protection, the entire protection function is automatically enabled.
Protect function	By writing a 4-bit protection code, the write or erase function can be individually inhibited for each block.

Note that addressing of operation commands is different from the case of standard commands due to the specific interface arrangements with the CPU as detailed operation of the user boot mode and RAM transfer mode is described later. Also note that the flash memory is written in 32-bit blocks. So, 32-bit (word) data transfer commands must be used in writing the flash memory.

(1) Block configuration

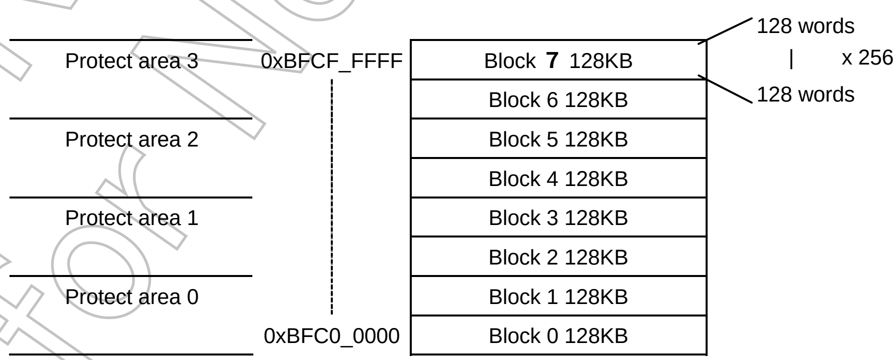


Fig.21.4 Block Configuration of Flash Memory

(2) Basic operation

Generally speaking, this flash memory device has the following two operation modes:

- The mode to read memory data (Read mode)
- The mode to automatically erase or rewrite memory data (Automatic operation)

Transition to the automatic mode is made by executing a command sequence while it is in the memory read mode. In the automatic operation mode, flash memory data cannot be read and any commands stored in the flash memory cannot be executed. In the automatic operation mode, any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated. During automatic operation, be sure not to cause any exceptions other than debug exceptions and reset while a DSU probe is connected. Any interrupt or exception generation cannot set the device to the read mode except when a hardware reset is generated.

1) Read

When data is to be read, the flash memory must be set to the read mode. The flash memory will be set to the read mode immediately after power is applied, when CPU reset is removed, or when an automatic operation is normally terminated. In order to return to the read mode from other modes or after an automatic operation has been abnormally terminated, either the Read/reset command (a software command to be described later) or a hardware reset is used. The device must also be in the read mode when any command written on the flash memory is to be executed.

- **Read/reset command and Read command (software reset)**

When an automatic operation is abnormally terminated, the flash memory cannot return to the read mode by itself (When FLCS<RDY/BSY> = 0, data read from the flash memory is undefined.) In this case, the Read/reset command can be used to return the flash memory to the read mode. Also, when a command that has not been completely written has to be canceled, the Read/reset command must be used to return to the read mode. The Read command is used to return to the read mode after executing the SW command to write the data "0x0000_00F0" to an arbitrary address of the flash memory.

- **With the Read/reset command, the device is returned to the read mode after completing the third bus write cycle.**

2) Command write

This flash memory uses the command control method. Commands are executed by executing a command sequence to the flash memory. The flash memory executes automatic operation commands according to the address and data combinations applied (refer to Command Sequence).

If it is desired to cancel a command write operation already in progress or when any incorrect command sequence has been entered, the Read/reset command is to be executed. Then, the flash memory will terminate the command execution and return to the read mode.

While commands are generally comprised of several bus cycles, the operation to apply the SW command to the flash memory is called "bus write cycle." The bus write cycles are to be in a specific sequential order and the flash memory will perform an automatic operation when the sequence of the bus write cycle data and address of a command write operation is in accordance with a predefined specific sequence. If any bus write cycle does not follow a predefined command write sequence, the flash memory will terminate the command execution and return to the read mode. The address [31:21] in each bus write cycle should be the virtual address [31:21] of command execution. It will be explained later for the address bits [20:8].

(Note 1) Command sequences are executed from outside the flash memory area.

(Note 2) The interval between bus write cycles for this device must be 15 system clock cycles or longer. The command sequencer in the flash memory device requires a certain time period to recognize a bus write cycle. If more than one bus write cycles are executed within this time period, normal operation cannot be expected. For adjusting the applicable bus write cycle interval using a software timer to be operated at the operating frequency, use the section 10) "ID-Read" to check for the appropriateness.

(Note 3) Between the bus write cycles, never use any load command (such as LW, LH, or LB) to the flash memory or perform a DMA transmission by specifying the flash area as the source address. Also, don't execute a Jump command to the flash memory. While a command sequence is being executed, don't generate any interrupt such as maskable interrupts (except debug exceptions when a DSU probe is connected).

If such an operation is made, it can result in an unexpected read access to the flash memory and the command sequencer may not be able to correctly recognize the command. While it could cause an abnormal termination of the command sequence, it is also possible that the written command is incorrectly recognized.

(Note 4) The SYNC command must be executed immediately after the SW command for each bus write cycle.

(Note 5) For the command sequencer to recognize a command, the device must be in the read mode prior to executing the command. Be sure to check before the first bus write cycle that the FLCS[0] RDY/BSY bit is set to "1." It is recommended to subsequently execute a Read command.

(Note 6) Upon issuing a command, if any address or data is incorrectly written, be sure to perform a system reset operation or issue a reset command to return to the read mode again.

3) Reset

Hardware reset

The flash memory has a reset input as the memory block and it is connected to the CPU reset signal. Therefore, when the RESET input pin of this device is set to V_{IL} or when the CPU is reset due to any overflow of the watch dog timer, the flash memory will return to the read mode terminating any automatic operation that may be in progress. The CPU reset is also used in returning to the read mode when an automatic operation is abnormally terminated or when any mode set by a command is to be canceled. It should also be noted that applying a hardware reset during an automatic operation can result in incorrect rewriting of data. In such a case, be sure to perform the rewriting again.

Refer to Section 21.2.1 "Reset Operation" for CPU reset operations. After a given reset input, the CPU will read the reset vector data from the flash memory and starts operation after the reset is removed.

4) Automatic Page Programming

Writing to a flash memory device is to make "1" data cells to "0" data cells. Any "0" data cell cannot be changed to a "1" data cell. For making "0" data cells to "1" data cells, it is necessary to perform an erase operation.

The automatic page programming function of this device writes data in 128 word blocks. A 128 word block is defined by a same [31:9] address and it starts from the address [8:0] = 0 and ends at the address [8:0] = 0x1FF. This programming unit is hereafter referred to as a "page."

Writing to data cells is automatically performed by an internal sequencer and no external control by the CPU is required. The state of automatic page programming (whether it is in writing operation or not) can be checked by the FLCS [0] <RDY/BSY> register.

Also, any new command sequence is not accepted while it is in the automatic page programming mode. If it is desired to interrupt the automatic page programming, use the hardware reset function. If the operation is stopped by a hardware reset operation, it is necessary to once erase the page and then perform the automatic page programming again because writing to the page has not been normally terminated.

The automatic page programming operation is allowed only once for a page already erased. No programming can be performed twice or more times irrespective of the data cell value whether it is "1" or "0." Note that rewriting to a page that has been once written requires execution of the automatic block erase or automatic chip erase command before executing the automatic page programming command again. Note that an attempt to rewrite a page twice or more without erasing the content can cause damages to the device.

No automatic verify operation is performed internally to the device. So, be sure to read the data programmed to confirm that it has been correctly written.

The automatic page programming operation starts when the fourth bus write cycle of the command cycle is completed. On and after the fifth bus write cycle, data will be written sequentially starting from the next address of the address specified in the fourth bus write cycle (in the fourth bus write cycle, the page top address will be command written) (32 bits of data is input at a time). Be sure to use the SW command in writing commands on and after the fourth bus cycle. In this, any SW command shall not be placed across word boundary. On and after the fifth bus write cycle, data is command written to the same page area. Even if it is desired to write the page only partially, it is required to perform the automatic page programming for the entire page. In this case, the address input for the fourth bus write cycle shall be set to the top address of the page. Be sure to perform command write operation with the input data set to "1" for the data cells not to be set to "0." For example, if the top address of a page is not to be written, set the input data of the fourth bus write cycle to 0xFFFFFFFF to command write the data.

Once the fourth bus cycle is executed, it is in the automatic programming operation. This condition can be checked by monitoring the register bit FLCS [0] <RDY/BSY> (See Table 21.16). Any new command sequence is not accepted while it is in automatic page programming mode. If it is desired to stop operation, use the hardware reset function. Be careful in doing so because data cannot be written normally if the operation is interrupted. When a single page has been command written normally terminating the automatic page writing process, the FLCS [0] <RDY/BSY> bit is set to "1" and it returns to the read mode.

When multiple pages are to be written, it is necessary to execute the page programming command for each page because the number of pages to be written by a single execution of the automatic page program command is limited to only one page. It is not allowed for automatic page programming to process input data across pages.

Data cannot be written to a protected block. When automatic programming is finished, it automatically returns to the read mode. This condition can be checked by monitoring FLCS [0] <RDY/BSY> (See Table 21.16). If automatic programming has failed, the flash memory is locked in the mode and will not return to the read mode. For returning to the read mode, it is necessary to use the reset command or hardware reset to reset the flash memory or the device. In this case, while writing to the address has failed, it is recommended not to use the device or not to use the block that includes the failed address.

Note: Software reset becomes ineffective in bus write cycles on and after the fourth bus write cycle of the automatic page programming command.

5) Automatic chip erase
The automatic chip erase operation starts when the sixth bus write cycle of the command cycle is completed.

This condition can be checked by monitoring FLCS [0] <RDY/BSY> (See Table 21.16). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic chip erase operation. If it is desired to stop operation, use the hardware reset function. If the operation is forced to stop, it is necessary to perform the automatic chip erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If all the blocks are protected, the automatic chip erase operation will not be performed and it returns to the read mode after completing the sixth bus read cycle of the command sequence. When an automatic chip erase operation is normally terminated, it automatically returns to the read mode. If an automatic chip erase operation has failed, the flash memory is locked in the mode and will not return to the read mode.

For returning to the read mode, it is necessary to use the reset command or hardware reset to reset the flash memory or the device. In this case, the failed block cannot be detected. It is recommended not to use the device anymore or to identify the failed block by using the block erase function for not to use the identified block anymore.

6) Automatic block erase (128 kB at a time)
The automatic block erase operation starts when the sixth bus write cycle of the command cycle is completed.

This status of the automatic block erase operation can be checked by monitoring FLCS [0] <RDY/BSY> (See Table 21.16). While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that data has been correctly erased. Any new command sequence is not accepted while it is in an automatic block erase operation. If it is desired to stop operation, use the hardware reset function. In this case, it is necessary to perform the automatic block erase operation again because the data erasing operation has not been normally terminated.

Also, any protected blocks cannot be erased. If an automatic block erase operation has failed, the flash memory is locked in the mode and will not return to the read mode. In this case, use the reset command or hardware reset to reset the flash memory or the device.

7) Automatic programming of protection bits
This device is implemented with four protection bits. The protection bits can be individually set in the automatic programming. The applicable protection bit is specified in the seventh bus write cycle. By automatically programming the protection bits, write and/or erase functions can be inhibited (for protection) individually for each block. The protection status of each block can be checked by the FLCS <BLPRO 3:0> register to be described later. This status of the automatic programming operation to set protection bits can be checked by monitoring FLCS <RDY/BSY> (See Table 21.16). Any new command sequence is not accepted while automatic programming is in progress to program the protection bits. If it is desired to stop the programming operation, use the hardware reset function. In this case, it is necessary to perform the programming operation again because the protection bits may not have been correctly programmed. If all the protection bits have been programmed, the flash memory cannot be read from any area outside the flash memory such as the internal RAM. In this condition, the FLCS <BLPRO 3:0> bits are set to "0 x F" indicating that it is in the protected state (See Table 21.16). After this, no command writing can be performed.

Note: Software reset is ineffective in the seventh bus write cycle of the automatic protection bit programming command. The FLCS <RDY/BSY> bit turns to "0" after entering the seventh bus write cycle.

8) Automatic erasing of protection bits
Different results will be obtained when the automatic protection bit erase command is executed depending on the status of the protection bits. It depends on the status of FLCS <BLPRO 3:0> before the command execution whether it is set to "0 x F" or to any other values. Be sure to check the value of FLCS <BLPRO 3:0> before executing the automatic protection bit erase command.

- **When FLCS <BLPRO 3:0> is set to "0 x F" (all the protection bits are programmed):**

When the automatic protection bit erase command is command written, the flash memory is automatically initialized within the device. When the seventh bus write cycle is completed, the entire area of the flash memory data cells is erased and then the protection bits are erased. This operation can be checked by monitoring FLCS <RDY/BSY>. If the automatic operation to erase protection bits is normally terminated, FLCS will be set to "0x01." While no automatic verify operation is performed internally to the device, be sure to read the data to confirm that it has been correctly erased. For returning to the read mode while the automatic operation after the seventh bus cycle is in progress, it is necessary to use the hardware reset to reset the flash memory or the device. If this is done, it is necessary to check the status of protection bits by FLCS <BLPRO 3:0> after retuning to the read mode and perform either the automatic protection bit erase, automatic chip erase, or automatic block erase operation, as appropriate.

- **When FLCS <BLPRO 3:0> is other than "0 x F" (not all the protection bits are programmed):**

The protection condition can be canceled by the automatic protection bit erase operation. With this device, protection bits can be erased handling two bits at a time. The target bits are specified in the seventh bus write cycle and when the command is completed, the device is in a condition the two bits are erased. The protection status of each block can be checked by FLCS <BLPRO 3:0> to be described later. This status of the programming operation for automatic protection bits can be checked by monitoring FLCS <RDY/BSY>. When the automatic operation to erase protection bits is normally terminated, the two protection bits of FLCS <BLPRO 3:0> selected for erasure are set to "0."

In any case, any new command sequence is not accepted while it is in an automatic operation to erase protection bits. If it is desired to stop the operation, use the hardware reset function. When the automatic operation to erase protection bits is normally terminated, it returns to the read mode.

The FLCS <RDY/BSY> bit is "0" while in automatic operation and it turns to "1" when the automatic operation is terminated.

9) Flash control/ status register

This resister is used to monitor the status of the flash memory and to indicate the block protection status.

Table 21.6 Flash Control Register

		7	6	5	4	3	2	1	0
FLCS	Bit Symbol	BLPRO3	BLPRO2	BLPRO1	BLPRO0				RDY/BSY
(0xFFFF_E520)	Read/Write	R				R	R	R	R
	After reset	0	0	0	0	0	0	0	1
	Function	Protection area setting (for each 256 kB) 0000:No blocks are protected xxx1:Block 0 is protected xx1x:Block 1 is protected x1xx:Block 2 is protected 1xxx:Block 3 is protected				Always reads "0."	Always reads "0."	Always reads "0."	Ready/Busy 0: in operation 1: Operation is terminated.
		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								

Bit 0: Ready/Busy flag bit

The RDY/BSY output is provided as a means to monitor the status of automatic operation. This bit is a function bit for the CPU to monitor the function. When the flash memory is in automatic operation, it outputs "0" to indicate that it is busy. When the automatic operation is terminated, it returns to the ready state and outputs "1" to accept the next command. If the automatic operation has failed, this bit maintains the "0" output. By applying a hardware reset, it returns to "1."

