

-----Table of Contents-----

Central Processing Unit (CPU)	13
Reset	16
Processor Mode	26
Clock Generating Circuit	29
Protection	37
Interrupts	38
Watchdog Timer	65
Timer	67
Serial I/O (Serial interface)	85
A-D Converter (A/D converter)	111
PWM Output Circuit	121
LPC Bus Interface	125
Serial interrupt output	143
I ² C-BUS Interface	154
PS2 Interface	185
Programmable I/O Ports	200
Electrical Characteristics	221
Flash Memory Version	234

The differences in M16C/6K (144-pin) group

Type name	M306K7F8LRP(In mass production)	M306K9FCLRP (In mass production)	M306KAFCLRP(Under development)
Pin numbers	144-pin	144-pin	144-pin
RAM	3K bytes	5K bytes	5K bytes
ROM	NEW DINOR Flash memory 68K bytes	NEW DINOR Flash memory 128K bytes	NEW DINOR Flash memory 128K bytes
Built-in ROM area	User ROM area Address 0EF00016 - 0FFFFFF16 Boot ROM area Address 0FF00016 - 0FFFFFF16	User ROM area Address 0E000016 - 0FFFFFF16 Boot ROM area Address 0FF00016 - 0FFFFFF16	User ROM area Address 0E000016 - 0FFFFFF16 Boot ROM area Address 0FF00016 - 0FFFFFF16
Address 03B416	Flash memory recognition register After reset 00000002	Flash memory recognition register After reset XXXXXX102	Flash memory recognition register After reset XXXXXX112
Address 03B716	Flash memory control register After reset XX0000012	Flash memory control register After reset 000000012	Flash memory control register After reset 000000012
The power supply for program/erase	Vcc 3.0 - 3.6V	Vcc 3.0 - 3.6V FVCC 3.0 - 3.6V	Vcc 3.0 - 3.6V
FVCC pin	Not exist	The input pin of power supply for program/erase	Not exist
PWM output circuit	14-bit X 4	8-bit X 6	8-bit X 6
I ² C bus interface	2 channels	3 channels	3 channels (I ² C bus interface pin of Channel 1 and 2 can changed.)
Key input interrupt	8 inputs shared with 1 interrupt request X 1 8 inputs (with event latch) shared with 1 interrupt request X 1 Detected only in the falling edge Can not be selected with 1 bit unit	8 inputs shared with 1 interrupt request X 1 8 inputs (with event latch) shared with 1 interrupt request X 1 Detected in either of the edges by the edge selection Can be selected with 1 bit unit	8 inputs shared with 1 interrupt request X 1 8 inputs (with event latch) shared with 1 interrupt request X 1 Detected in either of the edges by the edge selection Can be selected with 1 bit unit
DMAC	Exist (2 channels)	Exist (2 channels)	Not exist
D/A converter	Exist (8-bit X 2 channels)	Exist (8-bit X 2 channels)	Not exist
Comparator Circuit	Exist (8 channels)	Exist (8 channels)	Not exist
Interrupts	31 vector	31 vector	45 vector (add OBE int.)
Serial I/O	<ul style="list-style-type: none"> • UART or clock synchronous X 3 • clock synchronous X 2 	<ul style="list-style-type: none"> • UART or clock synchronous X 3 • clock synchronous X 2 	<ul style="list-style-type: none"> • UART or clock synchronous X 1 • clock synchronous X 2
Clock generation circuits	2 circuits	2 circuits	1 circuit

Pin configuration

Fig. AA-1 shows the pin configuration (top view).

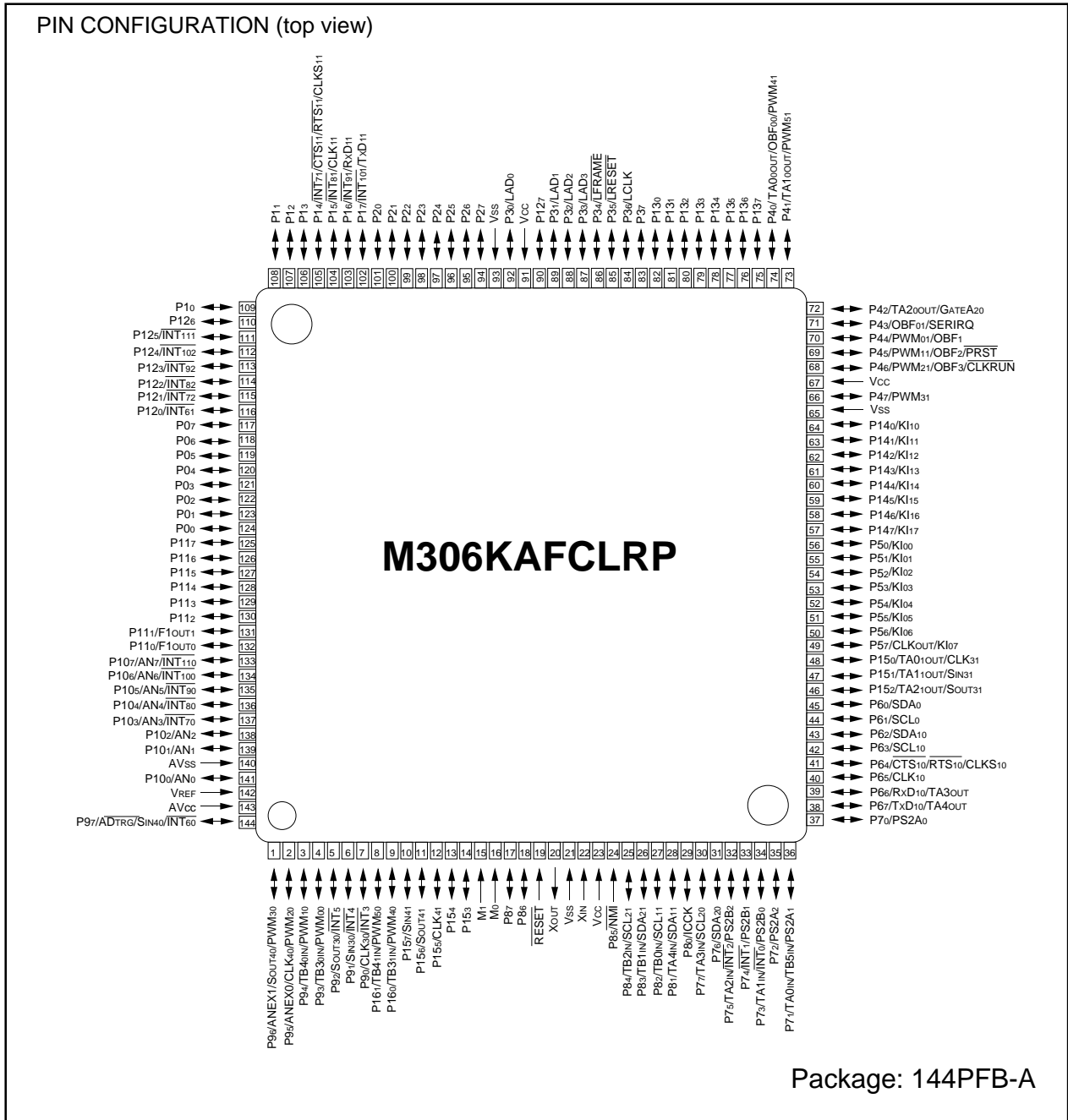


Fig. AA-1 Pin configuration (top view)

Block Diagram

Fig.AA-2 is a block diagram of the M16C/6KA (144-pin version) group.

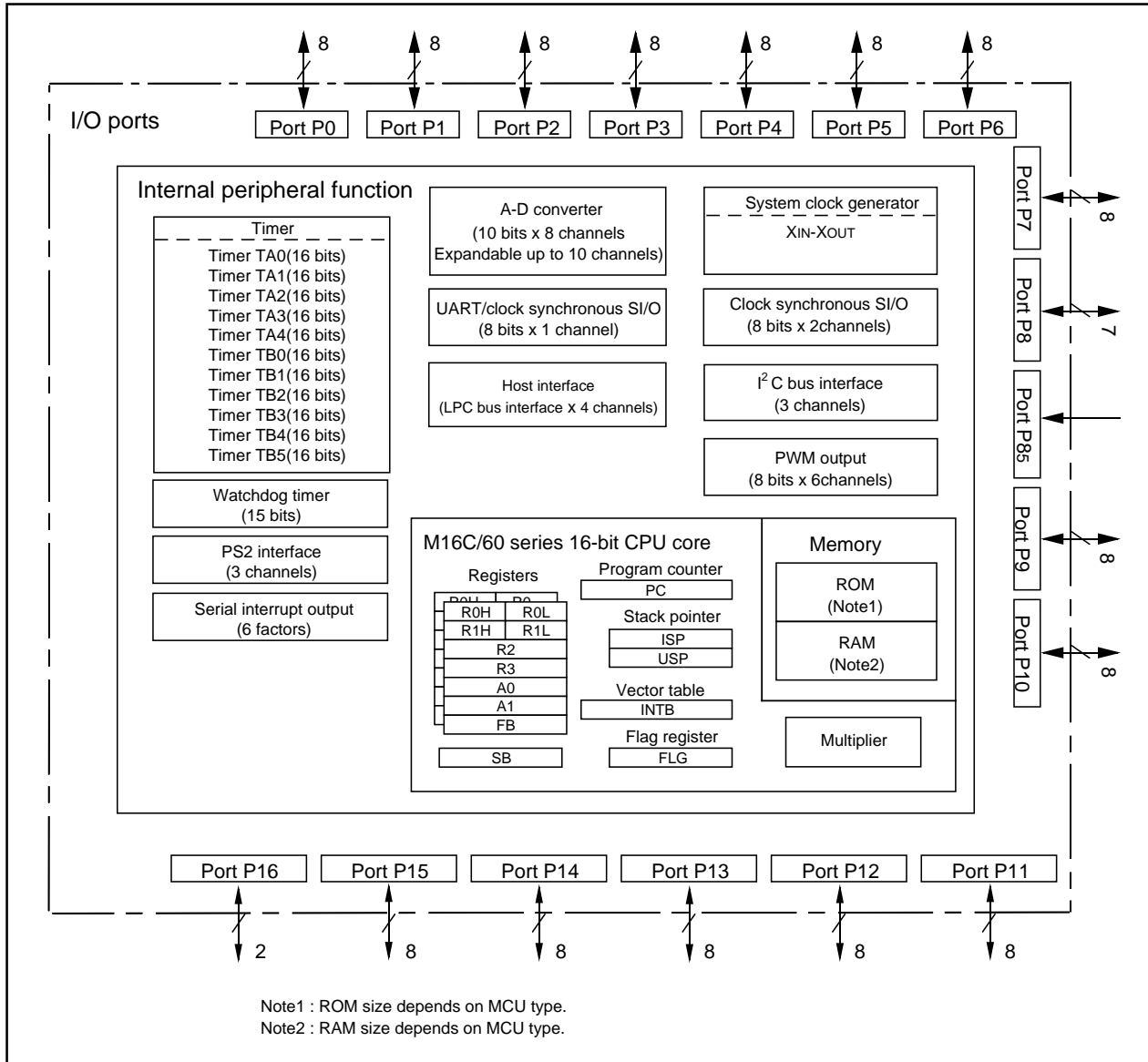


Fig.AA-2 Block diagram of M16C/6KA (144-pin version) group

Performance Outline

Table AA-1 is a performance outline of M16C/6KA (144-pin version) group.

Table AA-1 Performance outline of M16C/6KA (144-pin version) group

Item		Performance
Number of basic instructions		91 instructions
The Min. time of instruction execution		62.5ns (f(XIN)=16MHz, with 0 wait, Vcc=3.3V)
Memory capacity	ROM	(See the figure of ROM Expansion)
	RAM	5K bytes
I/O port	P0 to P10 (except P85)	8 bits x 10, 7 bits x 1
	P11 to P16	8 bitsx5, 2 bitsx1
Input port	P85	1 bit x 1
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits x 5
	TB0, TB1, TB2, TB3, TB4, TB5	16 bits x 6
Serial I/O	UART1	(UART or clock synchronous) x 1
	SI/O3, SI/O4	(Clock synchronous) x 2
A-D converter		10 bits x (8 + 2) channels
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		32 internal and 16 external sources, 4 software sources, 7 levels
Host interface		4 channels (LPC bus interface)
PWM		8 bits x 6
I ² C bus interface		3 channels
PS2 interface		3 channels
Serial interrupt output		6 factors (2 fixed factors, 4 programmable factors)
Clock generating circuit		1 built-in clock generation circuit (built-in feedback resistor, and external ceramic)
Power consumption		52.8mW (3.3V, f(XIN)=16MHz, with 0 wait)
I/O characteristics	I/O withstand voltage	3.3V
	Output current	5mA
Device configuration		CMOS high performance silicon gate
Package		144-pin plastic mold QFP

Renesas plans to release the following products in the M16C/6KA (144-pin version) group:

- (1) Support for flash memory version
- (2) ROM capacity
- (3) Package
 - 144PFB-A : Plastic molded QFP(flash memory version)

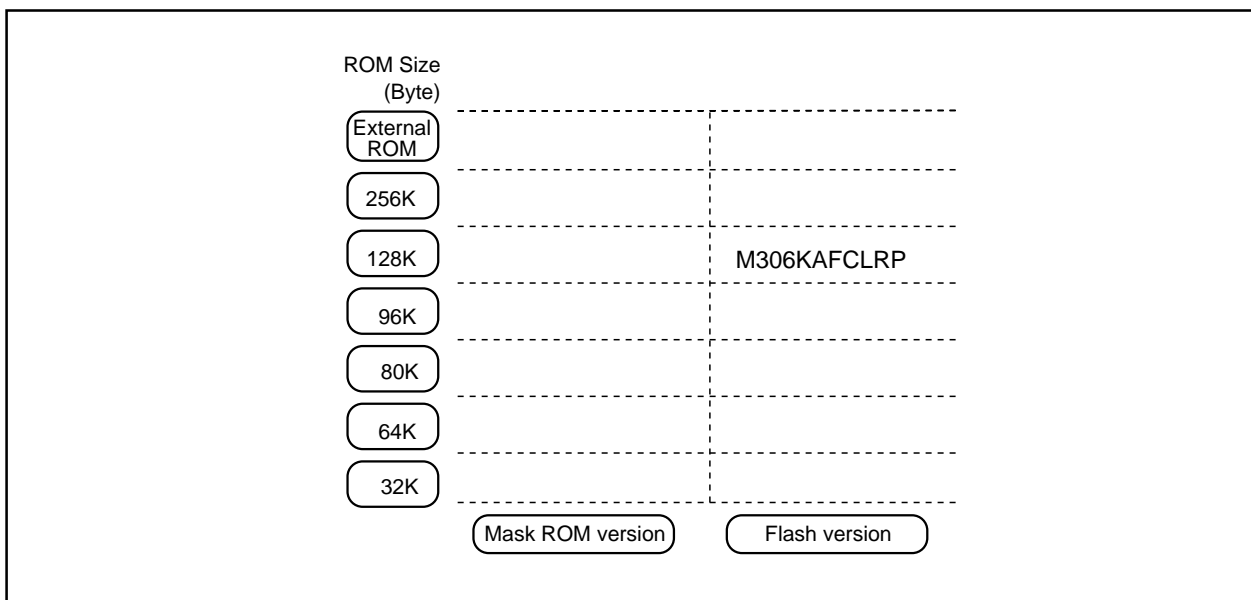


Fig.AA-3 ROM expansion

Table AA-2 Product list

From July 2004 up to now

Type No.	ROM size	RAM size	Package type	Host Interface	Remarks
M306KAFCLRP	128 bytes	5K bytes	144PFB-A	LPC	Flash memory (NEW DINOR) version

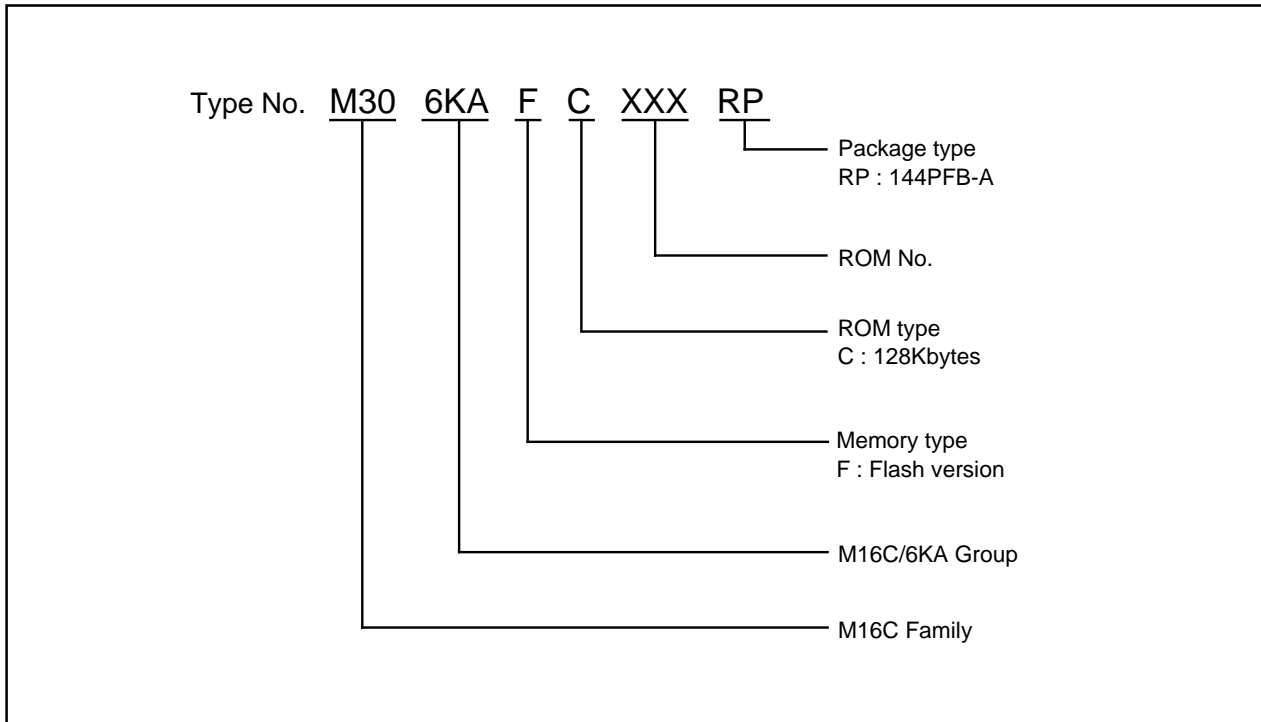


Fig.AA-4 Type No., memory size, and package

Pin Description

Pin name	Signal name	I/O type	Function
Vcc, Vss	Power supply input		Apply 3.0 to 3.6 V to Vcc . Apply 0V to Vss
$\overline{\text{RESET}}$	Reset input	Input	A "L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
Mo,M1	Chip mode setting	Input	Connect to Vss
AVCC	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to VCC.
AVSS	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vss.
VREF	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
P0 ₀ to P0 ₇	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor. This port supports CMOS input level. And output type supports CMOS 3 state or N channel open drain selectable.
P1 ₀ to P1 ₇	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as external interrupt pins or UART1 I/O pin selected by software.
P2 ₀ to P2 ₇	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type just supports CMOS 3 state only). P2 ₀ -P2 ₇ are available for directly driving LED's.
P3 ₀ to P3 ₇	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type just supports CMOS 3 state only). The port can be used for LPC bus interface I/O pins by software selection.
P4 ₀ to P4 ₇	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type just supports CMOS 3 state only). By software selecting, the port can also be used for LPC bus interface I/O pins, Timer A0 to A2 output pins PWM output pins or serial interrupt output I/O pins. P4 ₀ to P4 ₆ pins' level can be read regardless the setting of input port or output port. If P4 ₀ or P4 ₃ are used for output ports, the function that clears P4 ₀ or P4 ₃ to "0" after the read of output data buffer from host CPU is available.
P5 ₀ to P5 ₇	I/O port P5	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only). Key on wake interrupt 0 input function support. P5 ₇ in this port outputs a divide-by-8 or divide-by-32 clock of XIN selected by software.

Pin Description

Pin name	Signal name	I/O type	Function
P60 to P67	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that P60 to P63's output type is N channel open drain only; P64 to P67's output type is CMOS 3 state only; P60 to P63 no internal pull-up register support.) By software selecting, this port can be used for I ² C-BUS interface, UART1 input/output pin, timerA3, A4 output pin. When P60 to P63 used as I ² C-BUS interface SDA, SCL, the input level of these pins are CMOS/SMBUS selectable.
P70 to P77	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that P70 to P77 output type is N channel open drain only; no internal pull-up register support.) By software selecting, this port can be used for external interrupt input pin, timerA0 to A3 and timer B5 input pin, PS2 interface input/output pin, I ² C interface input/output pin. P70 to P75 pins' level can be read regardless of the setting of input port or output port.
P80 to P84, P86, P87, P85	I/O port P8 I/O port P85	Input/output Input/output Input/output Input	P80 to P84, P86, and P87 are I/O ports with the same functions as P0. (Except that P86 to P87's output type is CMOS 3 state only; P80 to P84's output type is N channel open drain only; P85 is input port only; the P80 to P84 and P85 are no internal pull-up register support.) By software selecting, this port can be used for timer A4, B0 to B2, I ² C-BUS interface I/O pins. The input level of P81 to P84 and SDA, SCL inputs can be switched to CMOS/SMBUS when these pins function as I ² C bus interface. P85 is an input-only port that also functions for NMI. The NMI interrupt is generated when the input at this pin changes from "H" to "L". The NMI function cannot be cancelled using software.
P90 to P97	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, the port can be used for external interrupt, timer B3 to B4, A-D converter extended input pins, A-D trigger, SI/O3, SI/O4 I/O pins, PWM, output pins.
P100 to P107	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) If the ports are set to input mode, the pull-up resistor can be set in bit unit. By software selecting, the port can be used for A-D converter, external interrupt input pins.

Pin Description

Pin name	Signal name	I/O type	Function
P110 to P117	I/O port P11	Input/output	This is an 8-bit I/O port equivalent to P0. By software selecting, P110, P111 also function as clock output pins, which the frequency is the same with XIN.
P120 to P127	I/O port P12	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, this port can be used for external interrupt input pin.
P130 to P137	I/O port P13	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is N channel open drain only; no internal pull-up register support.)
P140 to P147	I/O port P14	Input/output	This is an 8-bit I/O port equivalent to P0. The port can be used for key on wake-up interrupt 1 input pins. P140 to P143 are available for directly driving LED's. In input mode, the pull-up register can be set in one bit unit by software.
P150 to P157	I/O port P15	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, these ports can be used for timer A0 to A2's output or SI/O3 and SI/O4 I/O pins.
P160, P161	I/O port P16	Input/output	This is an 2-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, this port can be used for timer B3 and B4 input or PWM output pin.

Operation of Functional Blocks

The M16C/6KA (144-pin version) group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also peripheral units such as timers, serial I/O, A-D converter, host bus interface, PWM output, I²C BUS interface, PS2 interface and I/O ports are included.

The following explains each unit.

Memory

Fig.CA-1 is the memory map. The address space extends up to 1M bytes from address 00000₁₆ to FFFFF₁₆. There is 128K bytes of internal ROM from E0000₁₆ to FFFFF₁₆. The vector table for fixed interrupts such as the reset and $\overline{\text{NMI}}$ are mapped from FFFDC₁₆ to FFFFF₁₆. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 00400₁₆ to the address increasing direction RAM is allocated. For example, in the M306KAFCLRP, 5K bytes of internal RAM is mapped to the space from 00400₁₆ to 017FF₁₆. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped from 00000₁₆ to 003FF₁₆. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Fig.CA-2 to CA-5 are location of peripheral unit control registers. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is mapped from FFE00₁₆ to FFFDB₁₆. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

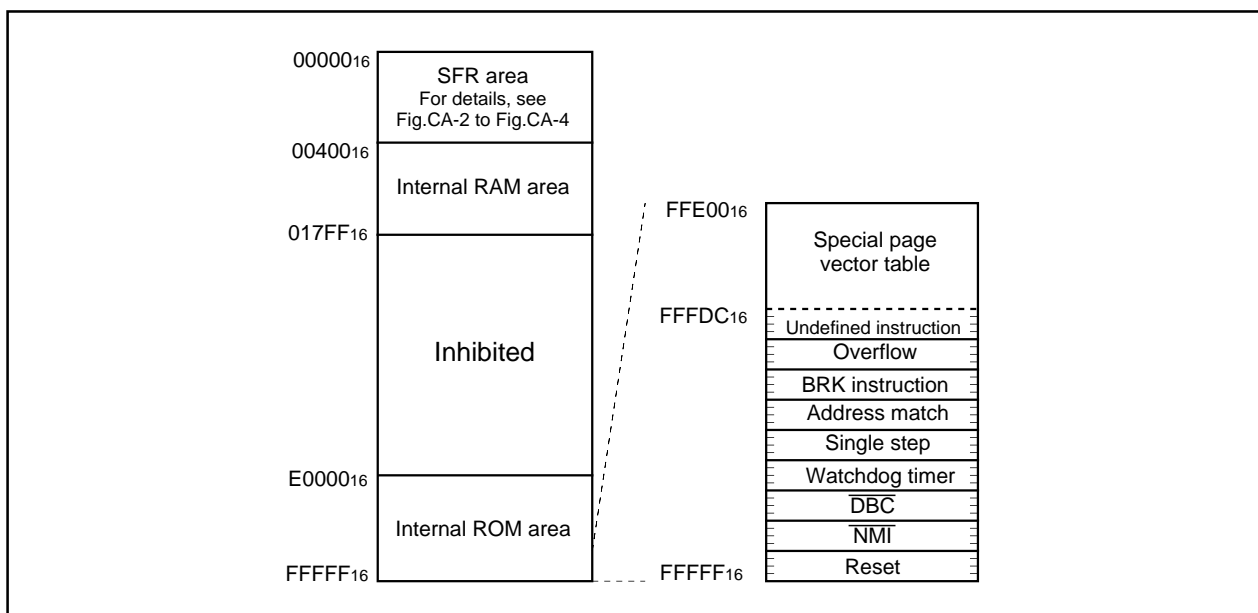


Fig.CA-1 Memory map

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Fig.BA-1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

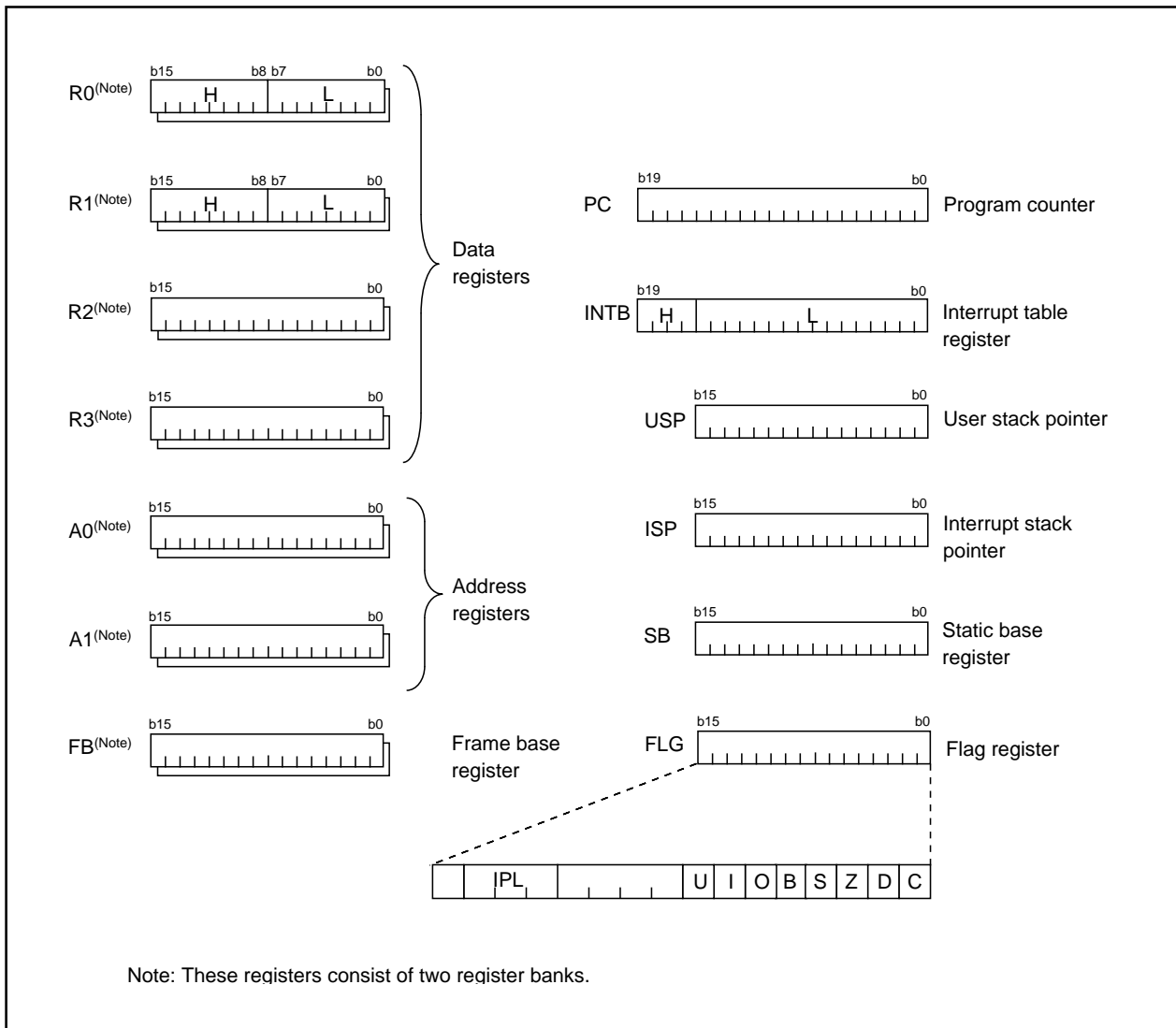


Fig.BA-1 Central processing unit register

(1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, and R3)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H/R1H), and low-order bits as (R0L/R1L). In some instructions, registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0/R3R1).

(2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

(3) Frame base register (FB)

Frame base register (FB) is configured with 16 bits, and is used for FB relative addressing.

(4) Program counter (PC)

Program counter (PC) is configured with 20 bits, indicating the address of an instruction to be executed.

(5) Interrupt table register (INTB)

Interrupt table register (INTB) is configured with 20 bits, indicating the start address of an interrupt vector table.

(6) Stack pointer (USP/ISP)

Stack pointer comes in two types: user stack pointer (USP) and interrupt stack pointer (ISP), each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag). This flag is located at the position of bit 7 in the flag register (FLG).

(7) Static base register (SB)

Static base register (SB) is configured with 16 bits, and is used for SB relative addressing.

(8) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Fig.BA-2 shows the flag register (FLG). The following explains the function of each flag:

- Bit 0: Carry flag (C flag)

This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

- Bit 1: Debug flag (D flag)

This flag enables a single-step interrupt.

When this flag is "1", a single-step interrupt is generated after instruction execution. This flag is cleared to "0" when the interrupt is acknowledged.

- Bit 2: Zero flag (Z flag)

This flag is set to "1" when an arithmetic operation resulted in 0; otherwise, cleared to "0".

- Bit 3: Sign flag (S flag)

This flag is set to "1" when an arithmetic operation resulted in a negative value; otherwise, cleared to "0".

- Bit 4: Register bank select flag (B flag)

This flag chooses a register bank. Register bank 0 is selected when this flag is "0"; register bank 1 is selected when this flag is "1".

- Bit 5: Overflow flag (O flag)

This flag is set to "1" when an arithmetic operation resulted in overflow; otherwise, cleared to "0".

- Bit 6: Interrupt enable flag (I flag)

This flag enables a maskable interrupt.

An interrupt is disabled when this flag is "0", and is enabled when this flag is "1". This flag is cleared to "0" when the interrupt is acknowledged.

- Bit 7: Stack pointer select flag (U flag)

Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.

This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt No. 0 to 31 is executed.

- Bits 8 to 11: Reserved area
- Bits 12 to 14: Processor interrupt priority level (IPL)

Processor interrupt priority level (IPL) is configured with the three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.

- Bit 15: Reserved area

The C, Z, S, and O flags are changed when instructions are executed. See the software manual for details.

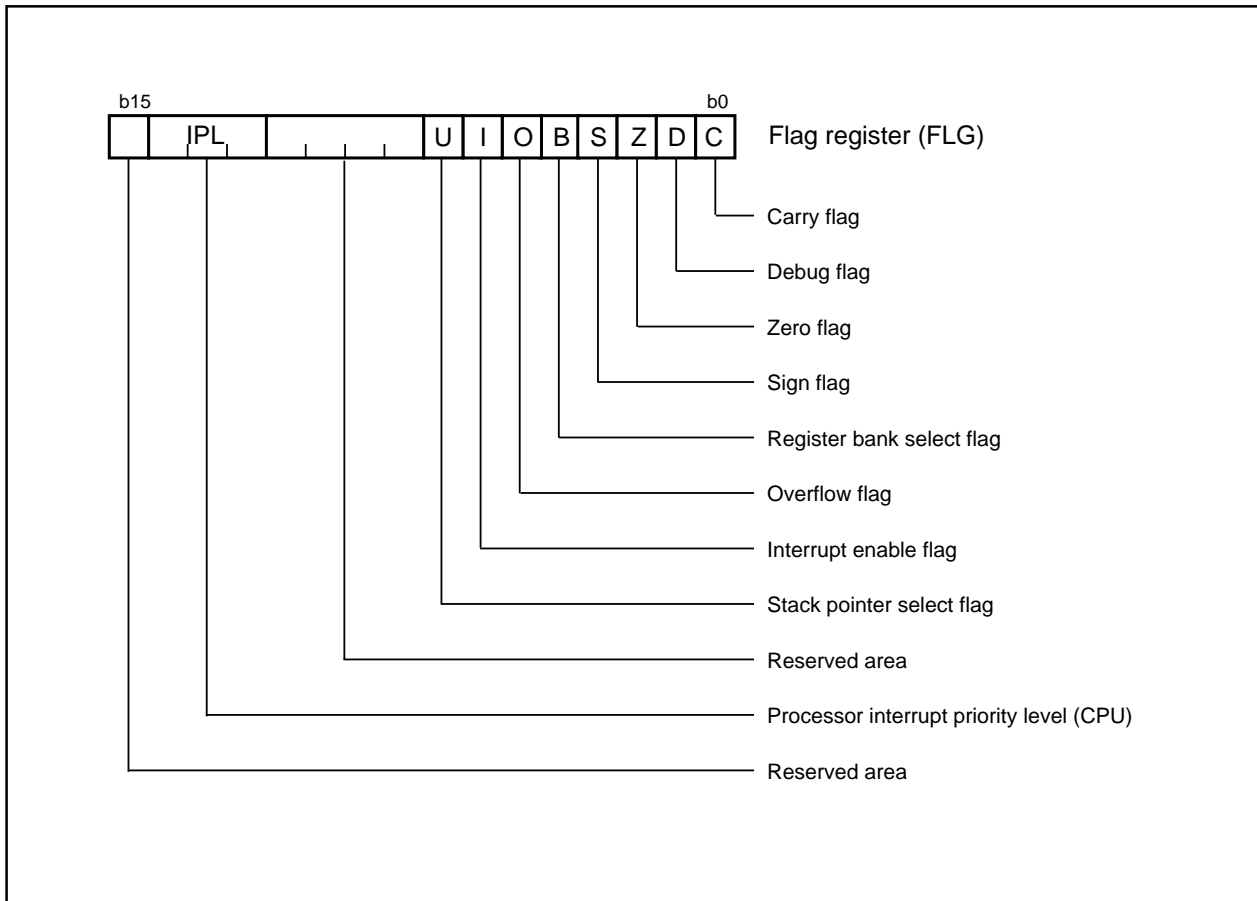


Fig.BA-2 Flag register (FLG)

Reset

There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See “Software Reset” for details of software resets.) This section explains the hardware reset.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level “L” (0.2V_{CC} max.) for at least 20 cycles. When the reset pin level is then returned to the “H” level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Fig.VB-1 shows the example reset circuit. Fig.VB-2 shows the reset sequence.

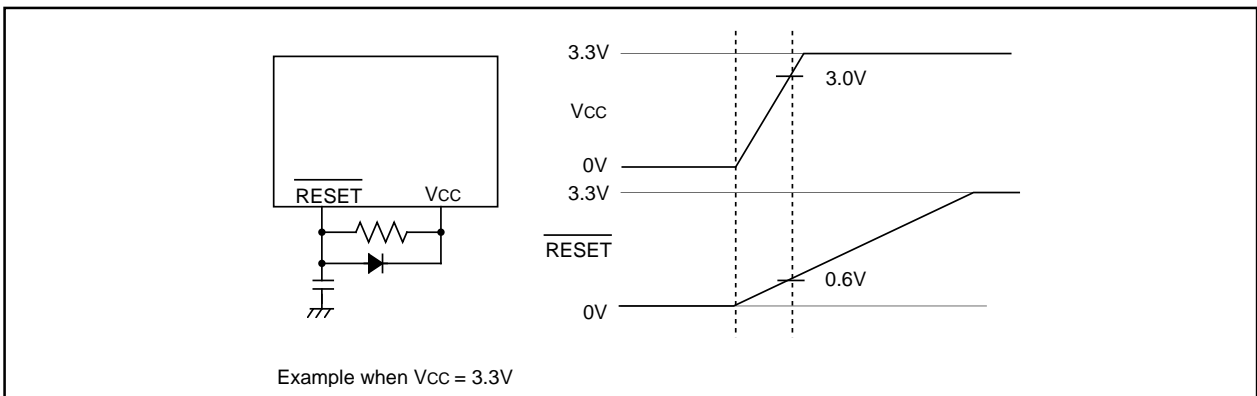


Fig.VB-1 Example reset circuit

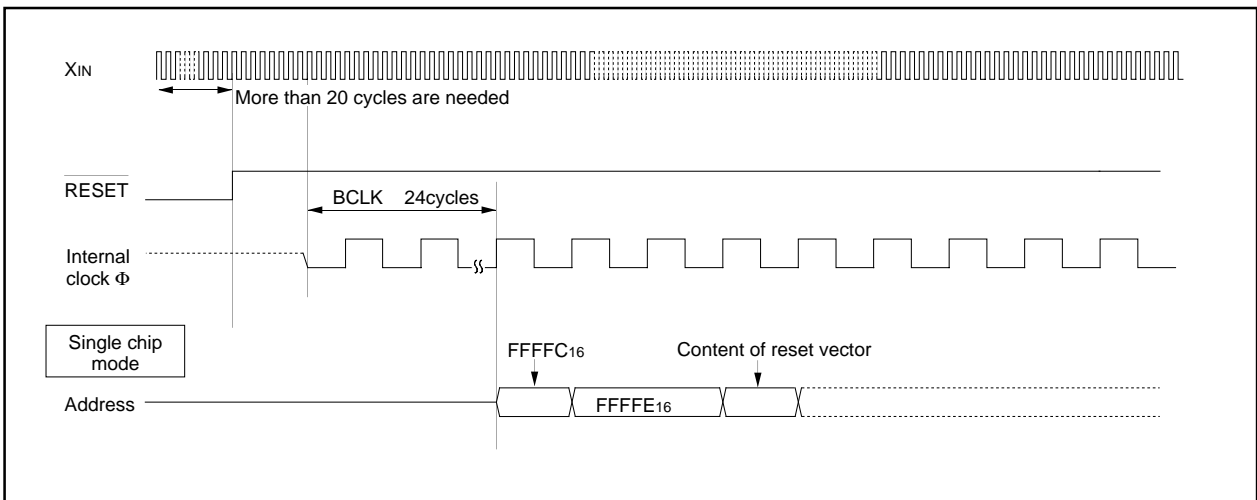


Fig.VB-2 Reset sequence

Table VB-1 shows the statuses of the other pins while the $\overline{\text{RESET}}$ pin level is "L". Fig.VB-3 and VB-4 show the internal status of the microcomputer immediately after the reset is cancelled.