(Note) Please issue it after confirming the command issue is always a ready state. A normal command not only is sent when the command is issued to a busy inside but also there is a possibility that the command after that cannot be input. In that case, please return by system reset or the reset command.

Bits [7:4]: Protection status bits (can be set to any combination of blocks)

Each of the protection bits (4 bits) represents the protection status of the corresponding block. When a bit is set to "1," it indicates that the block corresponding to the bit is protected. When the block is protected, data cannot be written to it.

10) ID-Read

Using the ID-Read command, you can obtain the type and other information on the flash memory contained in the device. The data to be loaded will be different depending on the address [15:14] of the fourth and subsequent bus write cycles (any input data other than 0xF can be used). On and after the fourth bus write cycle, when an LW command (to read an arbitrary flash memory area) is executed after an SW command, the ID value will be loaded (execute a SYNC command immediately after the LW command). Once the fourth bus write cycle of an ID-Read command has passed, the device will not automatically return to the read mode. In this condition, the set of the fourth bus write cycle and LW/SYNC commands can be repetitively executed. For returning to the read mode, reset the system or use the Read or Read/reset command.

(Important) The "interval between bus write cycles" between successive command sequences must be 15 system clock cycles or longer irrespective of the operating frequency used. This device doesn't have any function to automatically adjust the interval between bus write cycles regarding execution of multiple SW commands to the flash memory. Therefore, if an inadequate interval is used between two sets of bus write cycles, the flash memory cannot be written as expected. Prior to setting the device to work in the onboard programming mode, adjust the bus write cycle interval using a software timer, etc., to verify that the ID-Read command can be successfully executed at the operating frequency of the application program. In the onboard programming mode, use the bus write cycle interval at which the ID-Read command can be operated normally to execute command sequences to rewrite the flash memory.

(3) List of Command Sequences

Table 21.7 Flash Memory Access from the Internal CPU

Command sequence	First bus cycle Addr. / Data	Second bus cycle Addr. / Data	Third bus cycle Addr. / Data	Fourth bus cycle Addr. / Data	Fifth bus cycle Addr. / Data	Sixth bus cycle Addr. / Data	Seventh bus cycle Addr. / Data
Read	0xXX	RA					
	0xF0	RD					
Read/reset	0x55XX	0xAAXX	0x55XX	RA			
	0xAA	0x55	0xF0	RD			
ID-Read	0x55XX	0xAAXX	0x55XX	IA	0xXX	–	
	0xAA	0x55	0x90	0x00	ID	–	
Automatic page programming (note)	0x55XX	0xAAXX	0x55XX	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic chip erase	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	0x55XX	–
	0xAA	0x55	0x80	0xAA	0x55	0x10	–
Auto Block erase (note)	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	BA	–
	0xAA	0x55	0x80	0xAA	0x55	0x30	–
Protection bit programming	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	0x55XX	PBA
	0xAA	0x55	0x9A	0xAA	0x55	0x9A	0x9A
Protection bit erase	0x55XX	0xAAXX	0x55XX	0x55XX	0xAAXX	0x55XX	PBA
	0xAA	0x55	0x6A	0xAA	0x55	0x6A	0x6A

(4) Supplementary explanation

- RA: Read address
- **RD: Read data**
- IA: ID address
- **ID: ID data**
- PA: Program page address
  PD: Program data (32-bit data)
  After the fourth bus cycle, enter data in the order of the address for a page.
- BA: Block address
- PBA: Protection bit address

(Note 1) Always set "0" to the address bits [1:0] in the entire bus cycle. (Setting values to bits [7:2] are undefined.)

(Note 2) Bus cycles are "bus write cycles" except for the second bus cycle of the Read command, the fourth bus cycle of the Read/reset command, and the fifth bus cycle of the ID-Read command. Bus write cycles are executed by SW commands. Use "Data" in the table for the store data of SW commands. The address [31:16] in each bus write cycle should be the target flash memory address [31:16] of the command sequence. Use "Addr." in the table for the address [15:0].

(Note 3) In executing the bus write cycles, the interval between each bus write cycle shall be 15 system clocks or more.

(Note 4) The "Sync command" must be executed immediately after completing each bus write cycle.

(Note 5) Execute the "Sync command" immediately following the "LW command" after the fourth bus write cycle of the ID-Read command.

(5) Address bit configuration for bus write cycles

Table 21.8 Address Bit Configuration for Bus Write Cycles

<table>
<tr><td>Address</td><td>Addr [31:21]</td><td>Addr [20]</td><td>Addr [19]</td><td>Addr [18:17]</td><td>Addr [16]</td><td>Addr [15]</td><td>Addr [14]</td><td>Addr [13]</td><td>Addr [12:9]</td><td>Addr [8]</td><td>Addr [7:0]</td></tr>
<tr><td rowspan="2">Normal commands</td><td colspan="11">Normal bus write cycle address configuration</td></tr>
<tr><td>Flash area</td><td colspan="4">"0" is recommended</td><td colspan="5">Command</td><td>Addr [1:0]=0 (fixed), Others: 0 (recommended)</td></tr>
<tr><td rowspan="2">Block erase</td><td colspan="11">BA: Block address (Set the sixth bus write cycle address for block erase operation)</td></tr>
<tr><td>Flash area</td><td colspan="2">"0" is recommended</td><td>Block selection</td><td colspan="7">Addr [1:0]=0 (fixed), Others: 0 (recommended)</td></tr>
<tr><td rowspan="2">Auto page programming</td><td colspan="11">PA: Program page address (Set the fourth bus write cycle address for page programming operation)</td></tr>
<tr><td>Flash area</td><td colspan="2">"0" is recommended</td><td>Block selection</td><td colspan="5">Page selection</td><td colspan="2">Addr [1:0]=0 (fixed), Others: 0 (recommended)</td></tr>
<tr><td rowspan="2">ID-READ</td><td colspan="11">IA: ID address (Set the fourth bus write cycle address for ID-Read operation)</td></tr>
<tr><td>Flash area</td><td colspan="4">"0" is recommended</td><td colspan="2">ID address</td><td colspan="4">Addr [1:0]=0 (fixed), Others: 0 (recommended)</td></tr>
<tr><td rowspan="2">Protection bit programming</td><td colspan="11">PBA: Protection bit address (Set the seventh bus write cycle address for protection bit programming)</td></tr>
<tr><td>Flash area</td><td colspan="4">"0" is recommended</td><td colspan="2">Protection bit write
“00”: Block 0
“01”: Block 1
“10”: Block 2
“11”: Block 3</td><td colspan="4">Addr [1:0]=0 (fixed), Others: 0 (recommended)</td></tr>
<tr><td rowspan="2">Protection bit erase</td><td colspan="11">PBA: Protection bit address (Set the seventh bus write cycle address for protection bit erasure)</td></tr>
<tr><td>Flash area</td><td colspan="4">"0" is recommended</td><td>Erase protection for
0:Block 0,1
1:Block 2,3</td><td colspan="5">Addr [1:0]=0 (fixed), Others: 0 (recommended)</td></tr>
</table>

(Note) Table 21.17 "Flash Memory Access from the Internal CPU" can also be used.

(Note) Address setting can be performed according to the "Normal bus write cycle address configuration" from the first bus cycle.

(Note) "0" is recommended" can be changed as necessary.

Table 21.9 Block Erase Address Table

BA	Address Range		Size
	Flash Memory Address	When applied to the projected area	
Block 0	0xBFC0_0000~0xBFC1_FFFF	0x0000_0000~0x0001_FFFF	128 KB
Block 1	0xBFC2_0000~0xBFC3_FFFF	0x0002_0000~0x0003_FFFF	128 KB
Block 2	0xBFC4_0000~0xBFC5_FFFF	0x0004_0000~0x0005_FFFF	128 KB
Block 3	0xBFC6_0000~0xBFC7_FFFF	0x0006_0000~0x0007_FFFF	128 KB
Block 4	0xBFC8_0000~0xBFC9_FFFF	0x0008_0000~0x0009_FFFF	128 KB
Block 5	0xBFCA_0000~0xBFCB_FFFF	0x000A_0000~0x000B_FFFF	128 KB
Block 6	0xBFCC_0000~0xBFCD_FFFF	0x000C_0000~0x000D_FFFF	128 KB
Block 7	0xBFCE_0000~0xBFCF_FFFF	0x000E_0000~0x000F_FFFF	128 KB

Example: When BA0 is to be selected, any single address in the range 0xBFC0_0000 to 0xBFC1_FFFF may be entered.

As for the addresses from the first to the sixth bus cycles, specify the upper 4 bit with the corresponding flash memory addresses of the blocks to be erased.

Table 21.10 Protection Bit Programming Address Table

OPBA	The seventh bus write cycle address [15:14]	
	Address [15]	Address [14]
Block 0	0	0
Block 1	0	1
Block 2	1	0
Block 3	1	1

Table 21.11 Protection Bit Erase Address Table

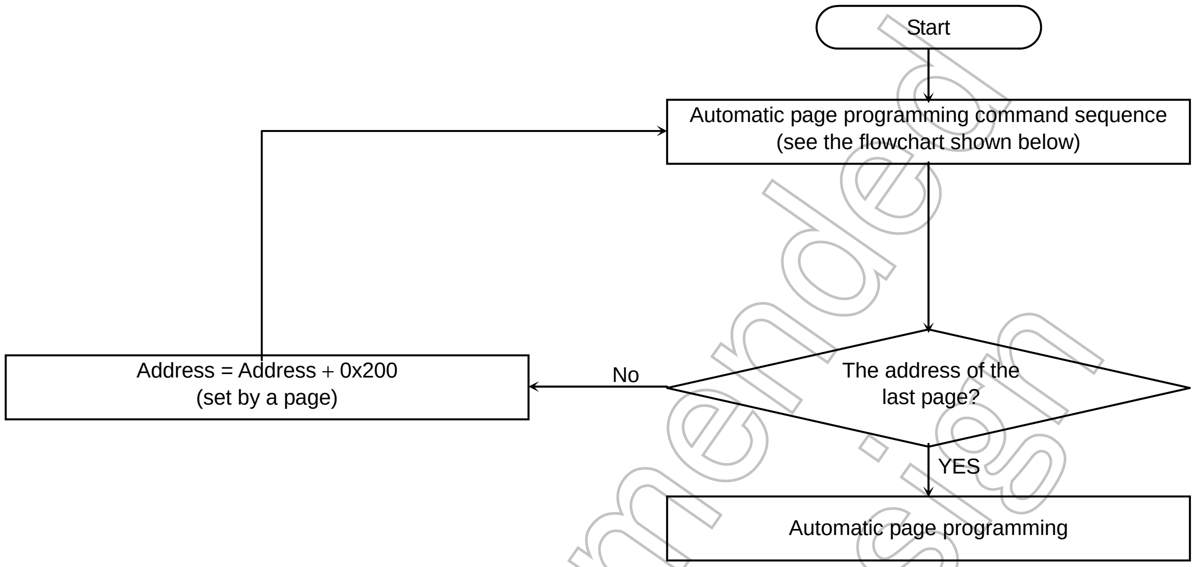
OPBA	The seventh bus write cycle address [15:14]	
	Address [15]	Address [14]
Block 0	0	X
Block 1	0	X
Block 2	1	X
Block 3	1	X

The protection bit erase command will erase bits 0 and 1 together.
The bits 2 and 3 are also erased together.

Table 21.12 The ID-Read command's fourth bus write cycle ID address (IA) and the data to be read by the following LW command (ID)

IA [15:14]	ID [7: 0]	Code
00b	0x98	Manufacturer code
01b	0x5A	Device code
10b	Reserved	---
11b	0x08	Macro code

(6) Flowchart



Automatic Page Programming Command Sequence (Address/ Command)

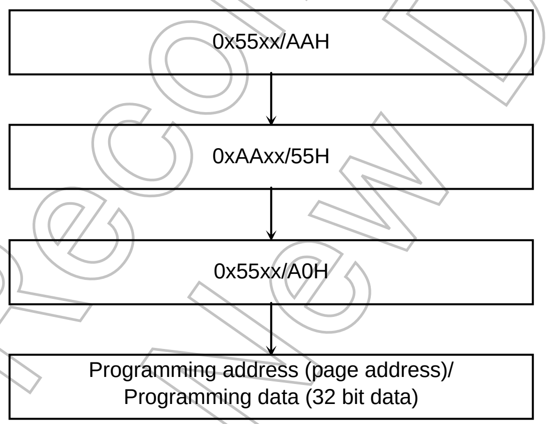


Fig. 21.5 Automatic Programming

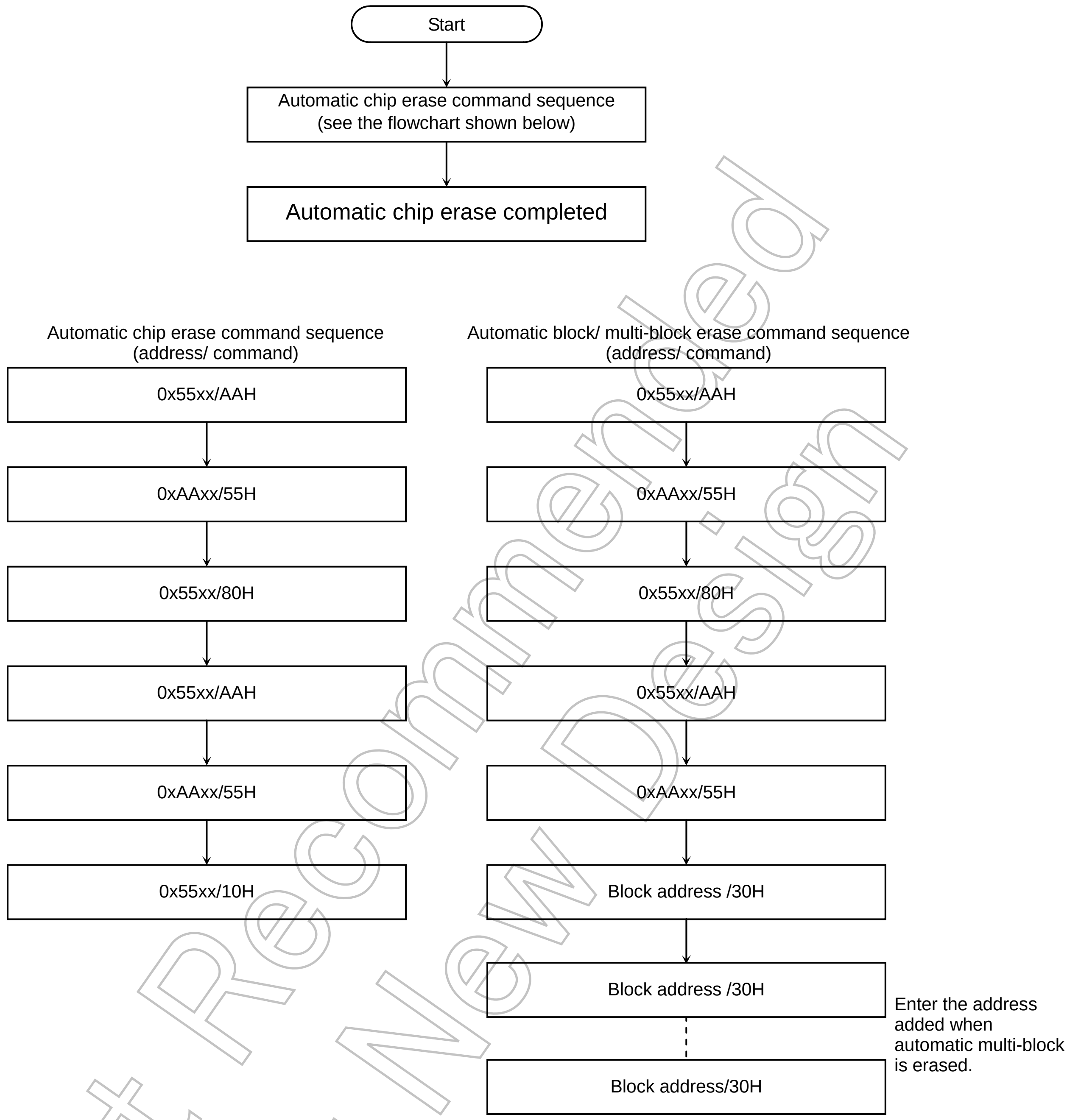


Fig.21.6 Automatic Erase

23. Electrical Characteristics

23.1 Absolute Maximum Ratings

The letter x in equations represents the cycle period of the fsys clock selected through the programming of the SYSCR1 <SYSCK> bit. The x value may vary if clock gear or low-speed oscillator is used. All relevant values in this chapter are calculated with the high-speed (fc) system clock (SYSCR1.<SYSCK> = 0) and a clock gear factor of 1/1 (SYSCR1.GEAR[1:0] = 00).

Parameter		Symbol	Rating	Unit
Supply voltage		Vcc15(Core)	– 0.3~3.0	V
		Vcc3(I/O)	– 0.3~3.9	
		AVCC(A/D)	– 0.3~3.9	
		FVCC3	– 0.3~3.9	
Input voltage		V_{IN}	– 0.3~V_{CC}+0.3	V
Low-level output current	Per pin	I_{OL}	5	mA
	Total	ΣI_{OL}	50	
High-level output current	Per pin	I_{OH}	-5	
	Total	ΣI_{OH}	50	
Power consumption (Ta = 85˚C)		PD	600	mW
Soldering temperature (10s)		T_{SOLDER}	260	˚C
Storage temperature		T_{STG}	-40~125	˚C
Operating Temperature	Except during Flash W/E	T_{OPR}	-20~85	˚C
	During Flash W/E		0~70	
Write/erase cycles		N_{EW}	100	cycle

V_{CC15}=DVCC15=CVCC15=FVCC15, V_{CC}3=DVCC3n (n=0~4),

AVCC=AVCC3m (m=1~2), V_{SS}=DVSS*=AVSS*=CVSS

Note: Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions. The equipment manufacturer should design so that no Absolute maximum rating value is exceeded with respect to current, voltage, power consumption, temperature, etc. Exposure to conditions beyond those listed above may cause permanent damage to the device or affect device reliability, which could increase potential risks of personal injury due to IC blowup and/or burning.

23.2 DC Electrical Characteristics (1/4)

Ta=-20~85˚C (n=0~4, m=1, 2)

<table>
<tr><th colspan="2">Parameter</th><th>Symbol</th><th>Rating</th><th>Min.</th><th>Typ. (Note 1)</th><th>Max.</th><th>Unit</th></tr>
<tr><td colspan="2" rowspan="2">Supply voltage
CVCC15=DVCC15
CVSS=DVSS=0V</td><td>DVCC15</td><td>fosc = 8~13.5MHz
fsys = 4MHz~54MH
PLLON、</td><td>1.35</td><td rowspan="2"></td><td>1.65</td><td rowspan="2">V</td></tr>
<tr><td>DVCC3n (n=0~4)</td><td>fsys = 4~54MHz</td><td>1.65</td><td>3.3</td></tr>
<tr><td rowspan="5">Low-level input voltage</td><td>P7~PA</td><td>V_{IL1}</td><td>2.7V≤AVCC32≤AVCC31≤3.3V</td><td rowspan="5">–0.3</td><td rowspan="5"></td><td>0.3AVCC31
0.3AVCC32</td><td rowspan="5">V</td></tr>
<tr><td>Normal port</td><td>V_{IL2}</td><td>1.65V≤DVCC3n≤3.3V (n=0~4)</td><td>0.3DVCC3n</td></tr>
<tr><td rowspan="2">Schmitt-Triggered port</td><td rowspan="2">V_{IL3}</td><td>1.65V≤DVCC3n≤3.3V(n=0~4)</td><td>0.2DVCC3n</td></tr>
<tr><td>1.35V≤DVCC15≤1.65V</td><td>0.1DVCC15</td></tr>
<tr><td>X1</td><td>V_{IL5}</td><td>1.35V≤CVCC15≤1.65V</td><td>0.1CVCC15</td></tr>
</table>

23.3 DC Electrical Characteristics (2/4)

Ta=-20~85˚C (n=0~4, m=1, 2)

Parameter		Symbol	Rating		Min.	Typ. (Note 1)	Max.	Unit
High-level input voltage	P7~PA	V_{IH1}	2.7V≤AVCC32≤AVCC31≤3.3V		0.7AVCC31 0.7AVCC32		AVCC31+0.3 AVCC32+0.3	V
	Normal port	V_{IH2}	1.65V≤DVCC3n≤3.3V(n=0~4)		0.7DVCC3n		DVCC3n+0.2	
	Schmitt-Triggered port	V_{IH3}	1.65V≤DVCC3n≤3.3V(n=0~4)		0.8DVCC3n			
			1.35V≤DVCC15≤1.65V		0.9DVCC15		DVCC15+0.2	
	X1	V_{IH4}	1.35V≤CVCC15≤1.65V		0.9CVCC15		CVCC15+0.2	
Low-level output voltage		V_{OL}	I_{OL}=2mA	DVCC3n≥2.7V			0.4	V
			I_{OL}=500µA	DVCC3n<2.7V			0.4	
High-level output voltage		V_{OH}	I_{OH}=−2mA	DVCC3n≥2.7V	2.4			
			I_{OH}=−500µA	DVCC3n<2.7V	0.8DVCC3n			

(Note 1) Ta=25˚C, DVCC15=1.5V, DVCC3n=3.0V, AVCC3m=3.3V, unless otherwise noted.

23.4 DC Electrical Characteristics (3/4)

Ta=-20~85°C (n=0~4, m=1, 2)

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
Input leakage current	I_{LI}	$0.0 \le V_{IN} \le$ DVCC15 $0.0 \le V_{IN} \le$ DVCC3n(n=0~4) $0.0 \le V_{IN} \le$ AVCC31 $0.0 \le V_{IN} \le$ AVCC32		0.02	±5	μA
Output leakage current	I_{LO}	$0.2 \le V_{IN} \le$ DVCC15–0.2 $0.2 \le V_{IN} \le$ DVCC3n–0.2(n=0~4) $0.2 \le V_{IN} \le$ AVCC31–0.2 $0.2 \le V_{IN} \le$ AVCC32–0.2		0.05	±10	μA
Power down voltage (@STOP)	V_{STOP} (DVCC15)		1.35		1.65	V
	V_{STOP2} (AVCC3)	V_{IL1}=0.3AVCC31,32 V_{IH1}=0.7AVCC31,32	2.7		3.3	V
	V_{STOP3} (DVCC3)	V_{IL2}=0.3DVCC3n, V_{IL3} = 0.1DVCC3n V_{IH2}=0.7DVCC3n, V_{IH3} = 0.9DVCC3n (n=0~4)	1.65		3.3	V
Pull-up resister at Reset	RRST	DVCC15=1.5V ± 0.15V	20	50	150	kΩ
Schmitt-Triggered port	VTH	1.65V≤DVCC3n≤3.3V(n=0~4) 1.35V≤DVCC15≤1.65V	0.3	0.6		V
Programmable pull-up/ pull-down resistor	PKH	DVCC3n=1.65V~3.3V(n=0~4) DVCC15=1.35V~1.65V	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=3.0V, AVCC3m=3.3V, unless otherwise noted.

23.5 DC Electrical Characteristics (4/4)

DVCC15=CVCC=FVCC15=1.5V±0.15V,
FVCC3=DVCC3n=3.0V±0.3V, AVCC3m=3.0V±0.3V,

Ta=-20~85˚C (n=0~4, m=1, 2)

Parameter	Symbol	Rating	Min.	Typ. (Note 1)	Max.	Unit
NORMAL(Note 2) Gear 1/1	I_{CC}	F_{sys}=54 MHz (f_{osc}=13.5 MHz, PLLON)		55	70	mA
IDLE(Doze)				18	28	
IDLE(Halt)				14	23	
STOP		DVCC15=FVCC15=CVCC15=1.35~1.65V DVCC3n=1.65~3.3V AVCC3m=2.7~3.3V FVCC3=2.7~3.3V		50	2000	μA

(Note 1) Ta=25˚C, DVCC15=1.5V, DVCC15x=1.5V, DVCC3n=3.0V, AVCC3m=3.3V, unless otherwise noted.

(Note 2) I_{CC} NORMAL

Measured with the CPU dhrystone operating and all the embedded peripheral I/O operating by the 4 system clock of external bus 16-bit width.

(Note 3) The currents flow through DVCC15, DVCC3n, CVCC15 and AVCC3m are included.

23.6 10-bit ADC Electrical Characteristics

DVCC15=CVCC15=1.35V~1.65V, CVCC3= DVCC3=AVCC3=VREFH=2.7V~3.3V,
AVCC=2.3V~2.7V, AVSS=DVSS, Ta=-20~85°C
AVCC3 load capacitance ≥3.3μF, VREFH load capacitance ≥3.3μF

Parameter		Symbol	Rating	Min	Typ	Max	Unit
Analog reference voltage (+)		VREFH		2.7	3.3	3.3	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 ≤1kΩ AIN load capacitance≥0.1μF Conversion time≥2.0μs @27MHz(ADCLK)		±2	±3	LSB
DNL error					±1	±2	
Offset error					±2	±4	
Full-scale error					±2	±4	
INL error		-	AIN resistance ≤10kΩ AIN load capacitance≥0.01μF Conversion time≥2.0μs @27MHz(ADCLK)		±2	±3	
DNL error					±1	±2	
Offset error					±2	±4	
Full-scale error					±2	±4	
INL error		-	AIN resistance ≤600Ω AIN load capacitance ≤30pF Conversion time ≥1.15μs @40MHz(ADCLK)		±2	±3	
DNL error					±1	±2	
Offset error					±2	±4	
Full-scale error					±2	±4	

(Note 1) 1LSB=(VREFH–VREFL)/ 1024[V]

23.7 AC Electrical Characteristics

[1] Separate bus mode

(1)DVCC15=CVCC15=1.35V~1.65V, DVCC3n=2.3V~3.3V

SYSCR3<ALESEL>=”0”, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	**x**		18.5		ns
2	A0-23 valid to $\overline{RD}$ / $\overline{WR}$ /$\overline{HWR}$ asserted	t_{AC}	(1+ALE)x-20		17		ns
3	A0 – 23 hold after $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	**x** -14		4.5		ns
4	A0 – 23valid to D0 – 15data in	t_{AD}		x(2+W+ALE)-42		50.5	ns
5	$\overline{RD}$ asserted to D0 – 15 data in	t_{RD}		x(1+W)-28		27.5	ns
6	$\overline{RD}$ pulse width low	t_{RR}	x(1+W)-10		45.5		ns
7	D0 – 15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to A0 – 23 output	t_{RAE}	x-15		3.5		ns
9	$\overline{WR}$ /$\overline{HWR}$ pulse width low	t_{WW}	x(1+W)-10		45.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-15 valid	t_{DO}		12.3		12.3	ns
11	D0-15 valid to $\overline{WR}$ /$\overline{HWR}$ negated	t_{DW}	x(1+W)-18		37.5		ns
12	D0 – 15 hold after $\overline{WR}$ /$\overline{HWR}$ rising	t_{WD}	x–15		3.5		ns
13	A0 - 23 valid to $\overline{WAIT}$ input	t_{AW}		x+(ALE)x+ (w-1)x-30		25.5	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$ / $\overline{WR}$ /$\overline{HWR}$	t_{CW}	x(TW-3)-1	x(TW-1)-30	17.5	25.5	ns

(Note) No. 1 to 14:

Internal 2 wait insertion, ALE “1” Clock, @54MHz

TW = W + 2N,
ALE=ALE output width
No. 21
(2W+2N)

W: Number of Auto wait insertion, 2N: Number of external wait insertion

TW = 2 + 2*1 = 4

AC measurement conditions:

Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels:High = 0.7DVCC33 V/Low 0.3DVCC33 V

(2) DVCC15=CVCC15=1.35V~1.65V, DVCC3n=1.65V~1.95V, DVCC15 ≤ DVCC3n+0.2V

SYSCR3<ALESEL> = “0”, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	x		18.5		ns
2	A0-23 valid to $\overline{RD}$ / $\overline{WR}$ /$\overline{HWR}$ asserted	t_{AC}	(1+ALE)x-20		17		ns
3	A0 – 23 hold after $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-7		11.5		ns
4	A0 – 23 valid to D0 – 15 data in	t_{AD}		x(2+W+ALE)-42		50.5	ns
5	$\overline{RD}$ asserted to D0 – 15 data in	t_{RD}		x(1+W)-28		27.5	ns
6	$\overline{RD}$ pulse width low	t_{RR}	x(1+W)-10		45.5		ns
7	D0 – 15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
8	$\overline{RD}$ negated to next A0 – 23 output	t_{RAE}	x-15		3.5		ns
9	$\overline{WR}$ /$\overline{HWR}$ pulse width low	t_{WW}	x(1+W)-10		45.5		ns
10	$\overline{WR}$ or $\overline{HWR}$ asserted to D0-15 valid	t_{DO}		12.3		12.3	ns
11	D0-15 valid to $\overline{WR}$ /$\overline{HWR}$ negated	t_{DW}	x(1+W)-18		37.5		ns
12	D0 – 15 hold after $\overline{WR}$ /$\overline{HWR}$ negated	t_{WD}	x–15		3.5		ns
13	A0 – 23 valid to $\overline{WAIT}$ input	t_{AW}		x+(ALE)x+(w-1) x-30		25.5	ns
14	$\overline{WAIT}$ hold after $\overline{RD}$ / $\overline{WR}$ /$\overline{HWR}$	t_{CW}	x(TW-3)-7	x(TW-1)-40	13.5	15.5	ns