Table VB-1 Pin status when $\overline{\text{RESET}}$ pin level is "L"

Pin name	Status
	CNV _{SS} = V _{SS} (Mo)
P0	I/O port (floating)
P1	I/O port (floating)
P2, P3, P40 to P43	I/O port (floating)
P44	I/O port (floating)
P45 to P47	I/O port (floating)
P50	I/O port (floating)
P51	I/O port (floating)
P52	I/O port (floating)
P53	I/O port (floating)
P54	I/O port (floating)
P55	I/O port (floating)
P56	I/O port (floating)
P57	I/O port (floating)
P6, P7, P80 to P84, P86, P87, P9, P10	I/O port (floating)
P11, P12, P13, P14	I/O port (floating)
P15, P16	I/O port (floating)

(1) Processor mode register 0	(000416)...	0016	(26) Timer B5 interrupt control register	(005416)...	XXXXXXXX?000
(2) Processor mode register 1	(000516)...	000000XX0	(27) OBE interrupt control register	(005516)...	XXXXXXXX?000
(3) System clock control register 0	(000616)...	01001000	(28) PS20 interrupt control register	(005616)...	XXXXXXXX?000
(4) System clock control register 1	(000716)...	00100000	(29) PS21 interrupt control register	(005716)...	XXXXXXXX?000
(5) Address match interrupt enable register	(000916)...	XXXXXXXX00	(30) PS22 interrupt control register	(005816)...	XXXXXXXX?000
(6) Protect register	(000A16)...	XXXXXXXX000	(31) UART1 receive interrupt control register	(005B16)...	XXXXXXXX?000
(7) Watchdog timer control register	(000F16)...	000???	(32) UART1 transmit interrupt control register	(005C16)...	XXXXXXXX?000
(8) Address match interrupt register 0	(001016)...	0016	(33) Key input 0 interrupt control register	(005F16)...	XXXXXXXX?000
	(001116)...	0016	(34) Key input 1 interrupt control register	(006016)...	XXXXXXXX?000
	(001216)...	XXXXXXXX0000	(35) SI/O3 interrupt control register	(006116)...	XXXXXXXX?000
(9) Address match interrupt register 1	(001416)...	0016	(36) SI/O4 interrupt control register	(006216)...	XXXXXXXX?000
	(001516)...	0016	(37) I ² C0 interrupt control register	(006316)...	XXXXXXXX?000
	(001616)...	XXXXXXXX0000	(38) SCL0, SDA0 interrupt control register	(006416)...	XXXXXXXX?000
(10) LRESET interrupt control register	(004116)...	XXXXXXXX?000	(39) I ² C1 interrupt control register	(006516)...	XXXXXXXX?000
(11) A-D interrupt control register	(004416)...	XXXXXXXX?000	(40) SCL1, SDA1 interrupt control register	(006616)...	XXXXXXXX?000
(12) IBF0 interrupt control register	(004516)...	XXXXXXXX?000	(41) I ² C2 interrupt control register	(006716)...	XXXXXXXX?000
(13) IBF1 interrupt control register	(004616)...	XXXXXXXX?000	(42) SCL2, SDA2 interrupt control register	(006816)...	XXXXXXXX?000
(14) IBF2 interrupt control register	(004716)...	XXXXXXXX?000	(43) INT0 interrupt control register	(006916)...	XX00?000
(15) IBF3 interrupt control register	(004816)...	XXXXXXXX?000	(44) INT1 interrupt control register	(006A16)...	XX00?000
(16) Timer A0 interrupt control register	(004A16)...	XXXXXXXX?000	(45) INT2 interrupt control register	(006B16)...	XX00?000
(17) Timer A1 interrupt control register	(004B16)...	XXXXXXXX?000	(46) INT3 interrupt control register	(006C16)...	XX00?000
(18) Timer A2 interrupt control register	(004C16)...	XXXXXXXX?000	(47) INT4 interrupt control register	(006D16)...	XX00?000
(19) Timer A3 interrupt control register	(004D16)...	XXXXXXXX?000	(48) INT5 interrupt control register	(006E16)...	XX00?000
(20) Timer A4 interrupt control register	(004E16)...	XXXXXXXX?000	(49) INT6 interrupt control register	(006F16)...	XX00?000
(21) Timer B0 interrupt control register	(004F16)...	XXXXXXXX?000	(50) INT7 interrupt control register	(007016)...	XX00?000
(22) Timer B1 interrupt control register	(005016)...	XXXXXXXX?000	(51) INT8 interrupt control register	(007116)...	XX00?000
(23) Timer B2 interrupt control register	(005116)...	XXXXXXXX?000	(52) INT9 interrupt control register	(007216)...	XX00?000
(24) Timer B3 interrupt control register	(005216)...	XXXXXXXX?000	(53) INT10 interrupt control register	(007316)...	XX00?000
(25) Timer B4 interrupt control register	(005316)...	XXXXXXXX?000	(54) INT11 interrupt control register	(007416)...	XX00?000

x : Nothing is mapped to this bit
? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Fig.VB-3 Device's internal status after a reset is cleared (1)

(55) PS20 shift register	(02A0 ₁₆)...	00 ₁₆	(85) PWM control register 0	(030C ₁₆)...	00 ₁₆
(56) PS20 status register	(02A1 ₁₆)...	00 ₁₆	(86) PWM control register 1	(030D ₁₆)...	00 ₁₆
(57) PS20 control register	(02A2 ₁₆)...	00 ₁₆	(87) I ² C2 address register	(0312 ₁₆)...	00 ₁₆
(58) PS21 shift register	(02A4 ₁₆)...	00 ₁₆	(88) I ² C2 control register 0	(0313 ₁₆)...	00 ₁₆
(59) PS21 status register	(02A5 ₁₆)...	00 ₁₆	(89) I ² C2 clock control register	(0314 ₁₆)...	00 ₁₆
(60) PS21 control register	(02A6 ₁₆)...	00 ₁₆	(90) I ² C2 start/stop condition control register	(0315 ₁₆)...	1A ₁₆
(61) PS22 shift register	(02A8 ₁₆)...	00 ₁₆	(91) I ² C2 control register 1	(0316 ₁₆)...	30 ₁₆
(62) PS22 status register	(02A9 ₁₆)...	00 ₁₆	(92) I ² C2 control register 2	(0317 ₁₆)...	00 ₁₆
(63) PS22 control register	(02AA ₁₆)...	00 ₁₆	(93) I ² C2 status register	(0318 ₁₆)...	0 0 1 0 0 0 0 X
(64) PS2 mode register	(02AC ₁₆)...	00 ₁₆	(94) I ² C0 address register	(0322 ₁₆)...	00 ₁₆
(65) Data bus buffer status register 0	(02C1 ₁₆)...	00 ₁₆	(95) I ² C0 control register 0	(0323 ₁₆)...	00 ₁₆
(66) Data bus buffer status register 1	(02C3 ₁₆)...	00 ₁₆	(96) I ² C0 clock control register	(0324 ₁₆)...	00 ₁₆
(67) Data bus buffer status register 2	(02C5 ₁₆)...	00 ₁₆	(97) I ² C0 start/stop condition control register	(0325 ₁₆)...	1A ₁₆
(68) Data bus buffer status register 3	(02C7 ₁₆)...	00 ₁₆	(98) I ² C0 control register 1	(0326 ₁₆)...	30 ₁₆
(69) Data bus buffer control register 1	(02C9 ₁₆)...	00 ₁₆	(99) I ² C0 control register 2	(0327 ₁₆)...	00 ₁₆
(70) GateA20 control register	(02CA ₁₆)...	00 ₁₆	(100) I ² C0 status register	(0328 ₁₆)...	0 0 1 0 0 0 0 X
(71) Port P11 direction register	(02E2 ₁₆)...	00 ₁₆	(101) I ² C1 address register	(0332 ₁₆)...	00 ₁₆
(72) Port P12 direction register	(02E3 ₁₆)...	00 ₁₆	(102) I ² C1 control register 0	(0333 ₁₆)...	00 ₁₆
(73) Port P13 direction register	(02E6 ₁₆)...	00 ₁₆	(103) I ² C1 clock control register	(0334 ₁₆)...	00 ₁₆
(74) Port P14 direction register	(02E7 ₁₆)...	00 ₁₆	(104) I ² C1 start/stop condition control register	(0335 ₁₆)...	1A ₁₆
(75) Port P15 direction register	(02EA ₁₆)...	00 ₁₆	(105) I ² C1 control register 1	(0336 ₁₆)...	30 ₁₆
(76) Port P16 direction register	(02EB ₁₆)...	X X X X X X X 0 0	(106) I ² C1 control register 2	(0337 ₁₆)...	00 ₁₆
(77) Port function selection register 0	(02F8 ₁₆)...	00 ₁₆	(107) I ² C1 status register	(0338 ₁₆)...	0 0 1 0 0 0 0 X
(78) Port function selection register 1	(02F9 ₁₆)...	00 ₁₆	(108) TimerB3,4,5 count start flag	(0340 ₁₆)...	0 0 0 X X X X X X X
(79) Port P4 input register	(02FA ₁₆)...	0 X X X X X X X X	(109) TimerB3 mode register	(035B ₁₆)...	0 0 ? X 0 0 0 0
(80) Port P7 input register	(02FB ₁₆)...	0 0 X X X X X X X	(110) TimerB4 mode register	(035C ₁₆)...	0 0 ? X 0 0 0 0
(81) Pull-up control register 3	(02FC ₁₆)...	00 ₁₆	(111) TimerB5 mode register	(035D ₁₆)...	0 0 ? X 0 0 0 0
(82) Pull-up control register 4	(02FD ₁₆)...	00 ₁₆	(112) Interrupt factor selection register 1	(035E ₁₆)...	00 ₁₆
(83) Port control register 1	(02FE ₁₆)...	00 ₁₆	(113) Interrupt factor selection register 0	(035F ₁₆)...	00 ₁₆
(84) Port control register 2	(02FF ₁₆)...	00 ₁₆	(114) SI/O3 control register	(0362 ₁₆)...	40 ₁₆
			(115) SI/O4 control register	(0366 ₁₆)...	40 ₁₆

x : Nothing is mapped to this bit
 ? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Fig.VB-4 Device's internal status after a reset is cleared (2)

(116) Count start flag	(0380 ₁₆)...	00 ₁₆	(138) Port P0 direction register	(03E2 ₁₆)...	00 ₁₆
(117) One-shot start flag	(0382 ₁₆)...	0 0 x 0 0 0 0 0	(139) Port P1 direction register	(03E3 ₁₆)...	00 ₁₆
(118) Trigger select flag	(0383 ₁₆)...	00 ₁₆	(140) Port P2 direction register	(03E6 ₁₆)...	00 ₁₆
(119) Up-down flag	(0384 ₁₆)...	00 ₁₆	(141) Port P3 direction register	(03E7 ₁₆)...	00 ₁₆
(120) Timer A0 mode register	(0396 ₁₆)...	00 ₁₆	(142) Port P4 direction register	(03EA ₁₆)...	00 ₁₆
(121) Timer A1 mode register	(0397 ₁₆)...	00 ₁₆	(143) Port P5 direction register	(03EB ₁₆)...	00 ₁₆
(122) Timer A2 mode register	(0398 ₁₆)...	00 ₁₆	(144) Port P6 direction register	(03EE ₁₆)...	00 ₁₆
(123) Timer A3 mode register	(0399 ₁₆)...	00 ₁₆	(145) Port P7 direction register	(03EF ₁₆)...	00 ₁₆
(124) Timer A4 mode register	(039A ₁₆)...	00 ₁₆	(146) Port P8 direction register	(03F2 ₁₆)...	0 0 x 0 0 0 0 0
(125) Timer B0 mode register	(039B ₁₆)...	0 0 ? x 0 0 0 0	(147) Port P9 direction register	(03F3 ₁₆)...	00 ₁₆
(126) Timer B1 mode register	(039C ₁₆)...	0 0 ? x 0 0 0 0	(148) Port P10 direction register	(03F6 ₁₆)...	00 ₁₆
(127) Timer B2 mode register	(039D ₁₆)...	0 0 ? x 0 0 0 0	(149) Pull-up control register 0	(03FC ₁₆)...	00 ₁₆
(128) UART1 transmit/receive mode register	(03A8 ₁₆)...	00 ₁₆	(150) Pull-up control register 1	(03FD ₁₆)...	00 ₁₆
(129) UART1 transmit/receive control register 0	(03AC ₁₆)...	0 0 0 0 1 0 0 0	(151) Pull-up control register 2	(03FE ₁₆)...	00 ₁₆
(130) UART1 transmit/receive control register 1	(03AD ₁₆)...	0 0 0 0 0 0 1 0	(152) Port control register 0	(03FF ₁₆)...	00 ₁₆
(131) UART transmit/receive control register 2	(03B0 ₁₆)...	x 0 0 0 0 0 0 0	(153) Data registers (R0/R1/R2/R3)		0000 ₁₆
(132) Flash memory recognition register (Note1)	(03B4 ₁₆)...	x x x x x x x 1 1	(154) Address registers (A0/A1)		0000 ₁₆
(133) Flash memory control register1 (Note1)	(03B5 ₁₆)...	0 0 0 x x x 0 x	(155) Frame base register (FB)		0000 ₁₆
(134) Flash memory control register0 (Note1)	(03B7 ₁₆)...	0 0 0 0 0 0 0 1	(156) Interrupt table register (INTB)		00000 ₁₆
(135) A-D control register 2	(03D4 ₁₆)...	0 0 0 1 0 0 0 0	(157) User stack pointer (USP)		0000 ₁₆
(136) A-D control register 0	(03D6 ₁₆)...	0 0 0 0 0 ? ? ?	(158) Interrupt stack pointer (ISP)		0000 ₁₆
(137) A-D control register 1	(03D7 ₁₆)...	00 ₁₆	(159) Static base register (SB)		0000 ₁₆
			(160) Flag register (FLG)		0000 ₁₆

x : Nothing is mapped to this bit
? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

(Note1) This register exists only in the flash memory version.

Fig.VB-5 Device's internal status after a reset is cleared (3)

(161) Serial interrupt control register 0	(02B0 ₁₆)...	00 ₁₆
(162) Serial interrupt control register 1	(02B1 ₁₆)...	00 ₁₆
(163) IRQ request register 0	(02B2 ₁₆)...	00 ₁₆
(164) IRQ request register 1	(02B3 ₁₆)...	00 ₁₆
(165) IRQ request register 2	(02B4 ₁₆)...	00 ₁₆
(166) IRQ request register 3	(02B5 ₁₆)...	00 ₁₆
(167) IRQ request register 4	(02B6 ₁₆)...	00 ₁₆
(168) Serial interrupt control register 2	(02B7 ₁₆)...	10 ₁₆
(169) LPC1 address register L	(02D0 ₁₆)...	00 ₁₆
(170) LPC1 address register H	(02D1 ₁₆)...	00 ₁₆
(171) LPC2 address register L	(02D2 ₁₆)...	00 ₁₆
(172) LPC2 address register H	(02D3 ₁₆)...	00 ₁₆
(173) LPC3 address register L	(02D4 ₁₆)...	00 ₁₆
(174) LPC3 address register H	(02D5 ₁₆)...	00 ₁₆
(175) LPC control register	(02D6 ₁₆)...	00 ₁₆
(176) Port function selection register 2	(02F1 ₁₆)...	00 ₁₆
(177) Pull-up resistor control register 5	(02F2 ₁₆)...	00 ₁₆
(178) Pull-up resistor control register 6	(02F3 ₁₆)...	00 ₁₆
(179) Key input interrupt 1 enable register	(02F4 ₁₆)...	00 ₁₆
(180) Key input interrupt 1 edge selection register	(02F5 ₁₆)...	00 ₁₆
(181) P14 event register	(02F6 ₁₆)...	00 ₁₆
(182) Port control register 3	(02F7 ₁₆)...	00 ₁₆

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Fig.VB-6 Device's internal status after a reset is cleared (4)

0000 ₁₆		0040 ₁₆	
0001 ₁₆		0041 ₁₆	LRESET interrupt control register (LRSTIC)
0002 ₁₆		0042 ₁₆	
0003 ₁₆		0043 ₁₆	
0004 ₁₆	Processor mode register 0 (PM0)	0044 ₁₆	A-D interrupt control register (A-DIC)
0005 ₁₆	Processor mode register 1 (PM1)	0045 ₁₆	IBF0 interrupt control register (IBF0IC)
0006 ₁₆	System clock control register 0 (CM0)	0046 ₁₆	IBF1 interrupt control register (IBF1IC)
0007 ₁₆	System clock control register 1 (CM1)	0047 ₁₆	IBF2 interrupt control register (IBF2IC)
0008 ₁₆		0048 ₁₆	IBF3 interrupt control register (IBF3IC)
0009 ₁₆	Address match interrupt enable register (AIER)	0049 ₁₆	
000A ₁₆	Protect register (PRCR)	004A ₁₆	Timer A0 interrupt control register (TA0IC)
000B ₁₆		004B ₁₆	Timer A1 interrupt control register (TA1IC)
000C ₁₆		004C ₁₆	Timer A2 interrupt control register (TA2IC)
000D ₁₆		004D ₁₆	Timer A3 interrupt control register (TA3IC)
000E ₁₆	Watchdog timer start register (WDTS)	004E ₁₆	Timer A4 interrupt control register (TA4IC)
000F ₁₆	Watchdog timer control register (WDC)	004F ₁₆	Timer B0 interrupt control register (TB0IC)
0010 ₁₆		0050 ₁₆	Timer B1 interrupt control register (TB1IC)
0011 ₁₆	Address match interrupt register 0 (RMAD0)	0051 ₁₆	Timer B2 interrupt control register (TB2IC)
0012 ₁₆		0052 ₁₆	Timer B3 interrupt control register (TB3IC)
0013 ₁₆		0053 ₁₆	Timer B4 interrupt control register (TB4IC)
0014 ₁₆		0054 ₁₆	Timer B5 interrupt control register (TB5IC)
0015 ₁₆	Address match interrupt register 1 (RMAD1)	0055 ₁₆	OBE interrupt control register (OBEIC)
0016 ₁₆		0056 ₁₆	PS20 interrupt control register (PS20IC)
0017 ₁₆		0057 ₁₆	PS21 interrupt control register (PS21IC)
0018 ₁₆		0058 ₁₆	PS22 interrupt control register (PS22IC)
0019 ₁₆		0059 ₁₆	
001A ₁₆		005A ₁₆	
001B ₁₆		005B ₁₆	UART1 receive interrupt control register (S1RIC)
001C ₁₆		005C ₁₆	UART1 transmit interrupt control register (S1TIC)
001D ₁₆		005D ₁₆	
001E ₁₆		005E ₁₆	
001F ₁₆		005F ₁₆	Key input interrupt 0 control register (KUP0IC)
0020 ₁₆		0060 ₁₆	Key input interrupt 1 control register (KUP1IC)
0021 ₁₆		0061 ₁₆	SI/O3 interrupt control register (S3IC)
0022 ₁₆		0062 ₁₆	SI/O4 interrupt control register (S4IC)
0023 ₁₆		0063 ₁₆	I ² C0 interrupt control register (IIC0IC)
0024 ₁₆		0064 ₁₆	SCL0,SDA0 interrupt control register (SCLDA0IC)
0025 ₁₆		0065 ₁₆	I ² C1 interrupt control register (IIC1IC)
0026 ₁₆		0066 ₁₆	SCL1,SDA1 interrupt control register (SCLDA1IC)
0027 ₁₆		0067 ₁₆	I ² C2 interrupt control register (IIC2IC)
0028 ₁₆		0068 ₁₆	SCL2,SDA2 interrupt control register (SCLDA2IC)
0029 ₁₆		0069 ₁₆	INT0 interrupt control register (INT0IC)
002A ₁₆		006A ₁₆	INT1 interrupt control register (INT1IC)
002B ₁₆		006B ₁₆	INT2 interrupt control register (INT2IC)
002C ₁₆		006C ₁₆	INT3 interrupt control register (INT3IC)
002D ₁₆		006D ₁₆	INT4 interrupt control register (INT4IC)
002E ₁₆		006E ₁₆	INT5 interrupt control register (INT5IC)
002F ₁₆		006F ₁₆	INT6 interrupt control register (INT6IC)
0030 ₁₆		0070 ₁₆	INT7 interrupt control register (INT7IC)
0031 ₁₆		0071 ₁₆	INT8 interrupt control register (INT8IC)
0032 ₁₆		0072 ₁₆	INT9 interrupt control register (INT9IC)
0033 ₁₆		0073 ₁₆	INT10 interrupt control register (INT10IC)
0034 ₁₆		0074 ₁₆	INT11 interrupt control register (INT11IC)
0035 ₁₆		≈	≈
0036 ₁₆			
0037 ₁₆		027E ₁₆	
0038 ₁₆		027F ₁₆	
0039 ₁₆			
003A ₁₆			
003B ₁₆			
003C ₁₆			
003D ₁₆			
003E ₁₆			
003F ₁₆			

Note 1: The areas that nothing are allocated in SFR are reserved. Read and Write to the areas are inhibited.

Fig.CA-2 Location of peripheral unit control registers (1)

0280 ₁₆		02C0 ₁₆	Data bus buffer register0 (DBB0)
0281 ₁₆		02C1 ₁₆	Data bus buffer status register0 (DBBSTS0)
0282 ₁₆		02C2 ₁₆	Data bus buffer register1 (DBB1)
0283 ₁₆		02C3 ₁₆	Data bus buffer status register1 (DBBSTS1)
0284 ₁₆		02C4 ₁₆	Data bus buffer register2 (DBB2)
0285 ₁₆		02C5 ₁₆	Data bus buffer status register2 (DBBSTS2)
0286 ₁₆		02C6 ₁₆	Data bus buffer register3 (DBB3)
0287 ₁₆		02C7 ₁₆	Data bus buffer status register3 (DBBSTS3)
0288 ₁₆		02C8 ₁₆	
0289 ₁₆		02C9 ₁₆	Data bus buffer control register1 (DBBCON1)
028A ₁₆		02CA ₁₆	Gate A20 control register (GA20CON)
028B ₁₆		02CB ₁₆	
028C ₁₆		02CC ₁₆	
028D ₁₆		02CD ₁₆	
028E ₁₆		02CE ₁₆	
028F ₁₆		02CF ₁₆	
0290 ₁₆		02D0 ₁₆	LPC1 address registerL (LPC1ADL)
0291 ₁₆		02D1 ₁₆	LPC1 address registerH (LPC1ADH)
0292 ₁₆		02D2 ₁₆	LPC2 address registerL (LPC2ADL)
0293 ₁₆		02D3 ₁₆	LPC2 address registerH (LPC2ADH)
0294 ₁₆		02D4 ₁₆	LPC3 address registerL (LPC3ADL)
0295 ₁₆		02D5 ₁₆	LPC3 address registerH (LPC3ADH)
0296 ₁₆		02D6 ₁₆	LPC control register (LPCCON)
0297 ₁₆		02D7 ₁₆	
0298 ₁₆		02D8 ₁₆	
0299 ₁₆		02D9 ₁₆	
029A ₁₆		02DA ₁₆	
029B ₁₆		02DB ₁₆	
029C ₁₆		02DC ₁₆	
029D ₁₆		02DD ₁₆	
029E ₁₆		02DE ₁₆	
029F ₁₆		02DF ₁₆	
02A0 ₁₆	PS20 shift register (PS20SR)	02E0 ₁₆	Port P11 (P11)
02A1 ₁₆	PS20 status register (PS20STS)	02E1 ₁₆	Port P12 (P12)
02A2 ₁₆	PS20 control register (PS20CON)	02E2 ₁₆	Port P11 direction register (PD11)
02A3 ₁₆		02E3 ₁₆	Port P12 direction register (PD12)
02A4 ₁₆	PS21 shift register (PS21SR)	02E4 ₁₆	Port P13 (P13)
02A5 ₁₆	PS21 status register (PS21STS)	02E5 ₁₆	Port P14 (P14)
02A6 ₁₆	PS21 control register (PS21CON)	02E6 ₁₆	Port P13 direction register (PD13)
02A7 ₁₆		02E7 ₁₆	Port P14 direction register (PD14)
02A8 ₁₆	PS22 shift register (PS22SR)	02E8 ₁₆	Port P15 (P15)
02A9 ₁₆	PS22 status register (PS22STS)	02E9 ₁₆	Port P16 (P16)
02AA ₁₆	PS22 control register (PS22CON)	02EA ₁₆	Port P15 direction register (PD15)
02AB ₁₆		02EB ₁₆	Port P16 direction register (PD16)
02AC ₁₆	PS2 mode register (PS2MOD)	02EC ₁₆	
02AD ₁₆		02ED ₁₆	
02AE ₁₆		02EE ₁₆	
02AF ₁₆		02EF ₁₆	
02B0 ₁₆	Serial Interrupt control register 0 (SERCON0)	02F0 ₁₆	
02B1 ₁₆	Serial Interrupt control register 1 (SERCON1)	02F1 ₁₆	Port function selection register 2 (PSL2)
02B2 ₁₆	IRQ request register 0 (IRQ0)	02F2 ₁₆	Pull-up resistor control register 5 (PUR5)
02B3 ₁₆	IRQ request register 1 (IRQ1)	02F3 ₁₆	Pull-up resistor control register 6 (PUR6)
02B4 ₁₆	IRQ request register 2 (IRQ2)	02F4 ₁₆	Key input interrupt 1 enable register (KIN1EN)
02B5 ₁₆	IRQ request register 3 (IRQ3)	02F5 ₁₆	Key input interrupt 1 edge selection register (KINSEL)
02B6 ₁₆	IRQ request register 4 (IRQ4)	02F6 ₁₆	P14 event register (P14EV)
02B7 ₁₆	Serial Interrupt control register 2 (SERCON2)	02F7 ₁₆	Port control register3 (PCR3)
02B8 ₁₆		02F8 ₁₆	Port function selection register0 (PSL0)
02B9 ₁₆		02F9 ₁₆	Port function selection register1 (PSL1)
02BA ₁₆		02FA ₁₆	Port P4 input register (P4PIN)
02BB ₁₆		02FB ₁₆	Port P7 input register (P7PIN)
02BC ₁₆		02FC ₁₆	Pull-up control register3 (PUR3)
02BD ₁₆		02FD ₁₆	Pull-up control register4 (PUR4)
02BE ₁₆		02FE ₁₆	Port control register1 (PCR1)
02BF ₁₆		02FF ₁₆	Port control register2 (PCR2)

Note 1: The areas that nothing are allocated in SFR are reserved. Read and Write to the areas are inhibited.

Fig.CA-3 Location of peripheral unit control registers (2)

0300 ₁₆	PWM0 prescaler (PREPWM0)	0340 ₁₆	TimerB3,4,5 count start flag (TBSR)
0301 ₁₆	PWM0 register (PWM0)	0341 ₁₆	
0302 ₁₆	PWM1 prescaler (PREPWM1)	0342 ₁₆	
0303 ₁₆	PWM1 register (PWM1)	0343 ₁₆	
0304 ₁₆	PWM2 prescaler (PREPWM2)	0344 ₁₆	
0305 ₁₆	PWM2 register (PWM2)	0345 ₁₆	
0306 ₁₆	PWM3 prescaler (PREPWM3)	0346 ₁₆	
0307 ₁₆	PWM3 register (PWM3)	0347 ₁₆	
0308 ₁₆	PWM4 prescaler (PREPWM4)	0348 ₁₆	
0309 ₁₆	PWM4 register (PWM4)	0349 ₁₆	
030A ₁₆	PWM5 prescaler (PREPWM5)	034A ₁₆	
030B ₁₆	PWM5 register (PWM5)	034B ₁₆	
030C ₁₆	PWM control register 0 (PWMCON0)	034C ₁₆	
030D ₁₆	PWM control register 1 (PWMCON1)	034D ₁₆	
030E ₁₆		034E ₁₆	
030F ₁₆		034F ₁₆	
0310 ₁₆	I ² C2 data shift register (S02)	0350 ₁₆	TimerB3 register (TB3)
0311 ₁₆		0351 ₁₆	
0312 ₁₆	I ² C2 address register (S0D2)	0352 ₁₆	TimerB4 register (TB4)
0313 ₁₆	I ² C2 control register 0 (S1D2)	0353 ₁₆	
0314 ₁₆	I ² C2 clock control register (S22)	0354 ₁₆	TimerB5 register (TB5)
0315 ₁₆	I ² C2 start/stop condition control register (S2D2)	0355 ₁₆	
0316 ₁₆	I ² C2 control register 1 (S3D2)	0356 ₁₆	
0317 ₁₆	I ² C2 control register 2 (S4D2)	0357 ₁₆	
0318 ₁₆	I ² C2 status register (S12)	0358 ₁₆	
0319 ₁₆		0359 ₁₆	
031A ₁₆		035A ₁₆	
031B ₁₆		035B ₁₆	TimerB3 mode register (TB3MR)
031C ₁₆		035C ₁₆	TimerB4 mode register (TB4MR)
031D ₁₆		035D ₁₆	TimerB5 mode register (TB5MR)
031E ₁₆		035E ₁₆	Interrupt event select register 1 (IFSR1)
031F ₁₆		035F ₁₆	Interrupt event select register 0 (IFSR0)
0320 ₁₆	I ² C0 data shift register (S00)	0360 ₁₆	SI/O3 transmit/receive register (S3TRR)
0321 ₁₆		0361 ₁₆	
0322 ₁₆	I ² C0 address register (S0D0)	0362 ₁₆	SI/O3 control register (S3C)
0323 ₁₆	I ² C0 control register0 (S1D0)	0363 ₁₆	SI/O3 communication speed register (S3BRG)
0324 ₁₆	I ² C0 clock control register (S20)	0364 ₁₆	SI/O4 transmit/receive register (S4TRR)
0325 ₁₆	I ² C0 start/stop condition control register (S2D0)	0365 ₁₆	
0326 ₁₆	I ² C0 control register1 (S3D0)	0366 ₁₆	SI/O4 control register (S4C)
0327 ₁₆	I ² C0 control register2 (S4D0)	0367 ₁₆	SI/O4 communication speed register (S4BRG)
0328 ₁₆	I ² C0 status register (S10)	0368 ₁₆	
0329 ₁₆		0369 ₁₆	
032A ₁₆		036A ₁₆	
032B ₁₆		036B ₁₆	
032C ₁₆		036C ₁₆	
032D ₁₆		036D ₁₆	
032E ₁₆		036E ₁₆	
032F ₁₆		036F ₁₆	
0330 ₁₆	I ² C1 data shift register (S01)	0370 ₁₆	
0331 ₁₆		0371 ₁₆	
0332 ₁₆	I ² C1 address register (S0D1)	0372 ₁₆	
0333 ₁₆	I ² C1 control register0 (S1D1)	0373 ₁₆	
0334 ₁₆	I ² C1 clock control register (S21)	0374 ₁₆	
0335 ₁₆	I ² C1 start/stop condition control register (S2D1)	0375 ₁₆	
0336 ₁₆	I ² C1 control register1 (S3D1)	0376 ₁₆	
0337 ₁₆	I ² C1 control register2 (S4D1)	0377 ₁₆	
0338 ₁₆	I ² C1 status register (S11)	0378 ₁₆	
0339 ₁₆		0379 ₁₆	
033A ₁₆		037A ₁₆	
033B ₁₆		037B ₁₆	
033C ₁₆		037C ₁₆	
033D ₁₆		037D ₁₆	
033E ₁₆		037E ₁₆	
033F ₁₆		037F ₁₆	

Note 1: The areas that nothing are allocated in SFR are reserved. Read and Write to the areas are inhibited.

Fig.CA-4 Location of peripheral unit control registers (3)

0380 ₁₆	Count start flag (TABSR)	03C0 ₁₆	A-D register0 (AD0)
0381 ₁₆		03C1 ₁₆	
0382 ₁₆	One-shot start flag (ONSF)	03C2 ₁₆	A-D register1 (AD1)
0383 ₁₆	Trigger select register (TRGSR)	03C3 ₁₆	
0384 ₁₆	Up-down flag (UDF)	03C4 ₁₆	A-D register2 (AD2)
0385 ₁₆		03C5 ₁₆	
0386 ₁₆	TimerA0 (TA0)	03C6 ₁₆	A-D register3 (AD3)
0387 ₁₆		03C7 ₁₆	
0388 ₁₆	TimerA1 (TA1)	03C8 ₁₆	A-D register4 (AD4)
0389 ₁₆		03C9 ₁₆	
038A ₁₆	TimerA2 (TA2)	03CA ₁₆	A-D register5 (AD5)
038B ₁₆		03CB ₁₆	
038C ₁₆	TimerA3 (TA3)	03CC ₁₆	A-D register6 (AD6)
038D ₁₆		03CD ₁₆	
038E ₁₆	TimerA4 (TA4)	03CE ₁₆	A-D register7 (AD7)
038F ₁₆		03CF ₁₆	
0390 ₁₆	TimerB0 (TB0)	03D0 ₁₆	
0391 ₁₆		03D1 ₁₆	
0392 ₁₆	TimerB1 (TB1)	03D2 ₁₆	
0393 ₁₆		03D3 ₁₆	
0394 ₁₆	TimerB2 (TB2)	03D4 ₁₆	A-D control register2 (ADCON2)
0395 ₁₆		03D5 ₁₆	
0396 ₁₆	TimerA0 mode register (TA0MR)	03D6 ₁₆	A-D control register0 (ADCON0)
0397 ₁₆	TimerA1 mode register (TA1MR)	03D7 ₁₆	A-D control register1 (ADCON1)
0398 ₁₆	TimerA2 mode register (TA2MR)	03D8 ₁₆	
0399 ₁₆	TimerA3 mode register (TA3MR)	03D9 ₁₆	
039A ₁₆	TimerA4 mode register (TA4MR)	03DA ₁₆	
039B ₁₆	TimerB0 mode register (TB0MR)	03DB ₁₆	
039C ₁₆	TimerB1 mode register (TB1MR)	03DC ₁₆	
039D ₁₆	TimerB2 mode register (TB2MR)	03DD ₁₆	
039E ₁₆		03DE ₁₆	
039F ₁₆		03DF ₁₆	
03A0 ₁₆		03E0 ₁₆	Port P0 (P0)
03A1 ₁₆		03E1 ₁₆	Port P1 (P1)
03A2 ₁₆		03E2 ₁₆	Port P0 direction register (P0D)
03A3 ₁₆		03E3 ₁₆	Port P1 direction register (P1D)
03A4 ₁₆		03E4 ₁₆	Port P2 (P2)
03A5 ₁₆		03E5 ₁₆	Port P3 (P3)
03A6 ₁₆		03E6 ₁₆	Port P2 direction register (P2D)
03A7 ₁₆		03E7 ₁₆	Port P3 direction register (P3D)
03A8 ₁₆	UART1 transmit/receive mode register (U1MR)	03E8 ₁₆	Port P4 (P4)
03A9 ₁₆	UART1 communication speed register (U1BRG)	03E9 ₁₆	Port P5 (P5)
03AA ₁₆	UART1 transmit buffer register (U1TB)	03EA ₁₆	Port P4 direction register (P4D)
03AB ₁₆		03EB ₁₆	Port P5 direction register (P5D)
03AC ₁₆	UART1 transmit/receive control register0 (U1C0)	03EC ₁₆	Port P6 (P6)
03AD ₁₆	UART1 transmit/receive control register1 (U1C1)	03ED ₁₆	Port P7 (P7)
03AE ₁₆	UART1 receive buffer register (U1RB)	03EE ₁₆	Port P6 direction register (P6D)
03AF ₁₆		03EF ₁₆	Port P7 direction register (P7D)
03B0 ₁₆	UART transmit/receive control register2 (UCON)	03F0 ₁₆	Port P8 (P8)
03B1 ₁₆		03F1 ₁₆	Port P9 (P9)
03B2 ₁₆		03F2 ₁₆	Port P8 direction register (P8D)
03B3 ₁₆		03F3 ₁₆	Port P9 direction register (P9D)
03B4 ₁₆	Flash memory identification register (FTR)	03F4 ₁₆	Port P10 (P10)
03B5 ₁₆	Flash memory control register1 (FMR1)	03F5 ₁₆	
03B6 ₁₆		03F6 ₁₆	Port P10 direction register (P10D)
03B7 ₁₆	Flash memory control register0 (FMR0)	03F7 ₁₆	
03B8 ₁₆		03F8 ₁₆	
03B9 ₁₆		03F9 ₁₆	
03BA ₁₆		03FA ₁₆	
03BB ₁₆		03FB ₁₆	
03BC ₁₆		03FC ₁₆	Pull-up control register0 (PUR0)
03BD ₁₆		03FD ₁₆	Pull-up control register1 (PUR1)
03BE ₁₆		03FE ₁₆	Pull-up control register2 (PUR2)
03BF ₁₆		03FF ₁₆	Port control register0 (PCR0)

Note 1: The areas that nothing are allocated in SFR are reserved. Read and Write to the areas are inhibited.

Fig.CA-5 Location of peripheral unit control registers (4)

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004₁₆) applies a (software) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are retained.

Processor Mode

(1) Types of Processor Mode

The single-chip mode is supported in processor mode.

- Single-chip mode

In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. Ports P0 to P16 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

Fig. BG-1 shows the structure of processor mode register 0 and processor mode register 1.

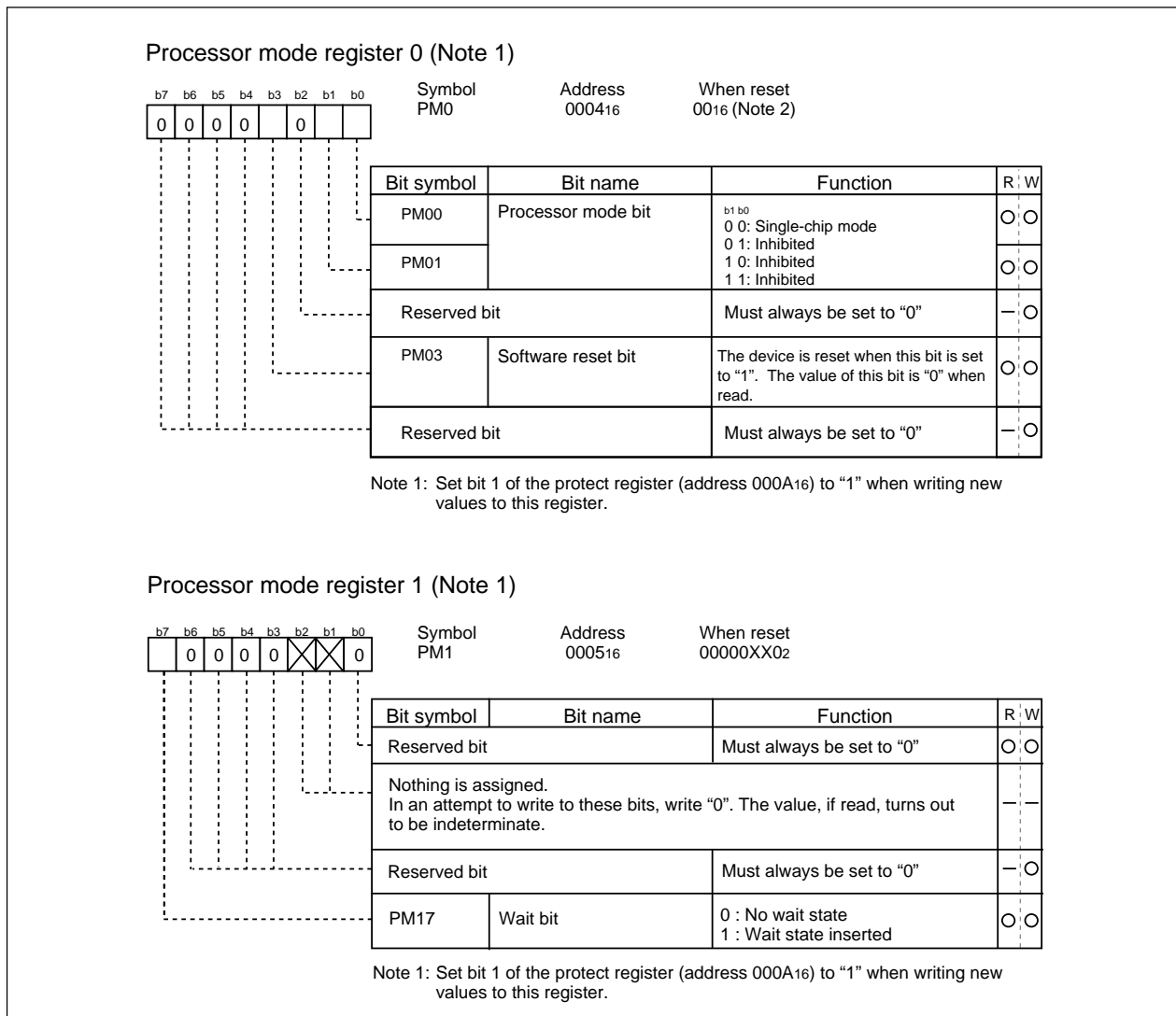


Fig.BG-1 Processor mode register 0 and 1

Bus control

(1) Software wait

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 000516) (Note) .