(Note)

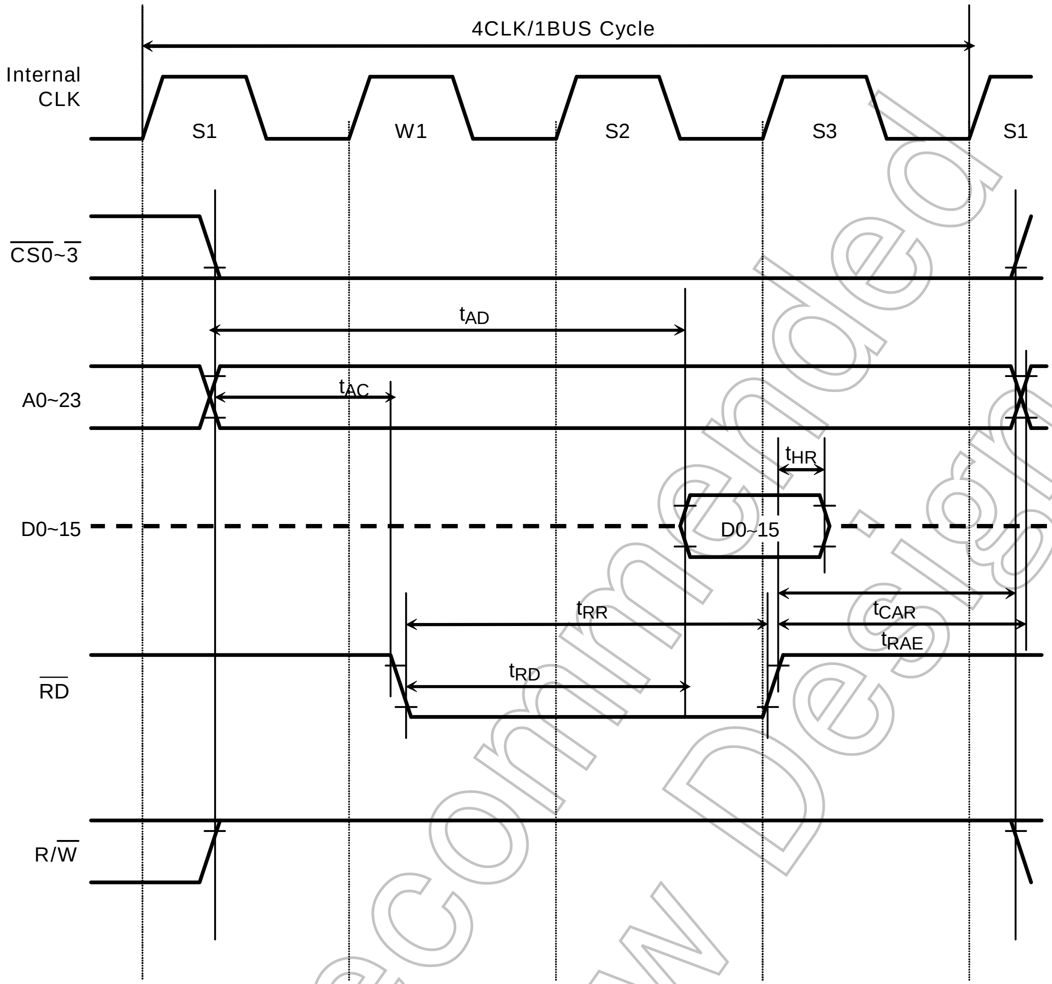
No. 1 to 14:

Internal 2 wait insertion, ALE “1” Clock, @54MHz
TW = W + 2N,
ALE=ALE output width
No. 21
(2W+2N)
W: Number of Auto wait insertion, 2N: Number of external wait insertion
TW = 2 + 2*1 = 4

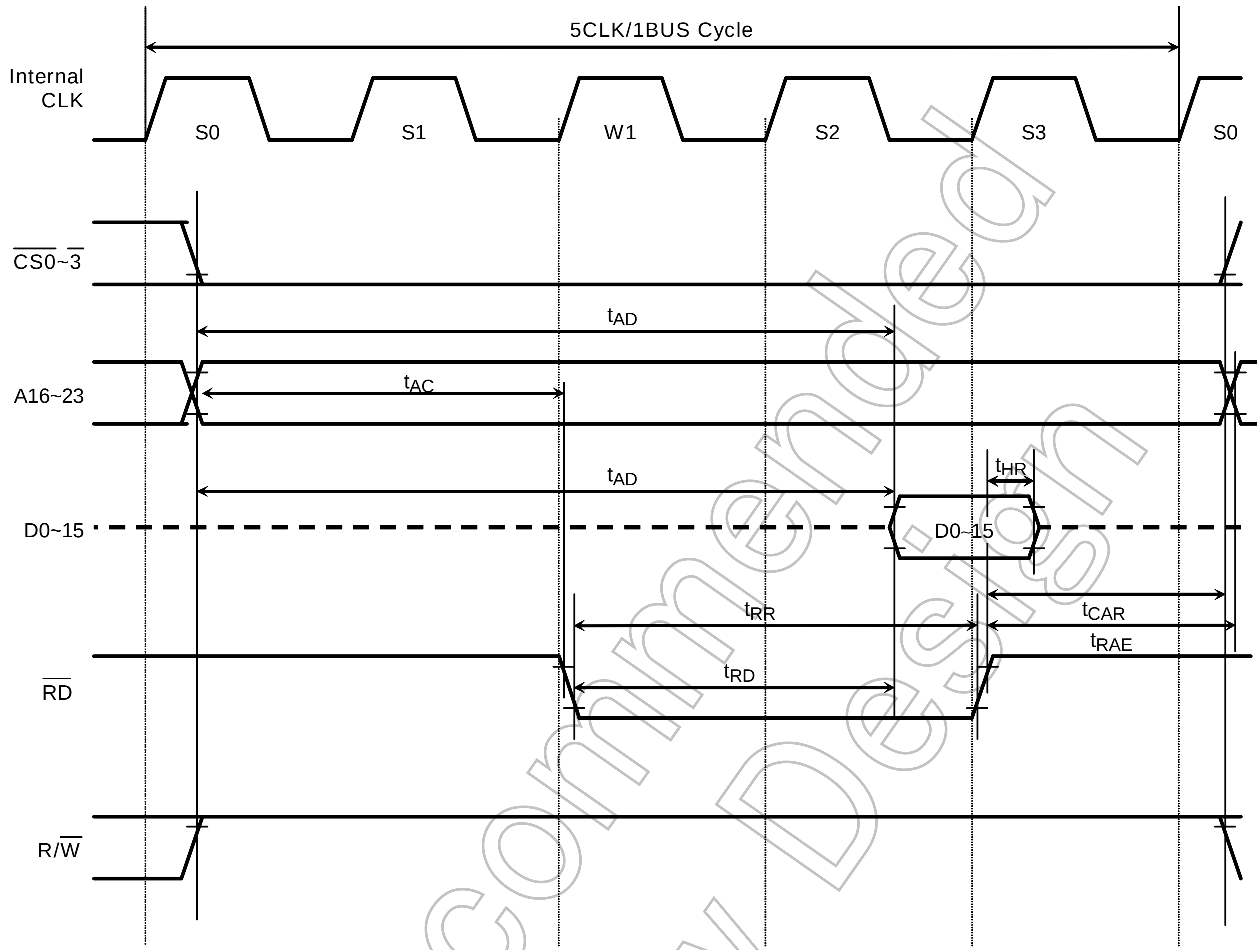
AC measurement conditions:

Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF
Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

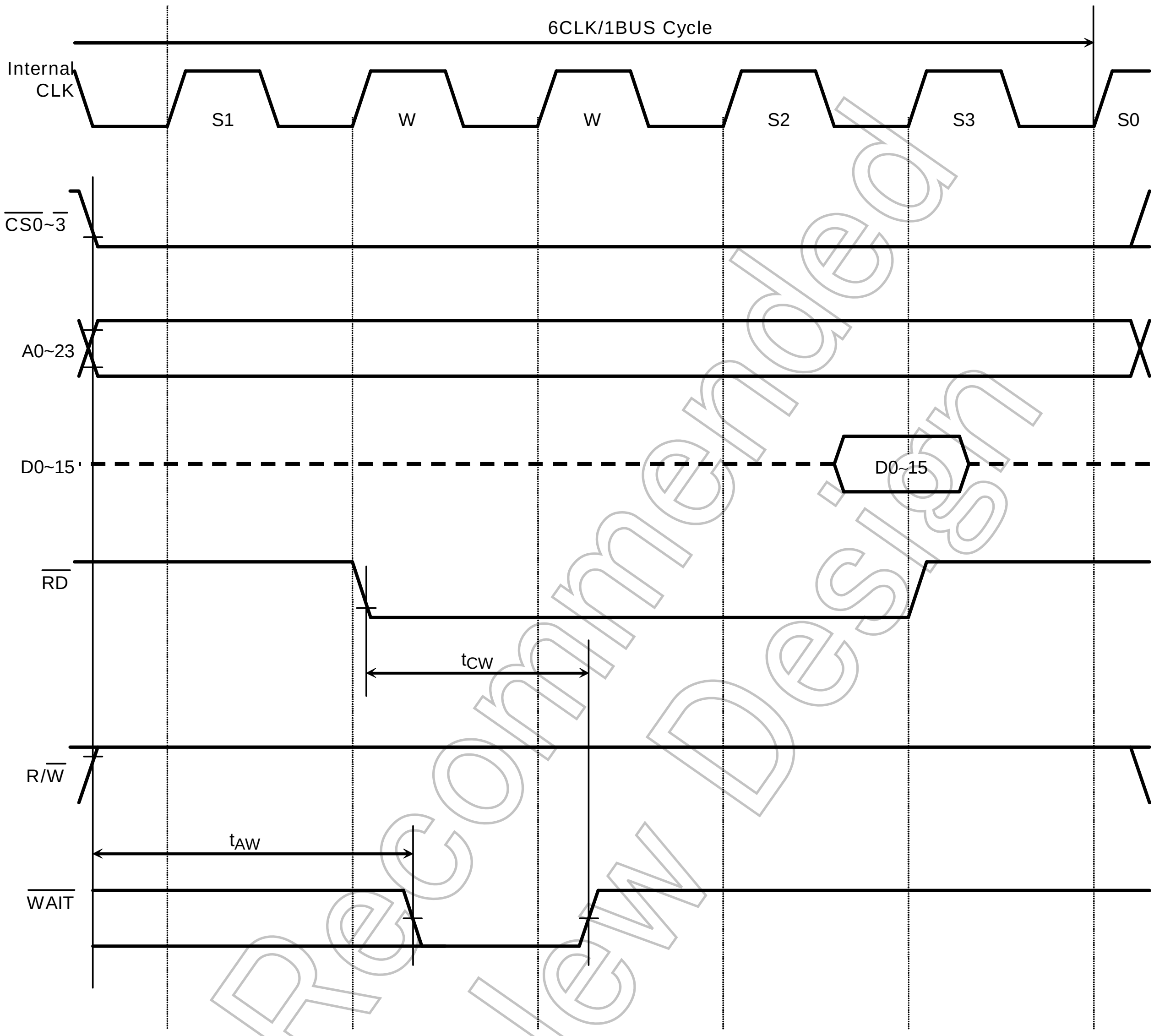
(1) Read cycle timing (SYSCR3<ALESEL>="0", 1 programmed wait state)



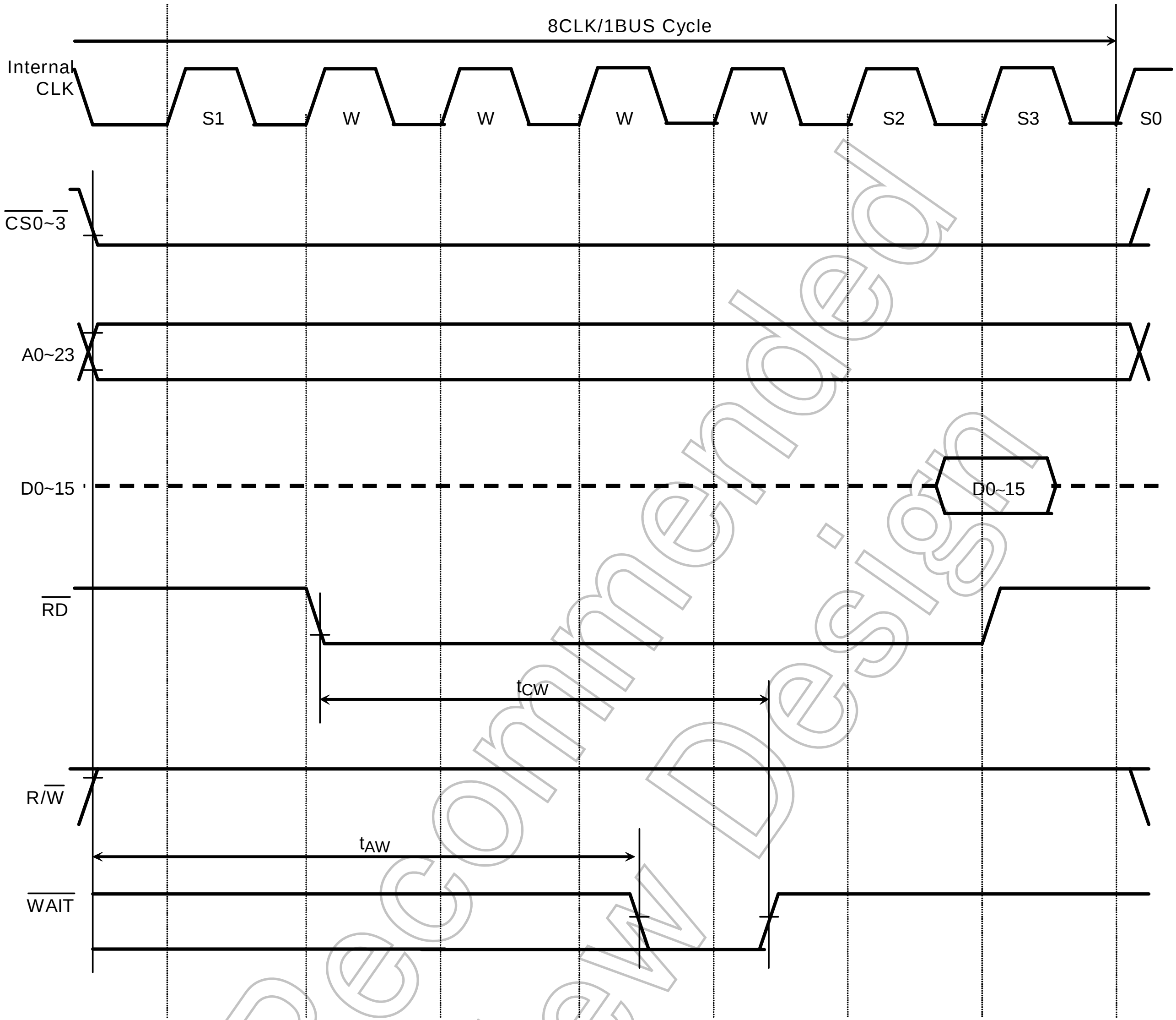
(2) Read timing (SYSCR3<ALESEL>="1", 1 programmed wait state)



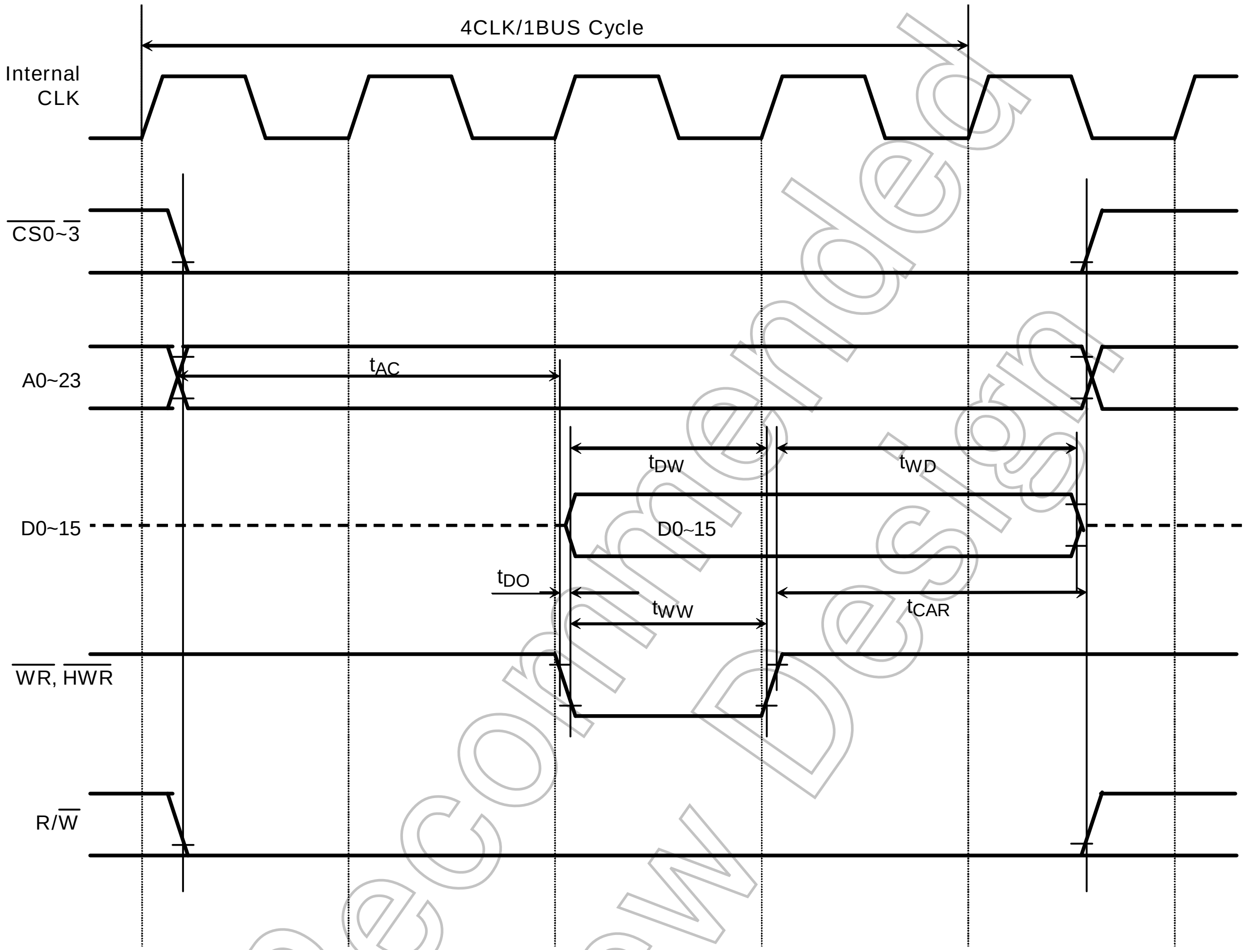
(3) Read timing (SYSCR3<ALESEL>="1", 2 wait (1+N externally generated wait states with N=1)



(4) Read timing (SYSCR3<ALESEL>="1", 4 wait (3+N externally generated wait states with N=1))



(5) Write timing (SYSCR3<ALESEL>="1", 0 wait state)



[2] Multiplex bus mode

(1) DVCC15=CVCC15=1.35V~1.65V, DVCC3n=2.3V~3.3V

1) ALE=1 clock cycle, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	X		18.5		ns
2	A0-15 valid to ALE low	t_{AL}	(ALE)x-12		6.5		ns
3	A0-15 hold after ALE low	t_{LA}	x-8		10.5		ns
4	ALE pulse width high	t_{LL}	(ALE)x-6		12.5		ns
5	ALE low to $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x-8		10.5		ns
6	$\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x-15		3.5		ns
7	A0-15 valid to $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x-20		17.0		ns
8	A16-23 valid to $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x-20		17.0		ns
9	A16-23 hold after $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-14		4.5		ns
10	A0-15 valid to D0-15 data in	t_{ADL}		x(2+W+ALE)-42		50.5	ns
11	A16-23 valid to D0-15 data in	t_{ADH}		x(2+W+ALE)-42		50.5	ns
12	$\overline{RD}$ asserted to D0-15 data in	t_{RD}		x(1+W)-28		27.5	ns
13	$\overline{RD}$ pulse width low	t_{RR}	x(1+W)-10		45.5		ns
14	D0-15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-15 output	t_{RAE}	x-15		3.5		ns
16	$\overline{WR}$ / $\overline{HWR}$ pulse width low	t_{WW}	x(1+W)-10		45.5		ns
17	D0-15 valid to $\overline{WR}$ or $\overline{HWR}$ negated	t_{DW}	x(1+W)-18		37.5		ns
18	D0-15 hold after $\overline{WR}$ or $\overline{HWR}$ negated	t_{WD}	x-15		3.5		ns
19	A16-23 valid to $\overline{WAIT}$ input	t_{AWH}		x+(ALE)x+(W-1)x -30		25.5	ns
20	A0-15 valid to $\overline{WAIT}$ input	t_{AWL}		x+(ALE)x+(W-1)x -30		25.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$	t_{CW}	x(TW-3)-1	x(TW-1)-30	17.5	25.5	ns

(Note)

No. 1 to 21:

Internal 2 wait insertion, ALE "1" Clock, @54MHz

TW = W + 2N,

W: Number of Auto wait insertion, 2N: Number of external wait insertion

ALE=ALE output width

TW = 2 + 2*1 = 4

AC measurement conditions:

Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

(2) DVCC15=CVCC15=1.35V~1.65V, DVCC15 ≤DVCC3n+0.2V

ALE=1 clock cycle, 2 programmed wait state

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			Min	Max	Min	Max	
1	System clock period (x)	t_{SYS}	x		18.5		ns
2	A0-15 valid to ALE low	t_{AL}	(ALE)x-12		6.5		ns
3	A0-15 hold after ALE low	t_{LA}	x-8		10.5		ns
4	ALE pulse width high	t_{LL}	(ALE)x-6		12.5		ns
5	ALE low to $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ asserted	t_{LC}	x-8		10.5		ns
6	$\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ negated to ALE high	t_{CL}	x-15		3.5		ns
7	A0-15 valid to $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACL}	2x-20		17.0		ns
8	A16-23 valid to $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ asserted	t_{ACH}	2x-20		17.0		ns
9	A16-23 hold after $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$ negated	t_{CAR}	x-7		11.5		ns
10	A0-15 valid to D0-15 data in	t_{ADL}		x(2+W+ALE)-42		50.5	ns
11	A16-23 valid to D0-15 data in	t_{ADH}		x(2+W+ALE)-42		50.5	ns
12	$\overline{RD}$ asserted to D0-15 data in	t_{RD}		x(1+W)-28		27.5	ns
13	$\overline{RD}$ pulse width low	t_{RR}	x(1+W)-10		45.5		ns
14	D0-15 hold after $\overline{RD}$ negated	t_{HR}	0		0		ns
15	$\overline{RD}$ negated to next A0-15 output	t_{RAE}	x-15		3.5		ns
16	$\overline{WR}$ / $\overline{HWR}$ pulse width low	t_{WW}	x(1+W)-10		45.5		ns
17	D0-15 valid to $\overline{WR}$ / $\overline{HWR}$ negated	t_{DW}	x(1+W)-18		37.5		ns
18	D0-15 hold after $\overline{WR}$ / $\overline{HWR}$ negated	t_{WD}	x-15		3.5		ns
19	A16-23 valid to $\overline{WAIT}$ input	t_{AWH}		x+(ALE)x+(W-1)x -30		25.5	ns
20	A0-15 valid to $\overline{WAIT}$ input	t_{AWL}		x+(ALE)x+(W-1)x -30		25.5	ns
21	$\overline{WAIT}$ hold after $\overline{RD}$ / $\overline{WR}$ or $\overline{HWR}$	t_{CW}	x(TW-3) - 7	x(TW-1) – 40	13.5	15.5	ns

(Note)

No. 1 to 21:

Internal 2 wait insertion, ALE “1” Clock, @54MHz

TW = W + 2N,

W: Number of Auto wait insertion, 2N: Number of external wait insertion
ALE=ALE output width

No. 21

(2W+2N)

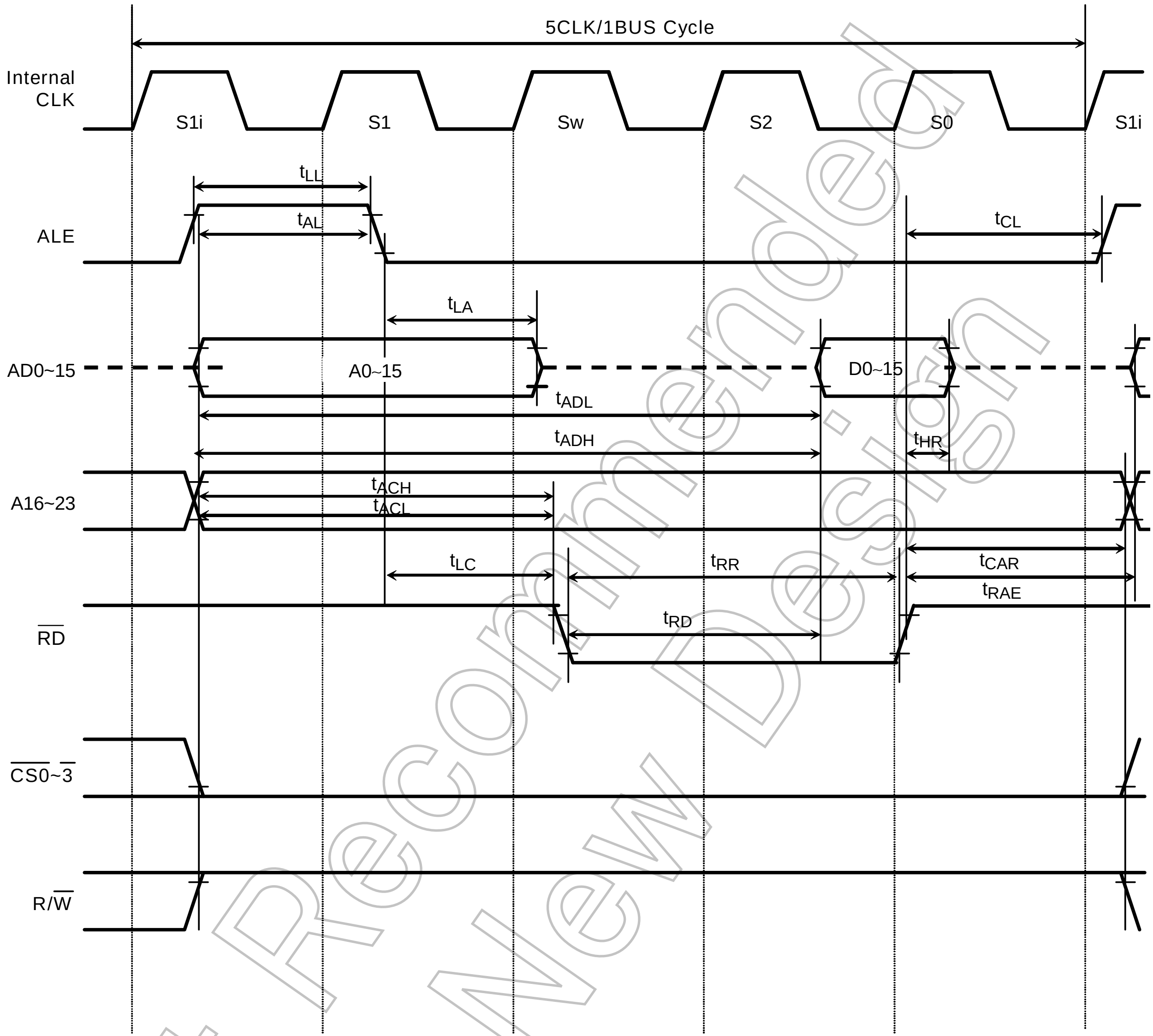
TW = 2 + 2*1 = 4

AC measurement conditions:

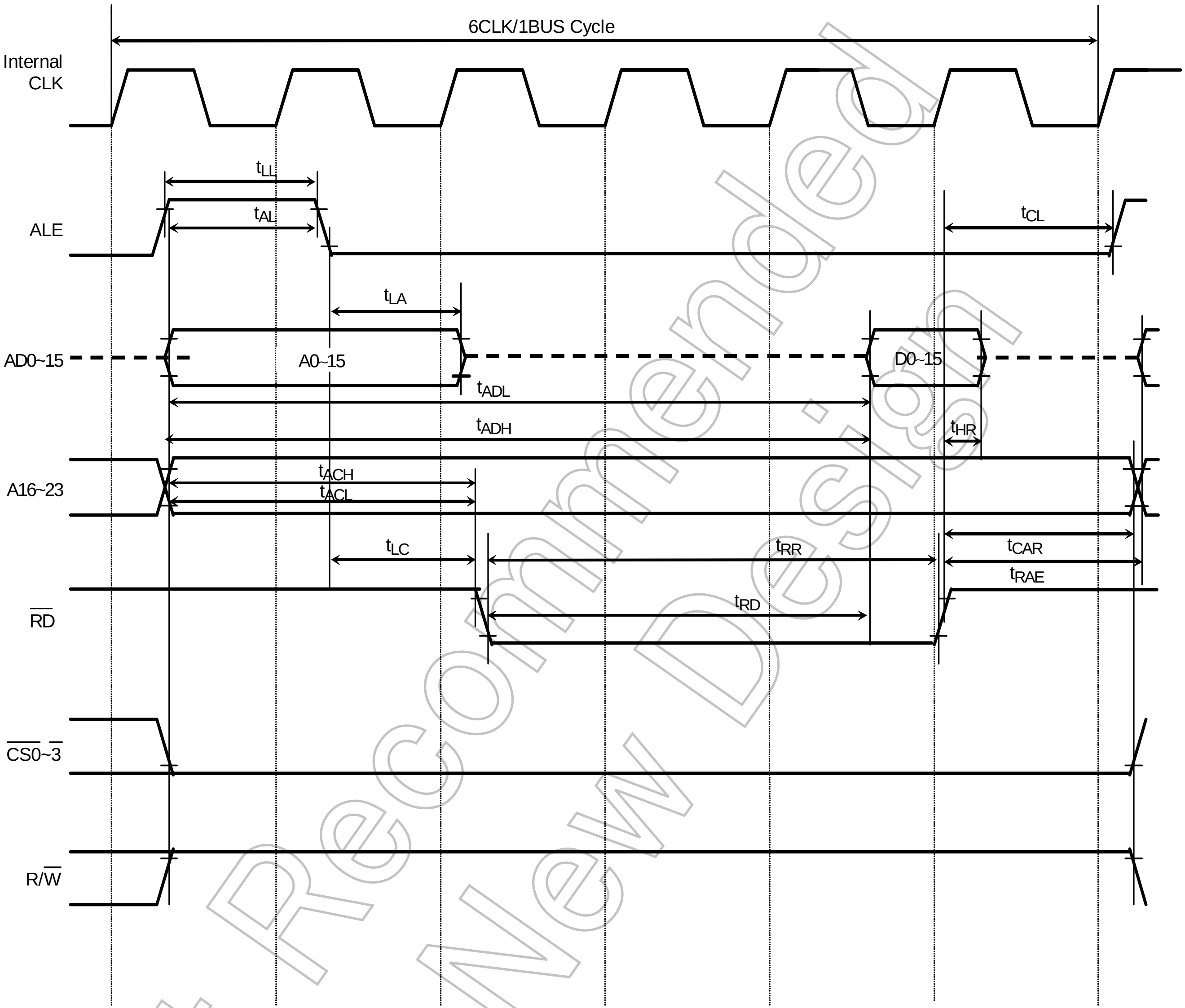
Output levels: High = 0.8DVCC33 V/Low 0.2DVCC33 V, CL = 30 pF

Input levels: High = 0.7DVCC33 V/Low 0.3DVCC33 V

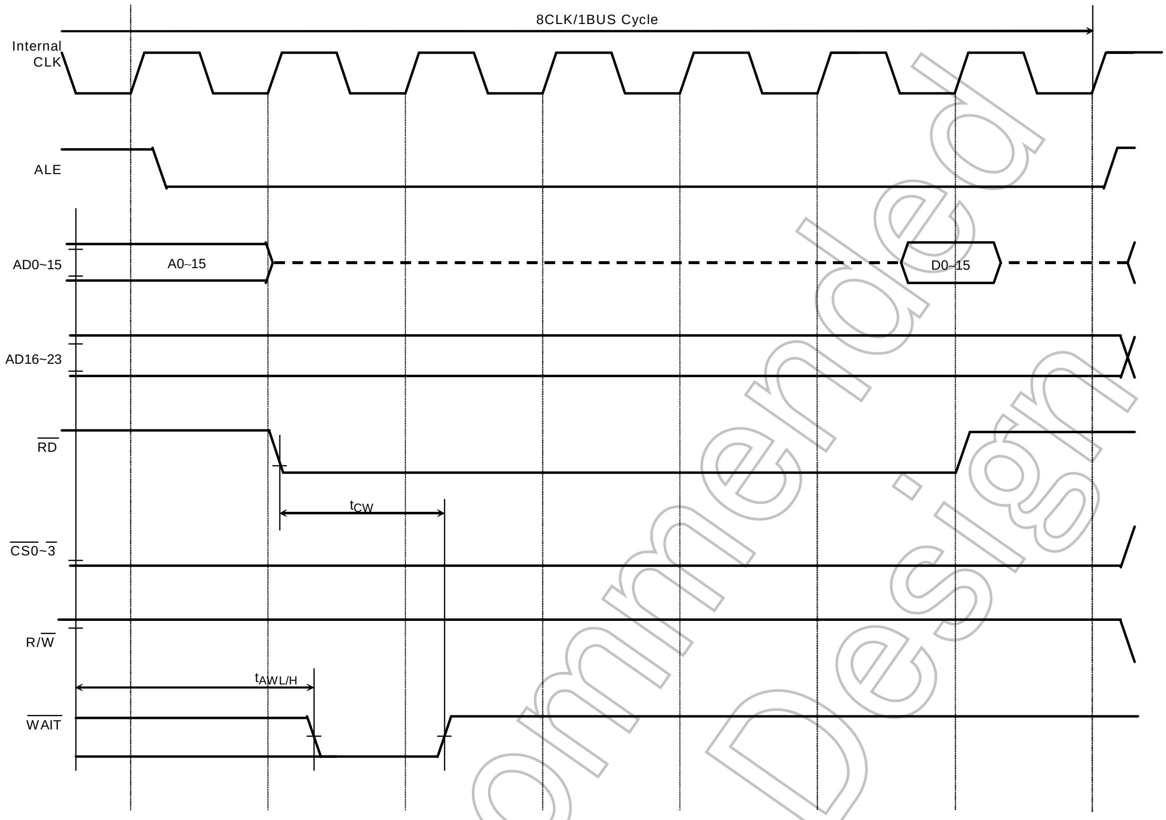
(1) Read timing (ALE=1 clock cycle, 1programmed wait state)



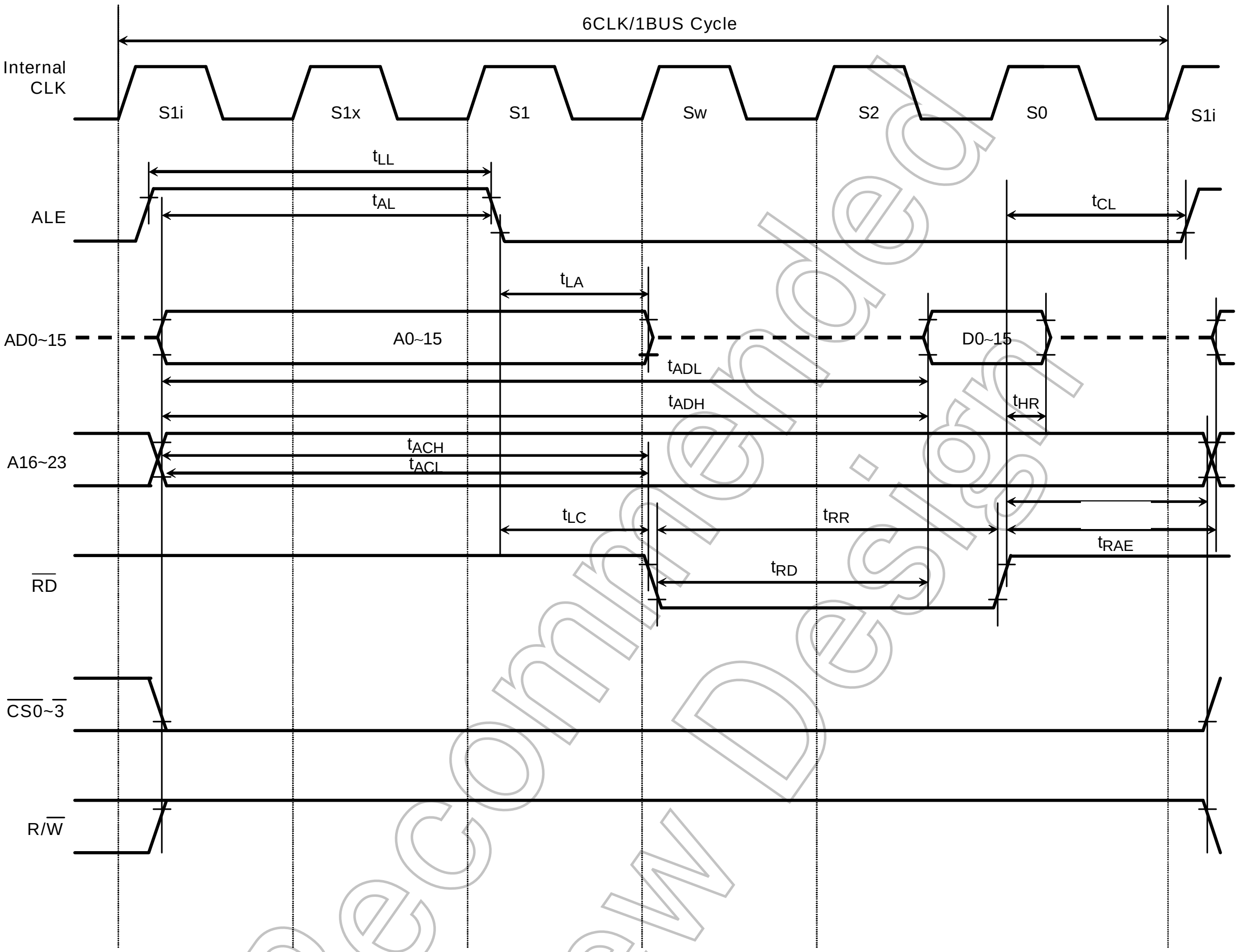
(2) Read timing (ALE=1 clock cycle, 2 programmed wait state)



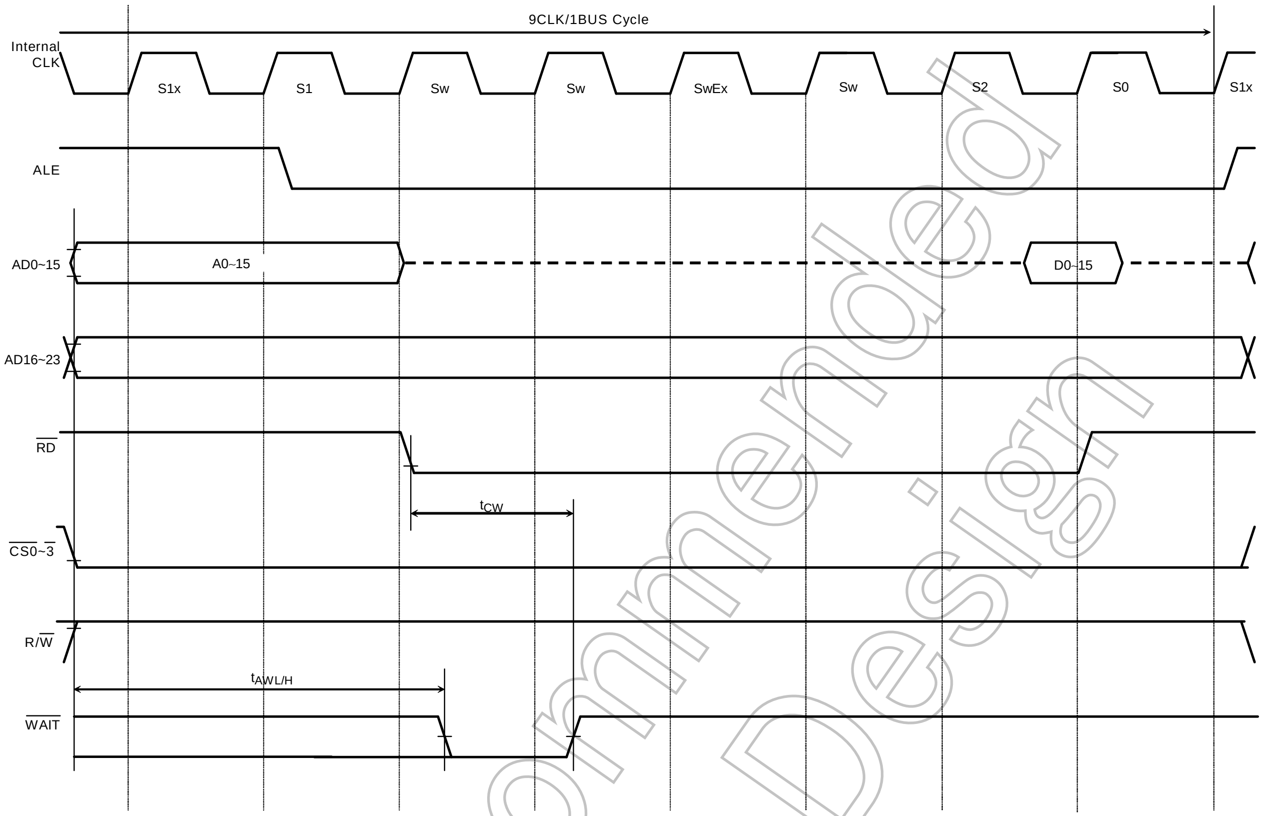
(3) Read timing (ALE = 1 clock cycle, 4 externally generated wait states (2+2N) with N=1



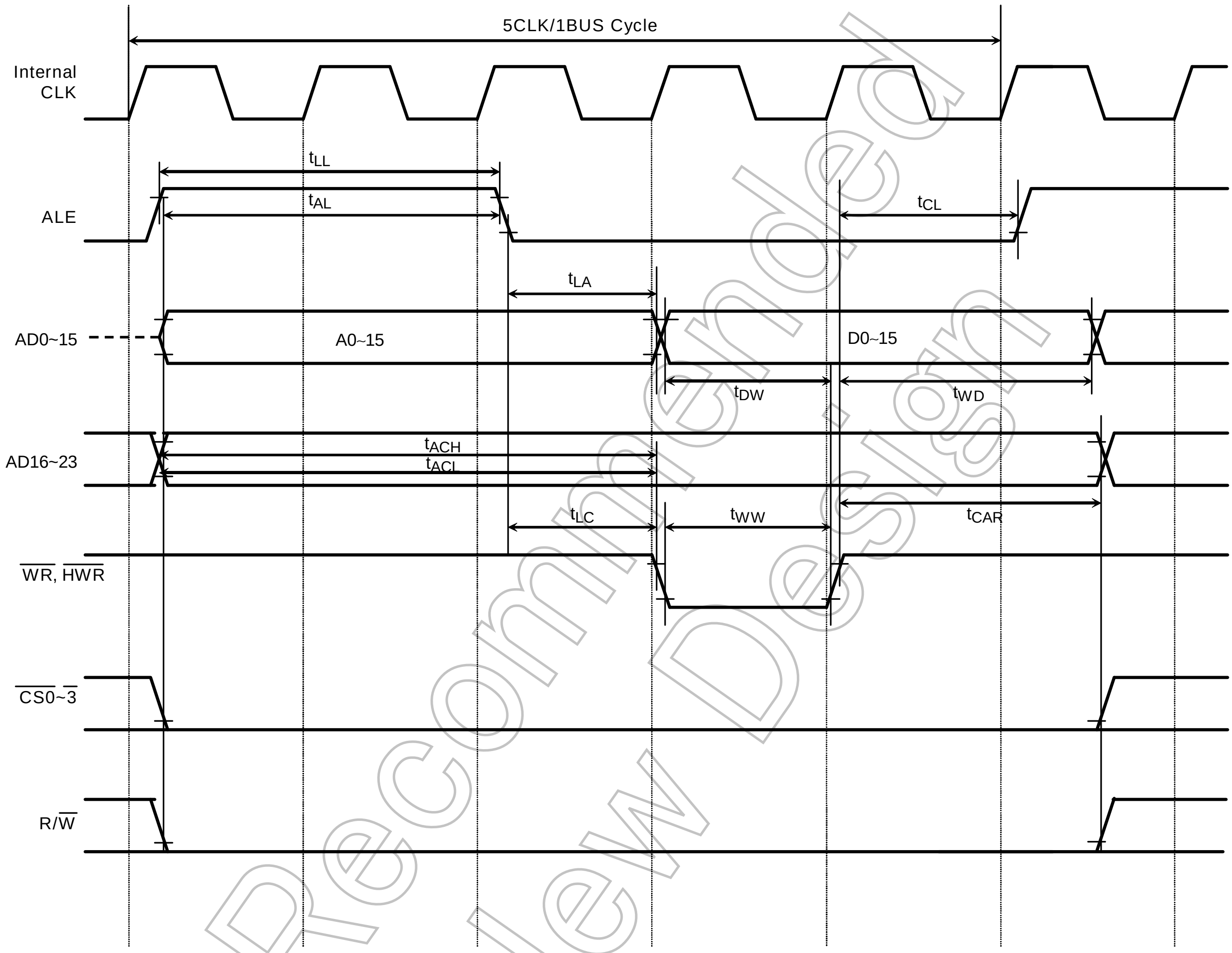
(4) Read timing (ALE = 2 clock cycle, 1programmed wait state)