A software wait is inserted in the internal ROM/RAM area by setting the wait bit of the processor mode register 1. When set to "0", each bus cycle is executed in one BCLK cycle. When set to "1", each bus cycle is executed in 2 BCLK cycles. After the microcomputer has been reset, this bit defaults to "0".

Set this bit after referring to the recommended operating conditions (main clock input oscillation frequency) of the electric characteristics.

The SFR area is always accessed in two BCLK cycles regardless of the setting of these control bits.

Table.EF-1 shows the software wait and bus cycles. Fig.EF-1 shows example bus timing when using software waits.

Note: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address 000A16) to "1".

Table.EF-1 Software waits and bus cycles

Area	Wait bit	Bus cycle
SFR	Invalid	2 BCLK cycles
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

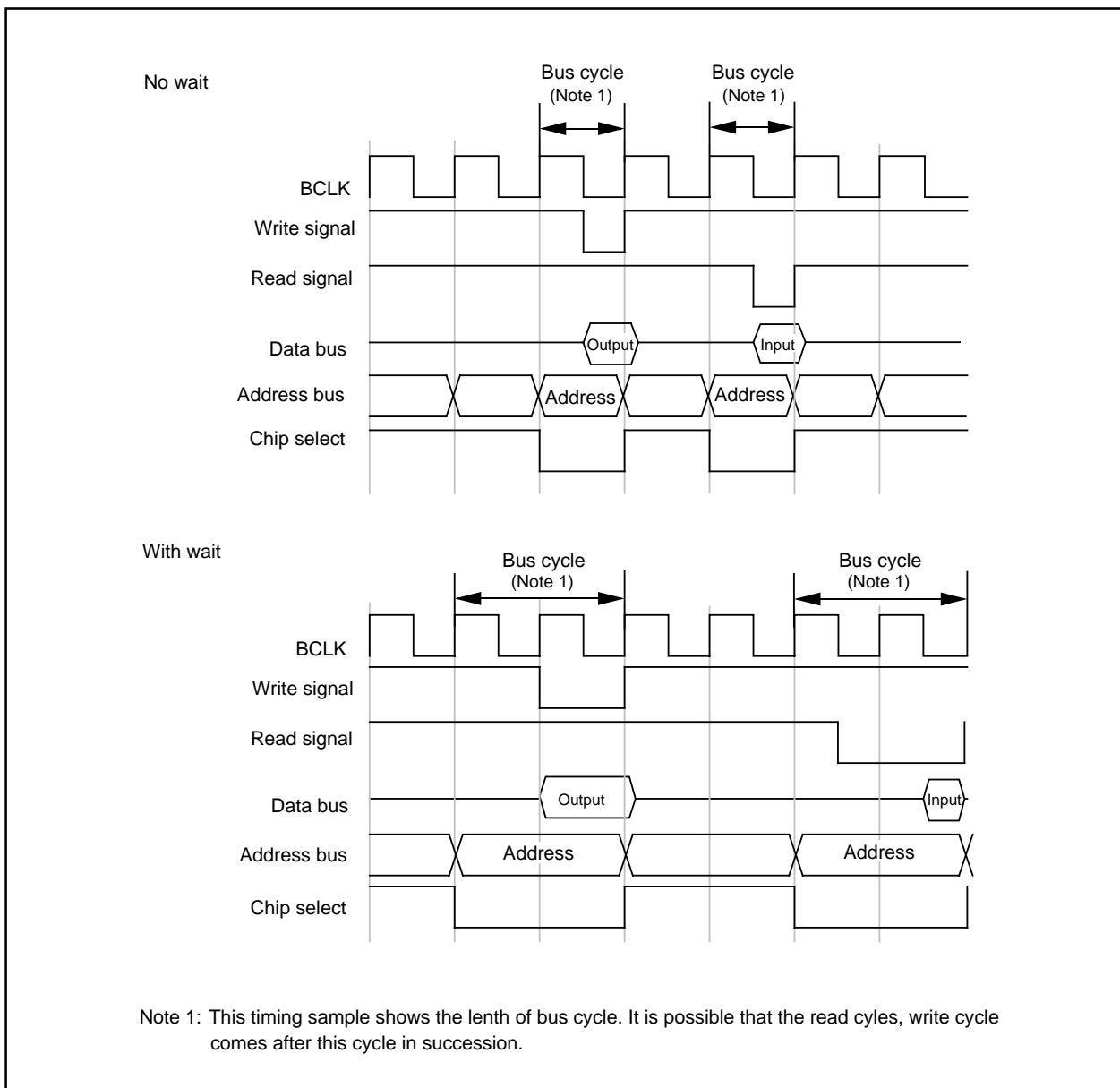


Fig.EF-1 Typical bus timings using software wait

Clock Generating Circuit

The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

Table.WA-1 Main clock generating circuits

	Main clock generating circuit
Use of clock	<ul style="list-style-type: none"> • CPU's operating clock source • Internal peripheral units' operating clock source
Usable oscillator	Ceramic oscillator
Pins to connect oscillator	XIN, XOUT
Oscillation stop/restart function	Available
Oscillator status immediately after reset	Oscillating
Other	Externally derived clock can be input

Example of oscillator circuit

Fig.WA-1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Fig.WA-1 vary with oscillator used. Use the values recommended by the manufacturer of your oscillator.

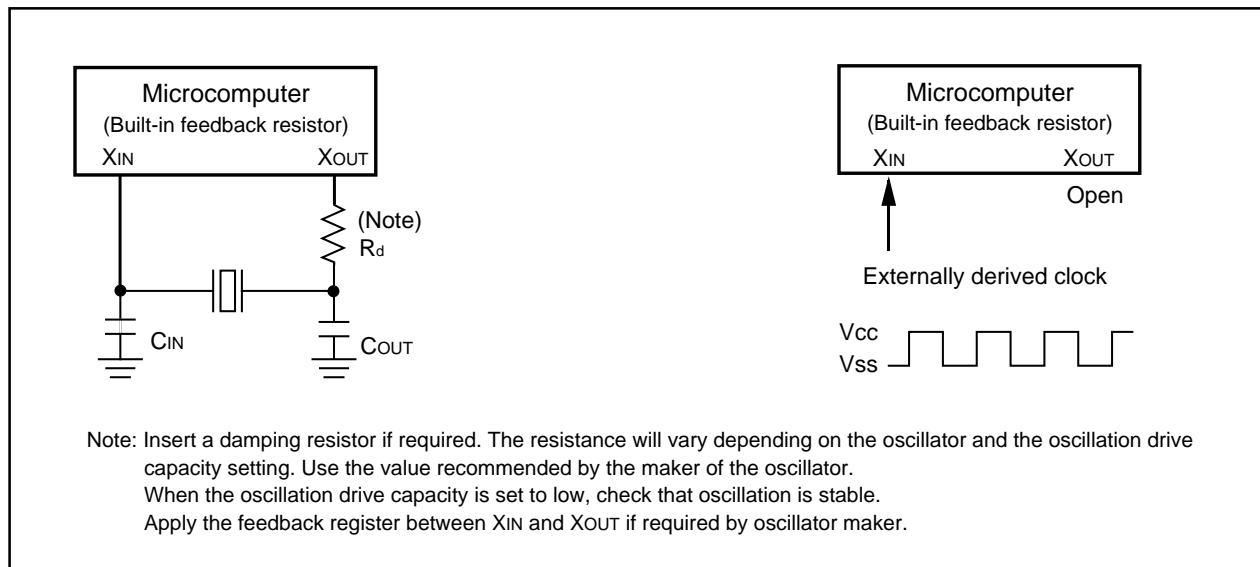


Fig.WA-1 Examples of main clock

Clock Control

Fig.WA-2 shows the block diagram of the clock generating circuit.

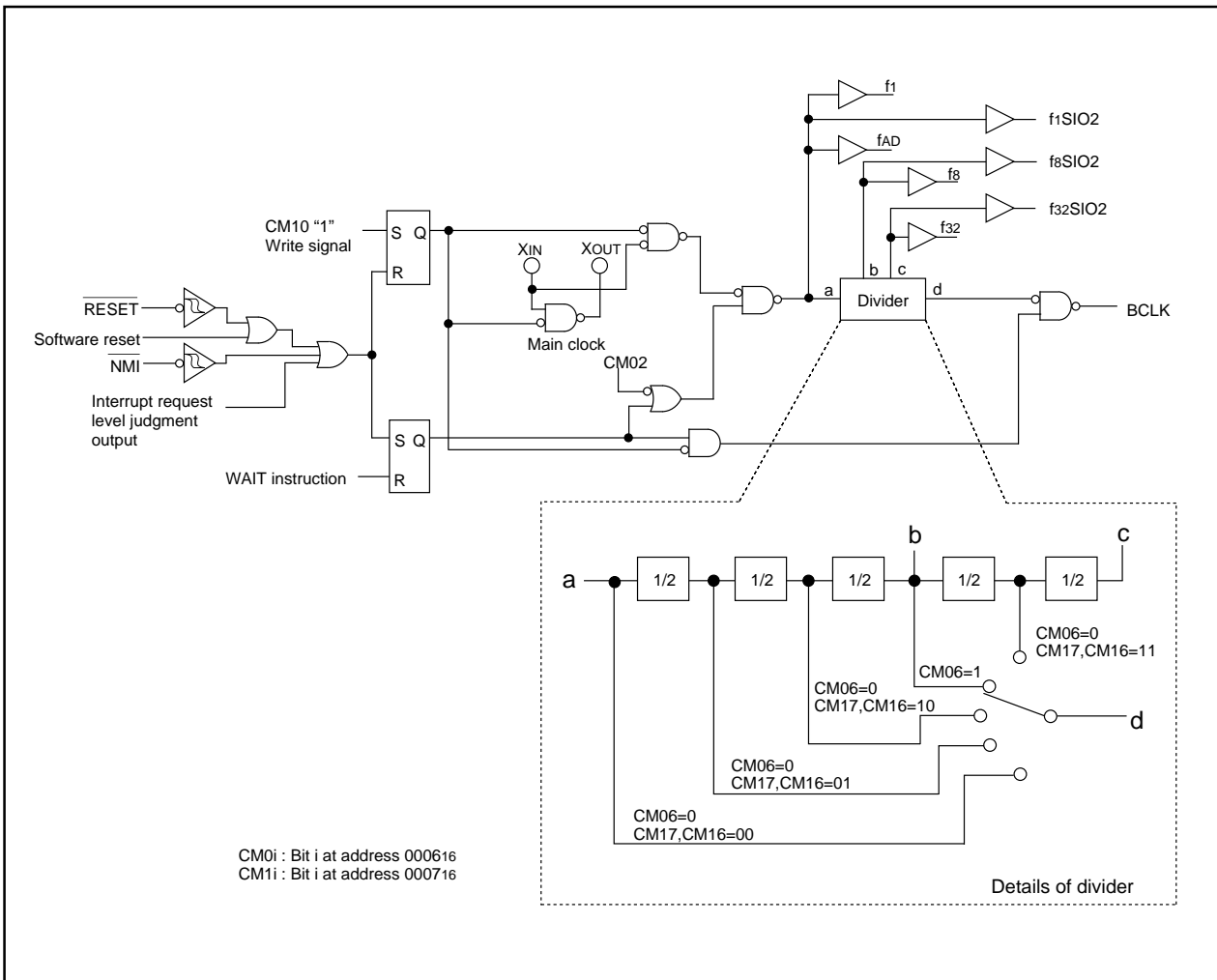


Fig.WA-2 Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006₁₆).

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007₁₆). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit defaults to "1" when shifting from high speed mode or mid-speed mode to stop mode and after a reset.

(2) BCLK

The BCLK is the clock that drives the CPU, and is either the main clock or is derived by dividing the main clock by 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset.

When shifting from high speed mode or mid-speed mode to stop mode, the main clock division select bit (bit 6 at 0006₁₆) is set to "1".

(3) Peripheral function clock

f1, f8, f32, f1SIO2, f8SIO2, f32SIO2, fAD

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006₁₆) to "1" and then executing a WAIT instruction.

Fig.WA-3 shows the system clock control registers 0 and 1.

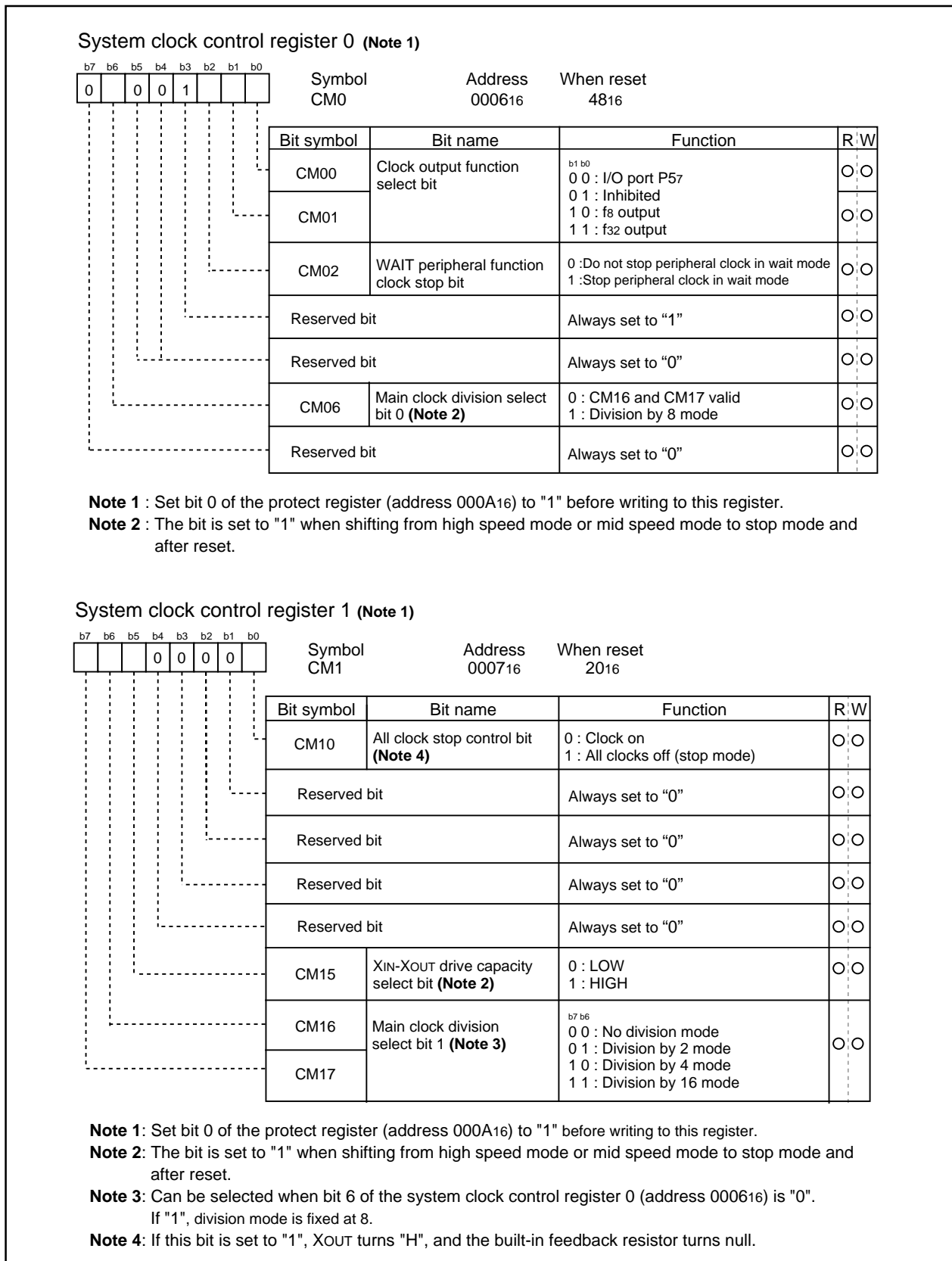


Fig.WA-3 System clock control registers 0 and 1

Clock Output

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006₁₆) enable f₈, f₃₂ to be output from the P57/CLKOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006₁₆) is set to "1", the output of f₈ and f₃₂ stops when a WAIT instruction is executed.

By setting the f₁ output function selection bits (bits 0 and 1 at address 02F1₁₆), the same frequency clock with f(XIN) can be output from P110 and P111.

Stop Mode

Writing "1" to the all-clock stop control bit (bit 0 at address 0007₁₆) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that VCC remains above 2V.

The oscillation, BCLK, f₁ to f₃₂, f₁SIO₂ to f₃₂SIO₂, and fAD stop in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UART1, SIO3,4 functions provided an external clock is selected. Table.WA-2 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled. After the restoration by interrupt, the corresponding interrupt routine will be processed. When shifting from high speed mode or mid-speed mode to stop mode, the main clock division select bit 0 (bit 6 at 0006₁₆) is set to "1".

Table.WA-2 Port status during stop mode

Pin		Single-chip mode
Port		Retains status before stop mode
CLKOUT	When f ₈ , f ₃₂ selected	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table.WA-3 shows the status of the ports in wait mode. Wait mode is cancelled by a hardware reset or interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts from interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

Table.WA-3 Port status during wait mode

Pin		Single-chip mode
CLKOUT	When f8, f32 selected	Does not stop when the WAIT peripheral function clock stop bit is "0". When the WAIT peripheral function clock stop bit is "1", the status immediately prior to entering wait mode is maintained.
Port		maintained the status immediately prior to entering wait mode

Status Transition Of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table.WA-4 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

After a reset, operation defaults to division by 8 mode. When shifting from high speed mode or mid-speed mode to stop mode, and after a reset main clock division select bit 0 (bit 6 at address 000616) is set to "1".

(1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

(2) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

(3) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. After reset, it works in this mode. Note that oscillation of the main clock must have stabilized before transferring from this mode to No-division, Division by 2 and Division by 4 mode.

(4) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

(5) No-division mode

The main clock is used as the BCLK.

Table.WA-4 Operating modes dictated by settings of system clock control registers 0 and 1

CM17	CM16	CM06	Operating mode of BCLK
0	1	0	Division by 2 mode
1	0	0	Division by 4 mode
Invalid	Invalid	1	Division by 8 mode
1	1	0	Division by 16 mode
0	0	0	No-division mode

Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Fig.WA-5 shows the protect register. The values in the processor mode register 0 (address 0004₁₆), processor mode register 1 (address 0005₁₆), system clock control register 0 (address 0006₁₆), system clock control register 1 (address 0007₁₆), can only be changed when the respective bit in the protect register is set to "1".

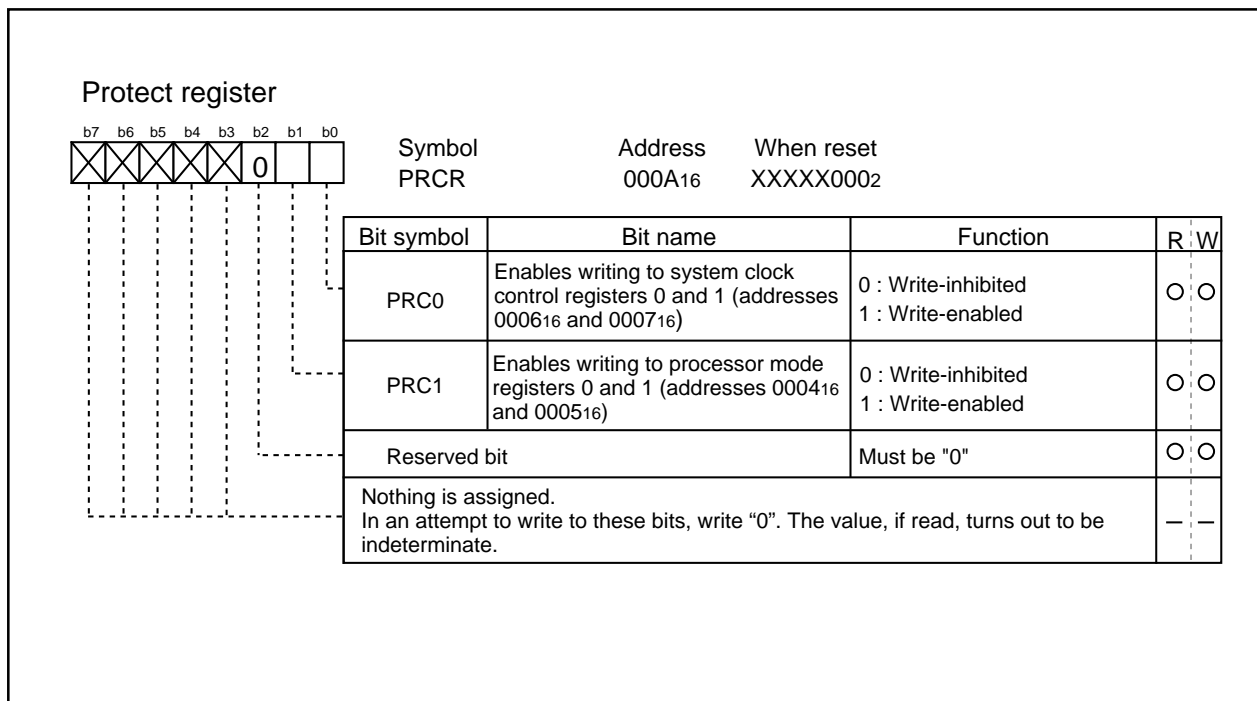


Fig.WA-5 Protect register

Overview of Interrupt

Type of Interrupts

Fig.DD-1 lists the types of interrupts.

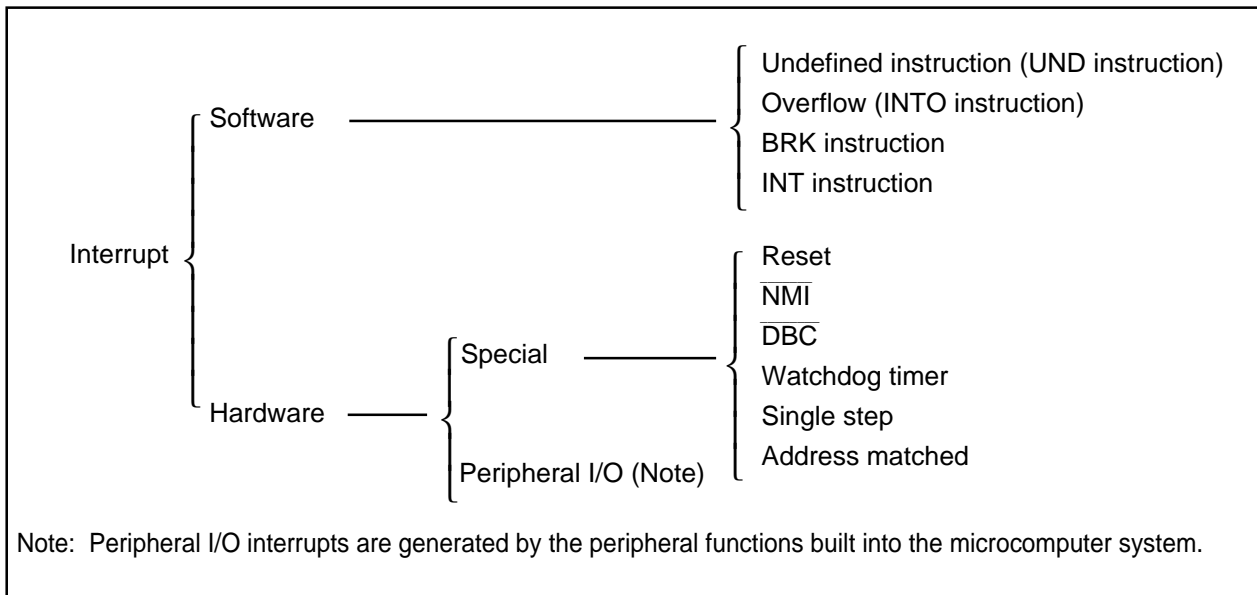


Fig.DD-1 Classification of interrupts

- Maskable interrupt : An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable interrupt : An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

- Undefined instruction interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

- Overflow interrupt

An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to “1”.

The following are instructions whose O flag changes by arithmetic:

ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

- BRK interrupt

A BRK interrupt occurs when executing the BRK instruction.

- INT interrupt

An INT interrupt occurs when assigning one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 52 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.

The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.

So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to “0” and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

(1) Special interrupts

Special interrupts are non-maskable interrupts.

- Reset

Reset occurs if an “L” is input to the $\overline{\text{RESET}}$ pin.

- $\overline{\text{NMI}}$ interrupt

An $\overline{\text{NMI}}$ interrupt occurs if an “L” is input to the $\overline{\text{NMI}}$ pin.

- $\overline{\text{DBC}}$ interrupt

This interrupt is exclusively for the debugger, do not use it in other circumstances.

- Watchdog timer interrupt

Generated by the watchdog timer.

- Single-step interrupt

This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1”, a single-step interrupt occurs after one instruction is executed.

- Address match interrupt

An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”.

If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs.

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors are also dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 52 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

1)Key-input interrupt 0

A key-input interrupt occurs if an “L” is input to the $\overline{\text{KI}}$ pin.

2)Key-input interrupt 1

A key-input interrupt occurs if an “L” or “H” is input to the $\overline{\text{KI}}$ pin.

3)A-D conversion interrupt

This is an interrupt that the A-D converter generates.

4)UART1, SI/O3 and SI/O4 transmission interrupt

These are interrupts that the serial I/O transmission generates.

5)UART1, SI/O3 and SI/O4 reception interrupt

These are interrupts that the serial I/O reception generates.

6)Timer A0 interrupt through timer A4 interrupt

These are interrupts that timer A generates

7)Timer B0 interrupt through timer B5 interrupt

These are interrupts that timer B generates.

8) $\overline{\text{INT0}}$ interrupt through $\overline{\text{INT11}}$ interrupt

An $\overline{\text{INT}}$ interrupt occurs if either a rising edge or a falling edge or both edges are input to the $\overline{\text{INT}}$ pin.

9) IBF0 to IBF3, OBE interrupt

These are interrupts that host bus interface generates.

10) $\overline{\text{LRESET}}$ interrupt

$\overline{\text{LRESET}}$ interrupt occurs if an "L" is input to $\overline{\text{LRESET}}$ pin.

11) I²C0, I²C1, I²C2, SCL0, SDA0, SCL1, SDA1, SCL2, SDA2 interrupt

These are interrupts that I²C bus interface generates.

12) PS20 to PS22 interrupt

These are interrupt that PS2 interface generates.

Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Fig.DD-2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

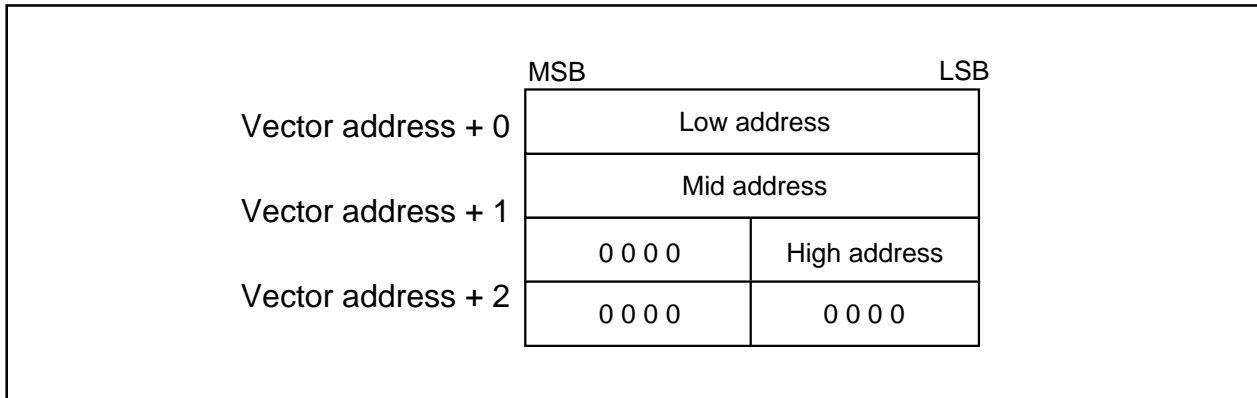


Fig.DD-2 Format for specifying interrupt vector addresses

- Fixed vector tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC₁₆ to FFFFF₁₆. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table.DD-1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table.DD-1 Interrupts assigned to the fixed vector tables and addresses of vector tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC ₁₆ to FFFDF ₁₆	Interrupt on UND instruction
Overflow	FFFE0 ₁₆ to FFFE3 ₁₆	Interrupt on INTO instruction
BRK instruction	FFFE4 ₁₆ to FFFE7 ₁₆	If the vector contains FF ₁₆ , program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8 ₁₆ to FFEF ₁₆	There is an address-matching interrupt enable bit
Single step (Note)	FFFE ₁₆ to FFEF ₁₆	Do not use
Watchdog timer	FFFF0 ₁₆ to FFFF3 ₁₆	
DBC (Note)	FFFF4 ₁₆ to FFFF7 ₁₆	Do not use
NMI	FFFF8 ₁₆ to FFFF ₁₆	External interrupt by input to NMI pin
Reset	FFFFC ₁₆ to FFFF ₁₆	

Note: Interrupts used for debugging purposes only.

- Variable vector tables

The addresses in the variable vector table can be modified, according to the user's settings. The start address of vector table is set to the interrupt table register (INTB). The 256-byte area subsequent that the start address is indicated by the INTB becomes the area for the variable vector tables. One vector table comprises 4 bytes. Set the first address of the interrupt routine in each vector table. Table.DD-2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table.DD-2 Interrupts assigned to the variable vector tables and addresses of vector

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked by I flag
Software interrupt number 1	+4 to +7 (Note 1)	$\overline{\text{LRESET}}$	
Software interrupt number 4	+16 to +19 (Note 1)	A-D	
Software interrupt number 5	+20 to +23 (Note 1)	IBF0	
Software interrupt number 6	+24 to +27 (Note 1)	IBF1	
Software interrupt number 7	+28 to +31 (Note 1)	IBF2	
Software interrupt number 8	+32 to +35 (Note 1)	IBF3	
Software interrupt number 10	+40 to +43 (Note 1)	Timer A0	
Software interrupt number 11	+44 to +47 (Note 1)	Timer A1	
Software interrupt number 12	+48 to +51 (Note 1)	Timer A2	
Software interrupt number 13	+52 to +55 (Note 1)	Timer A3	
Software interrupt number 14	+56 to +59 (Note 1)	Timer A4	
Software interrupt number 15	+60 to +63 (Note 1)	Timer B0	
Software interrupt number 16	+64 to +67 (Note 1)	Timer B1	
Software interrupt number 17	+68 to +71 (Note 1)	Timer B2	
Software interrupt number 18	+72 to +75 (Note 1)	Timer B3	
Software interrupt number 19	+76 to +79 (Note 1)	Timer B4	
Software interrupt number 20	+80 to +83 (Note 1)	Timer B5	
Software interrupt number 21	+84 to +87 (Note 1)	OBE	
Software interrupt number 22	+88 to +91 (Note 1)	PS20	
Software interrupt number 23	+92 to +95 (Note 1)	PS21	
Software interrupt number 24	+96 to +99 (Note 1)	PS22	
Software interrupt number 27	+108 to +111 (Note 1)	UART1 receive	
Software interrupt number 28	+112 to +115 (Note 1)	UART1 transmit	
Software interrupt number 31	+124 to +127 (Note 1)	Key input interrupt 0	

Note 1: Address relative to address in interrupt table register (INTB).

Table.DD-3 Interrupts assigned to the variable vector tables and addresses of vector

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 32	+128 to +131 (Note 1)	Key input interrupt 1	
Software interrupt number 33	+132 to +135 (Note 1)	SIO3	
Software interrupt number 34	+136 to +139 (Note 1)	SIO4	
Software interrupt number 35	+140 to +143 (Note 1)	I ² C0	
Software interrupt number 36	+144 to +149 (Note 1)	SCL0, SDA0	
Software interrupt number 37	+148 to +151 (Note 1)	I ² C1	
Software interrupt number 38	+152 to +155 (Note 1)	SCL1, SDA1	
Software interrupt number 39	+156 to +159 (Note 1)	I ² C2	
Software interrupt number 40	+160 to +163 (Note 1)	SCL2, SDA2	
Software interrupt number 41	+164 to +167 (Note 1)	$\overline{\text{INT0}}$	
Software interrupt number 42	+168 to +171 (Note 1)	$\overline{\text{INT1}}$	
Software interrupt number 43	+172 to +175 (Note 1)	$\overline{\text{INT2}}$	
Software interrupt number 44	+176 to +179 (Note 1)	$\overline{\text{INT3}}$	
Software interrupt number 45	+180 to +183 (Note 1)	$\overline{\text{INT4}}$	
Software interrupt number 46	+184 to +187 (Note 1)	$\overline{\text{INT5}}$	
Software interrupt number 47	+188 to +191 (Note 1)	$\overline{\text{INT6}}$	
Software interrupt number 48	+192 to +195 (Note 1)	$\overline{\text{INT7}}$	
Software interrupt number 49	+196 to +199 (Note 1)	$\overline{\text{INT8}}$	
Software interrupt number 50	+200 to +203 (Note 1)	$\overline{\text{INT9}}$	
Software interrupt number 51	+204 to +207 (Note 1)	$\overline{\text{INT10}}$	
Software interrupt number 52	+208 to +211 (Note 1)	$\overline{\text{INT11}}$	
Software interrupt number 53 to Software interrupt number 63	+212 to +215 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked by I flag

Note 1: Address relative to address in interrupt table register (INTB).

Interrupt Control

Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bits and processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bits are located in the interrupt control register of each interrupt. The interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

Fig.DD-3 and DD-4 shows the memory map of the interrupt control registers.

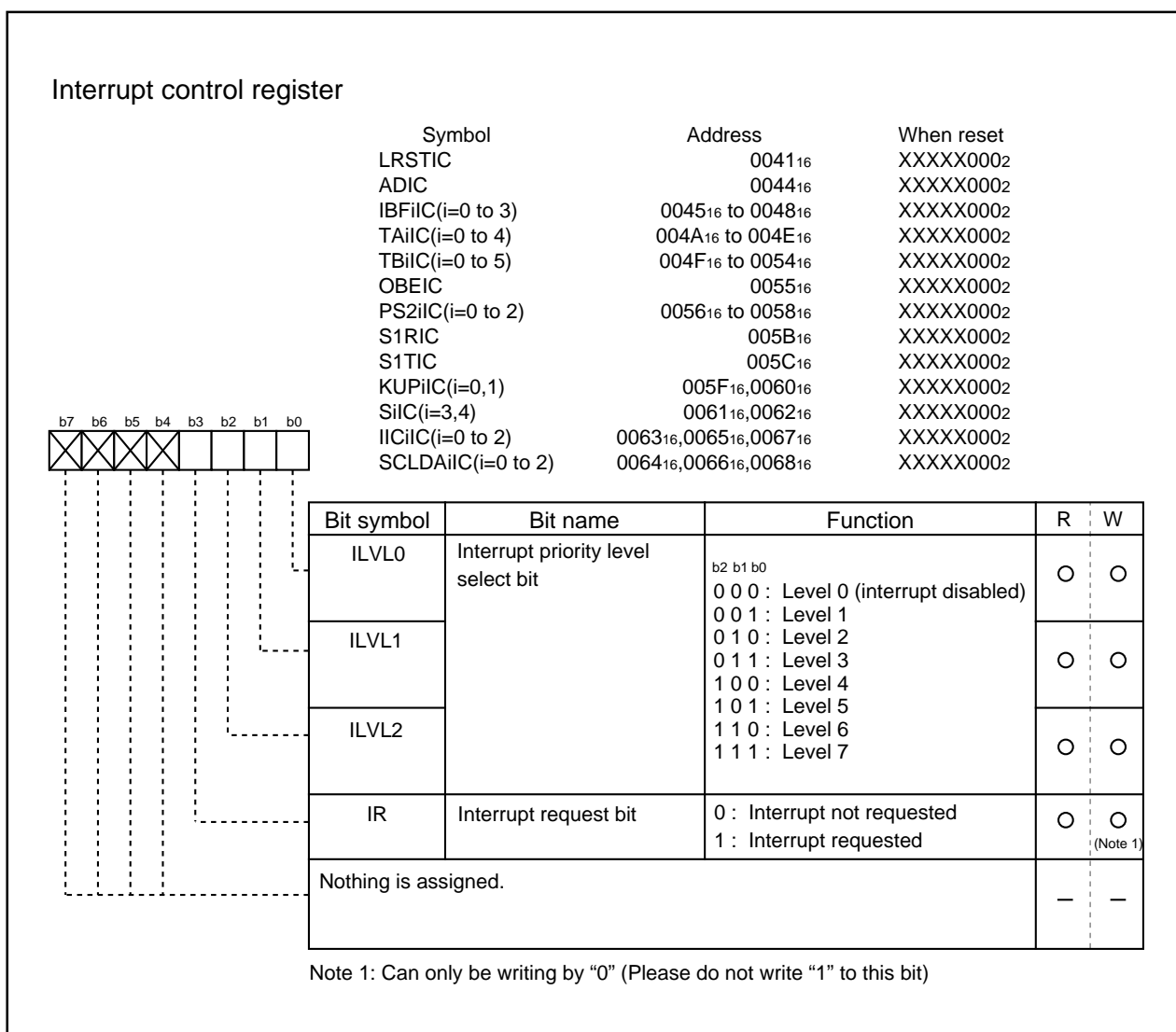


Fig.DD-3 Interrupt control registers(1)

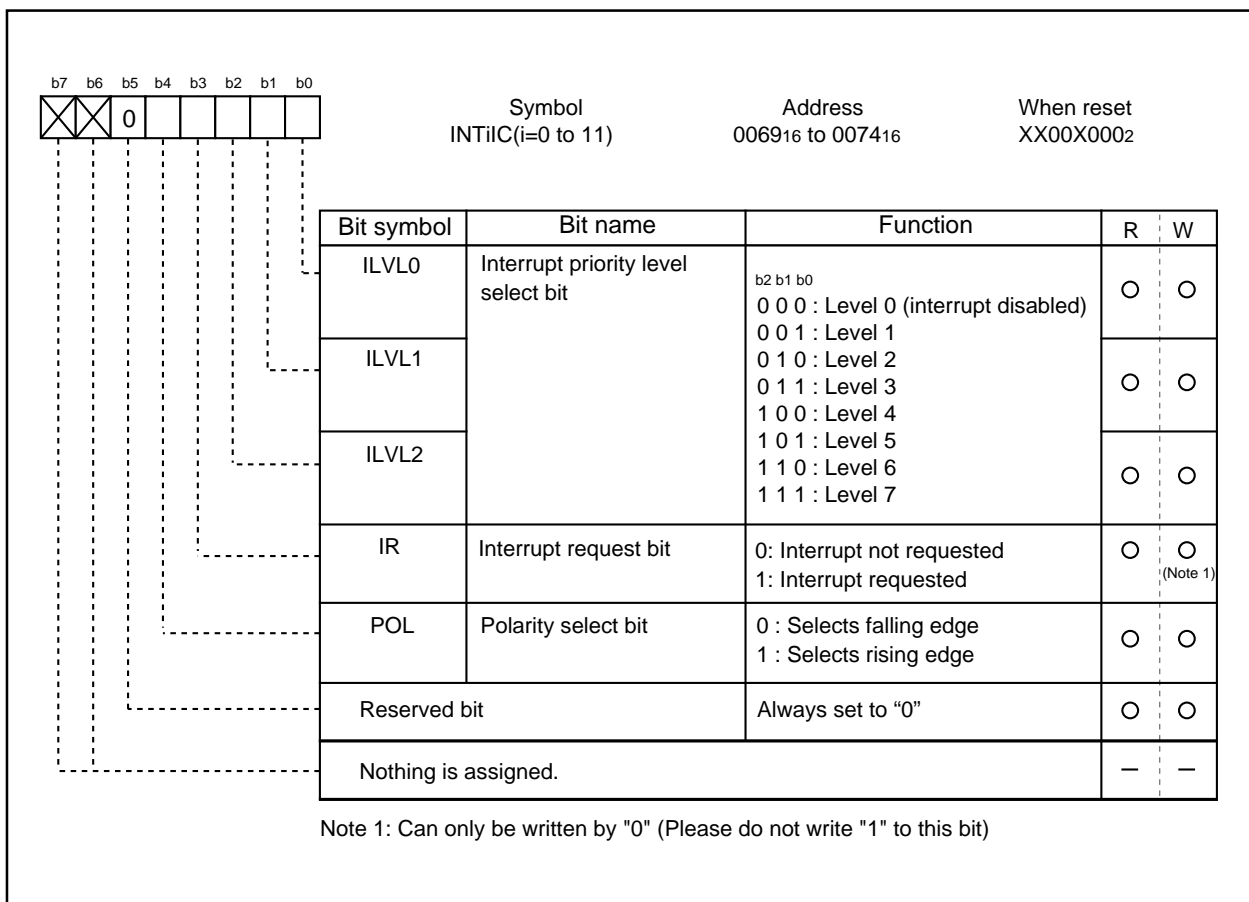


Fig.DD-4 Interrupt control registers (2)

Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to "1" enables all maskable interrupts; setting it to "0" disables all maskable interrupts. This flag is set to "0" after reset.

Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

Interrupt Priority Level Select Bits and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bits in the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to "0" disables the interrupt.

Table.DD-3 shows the settings of interrupt priority levels and Table.DD-4 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:

- interrupt enable flag (I flag) = 1
- interrupt request bit = 1
- interrupt priority level > processor interrupt priority level (IPL)

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bits, and the IPL are independent, and they are not affected each other.

Table.DD-4 Settings of interrupt priority levels


Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	————
0 0 1	Level 1	Low  High
0 1 0	Level 2	
0 1 1	Level 3	
1 0 0	Level 4	
1 0 1	Level 5	
1 1 0	Level 6	
1 1 1	Level 7	

Table.DD-5 Interrupt levels enabled according to the contents of the IPL

IPL	Enabled interrupt priority levels
IPL ₂ IPL ₁ IPL ₀ 0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

Rewrite the interrupt control register

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occurrence, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP                    ; Four NOP instructions are required when using HOLD function.
  NOP
  FSET  I           ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0     ; Dummy read.
  FSET  I           ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
  PUSHC FLG        ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG        ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

In the interrupt sequence, the processor carries out the following in sequence given:

- (1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000₁₆.
- (2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
- (3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0".
- (4) Saves the content of the temporary register (Note) within the CPU in the stack area.
- (5) Saves the content of the program counter (PC) in the stack area.
- (6) Sets the interrupt priority level of the accepted instruction in the IPL.

After the procession of interrupt sequence the processor executes instructions from the first address of the interrupt routine.

Note: This register cannot be utilized by the user.

Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Fig.DD-5 shows the interrupt response time.

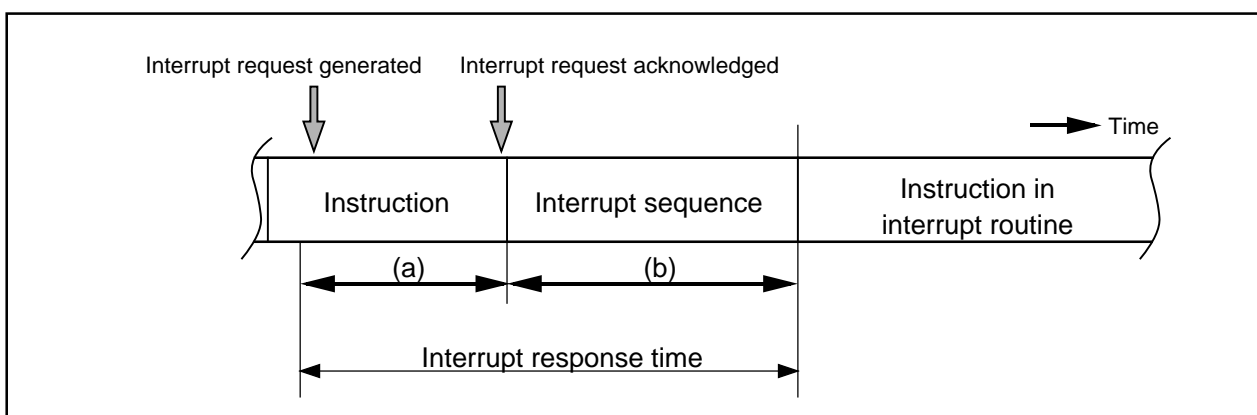


Fig.DD-5 Interrupt response time

Time (a) is dependent on the instruction under execution. 30 cycles is the maximum required for the DIVX instruction (without wait).

Time (b) is as shown in Table.DD-6

Table.DD-6 Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a \overline{DBC} interrupt; add 1 cycle in the case either of an address coincidence interrupt or of a single-step interrupt.

Note 2: Locate an interrupt vector address in an even address, if possible.

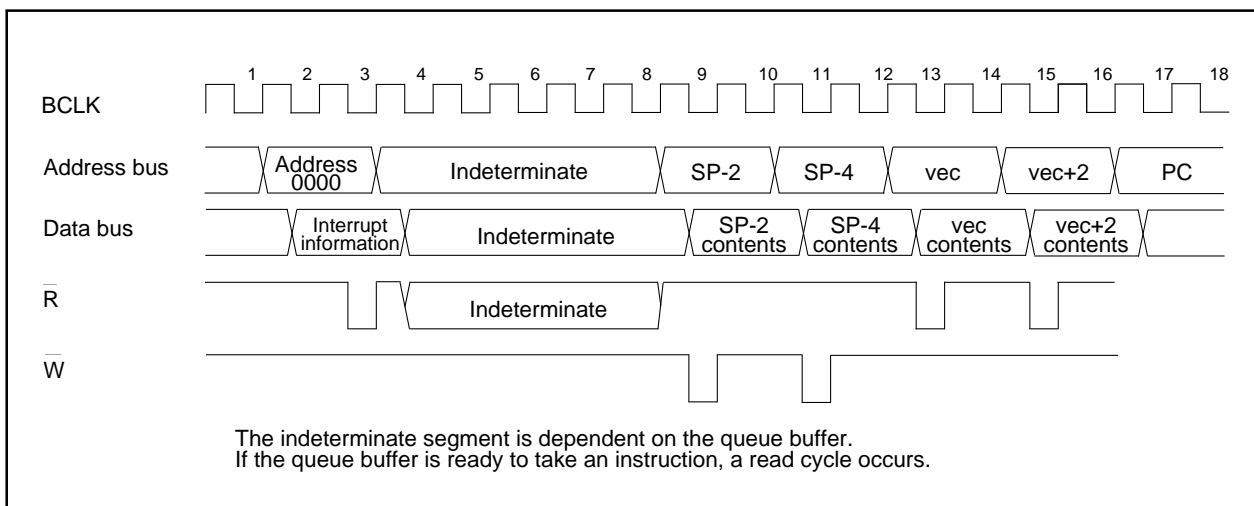


Fig.DD-6 Time required for executing the interrupt sequence

Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table.DD-7 is set in the IPL.

Table.DD-7 Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer, \overline{NMI}	7
Reset	0
Other	Not changed

Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Fig.DD-7 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).

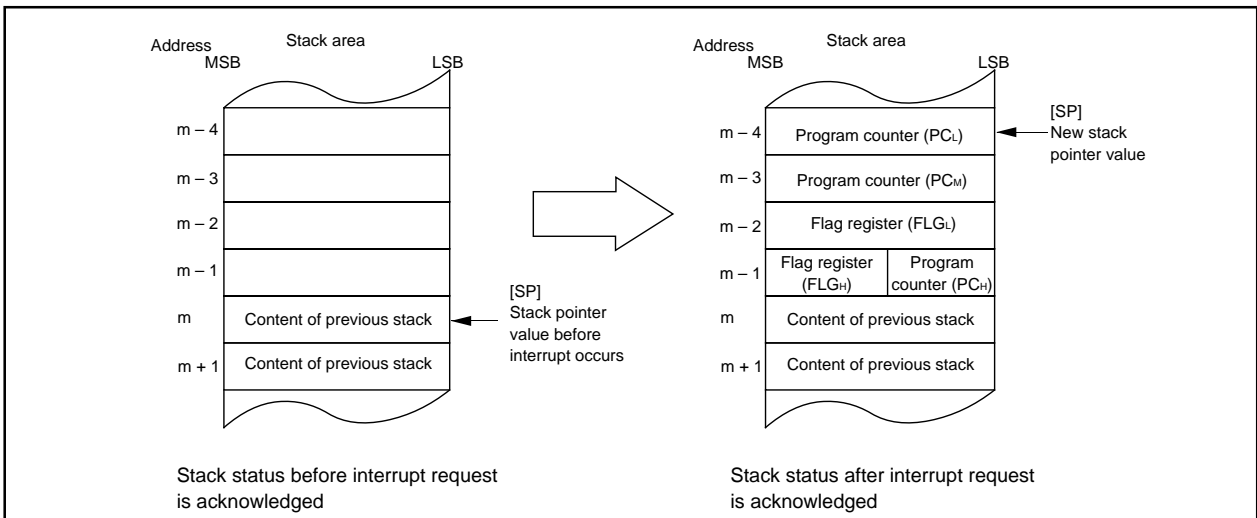


Fig.DD-7 State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer, at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Fig.DD-8 shows the operation of the saving registers.

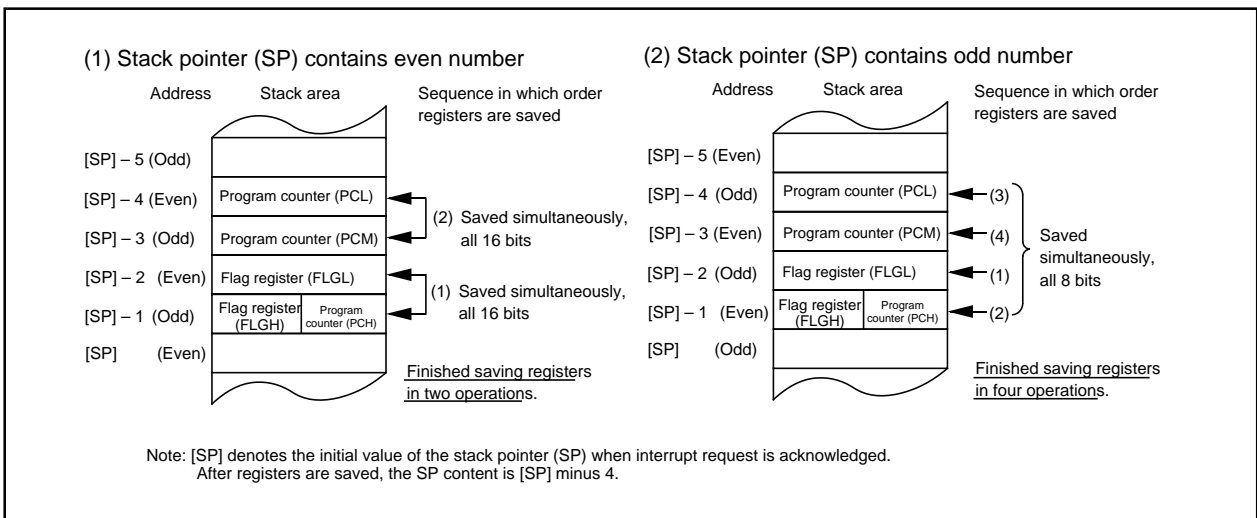


Fig.DD-8 Operation of saving registers

Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

Return the other registers saved by software within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bits. If the same interrupt priority level is assigned, however, the interrupt with higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Fig.DD-9 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine

Interrupt priority level judgement circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Fig.DD-10 shows the circuit that judges the interrupt priority level.

Reset > NMI > DBC > Watchdog timer > Peripheral I/O > Single step > Address match

Fig.DD-9 Hardware interrupts priorities

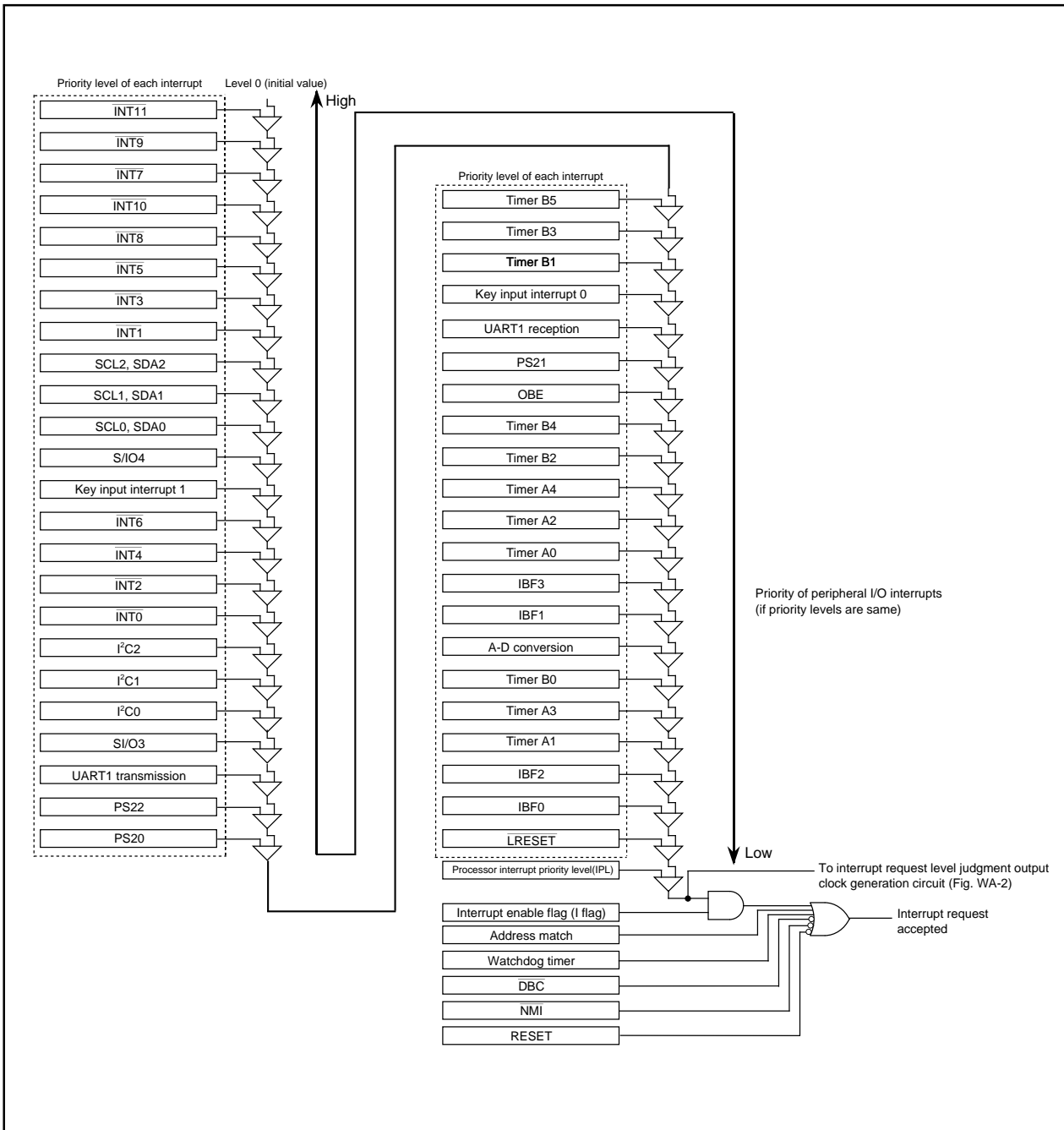


Fig.DD-10 Interrupt priority judgement circuit

INT Interrupt

$\overline{\text{INT}}_0$ to $\overline{\text{INT}}_{11}$ are triggered by the edges of external inputs. The edge polarity can be selected using the polarity select bit.

As for external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting "1" in the $\overline{\text{INT}}_i$ interrupt polarity switching bit of the interrupt factor selection register0,1 (035F₁₆, 035E₁₆). To select both edges, set the polarity switching bit of the corresponding interrupt control register to 'falling edge' ("0"). After the selection of interrupt edge, the corresponding interrupt request bit should be cleared to "0" before enabling the interrupt.

Fig.DD-11, Fig.DD-12 show the Interrupt factor selection register 0, 1.

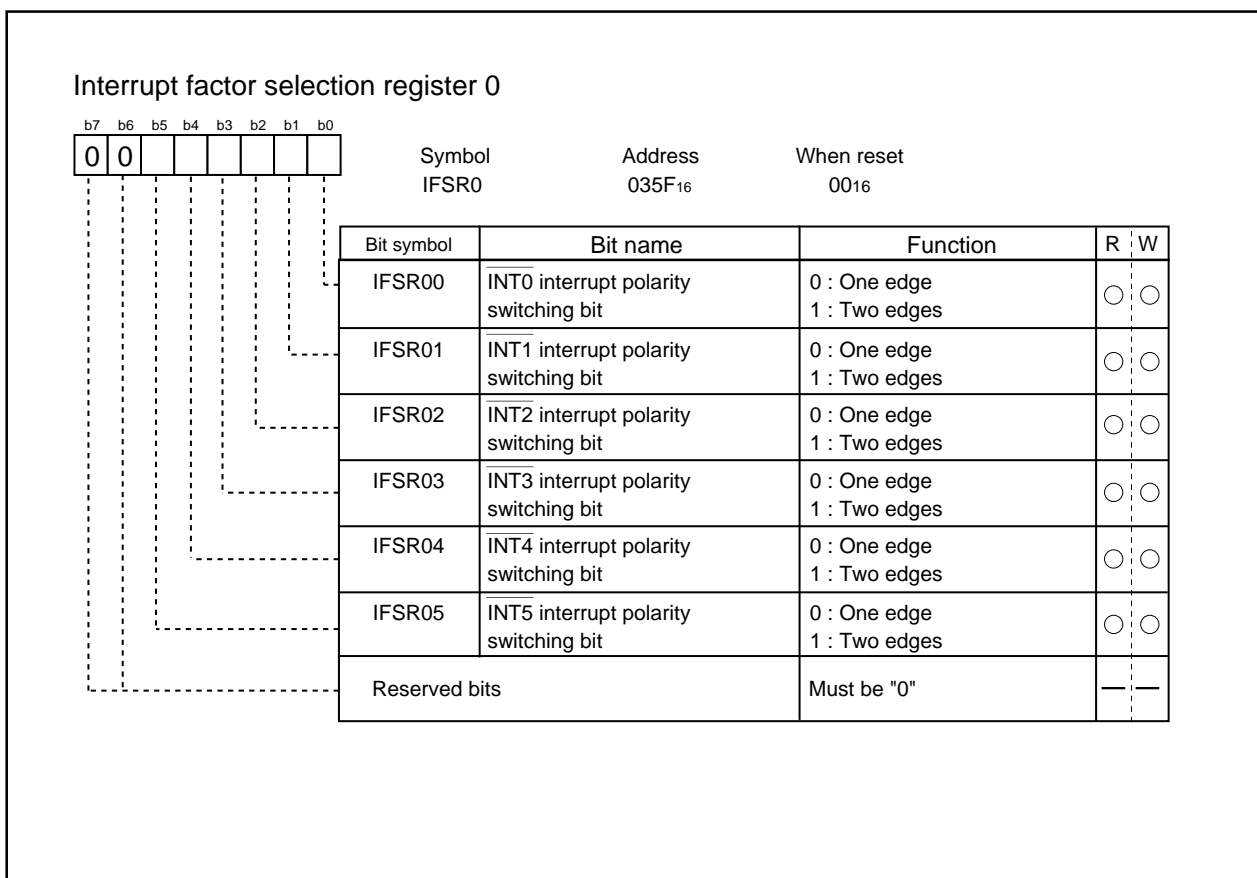


Fig.DD-11 Interrupt factor selection register(1)

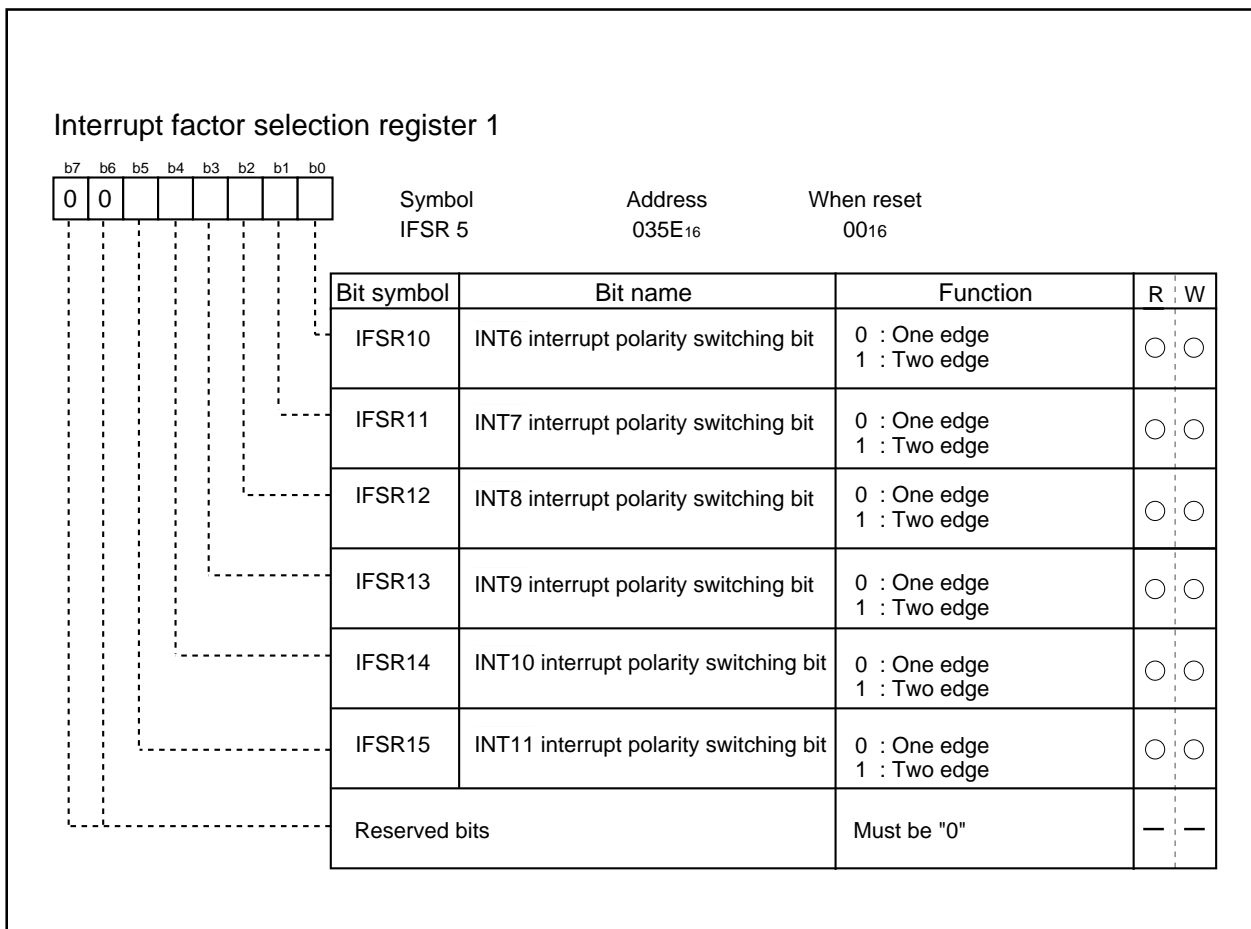


Fig. DD-12 Interrupt factor selection register(4)

NMI Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when the input to the P85/ $\overline{\text{NMI}}$ pin changes from “H” to “L”. The $\overline{\text{NMI}}$ interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address 03F016).

This pin cannot be used as a normal port input.

Key Input Interrupt 0

If the direction register of any of P50 to P57 is set for input and a falling edge is input to that port, a key input interrupt 0 is generated. A key input interrupt 0 can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. Fig.DD-13 shows the block diagram of the key input interrupt 0. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

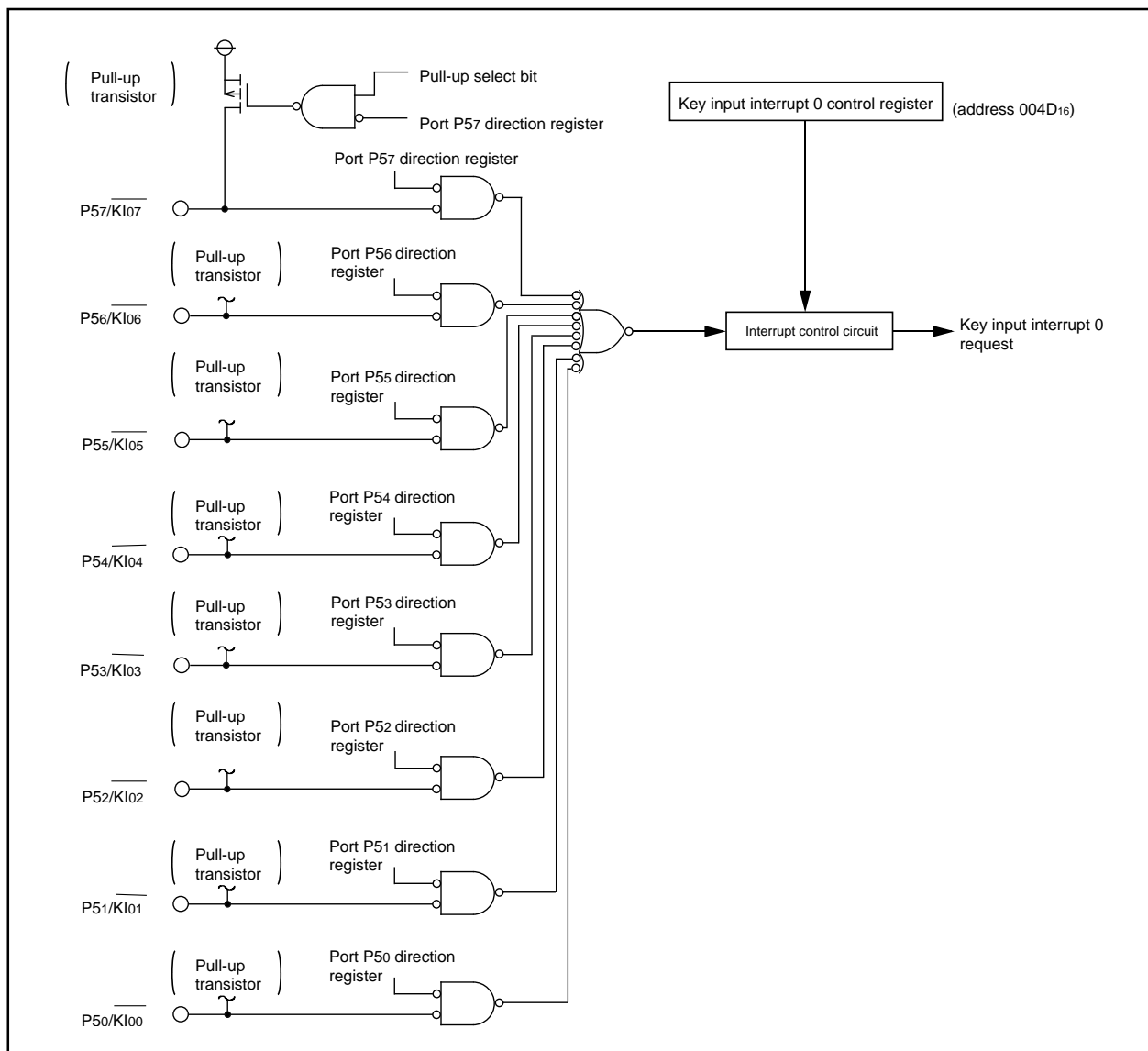


Fig.DD-13 Block diagram of key input interrupt 0

Key Input Interrupt 1

If any of the bits of key input interrupt 1 enable register (Address: 02F416) are set to “1”, the key input interrupt 1 request occurs when a falling or a rising edge is input to one of the corresponding pins.

The effective input edge of key input interrupt 1 is determined by the edge selection bit of key input interrupt 1 edge selection register (Address: 02F516). When the bit is set to “0”, at the falling edge, when the bit is set to “1”, at the rising edge of the input signal to the corresponding pin, the interrupt request occurs respectively.

When an effective rising edge or falling edge is input, “1” is set to the corresponding bit of P14 event register (Address: 02F616). By reading the register after the interrupt occurs, the pin, which the effective edge is input, can be confirmed even if the status of that pin has been changed.

At the completion of the reading of P14 event register, the bits, whose value is “1” in reading, will be cleared automatically. A dummy write clears the register too.

The registers, the block diagram and the timing of key input interrupt 1 are shown in Fig. DD-15, Fig. DD-16 and Fig. DD-17 respectively.

After changing the enable/disable setting of key input interrupt1 register or changing the effective edge by modifying key input interrupt 1 edge selection register, the value of P14 event register and interrupt request bit may become “1”. A dummy write to the P14 event register and a clear to the interrupt request bit should be done after changing the effective edge.

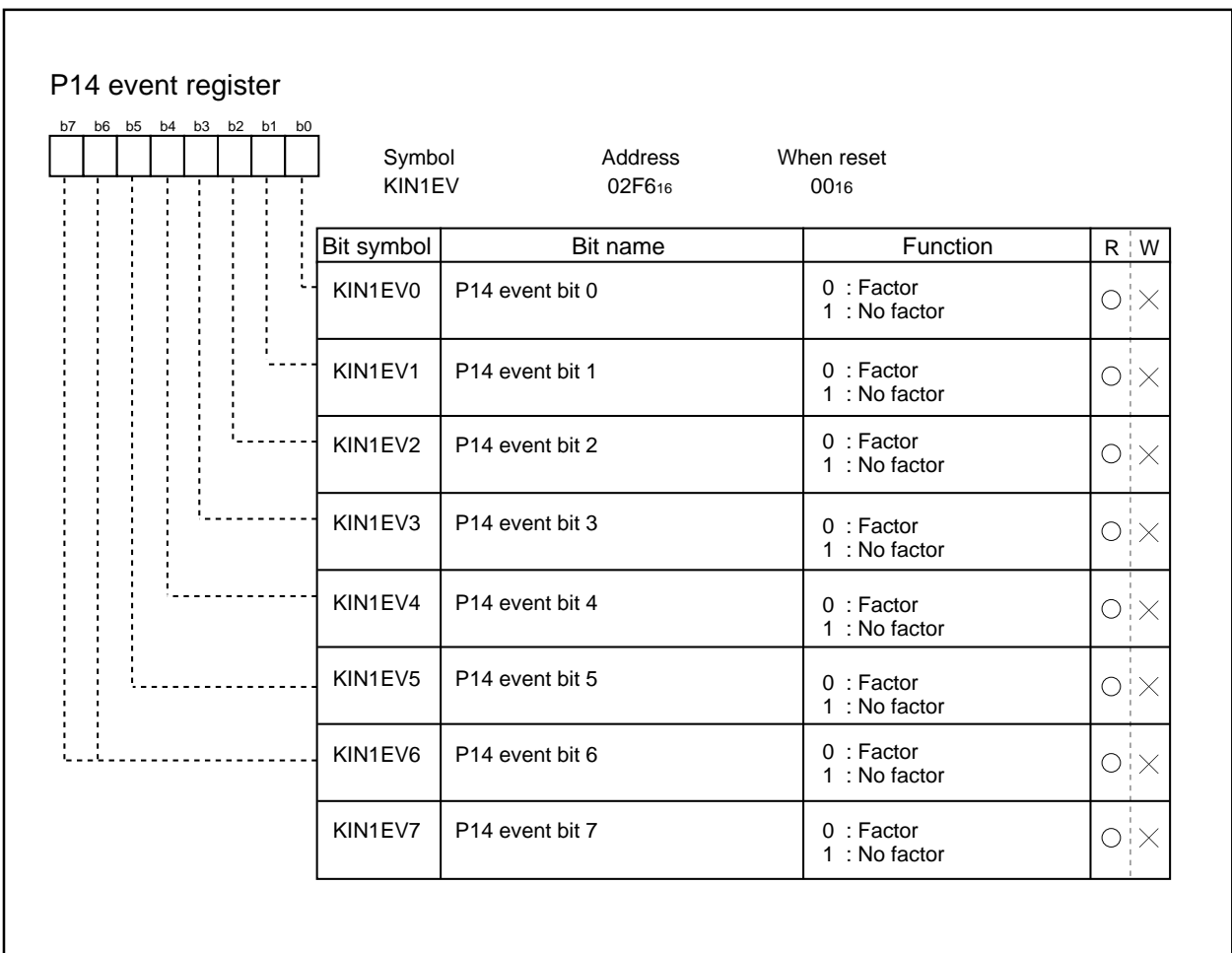


Fig.DD-14 P14 event register

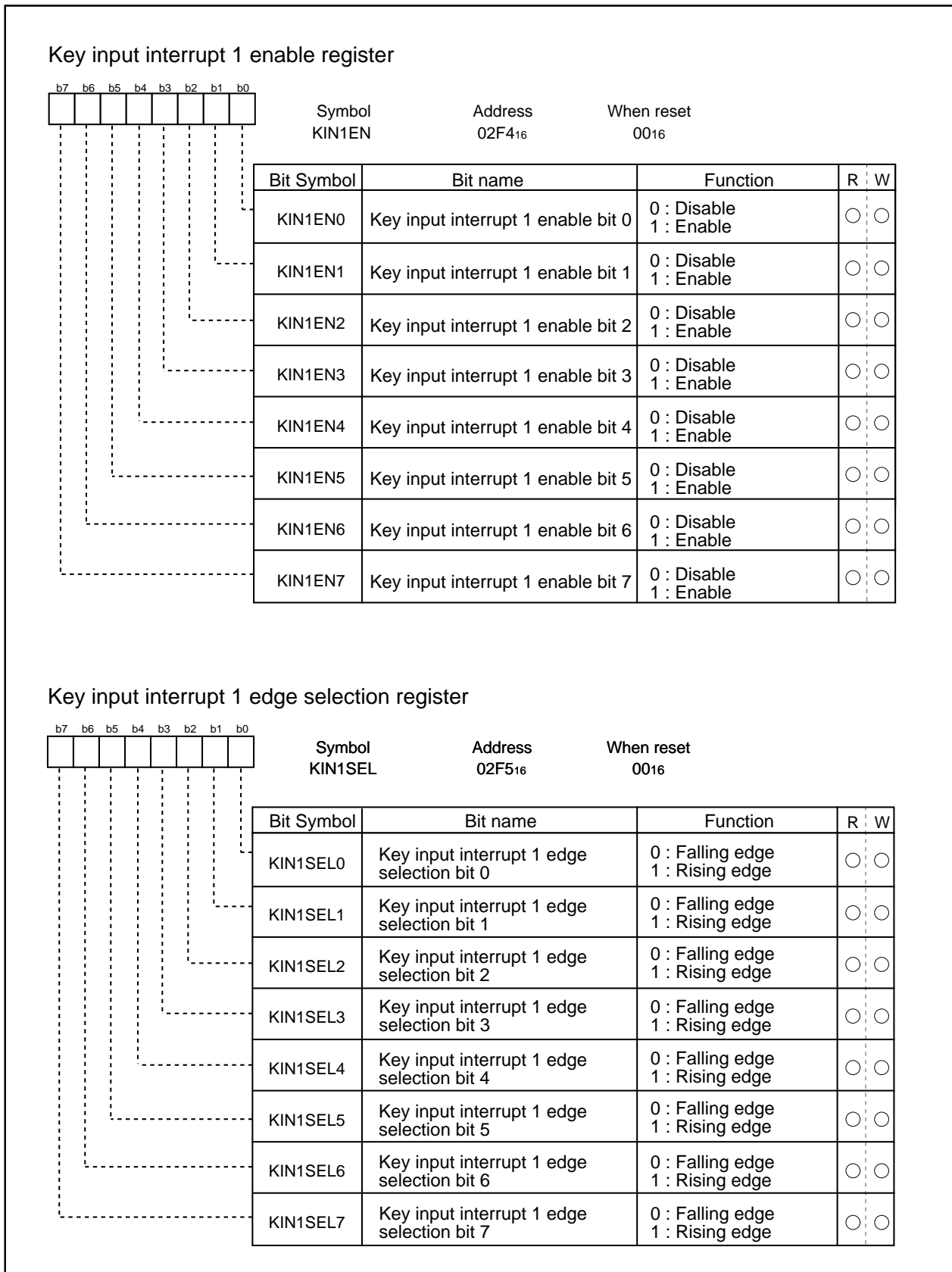


Fig.DD-15 Key input interrupt 1 registers

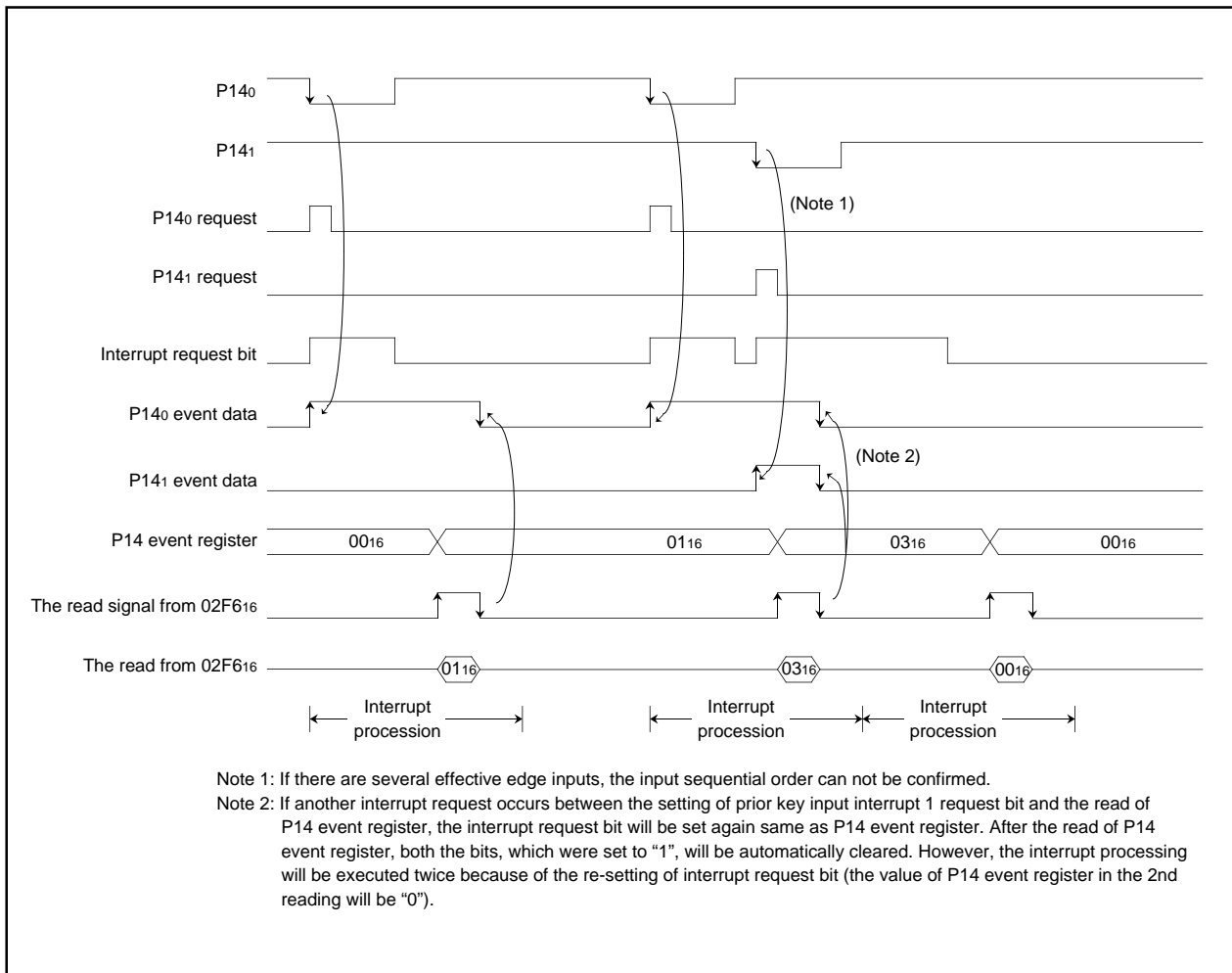


Fig.DD-17 The timing of key input interrupt 1

Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The stacked value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed.