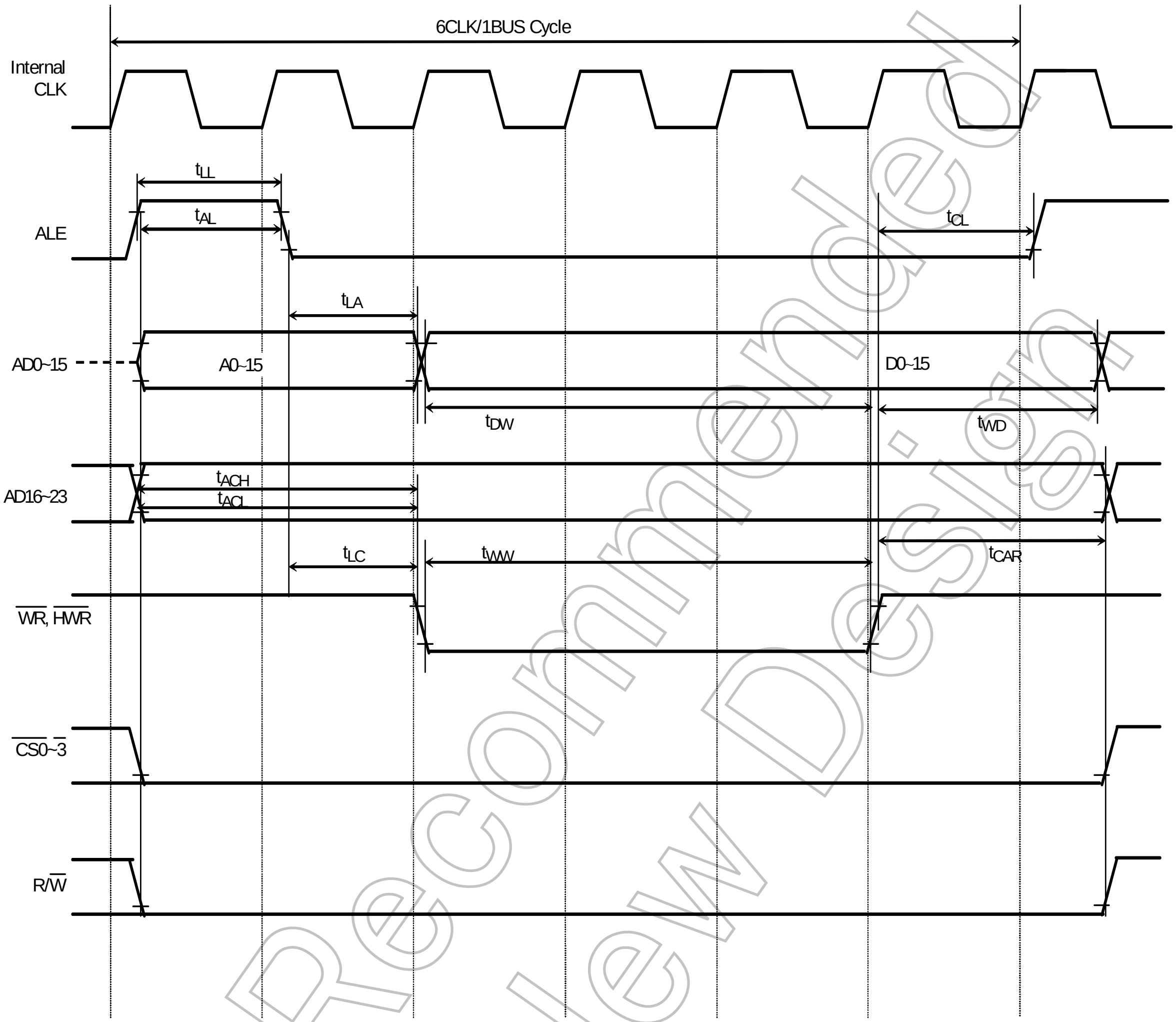
(5) Read timing (ALE = 2clock cycles, 4 externally generated wait states (2+2N) with N=1



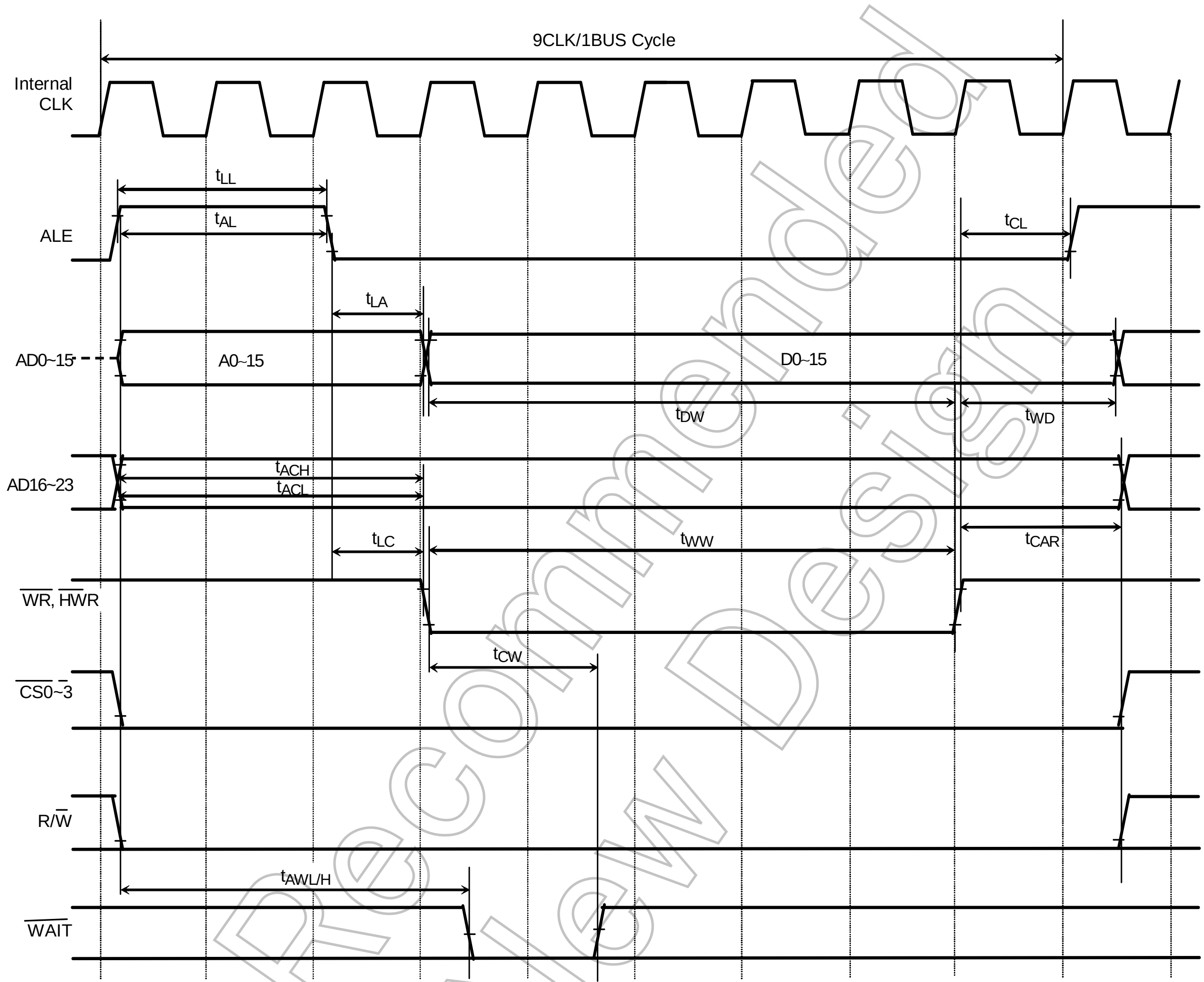
(6) Write timing (ALE = 2 clock timing, 0 wait state)



(7) Write timing (ALE = 1 clock cycle, 2 programmed wait state)



(8) Write timing (ALE = 2 clock cycle, 4 externally generated wait states (2+2N) with N=1)

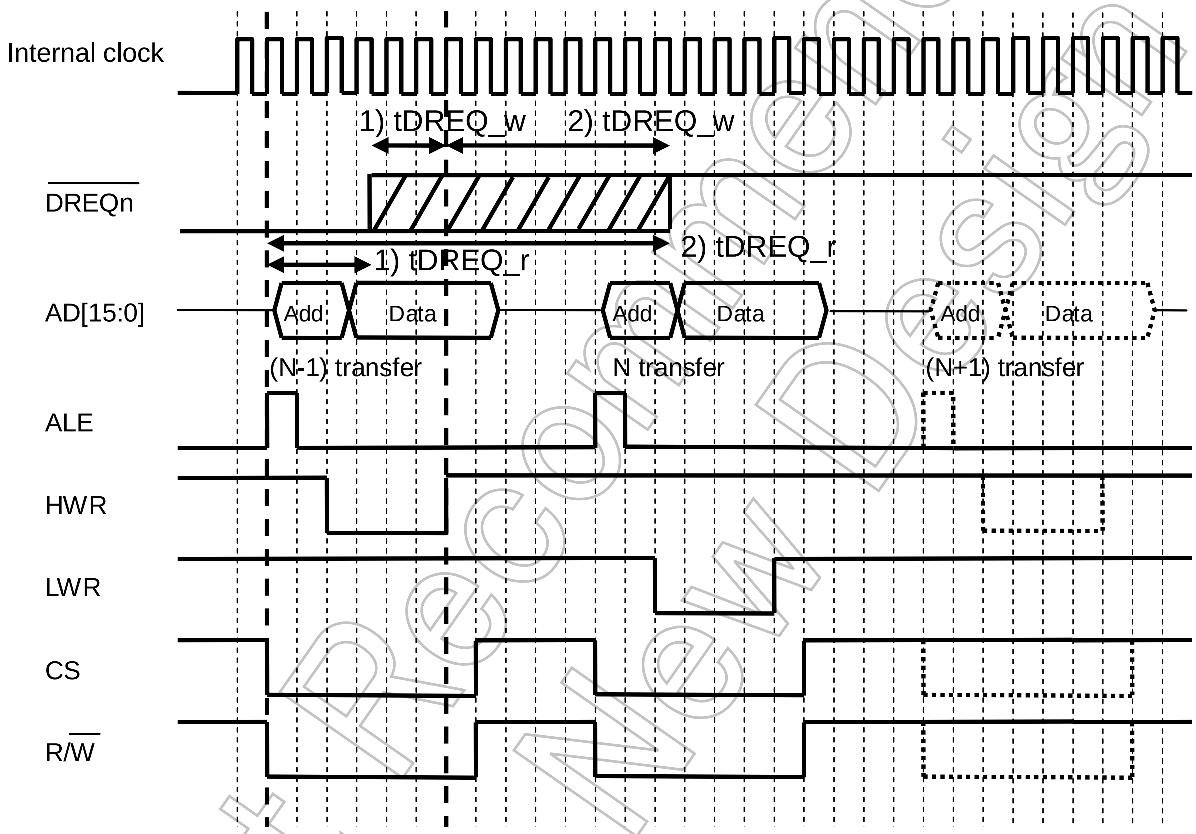


23.8 Transfer with DMA Request

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.

(1) DVCC15=CVCC15=1.35V~1.65V, AVCC3m=2.7V~3.3V
DVCC33=2.3V~3.3V, DVCC30/31/32/34=1.65V~3.3V, Ta= -20~85°C (m=1~2)

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			①Min	②Max	Min	Max	
2	$\overline{RD}$ asserted to $\overline{DREQn}$ negated (external device to on-chip RAM transfer)	tDREQ_r	(W+1)x	(2W+ALE+8)x-51	37	152.5	ns
3	$\overline{WR}$ / $\overline{HWR}$ rising to $\overline{DREQn}$ negated (on-chip RAM to external device transfer)	tDREQ_w	-(W+2)x	(5+WAIT)x-51.8	-55.5	59.2	ns

(2) DVCC15=CVCC15=1.35V~1.65V, AVCC3m =2.7V~3.3V
DVCC33=1.65V~1.95V, DVCC30/31/32/34=1.65V~3.3V, Ta=-20~85°C (m=1~2)

No.	Parameter	Symbol	Equation		54 MHz (fsys)		Unit
			①Min	②Max	Min	Max	
2	$\overline{RD}$ asserted to $\overline{DREQn}$ negated (external device to on-chip RAM transfer)	tDREQ_r	(W+1)x	(2W+ALE+8)x-56	37	147.5	ns
3	$\overline{WR}$ / $\overline{HWR}$ rising to $\overline{DREQn}$ negated (on-chip RAM to external device transfer)	tDREQ_w	-(W+2)x	(5+WAIT)x-56.8	-55.5	54.2	ns

W: number of wait
Ex.)
2 External wait +2N wait (N=1)
W=4
ALE: 1 is substituted for it at 1 clock cycle. 2 is substituted for it at 2 clock cycles.
The equations shown in the above table are calculated provided W=1 and ALE=1.

23.9 Serial Channel Timing

(1) I/O Interface mode (DVCC3=1.65V~3.3V)

In the table below, the letter x represents the fsys cycle period, which varies depending on the programming of the clock gear function.