Fig.DD-18 shows the address match interrupt-related registers.

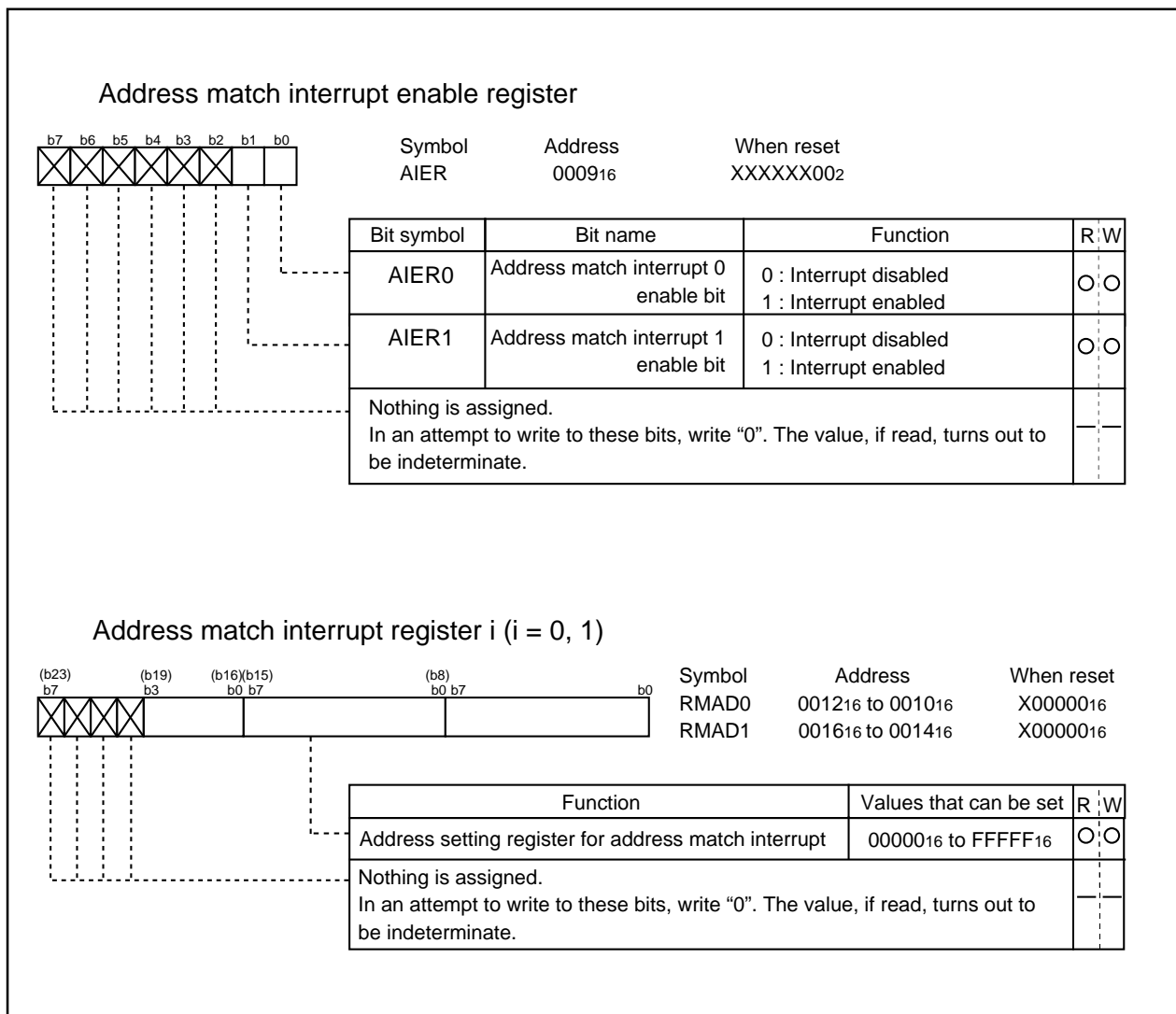


Fig.DD-18 Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 00000₁₆

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

The interrupt request bit of the certain interrupt written in address 00000₁₆ will then be set to "0".

Reading address 00000₁₆ by software sets the request bit, which the interrupt source is enabled with the highest priority, to "0".

Though the interrupt is generated, the interrupt routine may not be executed.

Hence do not read address 00000₁₆ by software.

(2) Setting the stack pointer

- The value of the stack pointer is initialized to 0000₁₆ right after the reset. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. When using the $\overline{\text{NMI}}$ interrupt, initialize the stack point at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.

(3) The $\overline{\text{NMI}}$ interrupt

- As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistor (pull-up) if unused. Be sure to work on it.
- The $\overline{\text{NMI}}$ pin also serves as P85, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the $\overline{\text{NMI}}$ interrupt is input.
- Do not reset the CPU with the input to the $\overline{\text{NMI}}$ pin being in the "L" state.
- Do not attempt to go into stop mode with the input to the $\overline{\text{NMI}}$ pin being in the "L" state. With the input to the $\overline{\text{NMI}}$ being in the "L" state, the CM10 is fixed to "0", so attempting to go into stop mode is turned down.
- Do not attempt to go into wait mode with the input to the $\overline{\text{NMI}}$ pin being in the "L" state. With the input to the $\overline{\text{NMI}}$ pin being in the "L" state, the CPU stops but the oscillation does not stop, so no power is saved. In this instance, the CPU is returned to the normal state by a later interrupt.
- Signals input to the $\overline{\text{NMI}}$ pin require an "L" level of 1 clock or more, from the operation clock of the CPU.

(4) External interrupt

- Either an "L" level or an "H" level of at least 380 ns width is necessary for the signal input to pins $\overline{\text{INT}}_0$ through $\overline{\text{INT}}_{11}$ regardless of the CPU operation clock.
- When the polarity of the $\overline{\text{INT}}_0$ to $\overline{\text{INT}}_{11}$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Fig.DD-19 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

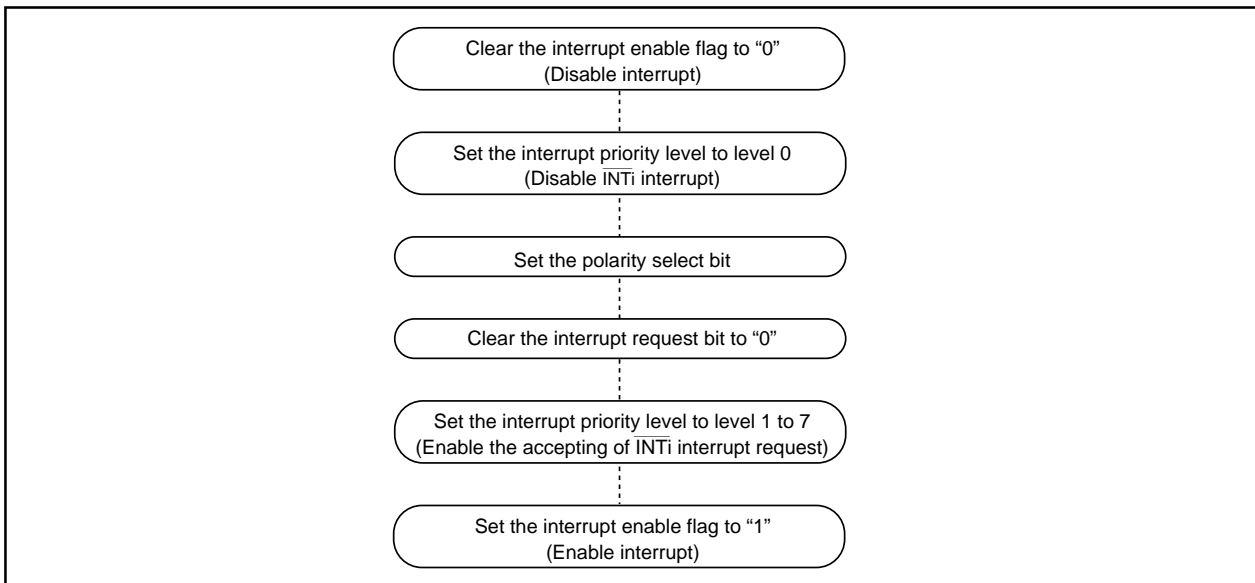


Fig.DD-19 Switching condition of $\overline{\text{INT}}$ interrupt request

(5) Rewrite the interrupt control register

• To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occurs, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

• When a instruction to rewrite the interrupt control register is executed when the interrupt is disabled, the

Example 1:

```

INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP                    ; Four NOP instructions are required when using HOLD function.
  NOP
  FSET  I           ; Enable interrupts.
  
```

Example 2:

```

INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0     ; Dummy read.
  FSET  I           ; Enable interrupts.
  