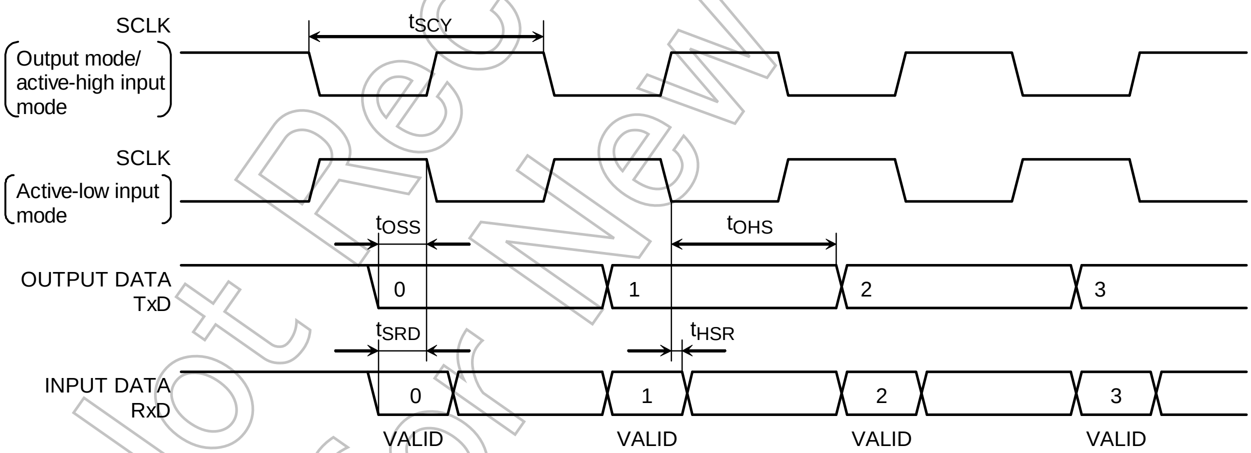
① SCLK input mode (SIO0~SIO6)

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
SCLK period	t_{SCY}	12x		222		ns
TxD data to SCLK rise or fall*	t_{OSS}	2x-35		2		ns
TxD data hold after SCLK rise or fall*	t_{OHS}	8x-15		133		ns
RxD data valid to SCLK rise or fall*	t_{SRD}	30		30		ns
RxD data hold after SCLK rise or fall*	t_{HSR}	2x+29		66		ns

*SCLK rise or fall: Measured relative to the programmed active edge of SCLK.

② SCLK output mode (SIO0~SIO6)

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
SCLK period (programmable)	t_{SCY}	8x		222		ns
TxD data to SCLK rise	t_{OSS}	4x-14		60		ns
TxD data hold after SCLK rise	t_{OHS}	4x-14		60		ns
RxD data valid to SCLK rise	t_{SRD}	45		45		ns
RxD data hold after SCLK rise	t_{HSR}	0		0		ns



23.10 SBI Timing

(1) I2C mode

In the table below, the letters x and t represent the fsys periods and ϕT0 respectively.

n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBI0CR1.

Parameter	Symbol	Equation		Standard mode f_{SYS} = 8 MHz n = 4		Fast mode f_{SYS} = 32 MHz n = 4		Unit
		Min	Max	Min	Max	Min	Max	
SCL clock frequency	t_{SCL}	0		0	100	0	400	kHz
Hold time for START condition	$t_{HD:STA}$			4.0		0.6		μs
SCL clock low width (Input) (Note 1)	t_{LOW}			4.7		1.3		μs
SCL clock high width (Input) (Note 2)	t_{HIGH}			4.0		0.6		μs
Setup time for a repeated START condition	$t_{SU;STA}$	(Note 5)		4.7		0.6		μs
Data hold time (Input) (Note 3, 4)	$t_{HD;DAT}$			0.0		0.0		μs
Data setup time	$t_{SU;DAT}$			250		100		ns
Setup time for STOP condition	$t_{SU;STO}$			4.0		0.6		μs
Bus free time between STOP and START conditions	t_{BUF}	(Note 5)		4.7		1.3		μs

Note 1) SCL clock low width (output) is calculated with: $(2^{(n-1)}+4)$ T.

Normal mode: 6usec@Typ(fsys=8MHZ, n=4)

Fast mode: 1.5usec@Typ(fsys=32MHZ, n=4)

Note 2) SCL high width (output) is calculated with: $(2^{(n-1)})$ T.

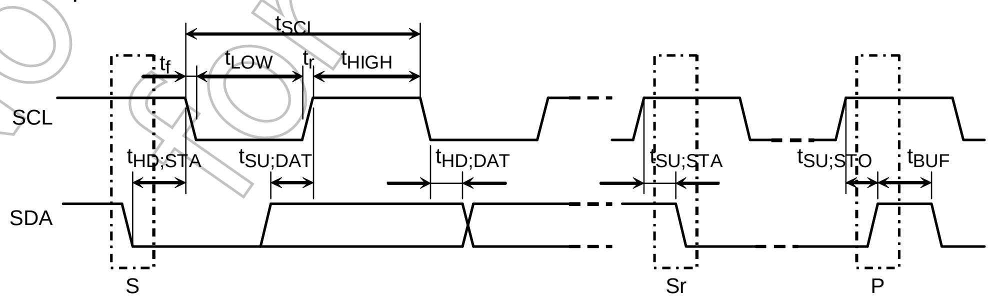
Normal mode: 4usec@Typ(fsys=8MHZ, n=4)

Fast mode: 1usec@Typ(fsys=32MHZ, n=4)

Note 3) The output data hold time is equal to 12x

Note 4) The Philips I²C-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 it to satisfy the input data hold time shown in the table, including tr/tf of the SCL and SDA lines.

Note 5) Software-dependent



S: START condition
Sr: Repeated START condition
P; STOP condition

Fast mode: fsys ≥ 20 MHz

Standard mode:fsys ≥ 4 MHz

(2) Clock-Synchronous 8-Bit SIO mode

In the table below, the letters x and t represent the fsys periods and ϕT0 respectively.
The letter n denotes the value of n programmed into the SCK (SCL output frequency select) field in the SBI0CR1.

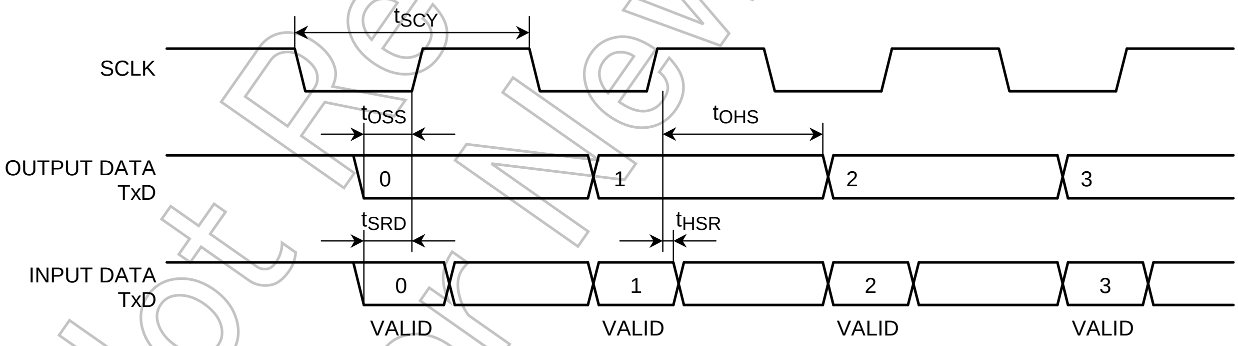
The electrical specifications below are for an SCLK signal with a 50% duty cycle.

③ SCK input mode

Parameter	Symbol	Equation Min	Equation Max	54 MHz Min	54 MHz Max	Unit
SCK period	t_{SCY}	16x		296		ns
TxD data to SCK rise	t_{OSS}	$(t_{SCY}/2)-(6x+20)$		17		ns
TxD data hold after SCK rise	t_{OHS}	$(t_{SCY}/2)+4x$		222		ns
RxD data valid to SCK rise	t_{SRD}	0		0		ns
RxD data hold after SCK rise	t_{HSR}	4x+10		84		ns

④ SCK output mode

Parameter	Symbol	Equation Min	Equation Max	54 MHz Min	54 MHz Max	Unit
SCK period (programmable)	t_{SCY}	16x		296		ns
TxD data to SCK rise	t_{OSS}	$(t_{SCY}/2)-20$		128		ns
TxD data hold after SCK rise	t_{OHS}	$(t_{SCY}/2)-20$		128		ns
RxD data valid to SCK rise	t_{SRD}	2x+30		67		ns
RxD data hold after SCK rise	t_{HSR}	0		0		ns



23.11 Event Counter

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Clock low pulse width	t_{VCKL}	2X+100		137		ns
Clock high pulse width	t_{VCKH}	2X+100		137		ns

23.12 Capture

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width	t_{CPL}	2X+100		137		ns
High pulse width	t_{CPH}	2X+100		137		ns

23.13 General Interrupt (INTC)

In the table below, the letter x represents the fsys cycle period.

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for INT0-INTA	t_{INTAL}	X+100		118.5		ns
High pulse width for INT0-INTA	t_{INTAH}	X+100		118.5		ns

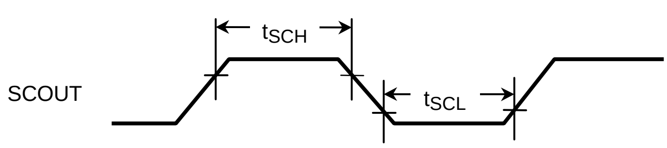
23.14 $\overline{\text{NMI}}$/STOP Release Interrupt

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for $\overline{\text{NMI}}$ and INT0-INT4	t_{INTBL}	100		100		ns
High pulse width for INT0-INT4	t_{INTBH}	100		100		ns

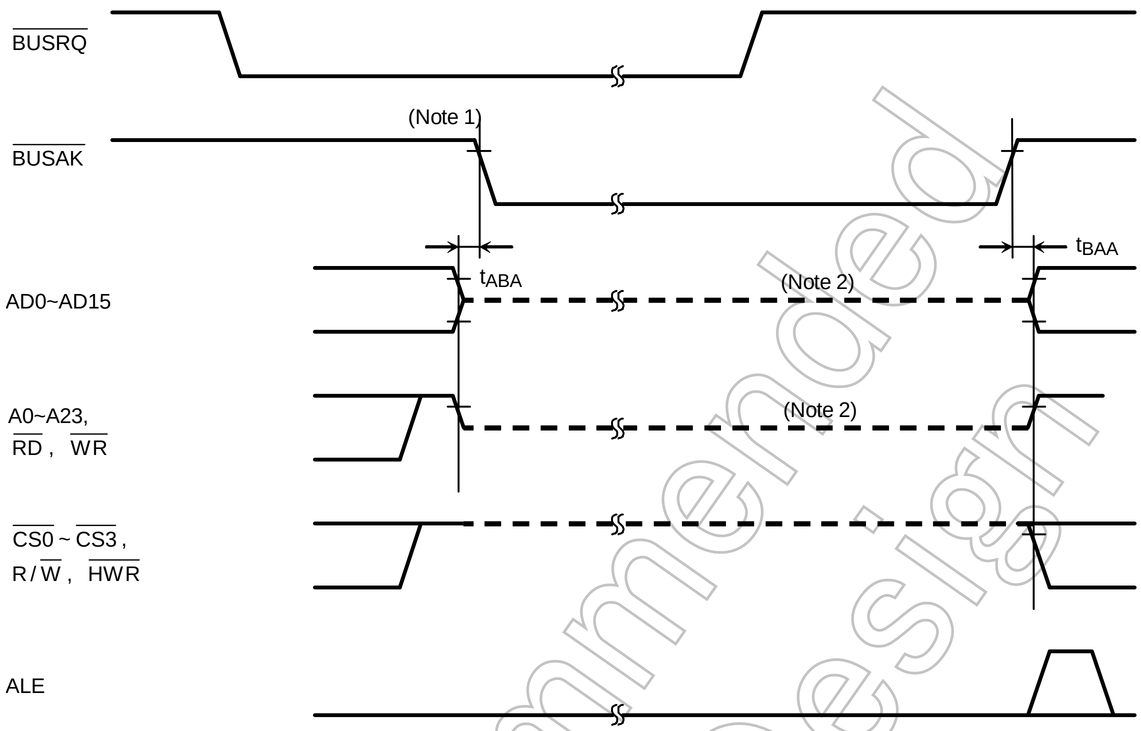
23.15 SCOUT Pin

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Clock high pulse width	t_{SCH}	0.5T–5		4.3		ns
Clock low pulse width	t_{SCL}	0.5T–5		4.3		ns

Note: In the above table, the letter T represents the cycle period of the SCOUT output clock.



23.16 Bus Request and Bus Acknowledge Signals



Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Bus float to $\overline{\text{BUSAK}}$ fall	t_{ABA}	0	80	0	80	ns
Bus float to $\overline{\text{BUSAK}}$ rise	t_{BAA}	0	80	0	80	ns

(Note 1) If the current bus cycle has not terminated due to wait-state insertion, the TMP19A63 does not respond to $\overline{\text{BUSRQ}}$ low 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. In case of using the external load capacitance to maintain the bus at a predefined state, the equipment manufacturer needs to consider the additional time (determined by the CR constant) required for the signal transmission through the external load capacitances. The on-chip, integrated programmable pullup/pulldown resistors remain active, depending on internal signal states.

23.17 KWUP Input

With Pull up

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY0~D	tky_{TBL}	X+100		118		ns
High pulse width for KEY0~D	tky_{TBH}	X+100		118		ns

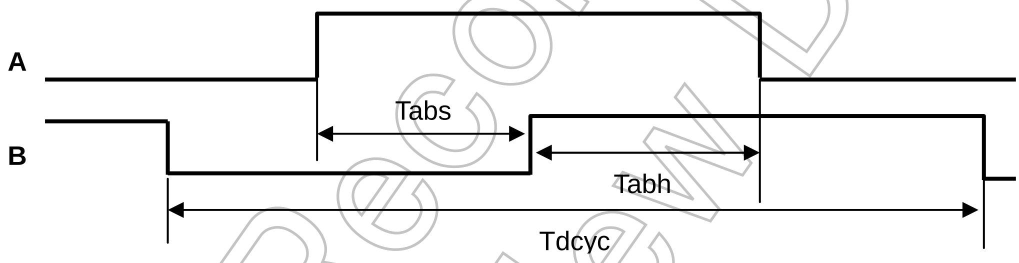
Without pull up

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Low pulse width for KEY0~D	tky_{TBL}	100		100		ns

23.18 Dual Pulse Input

Parameter	Symbol	Equation		54 MHz		Unit
		Min	Max	Min	Max	
Dual input pulse period	Tdcyc	8Y		296		ns
Dual input pulse setup	Tabs	Y+20		57		ns
Dual input pulse hold	Tabh	Y+20		57		ns

Y: fsys/2



24. Package

P-TFBGA289-1111-0.50A