```

Example 3:

```

INT_SWITCH3:
  PUSHC FLG         ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG         ; Enable interrupts.
  
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. Bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the prescaler.

With XIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (16 or 128)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

For example, suppose that BCLK runs at 16 MHz and that 16 has been chosen for the dividing ratio of the prescaler, then the watchdog timer's period becomes approximately 32.7 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E16).

Fig.DG-1 shows the block diagram of the watchdog timer. Fig.DG-2 shows the watchdog timer-related registers.

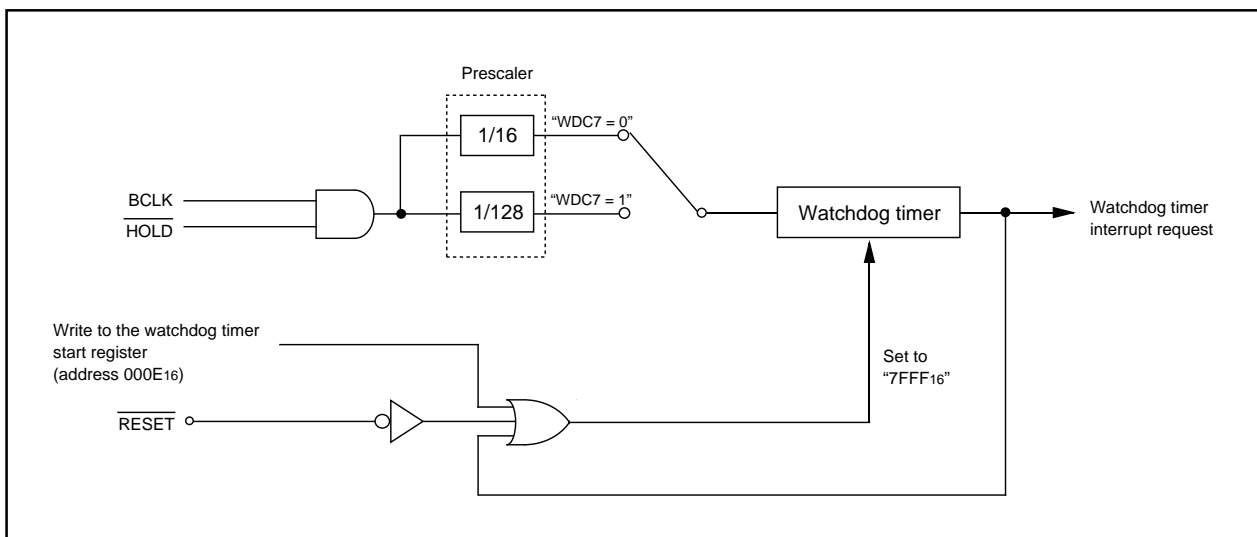


Fig.DG-1 Block diagram of watchdog timer

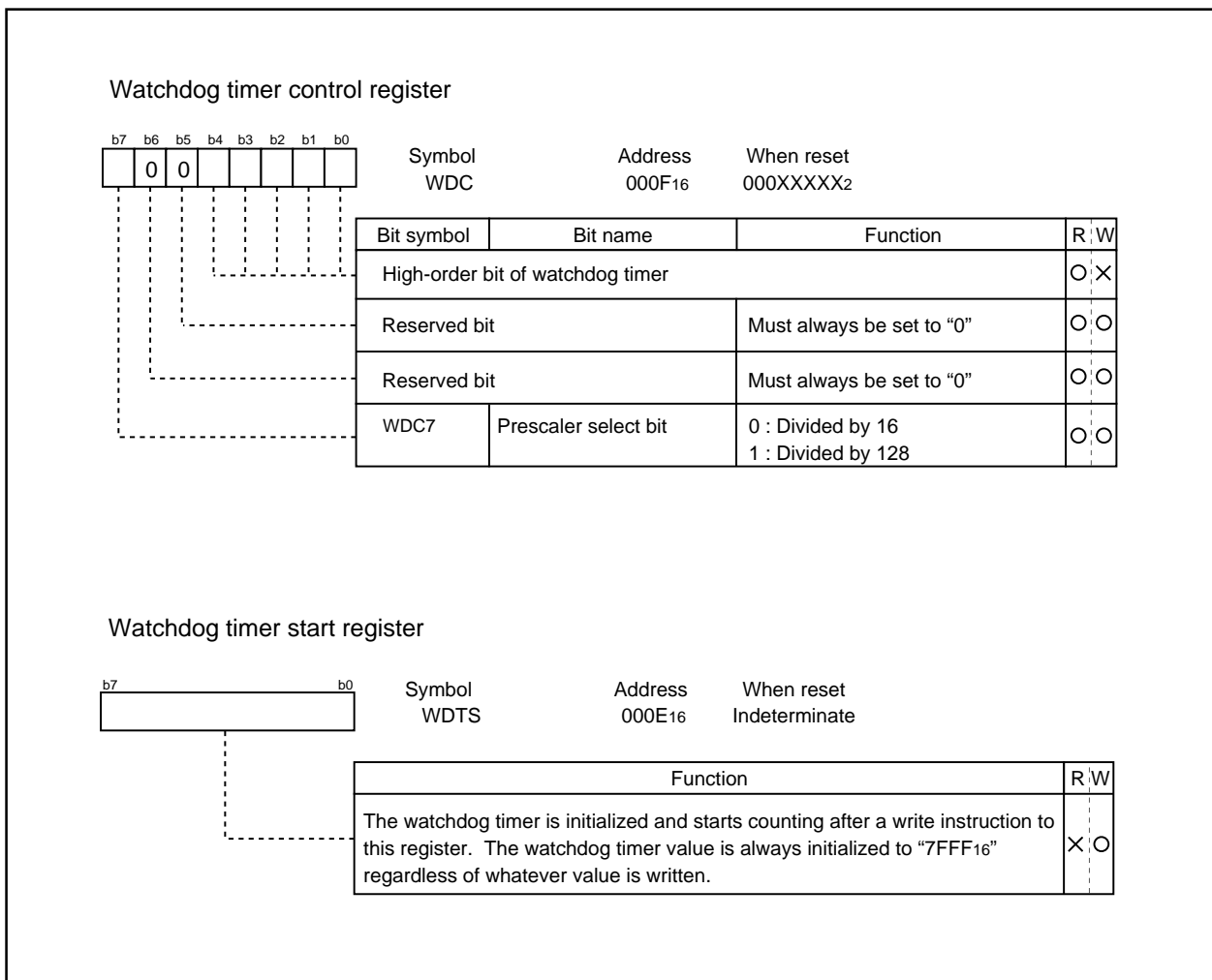


Fig.DG-2 Watchdog timer control and start registers

Timer

There are eleven 16-bit timers. These timers can be classified by function into timers A (five) and timers B (six). All these timers function independently. Fig.FB-1 and FB-2 show the block diagram of timers.

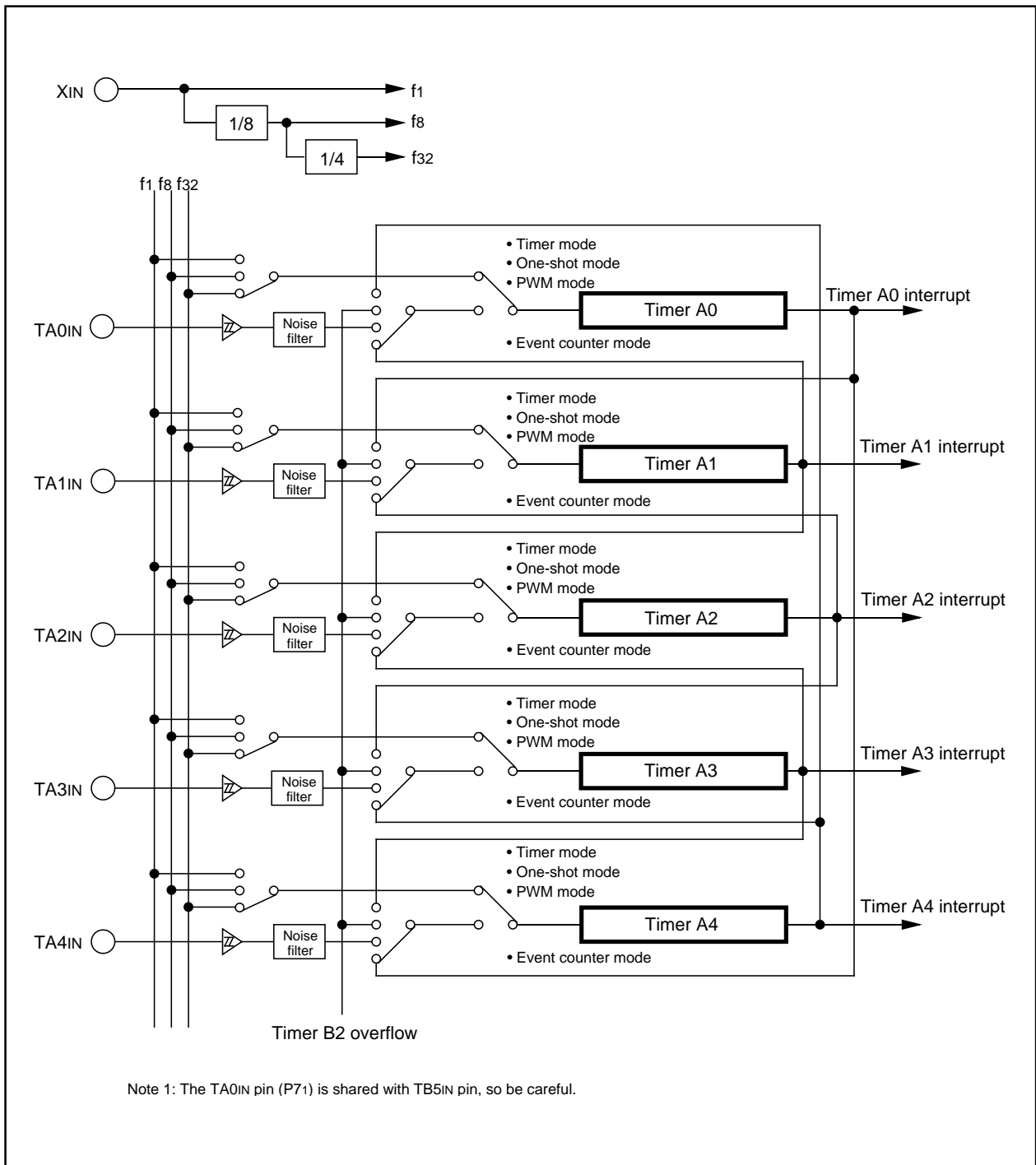


Fig.FB-1 Timer A block diagram

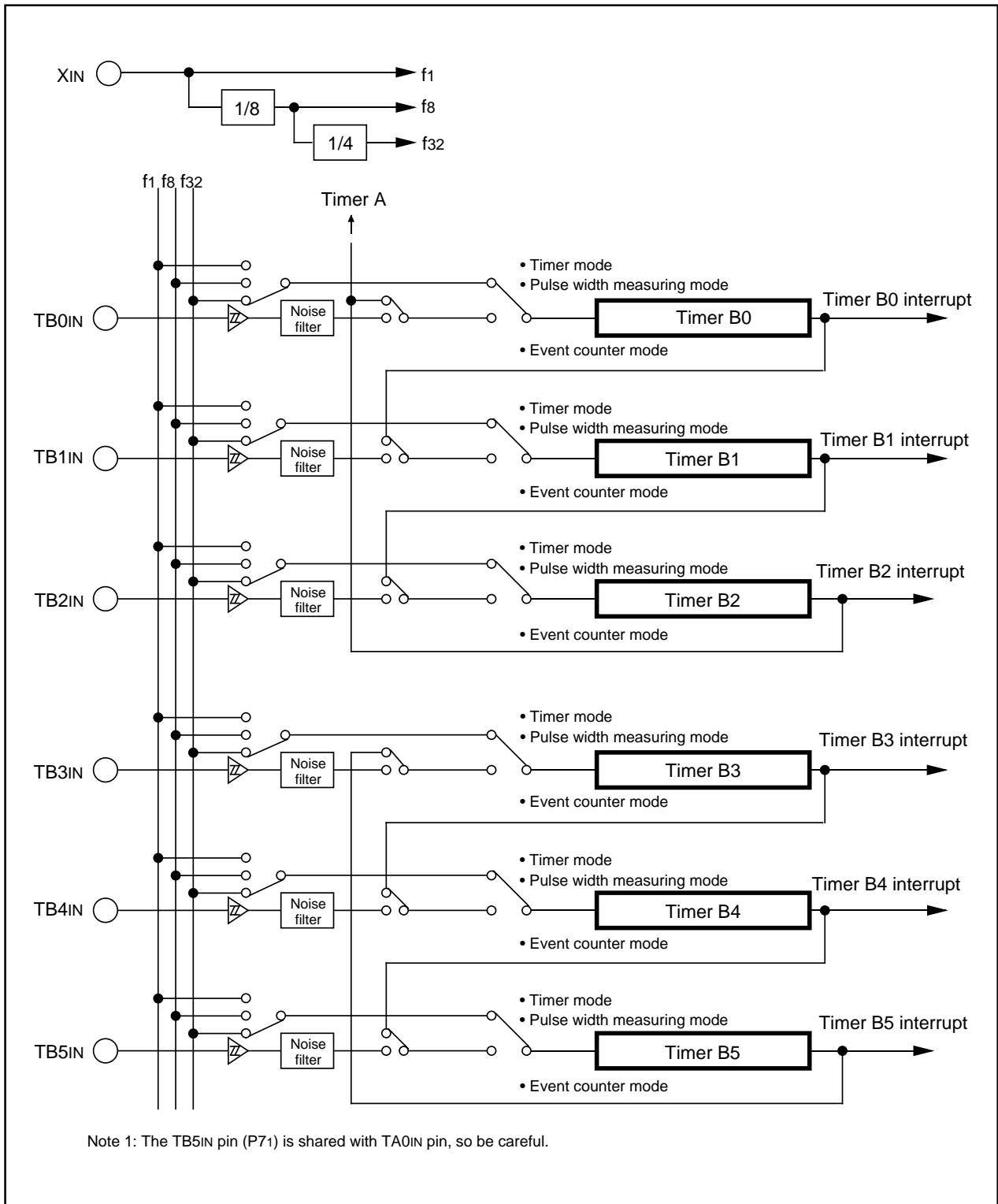


Fig.FB-2 Timer B block diagram

Timer A

Fig.FB-3 shows the block diagram of timer A. Fig.FB-4 to FB-6 show the timer A-related registers.

Except in event counter mode, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches "0000₁₆".
- Pulse width modulation (PWM) mode: The timer continually outputs pulse with arbitrary width.

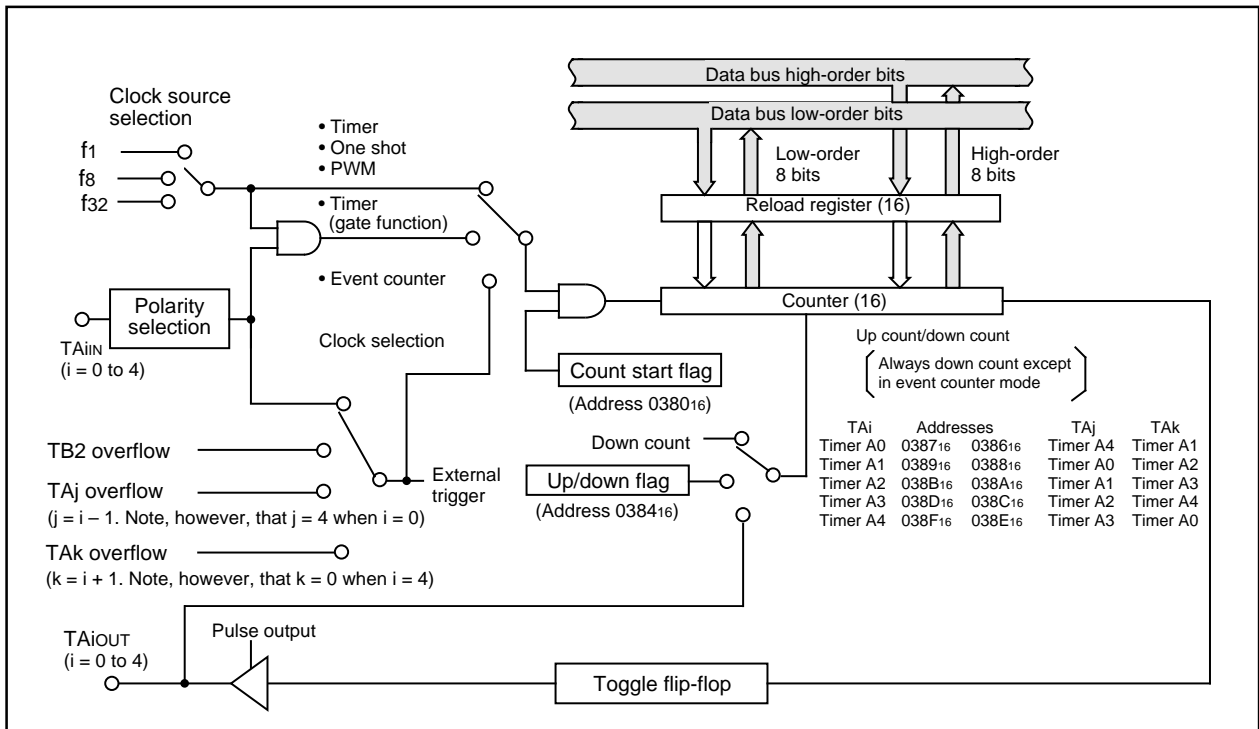


Fig.FB-3 Block diagram of timer A

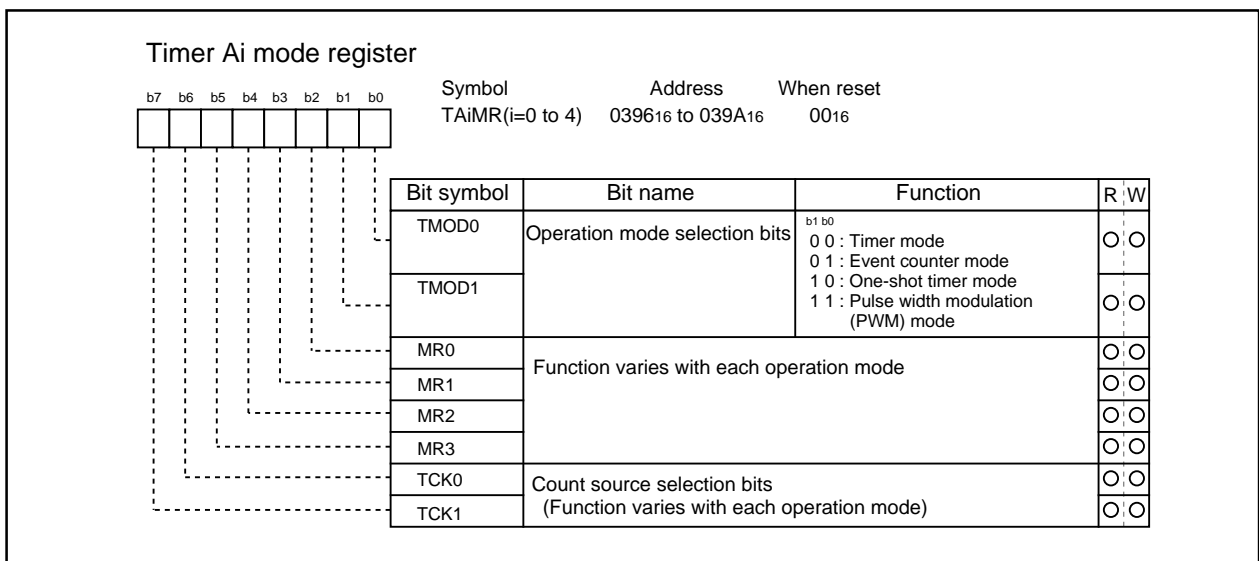


Fig.FB-4 Timer A-related registers (1)

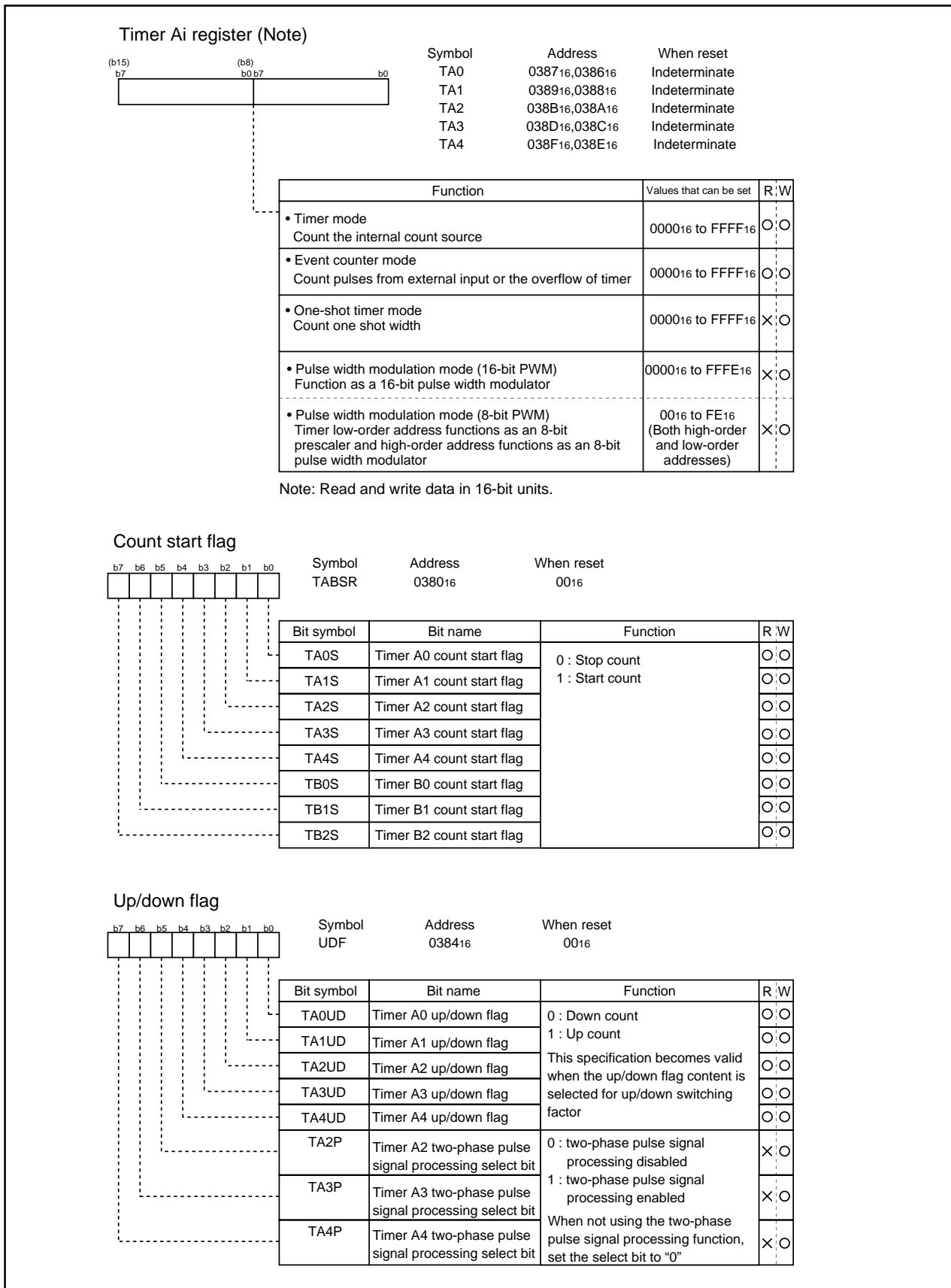


Fig.FB-5 Timer A-related registers (2)

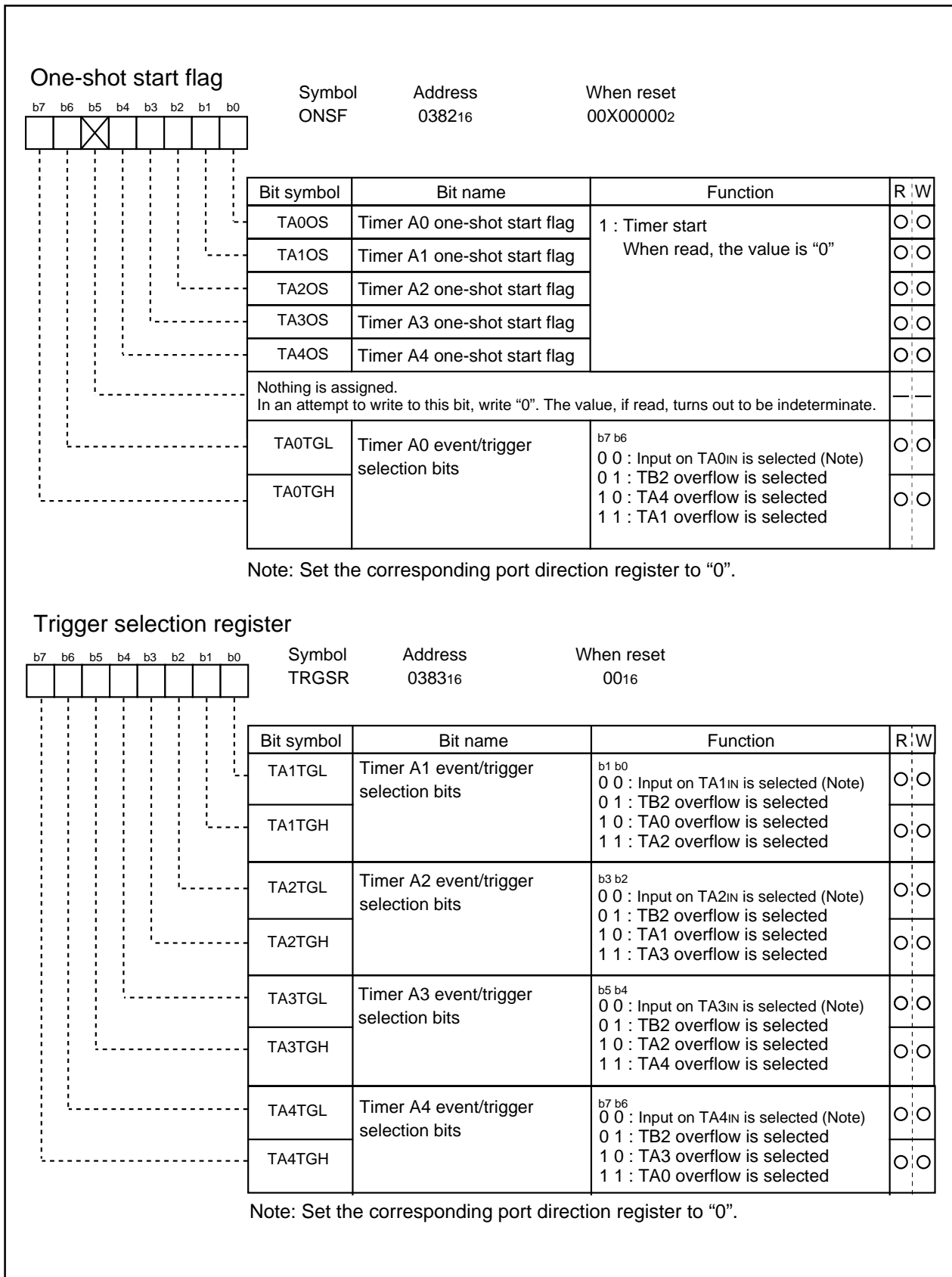


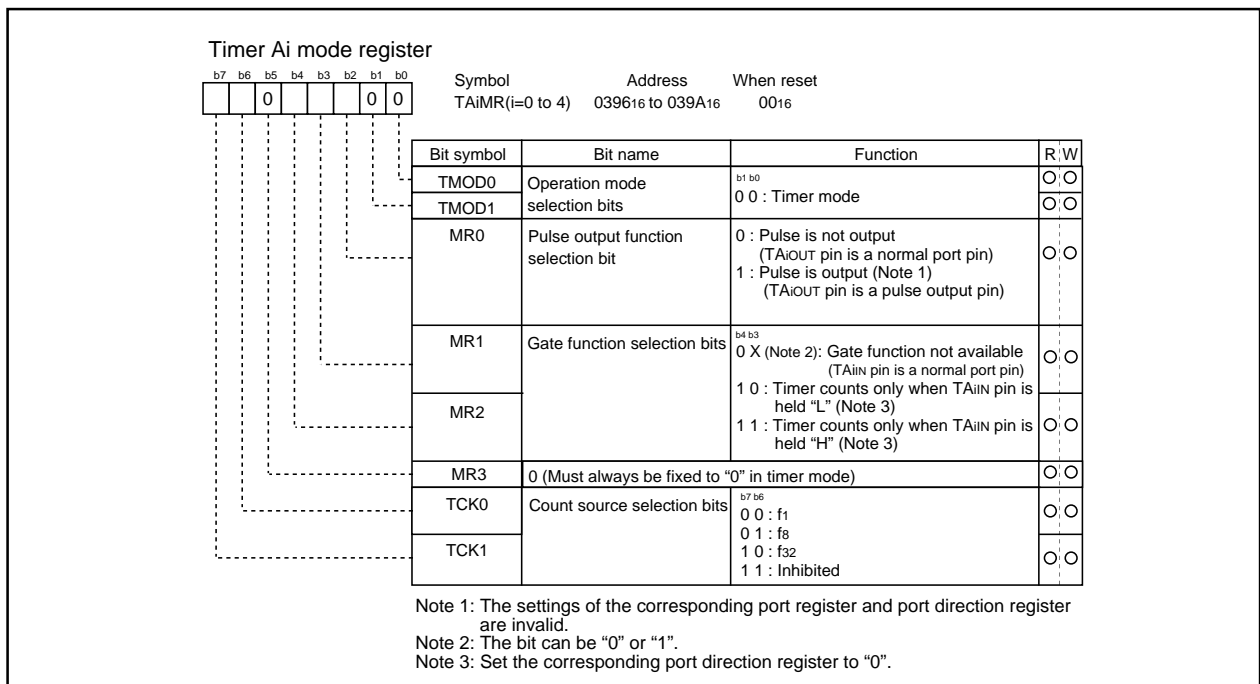
Fig.FB-6 Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table.FB-1) Fig.FB-7 shows the timer Ai mode register in timer mode.

Table.FB-1 Specifications of timer mode

Item	Specification
Count source	f1, f8, f32
Count operation	<ul style="list-style-type: none"> Down count When the timer underflows, it reloads the reload register contents and then continuing counting
Divide ratio	$1/(n+1)$ n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	<ul style="list-style-type: none"> Gate function Counting can be started and stopped by the TAiIN pin's input signal Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed

**Fig.FB-7 Timer Ai mode register in timer mode**

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signals. Table.FB-2 lists timer specifications and Fig. FB-8 shows the timer Ai mode register in event count mode when counting a single-phase external signal.

Table.FB-3 lists timer specifications and Fig. FB-8 shows the timer Ai mode register in event count mode when counting a two-phase external signals.

Table.FB-2 Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	<ul style="list-style-type: none"> External signal input to TAIIN pin (effective edge can be selected by software) TB2 overflow, TAJ overflow
Count operation	<ul style="list-style-type: none"> Up count or down count can be selected by external signal or software When the timer overflows or underflows, it reloads the reload register contents and then continuing counting (Note)
Divide ratio	$1 / (FFFF_{16} - n + 1)$ for up count $1 / (n + 1)$ for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	<ul style="list-style-type: none"> Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it Pulse output function Each time the timer overflows or underflows, the TAIOUT pin's polarity is reversed

Note: This does not apply when the free-run function is selected.

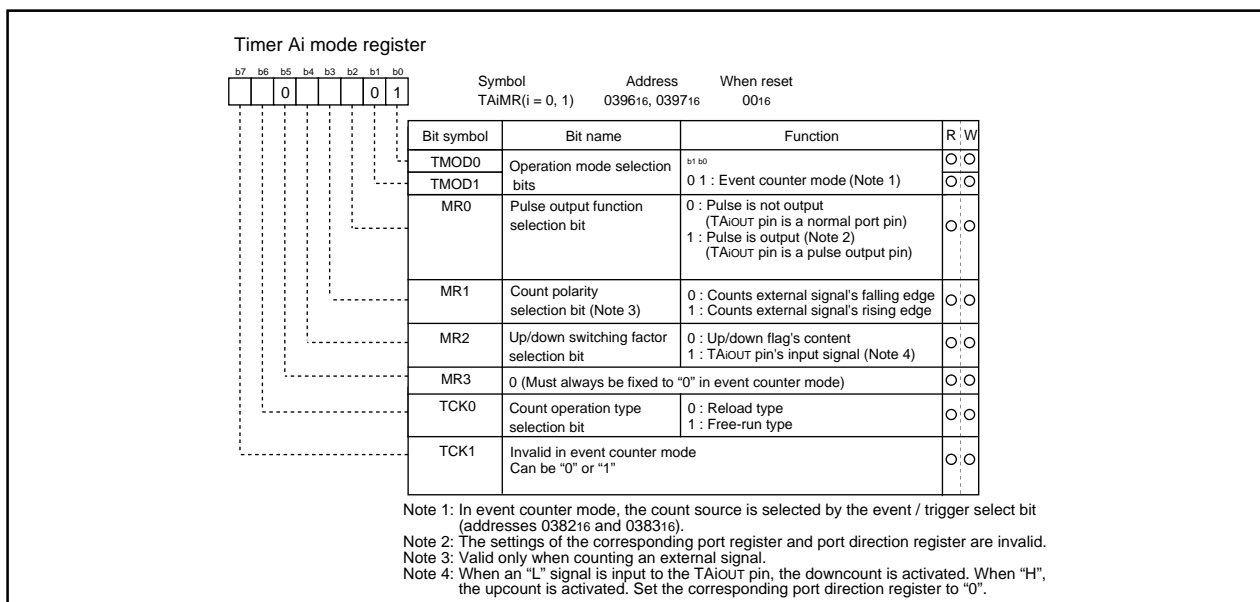
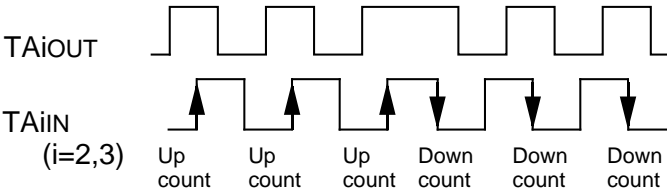
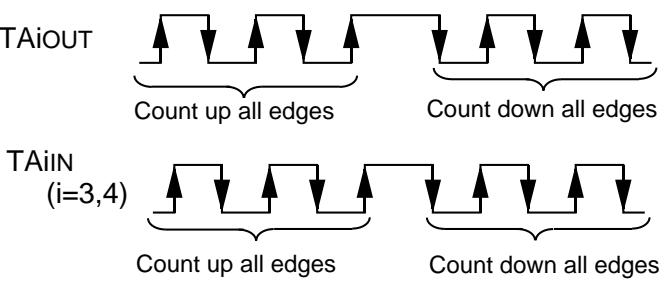
**Fig.FB-8** Timer Ai mode register in event counter mode

Table.FB-3 Timer specifications in event counter mode (when processing two-phase pulse signals with timers A2, A3, and A4)

Item	Specification
Count source	<ul style="list-style-type: none"> Two-phase pulse signals input to TAIiN and TAIOUT pin
Count operation	<ul style="list-style-type: none"> Up count or down count can be selected by two-phase pulse signals When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	$1 / (FFFF_{16} - n + 1)$ for up count $1 / (n + 1)$ for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	<ul style="list-style-type: none"> When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function	<ul style="list-style-type: none"> Normal processing operation The timer up-counts by the rising edge of TAIiN pin and down-counts by the falling edge fo TAIiN pin during the "H" level period of input signal in TAIOUT pin.  Multiply-by-4 processing operation If the phase relationship is such that the TAIiN pin goes "H" when the input signal on the TAIOUT pin is "H", the timer counts up rising and falling edges on the TAIOUT and TAIiN pins. If the phase relationship is such that the TAIiN pin goes "L" when the input signal on the TAIOUT pin is "H", the timer counts down rising and falling edges on the TAIOUT and TAIiN pins. 

Note: This does not apply when the free-run function is selected.

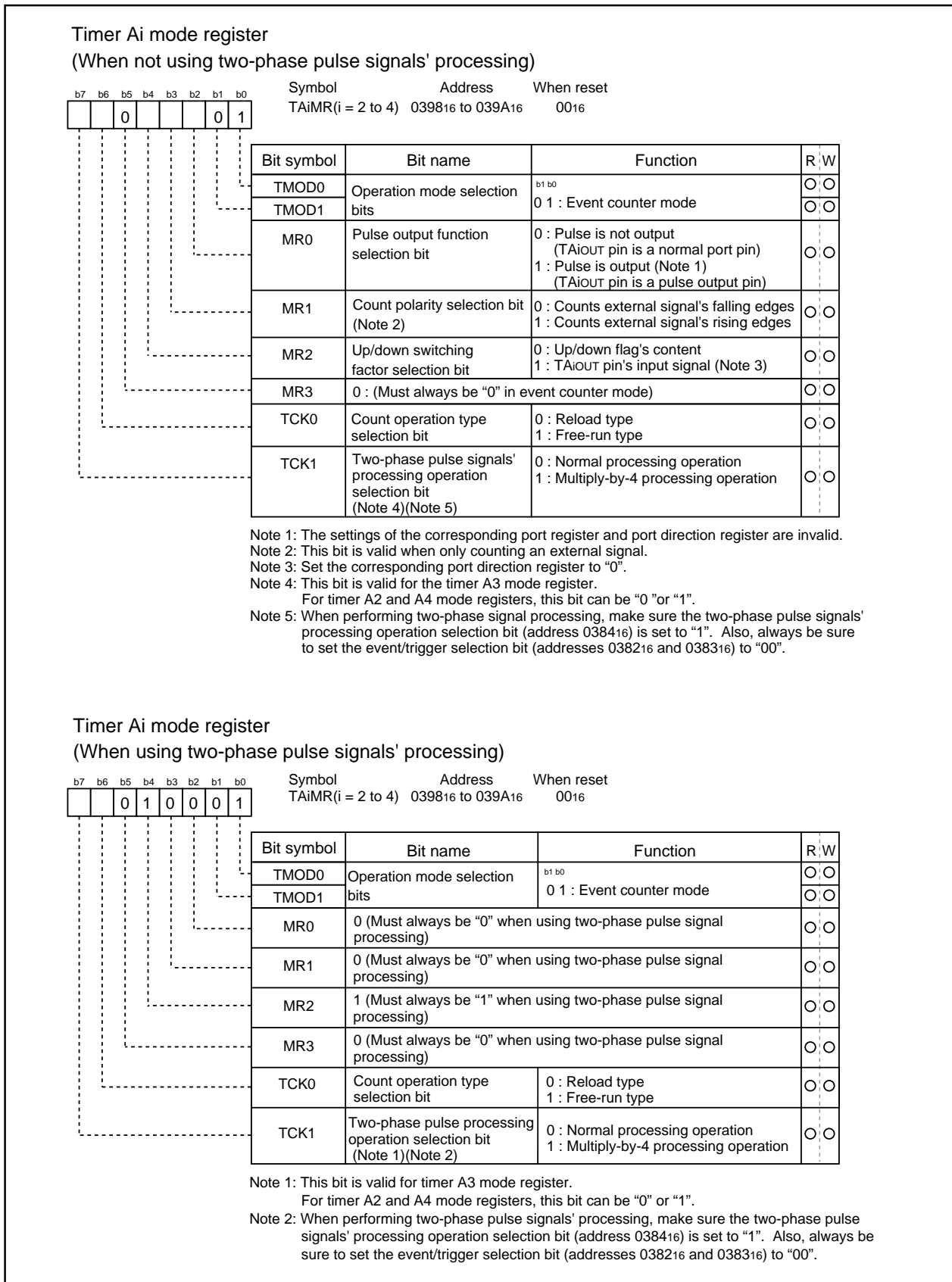


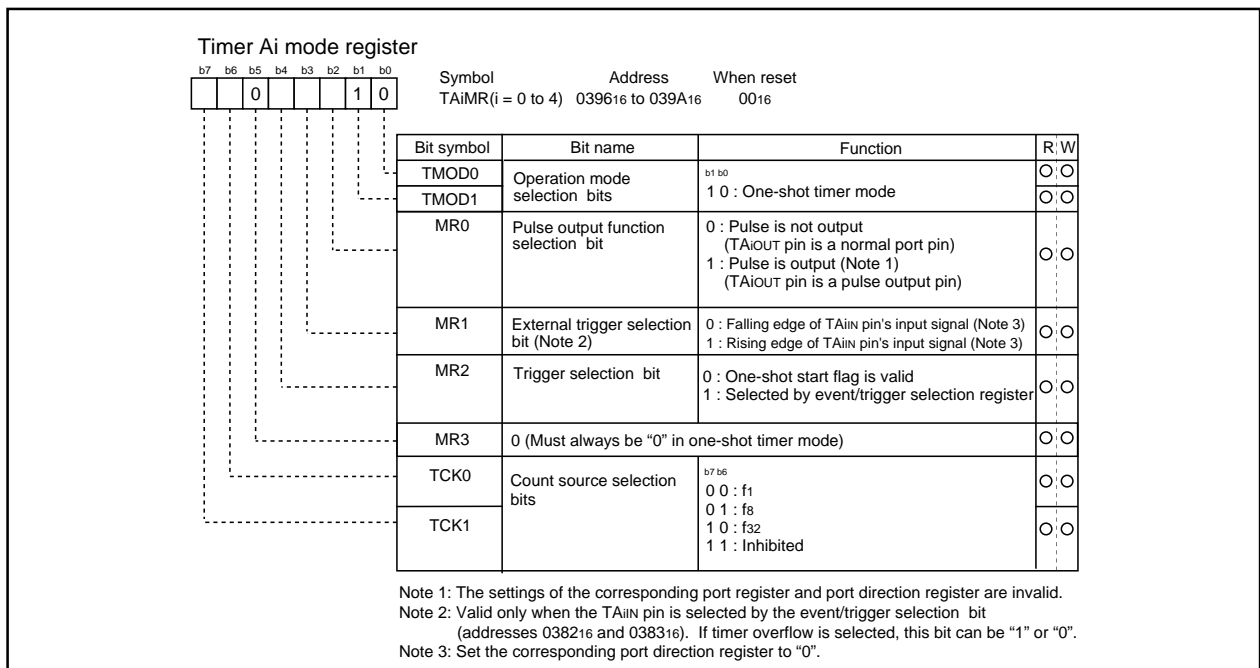
Fig.FB-9 Timer Ai mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table.FB-4) When a trigger occurs, the timer starts to operate for a given period. Fig.FB-10 shows the timer Ai mode register in one-shot timer mode.

Table.FB-4 Timer specifications in one-shot timer mode

Item	Specification
Count source	f1, f8, f32
Count operation	<ul style="list-style-type: none"> The timer counts down When the count reaches 0000₁₆, the timer stops counting after reloading a new count If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	<ul style="list-style-type: none"> An external trigger is input The timer overflows The one-shot start flag is set (= 1)
Count stop condition	<ul style="list-style-type: none"> A new count is reloaded after the count has reached 0000₁₆ The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000 ₁₆
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

**Fig.FB-10 Timer Ai mode register in one-shot timer mode**

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table.FB-5) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Fig.FB-11 shows the timer Ai mode register in pulse width modulation mode. Fig.FB-12 shows the example of how a 16-bit pulse width modulator operates. Fig.FB-13 shows the example of how an 8-bit pulse width modulator operates.

Table.FB-5 Timer specifications in pulse width modulation mode

Item	Specification
Count source	f1, f8, f32
Count operation	<ul style="list-style-type: none"> The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) The timer reloads a new count at a rising edge of PWM pulse and continues counting The timer is not affected by a trigger that occurs when counting
16-bit PWM	<ul style="list-style-type: none"> High level width n / f_i n : Set value Cycle time $(2^{16}-1) / f_i$ fixed
8-bit PWM	<ul style="list-style-type: none"> High level width $n \times (m+1) / f_i$ n : values set to timer Ai register's high-order address Cycle time $(2^8-1) \times (m+1) / f_i$ m : values set to timer Ai register's low-order address
Count start condition	<ul style="list-style-type: none"> External trigger is input The timer overflows The count start flag is set (= 1)
Count stop condition	<ul style="list-style-type: none"> The count start flag is reset (= 0)
Interrupt request generation timing	The falling edge of PWM pulse
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

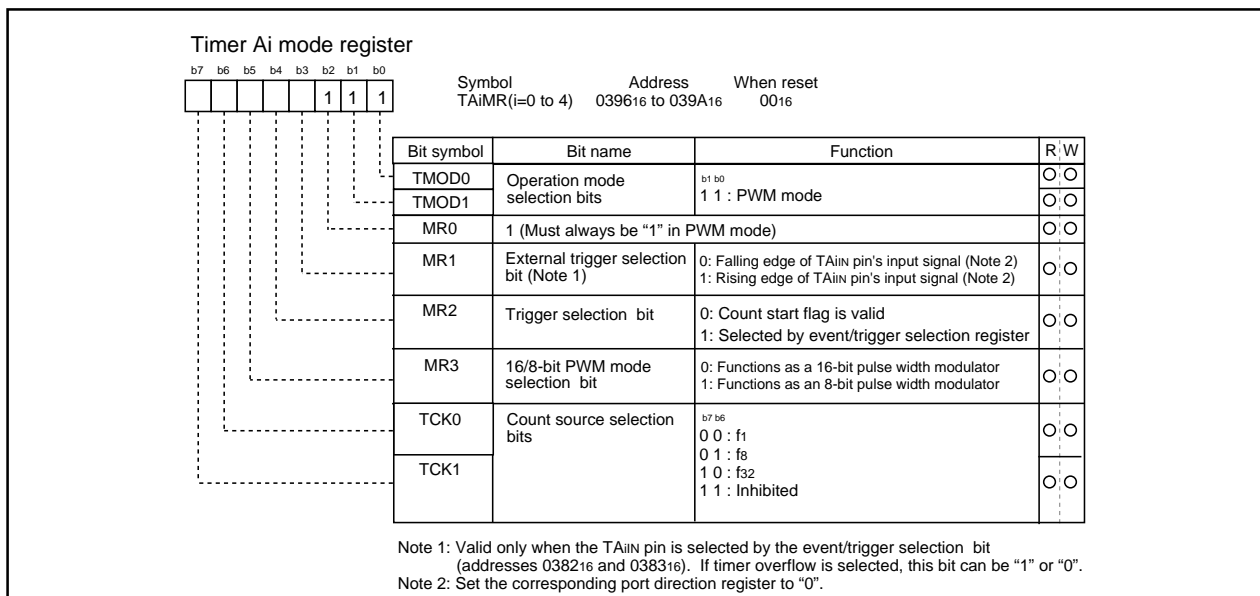


Fig.FB-11 Timer Ai mode register in pulse width modulation mode

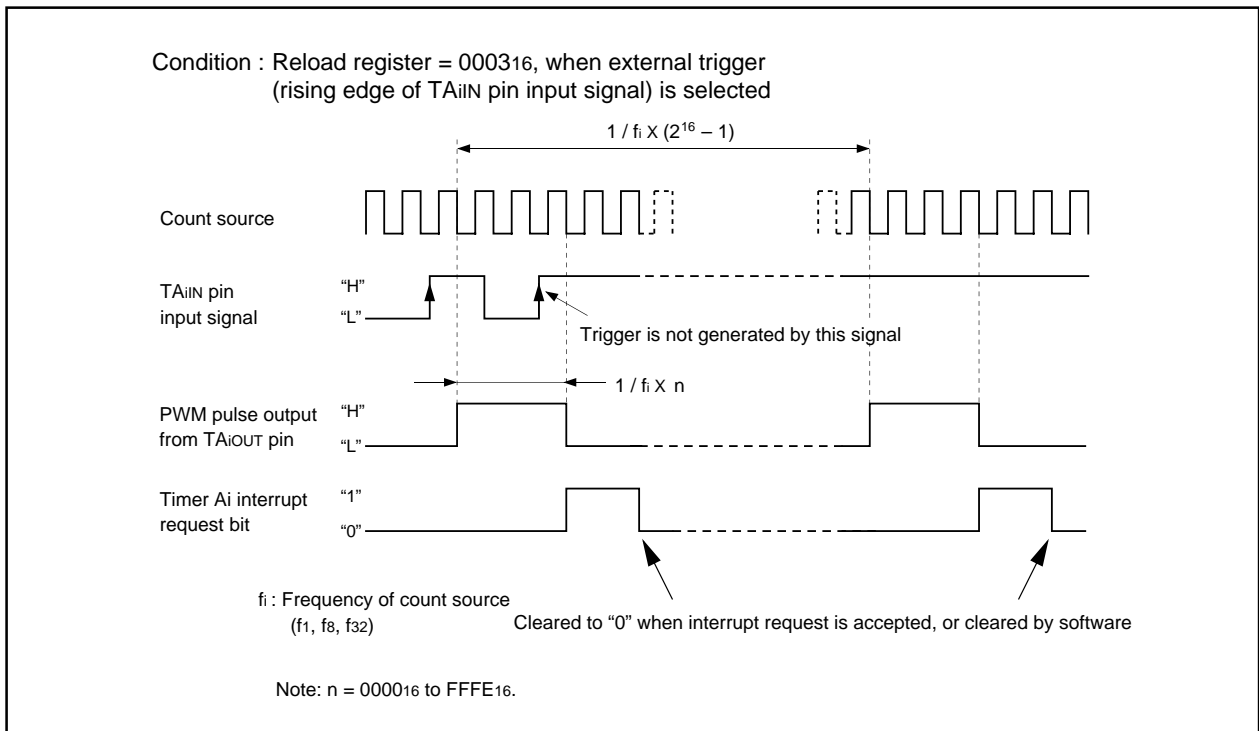


Fig.FB-12 Example of how a 16-bit pulse width modulator operates

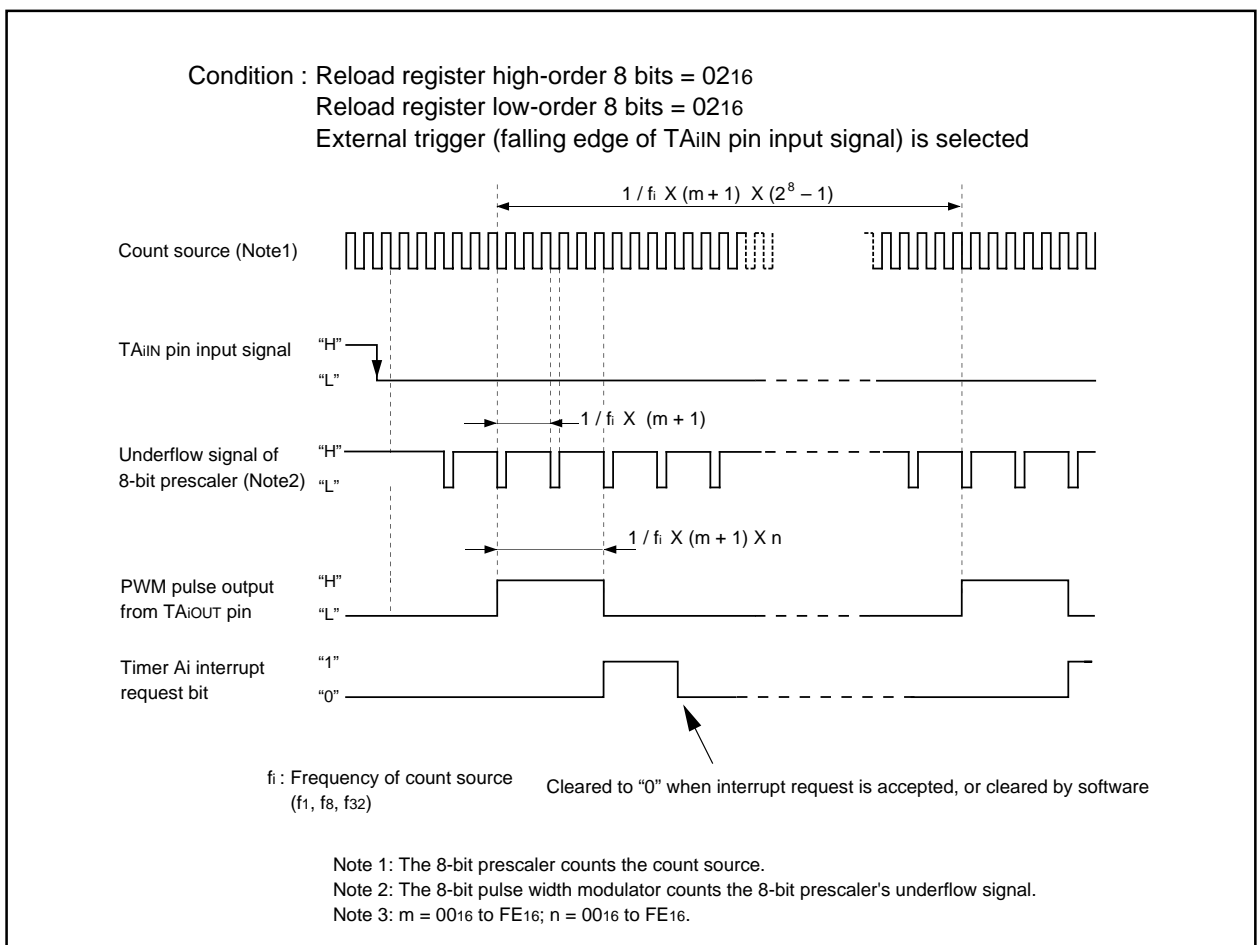


Fig.FB-13 Example of how an 8-bit pulse width modulator operates

Timer B

Fig.FB-14 shows the block diagram of timer B. Fig.FB-15 and FB-16 show the timer B-related registers. Use the timer Bi mode register (i = 0 to 5) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

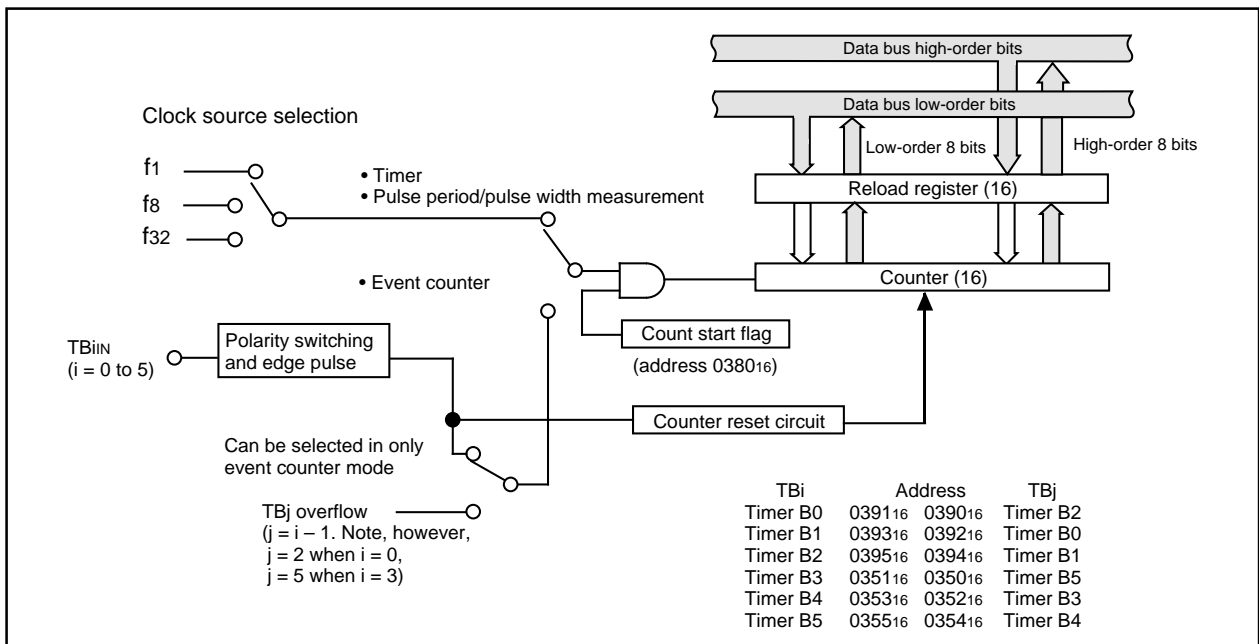


Fig.FB-14 Block diagram of timer B

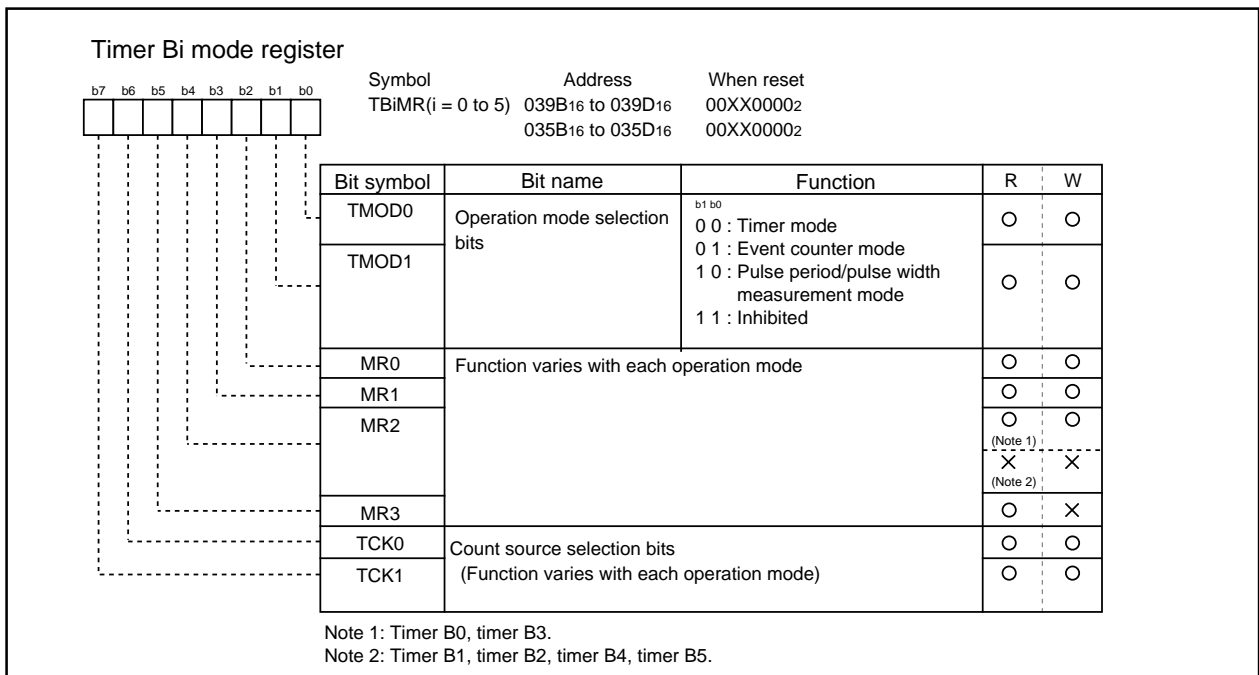


Fig.FB-15 Timer B-related registers (1)

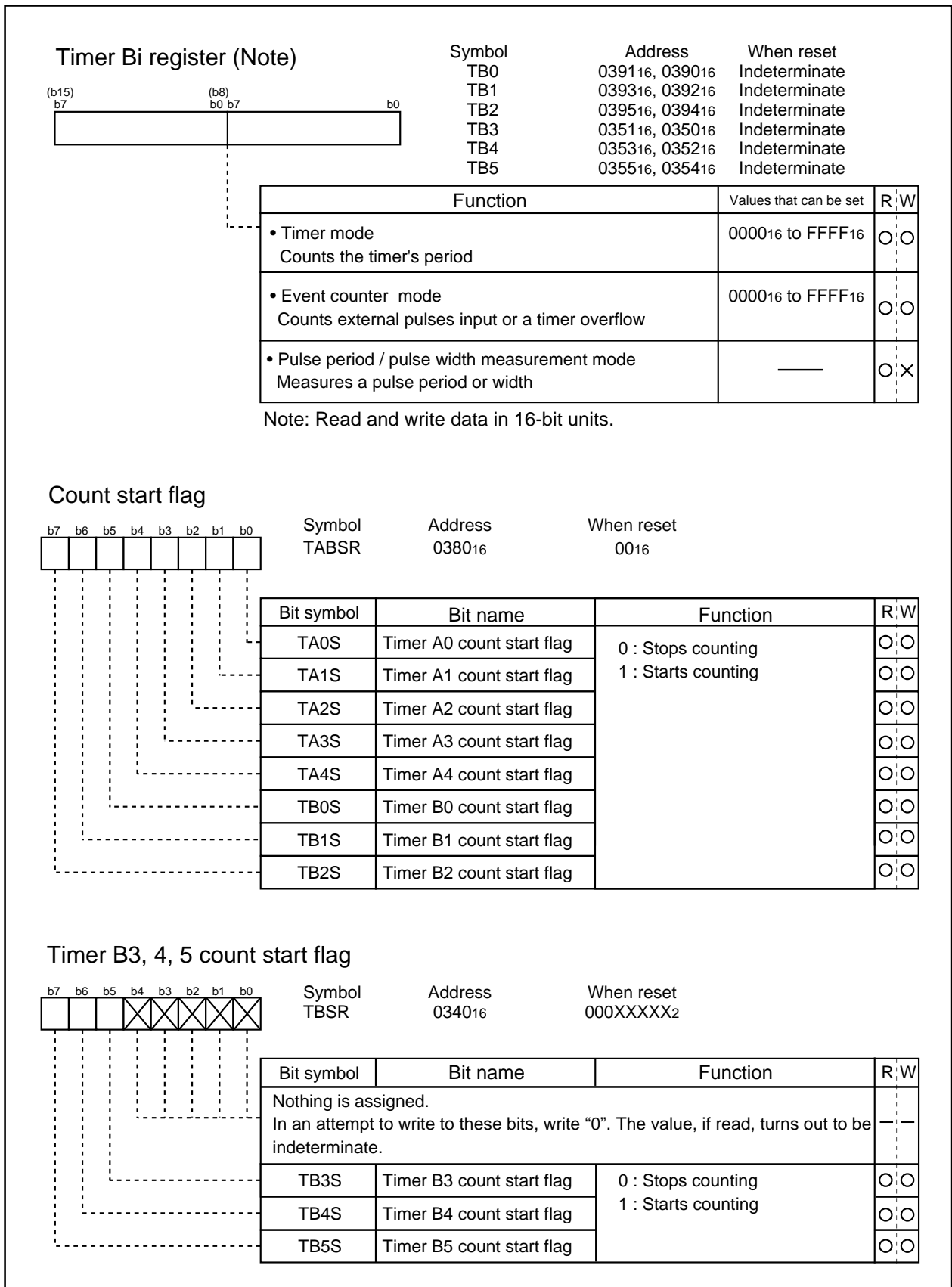


Fig.FB-16 Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table.FB-6.) Fig.FB-17 shows the timer Bi mode register in timer mode.

Table.FB-6 Timer specifications in timer mode

Item	Specification
Count source	f1, f8, f32
Count operation	<ul style="list-style-type: none"> •Counts down •When the timer underflows, it reloads the reload register contents and then continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> •When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter <ul style="list-style-type: none"> • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

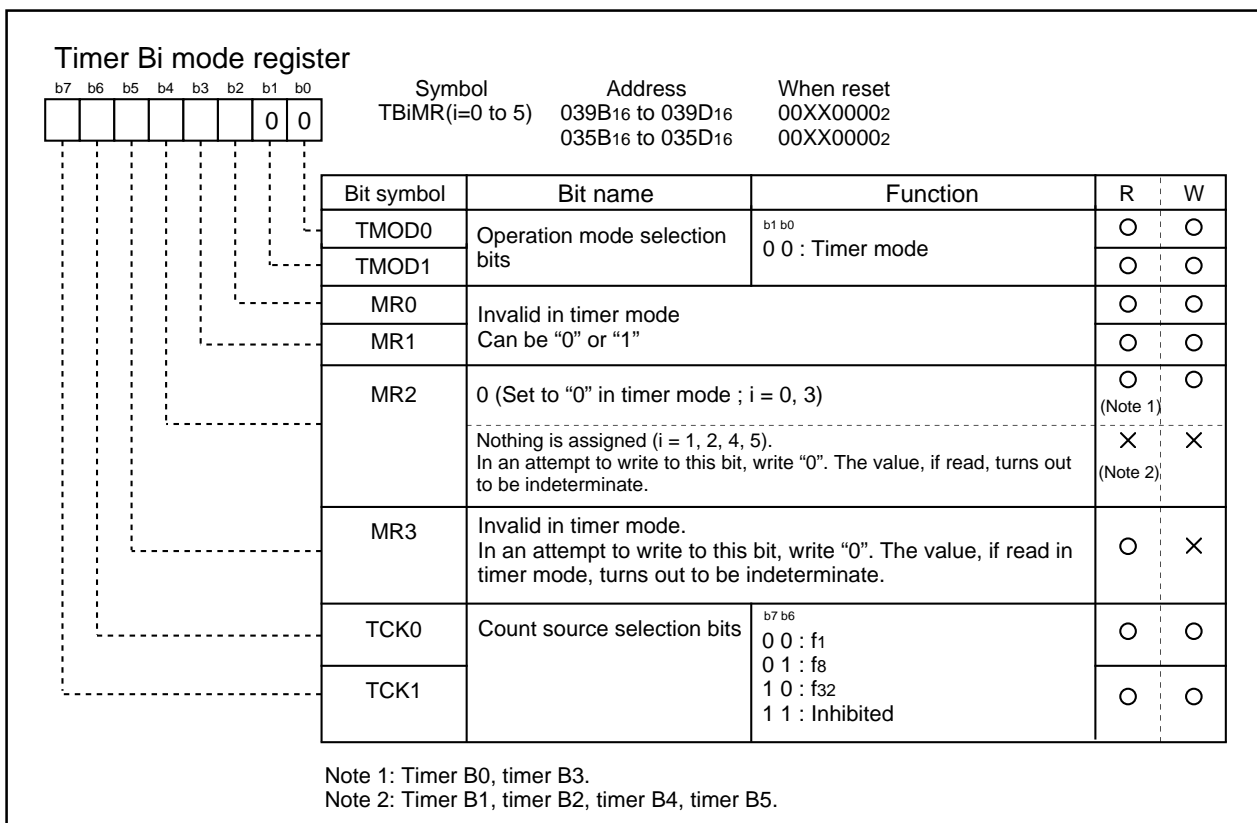


Fig.FB-17 Timer Bi mode register in timer mode

(3) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table.FB-8) Fig.FB-19 shows the timer Bi mode register in pulse period/pulse width measurement mode. Fig.FB-20 shows the operation timing when measuring a pulse period. Fig.FB-21 shows the operation timing when measuring a pulse width.

Table.FB-8 Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f1, f8, f32
Count operation	<ul style="list-style-type: none"> •Up count •At measurement pulse's effective edge, after the count value is transferred to reload register, it is cleared to "0000₁₆" and then continues counting.
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	<ul style="list-style-type: none"> •When measurement pulse's effective edge is input (Note 1) •When an overflow occurs. (Simultaneously, the timer Bi overflow flag becomes "1". The timer Bi overflow flag becomes "0" when the count start flag is "1" and a value is written to the timer Bi mode register.)
TBiIN pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1 : An interrupt request is not generated when the first effective edge is input after the timer has started counting.

Note 2 : After count starts, the value read out from the timer Bi register is indeterminate until the second effective edge is input .

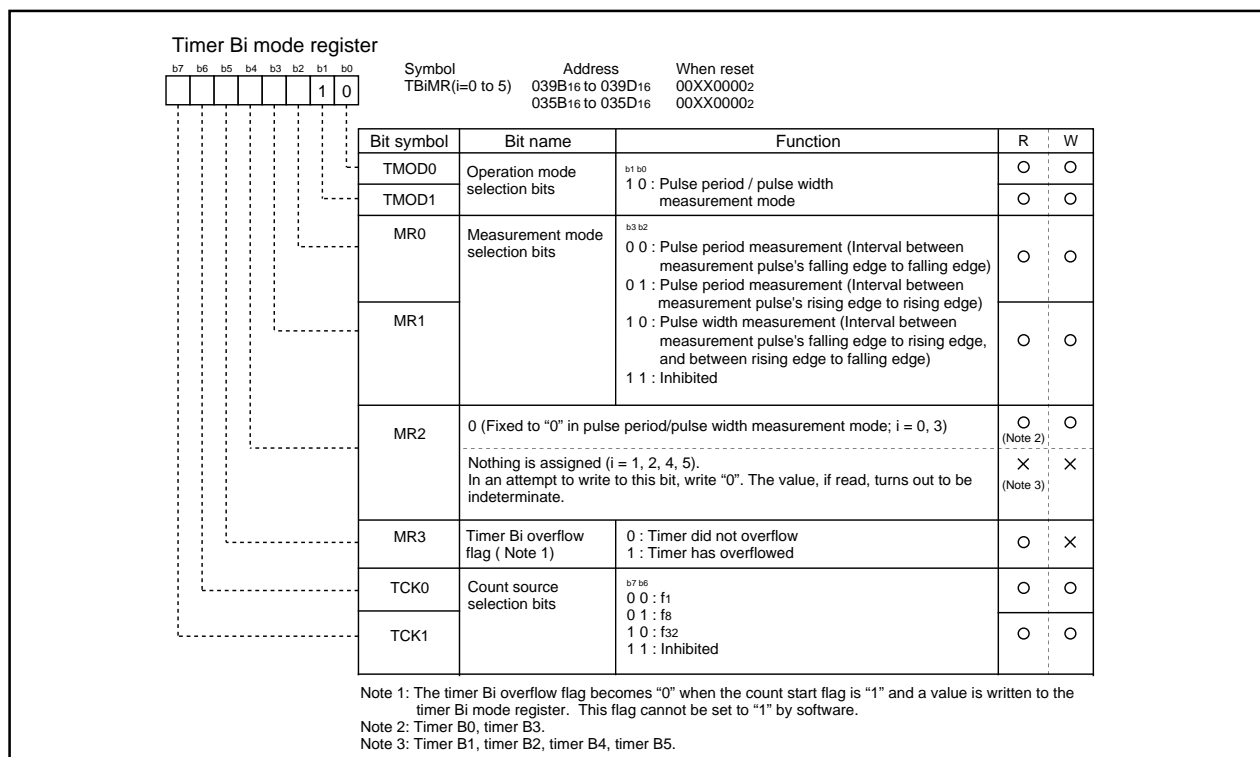


Fig.FB-19 Timer Bi mode register in pulse period/pulse width measurement mode

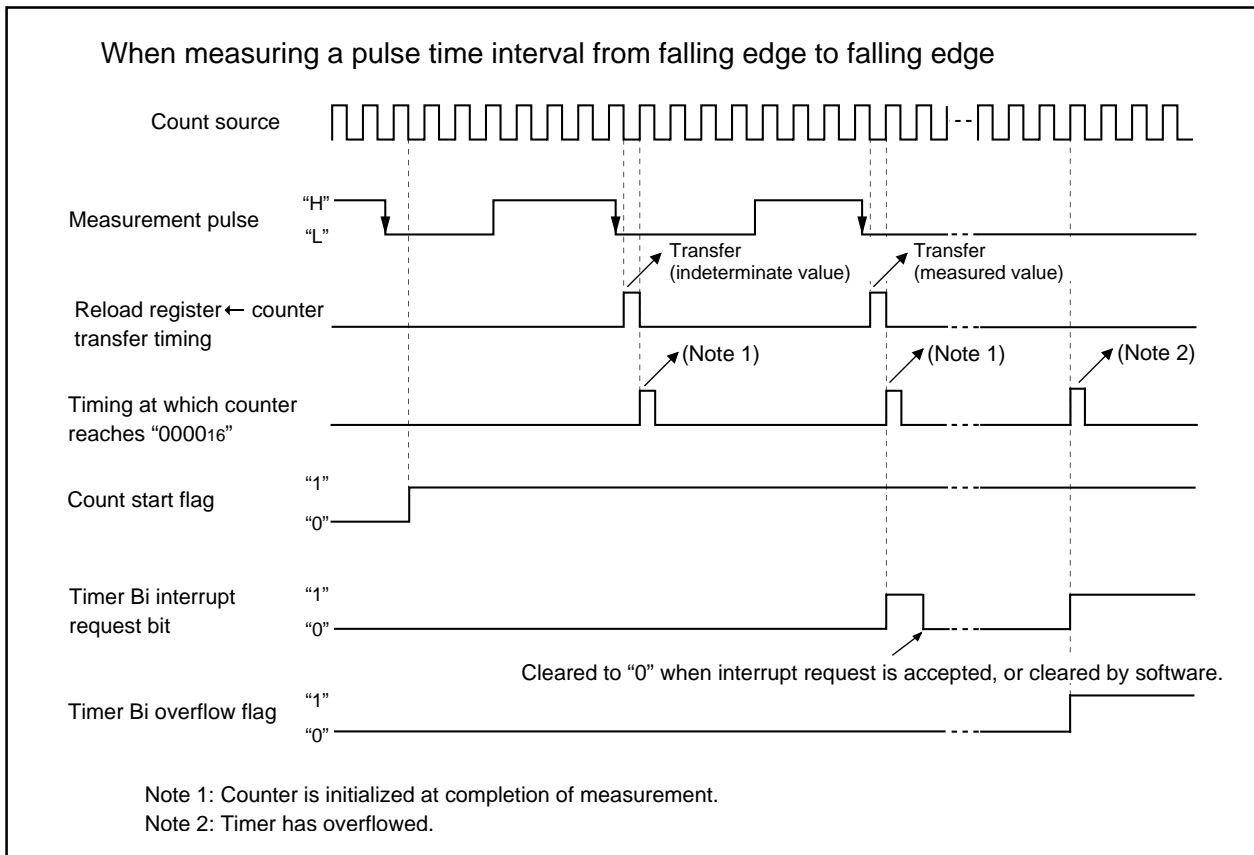


Fig.FB-20 Operation timing when measuring a pulse period

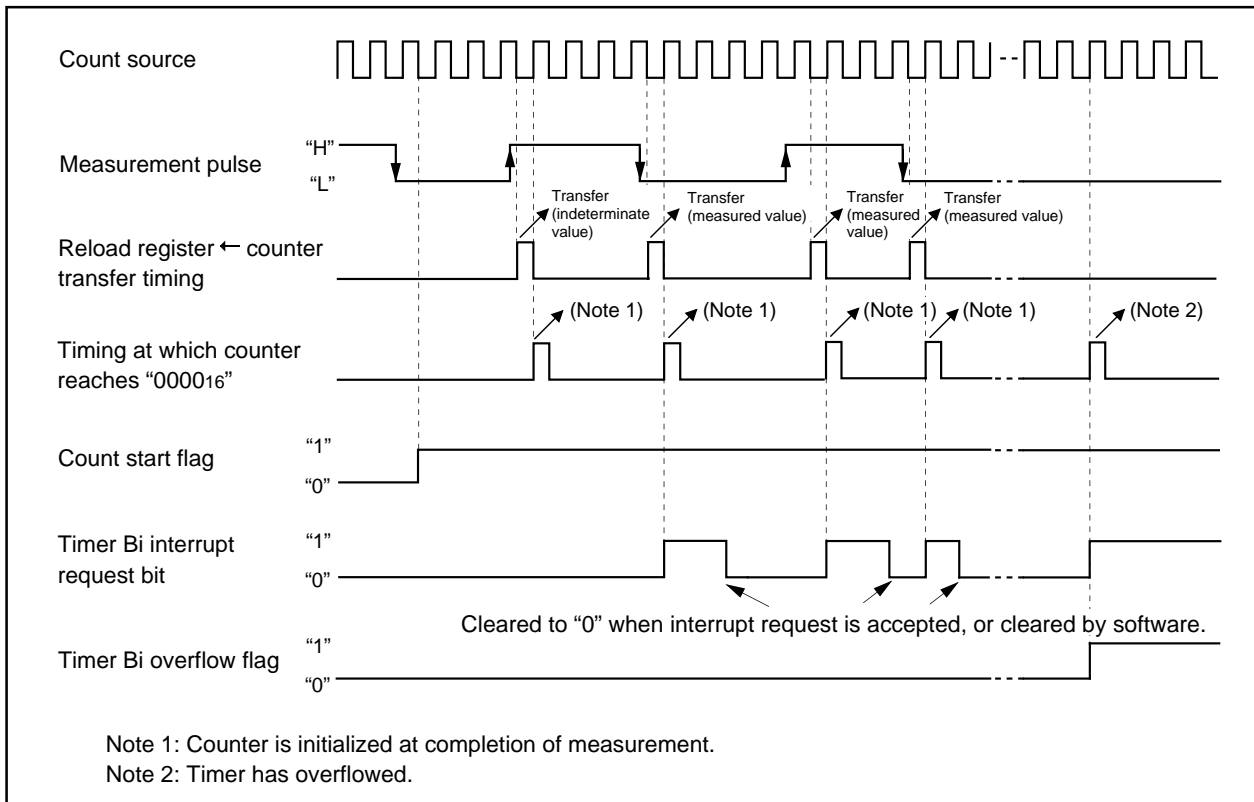


Fig.FB-21 Operation timing when measuring a pulse width

Serial I/O

Serial I/O is configured as three channels: UART1, S I/O3 and S I/O4.

UART1

Fig.GA-1 shows the block diagram of UART1. Fig.GA-2 shows the block diagram of the transmit/receive unit.

UART1 has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode selection bits (bits 0 to 2 at address 03A8₁₆) determine whether UART1 is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART1 has almost the same functions.

Table.GA-1 shows the functions of UART1, and Fig.GA-3 to GA-7 show the registers related to UART1.

Table.GA-1 Functions of UART1

Function	UART1
CLK polarity selection	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)
Continuous receive mode selection	Possible (Note 1)
Transfer clock output from multiple pins selection	Possible (Note 1)
Serial data logic switch	Possible
TxD, RxD I/O polarity switch	Possible
TxD, RxD port output format	CMOS output

Note 1: Only in clock synchronous serial I/O mode.

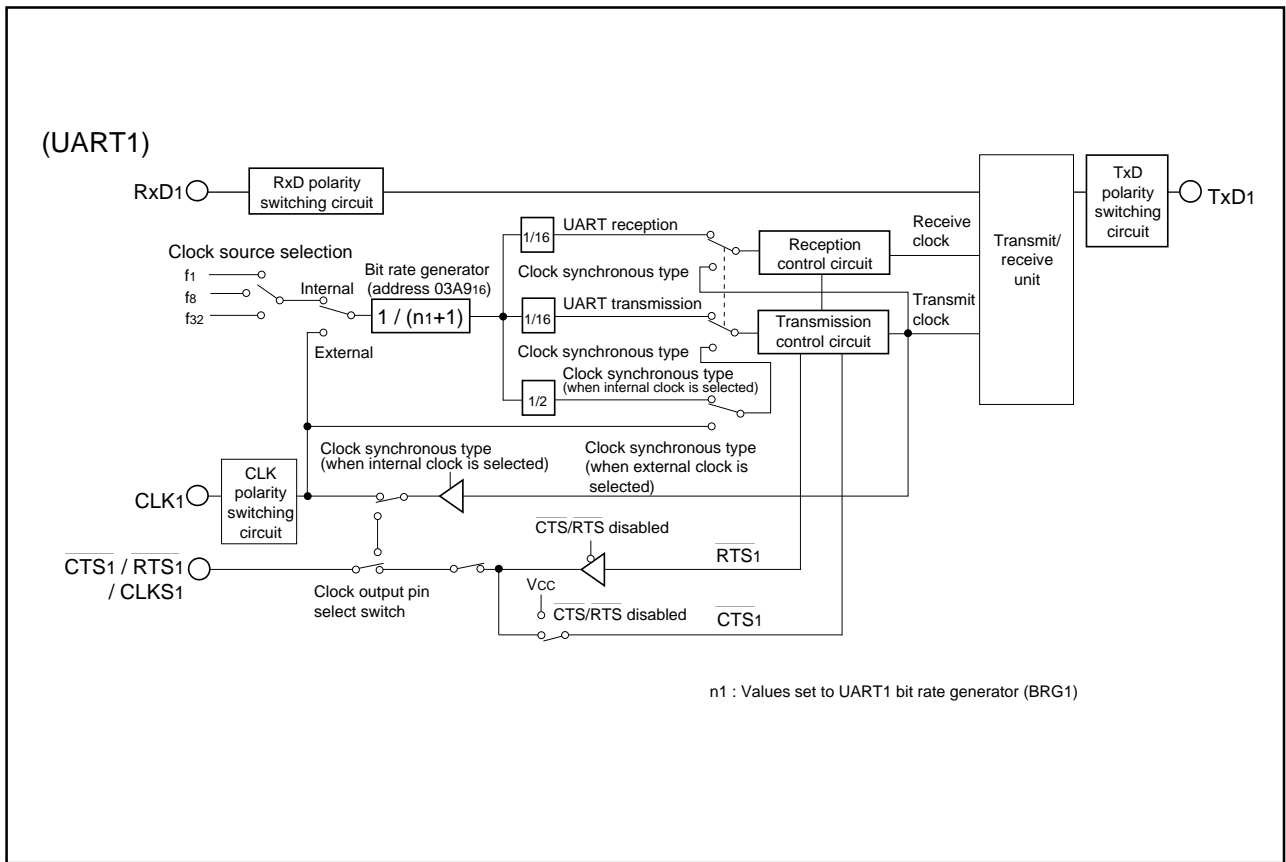


Fig.GA-1 Block diagram of UART1

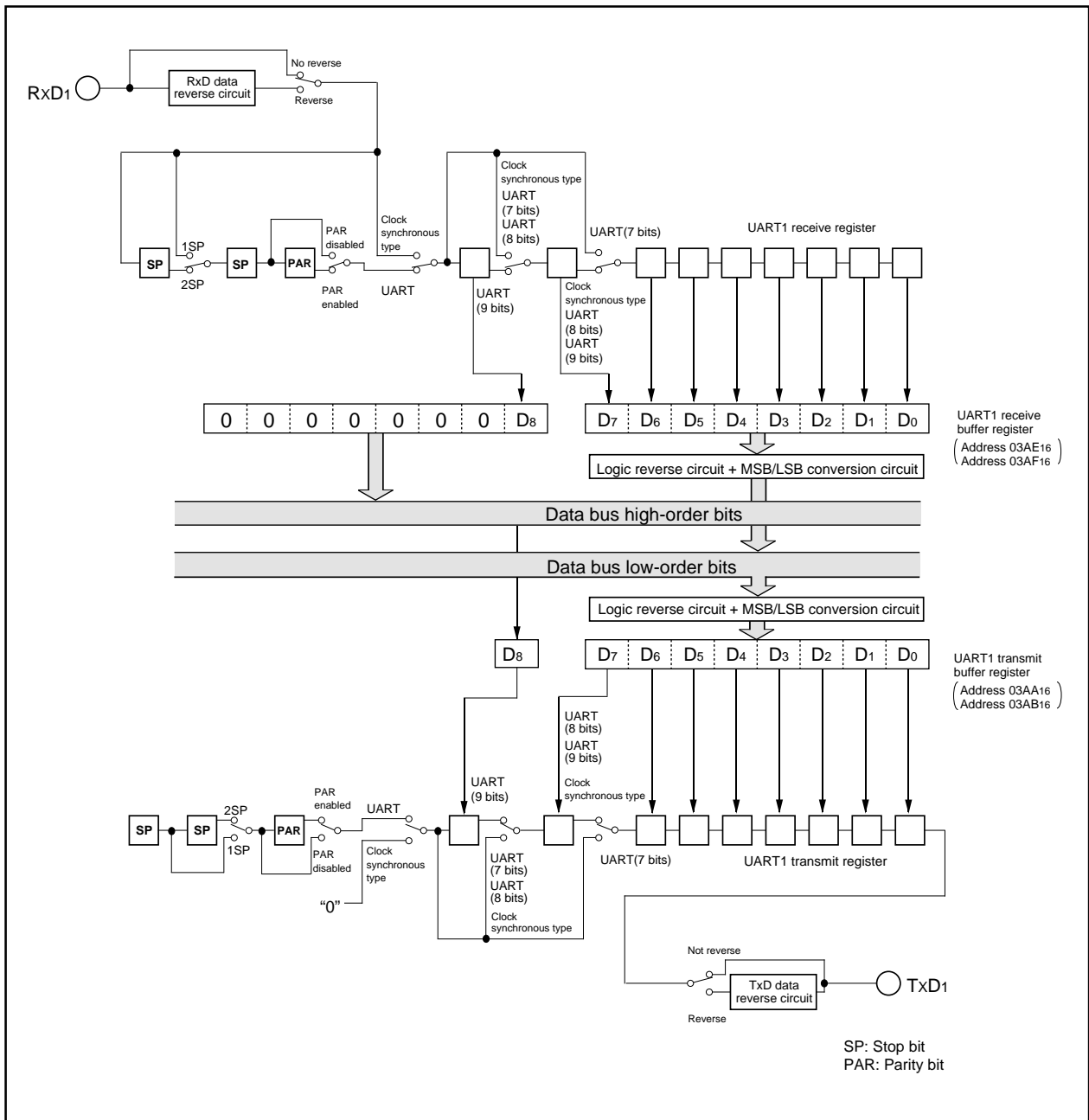


Fig.GA-2 Block diagram of UART1 transmit/receive unit

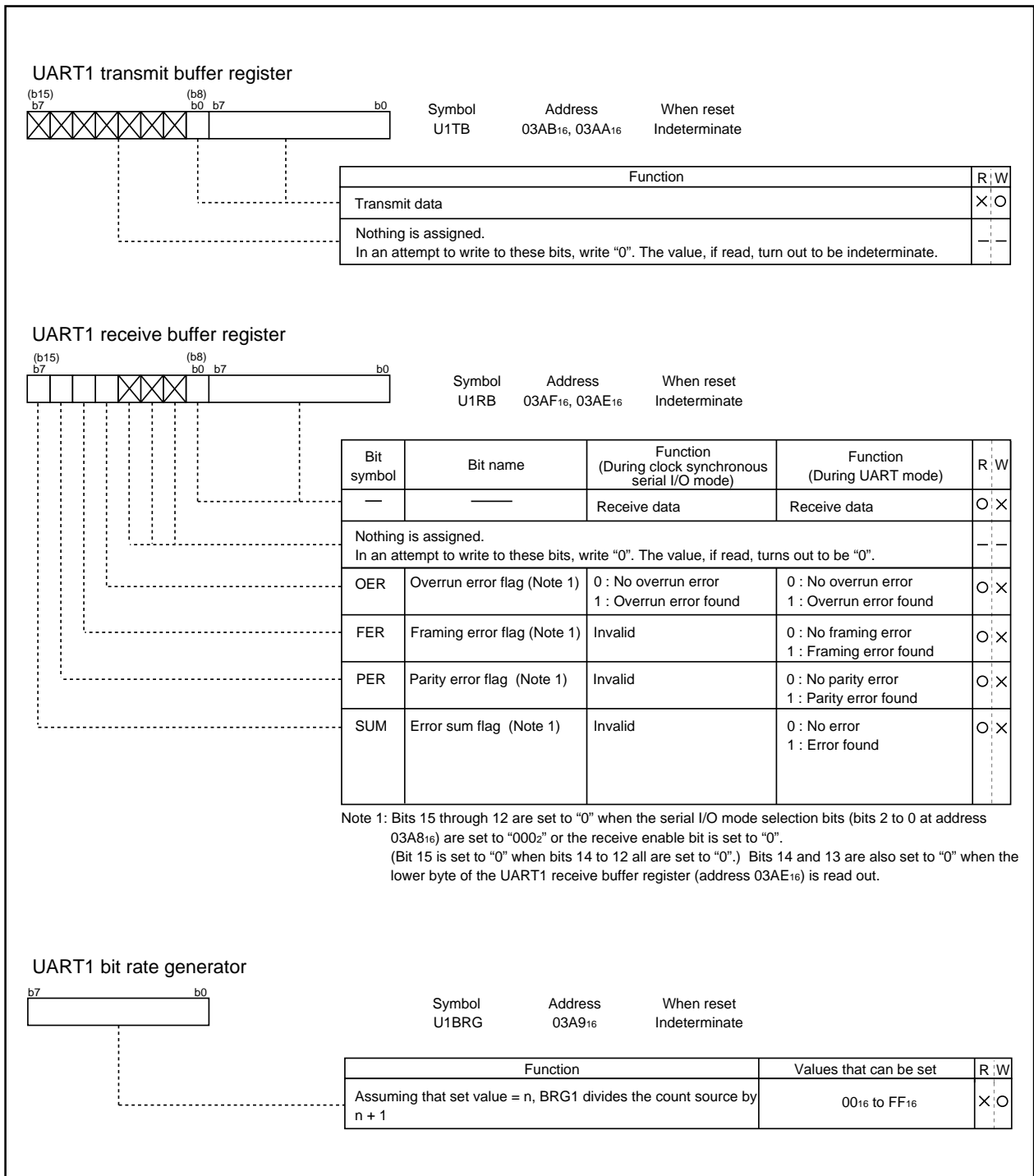


Fig.GA-3 Serial I/O-related registers (1)

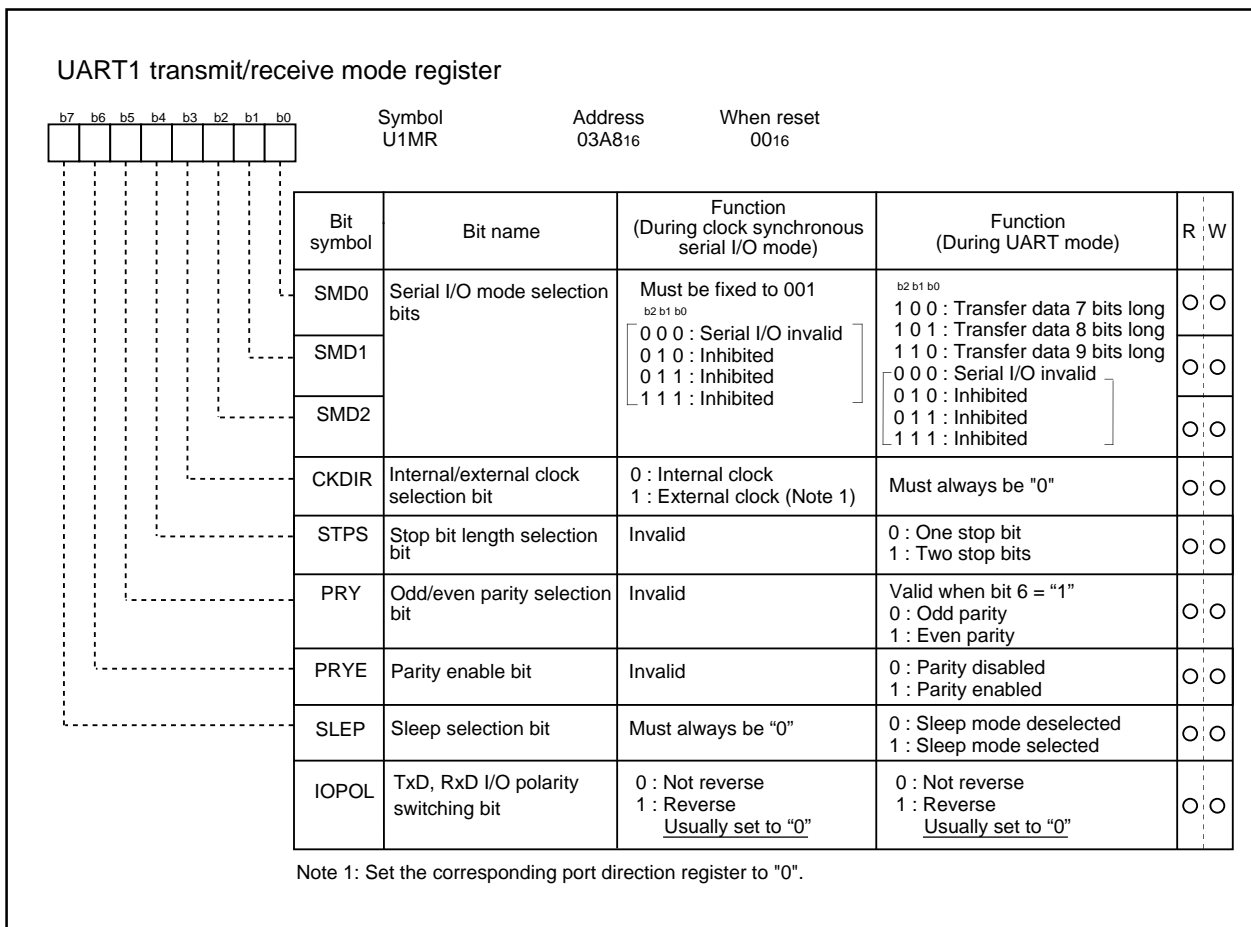


Fig.GA-4 Serial I/O-related registers (2)

UART1 transmit/receive control register 0

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source selection bits	$b_1 b_0$ 0 0 : f ₁ is selected 0 1 : f ₈ is selected 1 0 : f ₃₂ is selected 1 1 : Inhibited	$b_1 b_0$ 0 0 : f ₁ is selected 0 1 : f ₈ is selected 1 0 : f ₃₂ is selected 1 1 : Inhibited	○	○
CLK1					
CRS	CTS/RTS function selection bit	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = "0" 0 : CTS function is selected (Note 1) 1 : RTS function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	CTS/RTS disable bit	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (Pins function as programmable I/O port)	0 : CTS/RTS function enabled 1 : CTS/RTS function disabled (Pins function as programmable I/O port)	○	○
NCH	Data output selection bit	0 : TxD ₁ pin is CMOS output 1 : TxD ₁ pin is N-channel open- drain output	0 : TxD ₁ pin is CMOS output 1 : TxD ₁ pin is N-channel open-drain output	○	○
CKPOL	CLK polarity selection bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be "0"	○	○
UFORM	Transfer format selection bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding port direction register to "0".
 Note 2: The settings of the corresponding port register and port direction register are invalid.
 Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.

Fig.GA-5 Serial I/O-related registers (3)

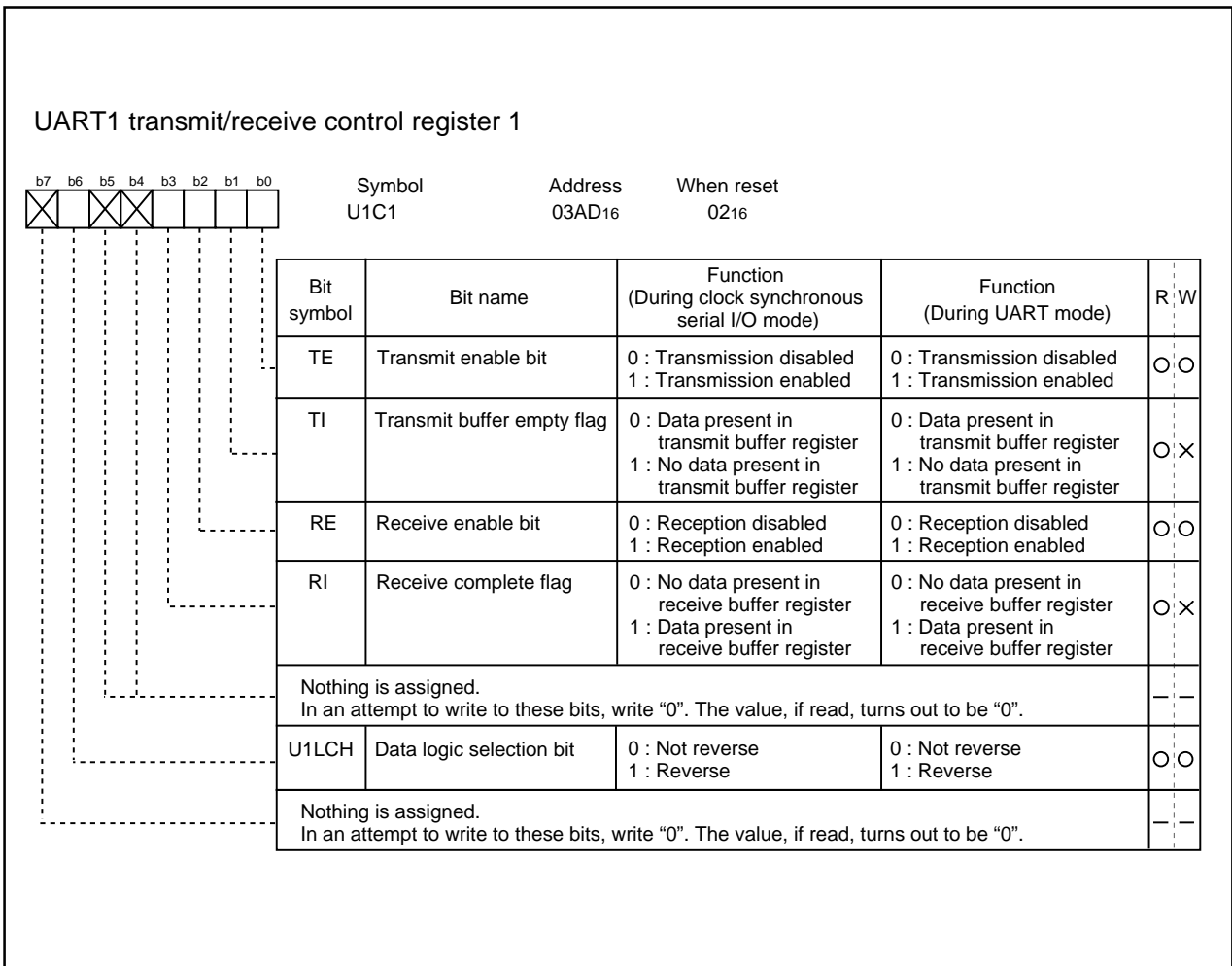


Fig.GA-6 Serial I/O-related registers (4)

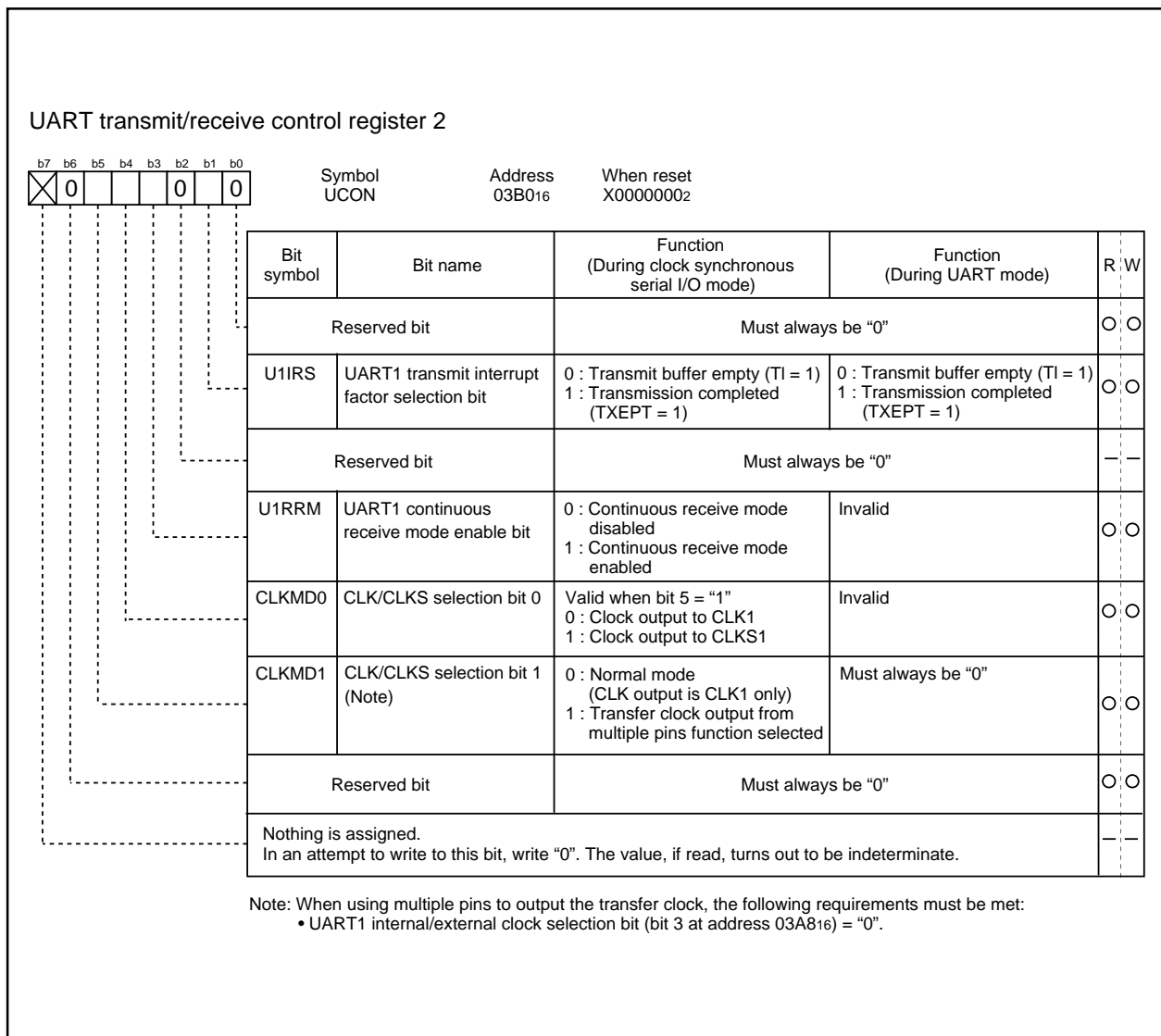


Fig.GA-7 Serial I/O-related registers (5)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables.GA-2 and GA-3 list the specifications of the clock synchronous serial I/O mode. Fig.GA-8 shows the UART1 transmit/receive mode register.

Table.GA-2 Specifications of clock synchronous serial I/O mode (1)

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> • When internal clock is selected (bit 3 at address 03A8₁₆ = "0") : $f_i / 2^{(n+1)}$ (Note 1) $f_i = f_1, f_8, f_{32}$ • When external clock is selected (bit 3 at address 03A8₁₆ = "1") : Input from CLK₁ pin
Transmission/reception control	<ul style="list-style-type: none"> • Selecting from $\overline{\text{CTS}}$ function/RTS function/Disable $\overline{\text{CTS}}$, RTS function
Transmission start condition	<ul style="list-style-type: none"> • To start transmission, the following requirements must be met: <ul style="list-style-type: none"> – Transmit enable bit (bit 0 at address 03AD₁₆) = "1" – Transmit buffer empty flag (bit 1 at address 03AD₁₆) = "0" – When $\overline{\text{CTS}}$ function selected, $\overline{\text{CTS}}$ input level = "L" • Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> – CLK₁ polarity select bit (bit 6 at address 03AC₁₆) = "0": CLK₁ input level = "H" – CLK₁ polarity select bit (bit 6 at address 03AC₁₆) = "1": CLK₁ input level = "L"
Reception start condition	<ul style="list-style-type: none"> • To start reception, the following requirements must be met: <ul style="list-style-type: none"> – Receive enable bit (bit 2 at address 03AD₁₆) = "1" – Transmit enable bit (bit 0 at address 03AD₁₆) = "1" – Transmit buffer empty flag (bit 1 at address 03AD₁₆) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> – CLK₁ polarity select bit (bit 6 at address 03AC₁₆) = "0": CLK₁ input level = "H" – CLK₁ polarity select bit (bit 6 at address 03AC₁₆) = "1": CLK₁ input level = "L"
Interrupt request generation timing	<ul style="list-style-type: none"> • When transmitting <ul style="list-style-type: none"> – Transmit interrupt factor selection bit (bit 1 at address 03B0₁₆) = "0": At the completion of data transmission from UART1 transfer buffer register to UART1 transmit register – Transmit interrupt factor selection bit (bit 1 at address 03B0₁₆) = "1": At the completion of data transmission from UART1 transfer register is completed • When receiving <ul style="list-style-type: none"> – At the completion of data transferring from UART1 receive register to UART1 receive buffer register
Error detection	<ul style="list-style-type: none"> • Overrun error (Note 2) This error occurs when bit 7 of next data is received before the contents of UART1 receive buffer register are read out.

Note 1: "n" denotes the value 00₁₆ to FF₁₆ that is set to the UART bit rate generator.

Note 2: If an overrun error occurs, the UART1 receive buffer will have the next data written in. Note also that the UART1 receive interrupt request bit is not set to "1".

Table.GA-3 Specifications of clock synchronous serial I/O mode (2)

Item	Specification
Function selection	<ul style="list-style-type: none"><li data-bbox="486 400 1409 510">• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected<li data-bbox="486 517 1409 589">• LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected<li data-bbox="486 595 1409 667">• Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register<li data-bbox="486 674 1409 784">• Transfer clock output from multiple pins selection UART1 transfer clock can be chosen by software to be output from one of the two pins set

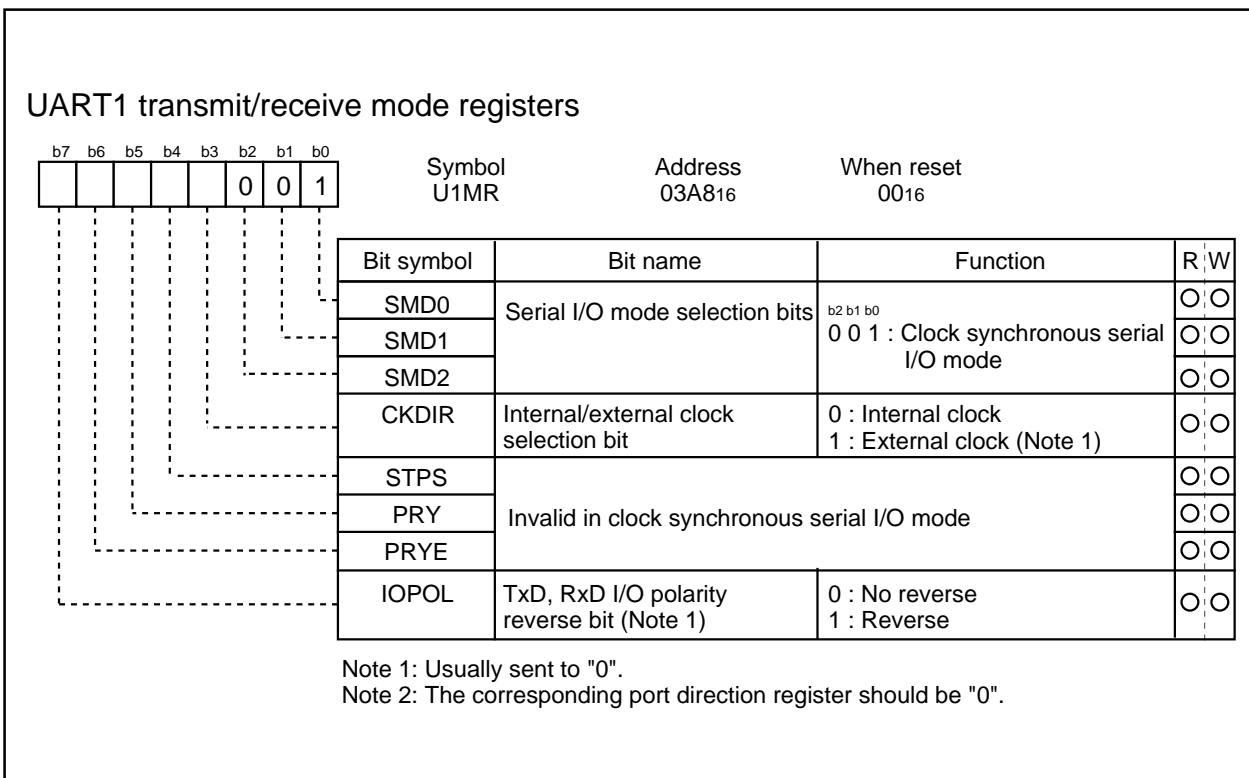


Fig.GA-8 UART1 transmit/receive mode register in clock synchronous serial I/O mode

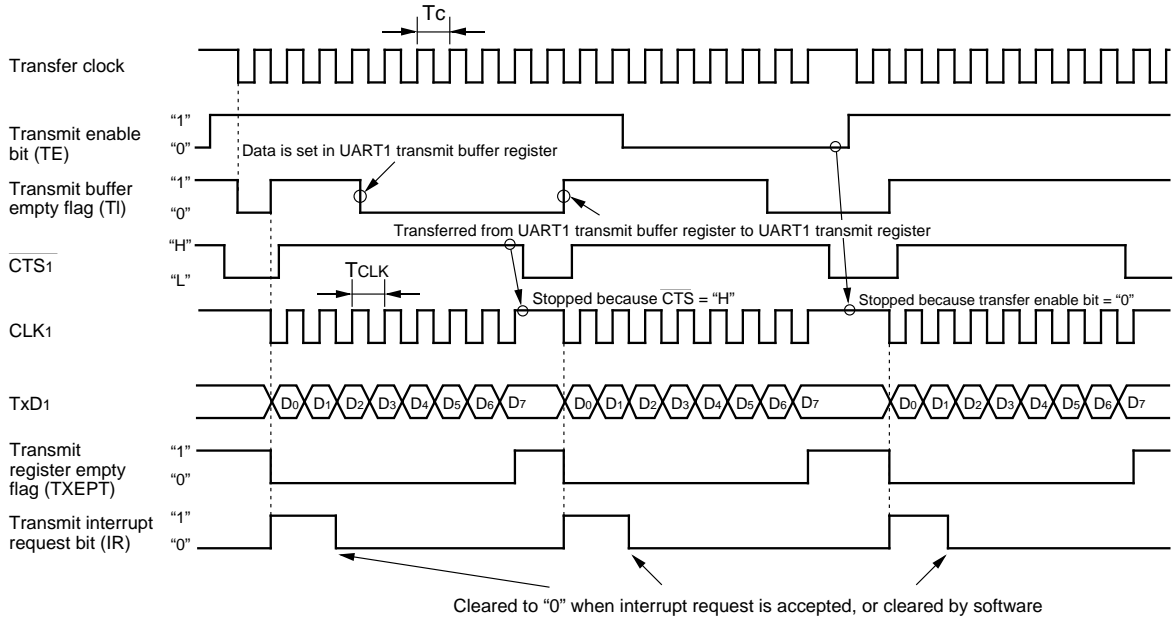
Table.GA-4 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions that the transfer clock output from multiple pins are not selected. Note that for a period from when the UART1 operation mode is selected to when transfer starts, the TxD1 pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table.GA-4 Input/output pin functions in clock synchronous serial I/O mode

(The function that the transfer clock output from multiple pin is not selected.)

Pin name	Function	Method of selection
TxD1	Serial data output	(Outputs dummy data when performing reception only)
RxD1	Serial data input	The corresponding bit of port direction register = "0" (Can be used as an input port when performing transmission only)
CLK1	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A816) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A816) = "1" The corresponding bit of port direction register = "0"
$\overline{\text{CTS}}_1/\overline{\text{RTS}}_1$	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03AC16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit (bit 2 at address 03AC16) = "0" The corresponding port direction bit = "0"
	RTS output	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03AC16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit (bit 2 at address 03AC16) = "1"
	Programmable I/O port	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03AC16) = "1"

• Example of transmit timing (when internal clock is selected)



Shown in () are bit symbols.

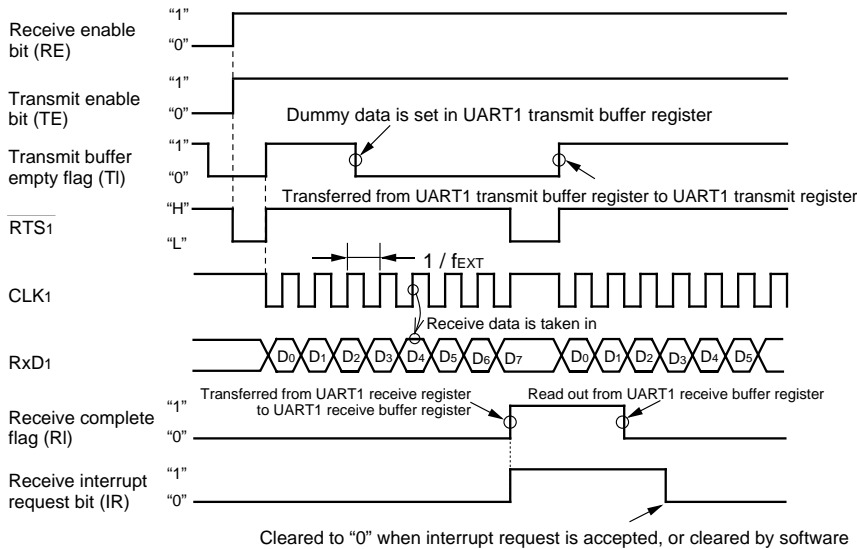
The above timing applies to the following settings:

- Internal clock is selected.
- CTS function is selected.
- CLK polarity selection bit = "0".
- Transmit interrupt factor selection bit = "0".

$$T_c = T_{CLK} = 2(n + 1) / f_i$$

fi: frequency of BRG1 count source (f1, f8, f32)
n: value set to BRG1

• Example of receive timing (when external clock is selected)



Shown in () are bit symbols.

The above timing applies to the following settings:

- External clock is selected.
- RTS function is selected.
- CLK polarity selection bit = "0".

fEXT: frequency of external clock

The following conditions should be matched when the input level of CLK1 pin is "H" before the data reception.

- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UART1 transmit buffer register

Fig.GA-9 Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity selection function

As shown in Fig.GA-10, the CLK polarity selection bit (bit 6 at address 03AC16) allows to select the polarity of the transfer clock.

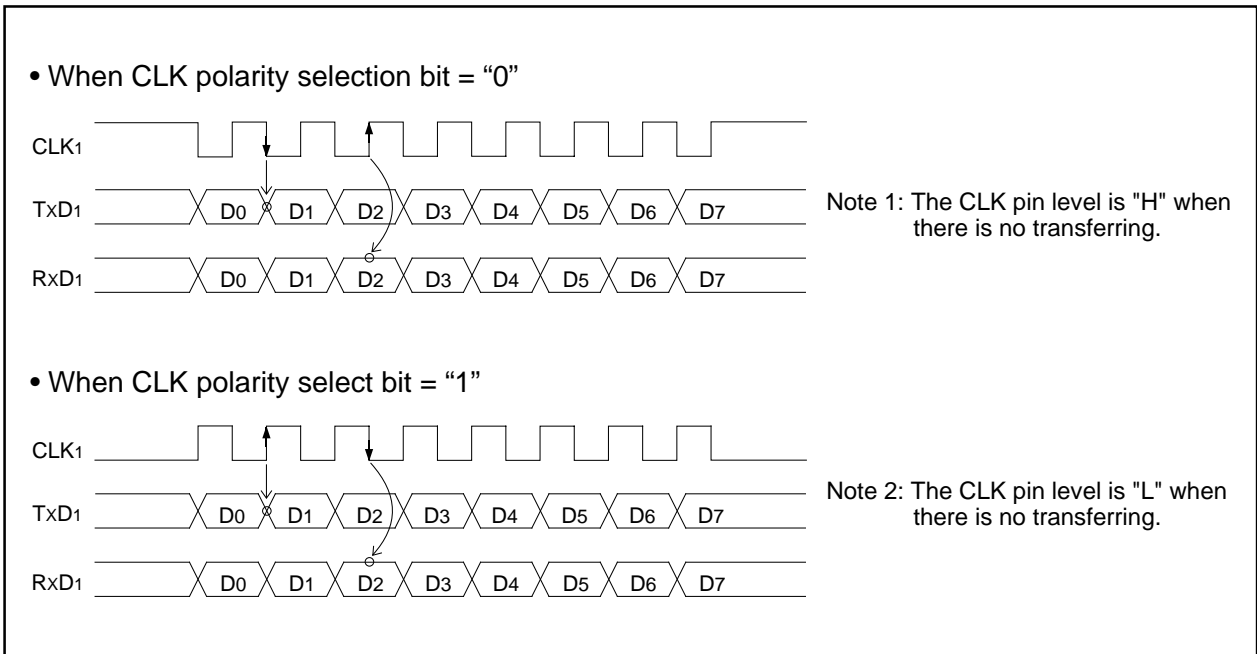


Fig.GA-10 Polarity of transfer clock

(b) LSB first/MSB first selection function

As shown in Fig.GA-11, when the transfer format selection bit (bit 7 at address 03AC16) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

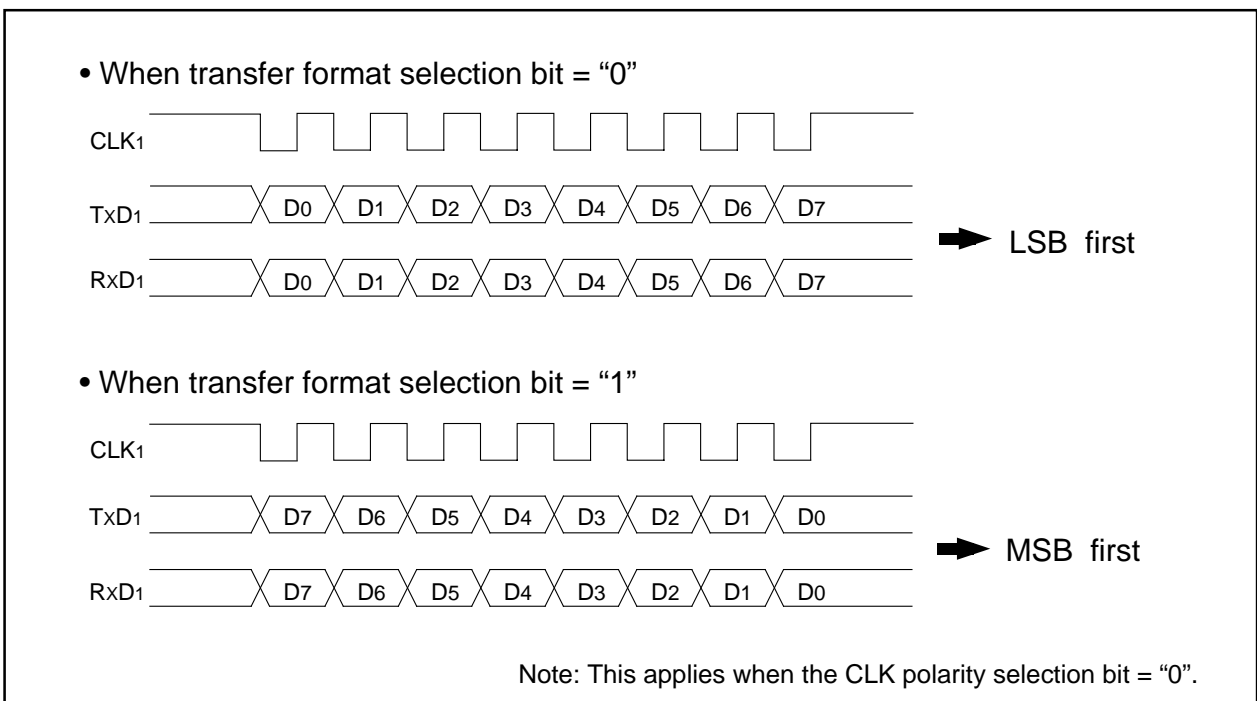


Fig.GA-11 Transfer format

(c) Transfer clock output from multiple pins function (UART1)

This function allows to set two transfer clock output pins and chooses one to output a clock by the setting of CLK and CLKS selection bits (bits 4 and 5 at address 03B016). (See Fig.GA-12) The function is valid only when the UART1 internal clock is selected. Note that when this function is selected, $\overline{\text{CTS}}/\overline{\text{RTS}}$ function cannot be used.

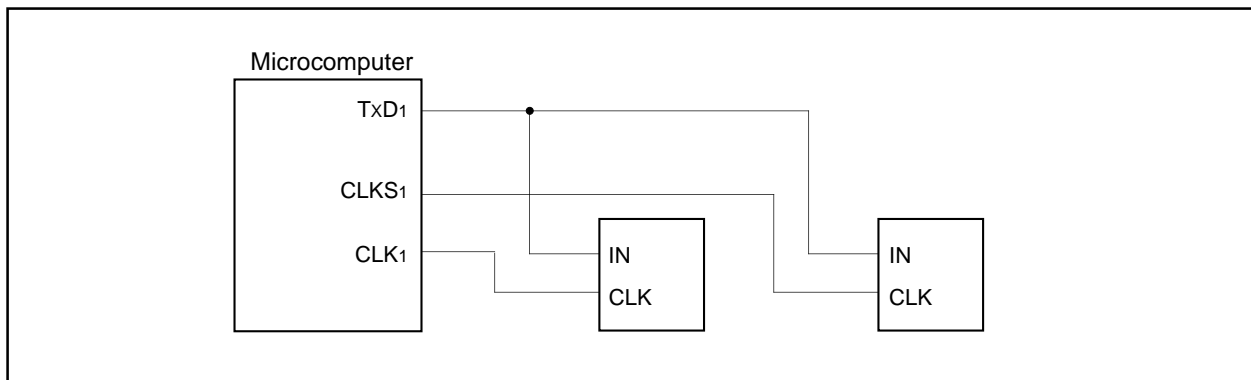


Fig.GA-12 The sample of transfer clock output from the multiple pins function

(d) Continuous receive mode

If the continuous receive mode enable bit (bit 3 at address 03B016) are set to "1", the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

(2) Clock asynchronous serial I/O (UART) mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables.GA-5 and GA-6 list the specifications of the UART mode. Fig.GA-13 shows the UART1 transmit/receive mode register.

Table.GA-5 Specifications of UART Mode (1)

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	<ul style="list-style-type: none"> • When internal clock is selected (bit 3 at address 03A8₁₆ = "0") : $f_i/16(n+1)$ (Note 1) $f_i = f_1, f_8, f_{32}$ • When external clock is selected (bit 3 at address 03A8₁₆ = "1") : $f_{EXT}/16(n+1)$(Note 1) (Note 2)
Transmission/reception control	<ul style="list-style-type: none"> • Selecting from Disable \overline{CTS}, RTS function
Transmission start condition	<ul style="list-style-type: none"> • To start transmission, the following requirements must be met: <ul style="list-style-type: none"> - Transmit enable bit (bit 0 at address 03AD₁₆) = "1" - Transmit buffer empty flag (bit 1 at address 03AD₁₆) = "0" - When \overline{CTS} function is selected \overline{CTS} input level = "L"
Reception start condition	<ul style="list-style-type: none"> • To start reception, the following requirements must be met: <ul style="list-style-type: none"> - Receive enable bit (bit 2 at address 03AD₁₆) = "1" - Start bit detection
Interrupt request generation timing	<ul style="list-style-type: none"> • When transmitting <ul style="list-style-type: none"> - Transmit interrupt factor selection bit (bit 1 at address 03B0₁₆) = "0": At the completion of data transferring from UART1 transfer buffer register to UART1 transmit register - Transmit interrupt factor selection bit (bit 1 at address 03B0₁₆) = "1": At the completion of data transmission from UART1 transfer register • When receiving <ul style="list-style-type: none"> - At the completion of data transferring from UART1 receive register to UART1 receive buffer register
Error detection	<ul style="list-style-type: none"> • Overrun error (Note 3) This error occurs when the bit prior to the stop bit of next data is received before the contents of UART1 receive buffer register are read out. • Framing error This error occurs when the number set for stop bits is not detected • Parity error This error occurs in the case that parity is enabled and the number of "1" in parity bit and character bits does not match the number of "1" in parity odd/even setting. • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered

Note 1: 'n' denotes the value 00₁₆ to FF₁₆ that is set to the UART1 bit rate register.

Note 2: f_{EXT} is input from the CLK1 pin.

Note 3: If an overrun error occurs, the UART1 receive buffer will have the next data written in. Also note that the UART1 receive interrupt request bit is not set to "1".

Table.GA-6 Specifications of UART Mode (2)

Item	Specification
Function selection	<ul style="list-style-type: none"><li data-bbox="507 405 1428 517">• Serial data logic switch This function is reversing logic value of transferring data. Start bit, and stop bit are not reversed.<li data-bbox="507 524 1428 629">• TXD, RXD I/O polarity switch This function is reversing TXD port output and RXD port input. All I/O data level are reversed.

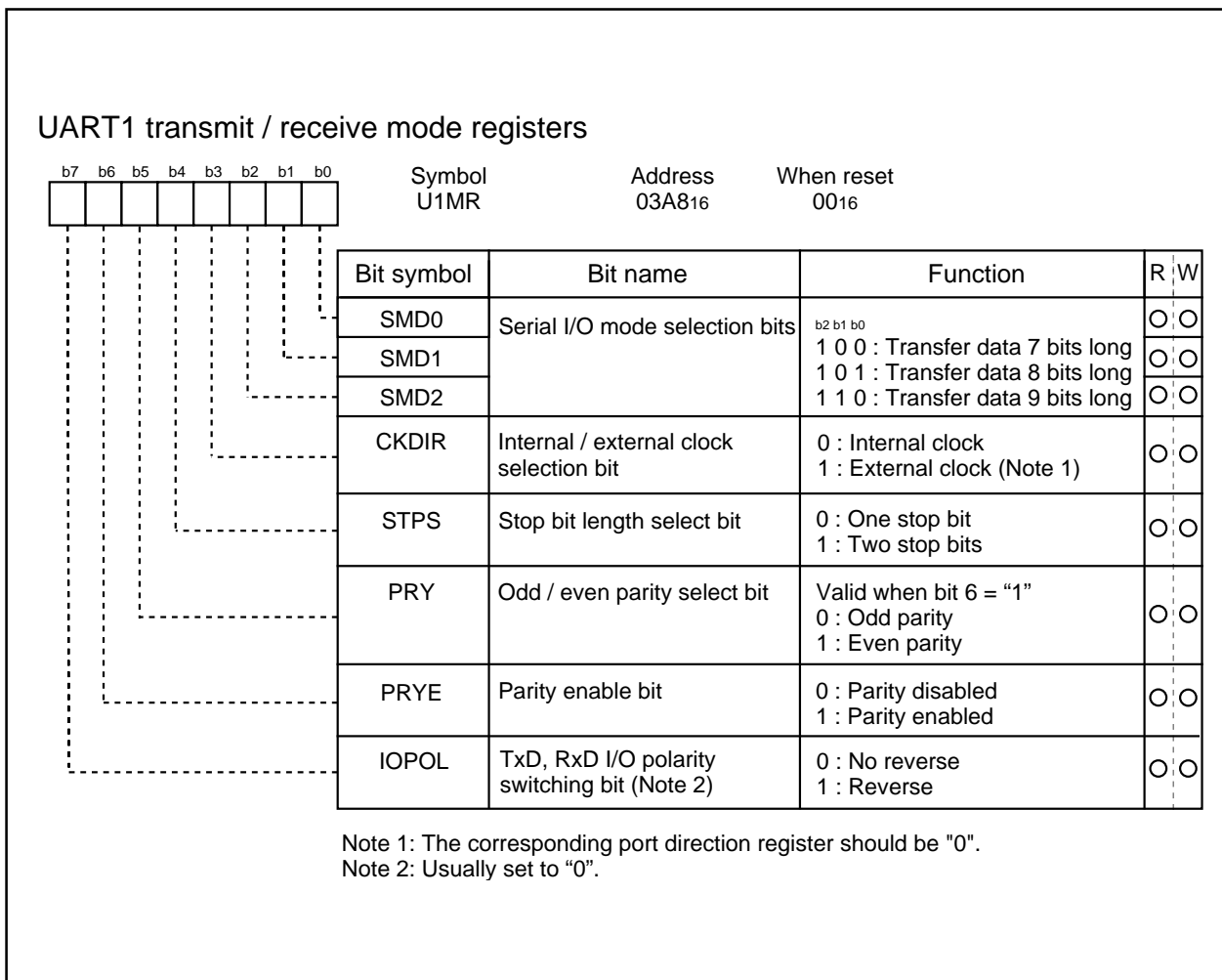


Fig.GA-13 UARTi transmit/receive mode register in UART mode

Table.GA-7 lists the functions of the input/output pins during UART mode. Note that for a period from when the UART1 operation mode is selected to when transfer starts, the TxD1 pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table.GA-7 Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxD1	Serial data output	
RxD1	Serial data input	Corresponding port direction register bit = "0". (Can be used as an input port when performing transmission only)
CLK1	Programmable I/O port	Internal/external clock selection bit (bit 3 at address 03A816) = "0"
	Transfer clock input	Internal/external clock selection bit (bit 3 at address 03A816) = "1" Corresponding port direction register bit = "0"
$\overline{\text{CTS}}_1/\overline{\text{RTS}}_1$	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03AC16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit (bit 2 at address 03AC16) = "0" Corresponding port direction register bit = "0"
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03AC16) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function selection bit (bit 2 at address 03AC16) = "1"
	Programmable I/O port	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03AC16) = "1"

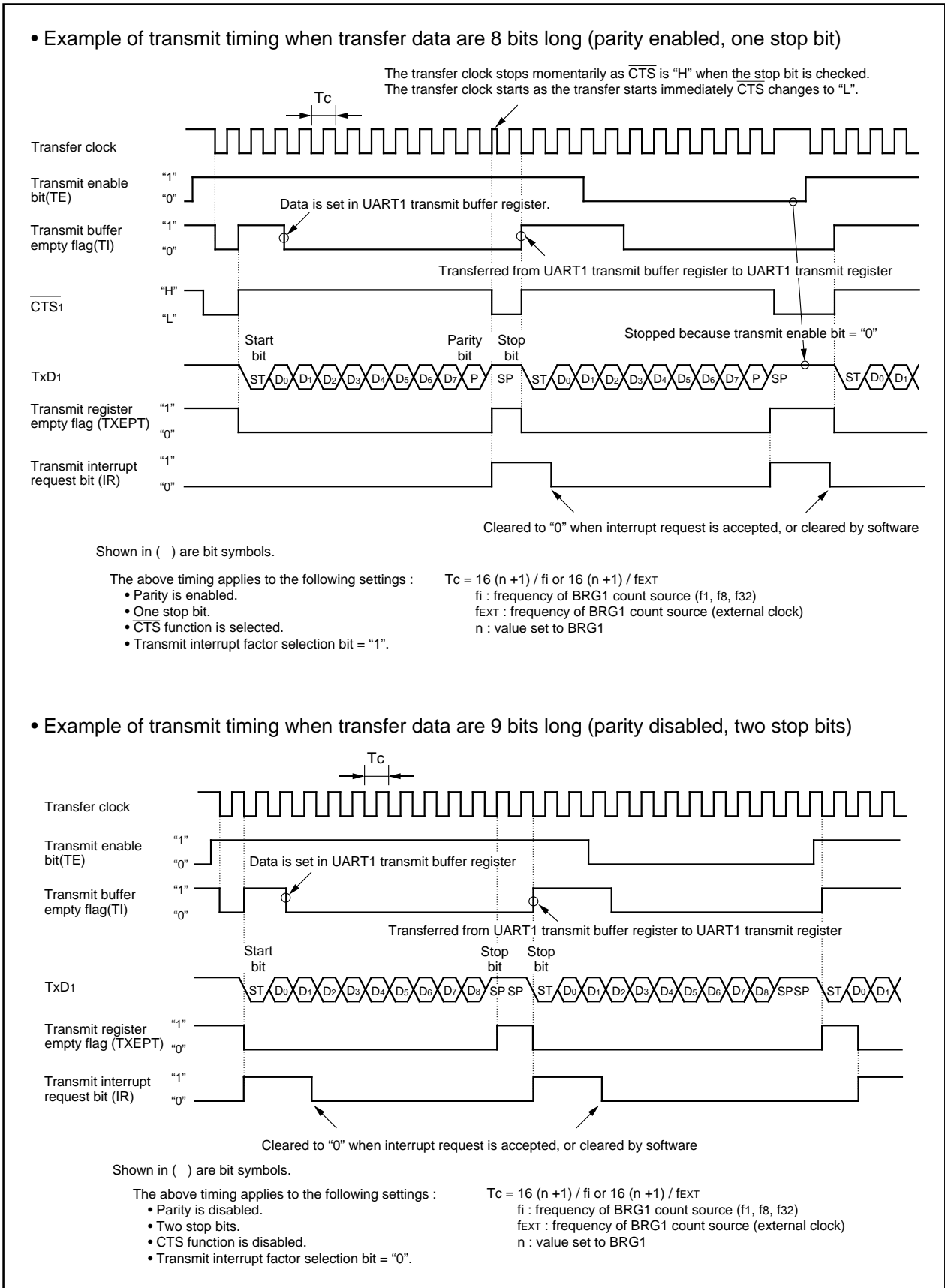


Fig.GA-14 Typical transmit timings in UART mode

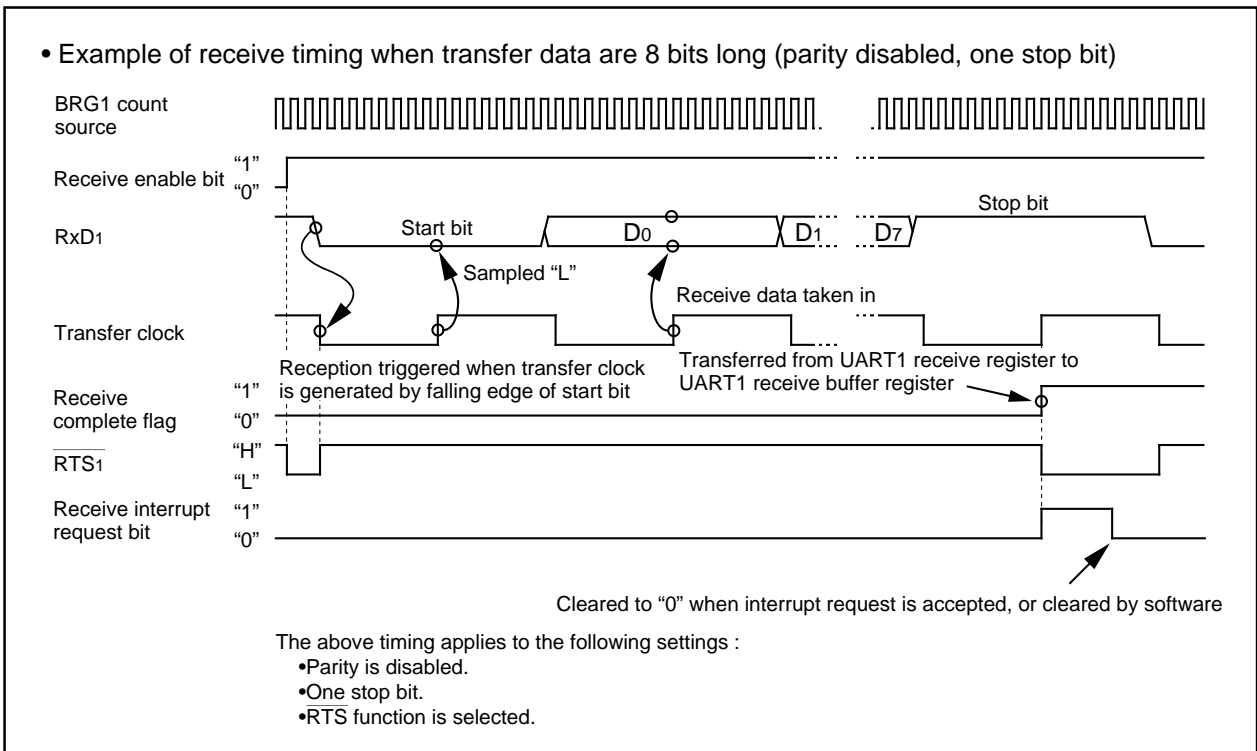


Fig.GA-15 Typical receive timing in UART mode

(a) Function for switching serial data logic

When the data logic selection bit is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Fig.GA-16 shows the example of timing for switching serial data logic.

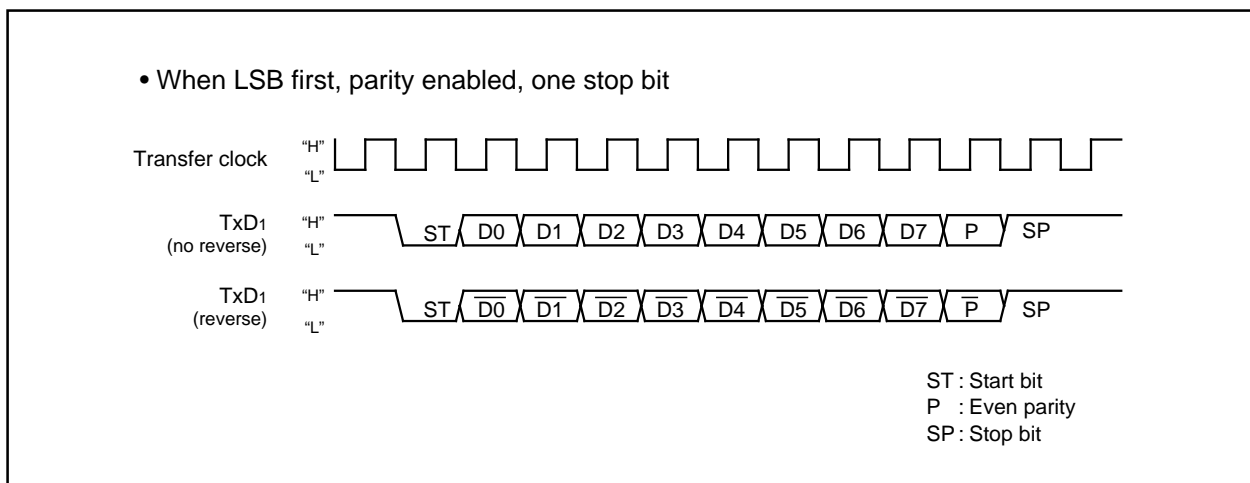


Fig.GA-16 Timing for switching serial data logic

(b) TxD, RxD I/O polarity switching function

This function is to reverse TxD pin output and RxD pin input. The level of any data to be input or output (including the start bit, stop bit(s), and parity bit) is reversed. Set this function to "0" (not to reverse) for usual use.

S I/O3, 4

S I/O 3 and S I/O 4 are exclusive clock-synchronous serial I/Os.

Fig.GA-17 shows the S I/O 3, 4 block diagram, and Fig.GA-18 shows the S I/O 3, 4 control register.

Table.GA-8 shows the specifications of S I/O 3, 4.

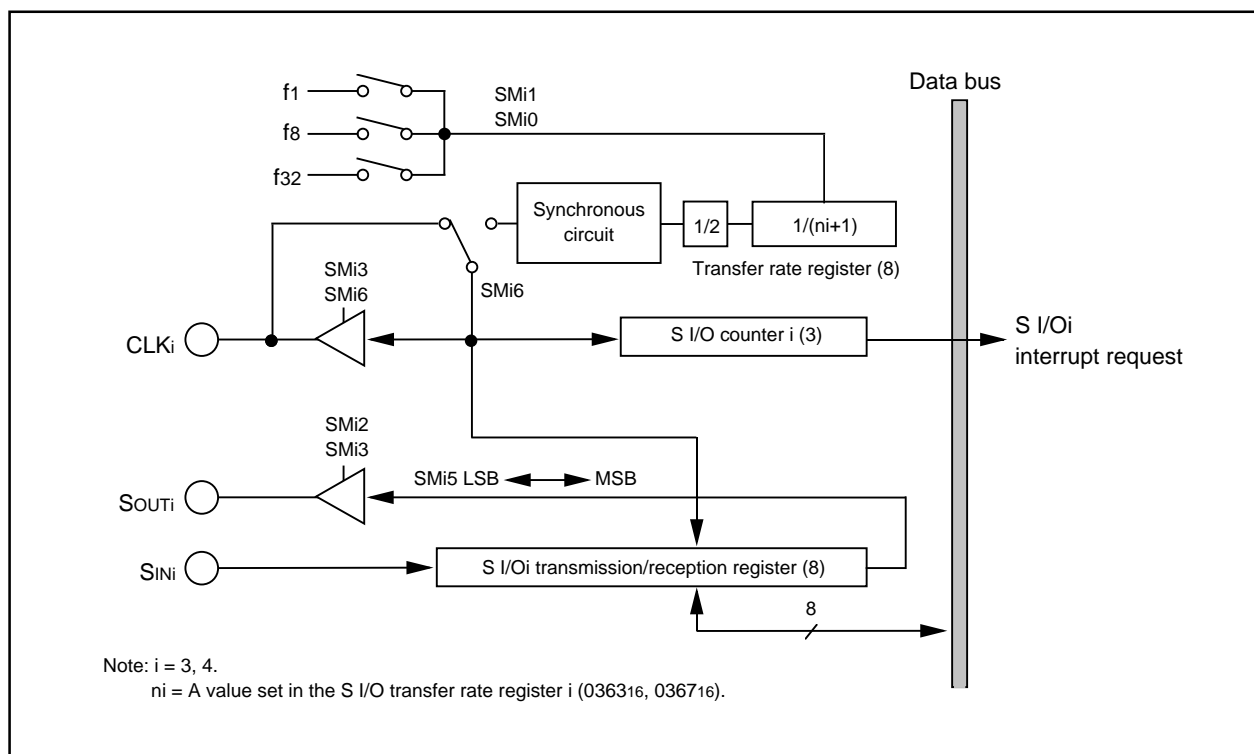


Fig.GA-17 S I/O3, 4 block diagram

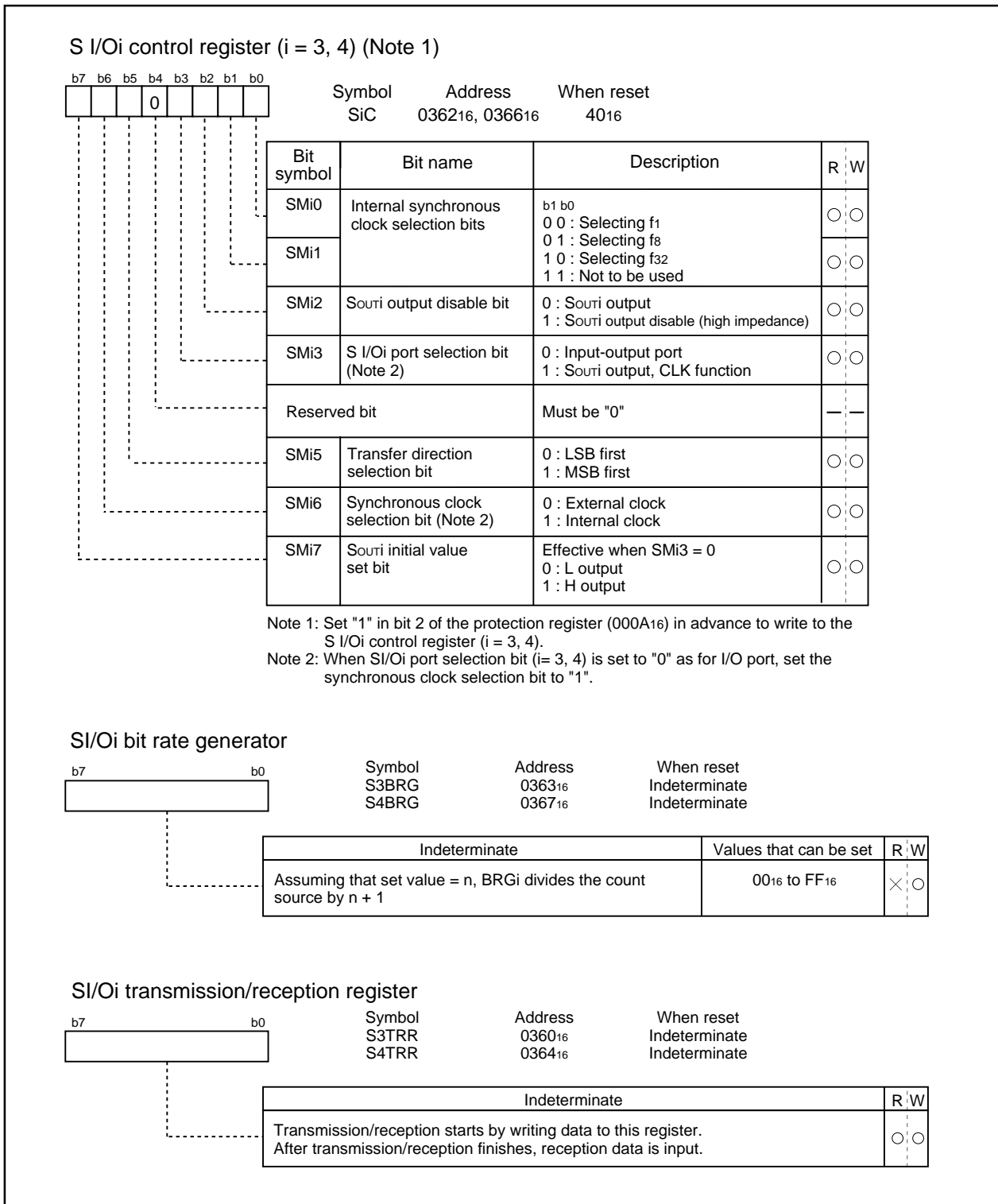


Fig.GA-18 S I/O3, 4 control registers

Table.GA-8 Specifications of S I/O3, 4

Item	Specifications
Transfer data format	<ul style="list-style-type: none"> Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> With the internal clock selected (bit 6 of 0362₁₆, 0366₁₆ = "1"): f₁/2(ni+1), f₈/2(ni+1), f₃₂/2(ni+1) (Note 1) With the external clock selected (bit 6 of 0362₁₆, 0366₁₆ = 0): Input from the CLKi terminal (Note 2)
Conditions for transmission/reception start	<ul style="list-style-type: none"> To start transmit/reception, the following requirements must be met: <ul style="list-style-type: none"> Select the synchronous clock (use bit 6 of 0362₁₆, 0366₁₆). Select a frequency dividing ratio if the internal clock has been selected (use bits 0 and 1 of 0362₁₆, 0366₁₆). SOUTi initial value set bit (use bit 7 of 0362₁₆, 0366₁₆) = 1. S I/Oi port select bit (bit 3 of 0362₁₆, 0366₁₆) = 1. Select the transfer direction (use bit 5 of 0362₁₆, 0366₁₆) Write transfer data to SI/Oi transmission/reception register(0360₁₆, 0364₁₆) To use S I/Oi interrupts, the following requirements must be met: <ul style="list-style-type: none"> S I/Oi interrupt request bit (bit 3 of 0049₁₆, 0048₁₆) = 0.
Interrupt request generation timing	<ul style="list-style-type: none"> At the rising edge of the last transfer clock (Note3)
Select function	<ul style="list-style-type: none"> LSB first or MSB first selection Whether transmission/reception begins with bit 0 (LSB) or bit 7 (MSB) can be selected. The SOUTi default value setting function If the transfer clock is selected to external clock, the output level of SOUTi pin can be selected when it is not in transferring please refer to Fig.GA-30.
Precaution	<ul style="list-style-type: none"> The SI/Oi (i=3,4) is different from UART0 to 2 that the register and buffer can not be separated, so don't write the next transfer data to the transmission/reception register(0360₁₆, 0364₁₆) during transferring. If the transfer clock is selected to internal clock, at the end of transferring, the SOUTi holds the last data during the last 1/2 transfer clock, and then to high impedance. If the transmission/reception register(0360₁₆, 0364₁₆) is written during the period, the SOUTi becomes the high impedance right the writing, the data hold time will be shortened.

Note 1: n is a value from 00₁₆ through FF₁₆ set in the S I/Oi transfer rate register (i = 3, 4).

Note 2: With the external clock selected:

- Please write to the SI/Oi transmission/reception register(0360₁₆, 0364₁₆) under the status that the CLKi pin is input to "H" level. Also please write to the bit 7(SOUTi default value setting bit) under the status that the CLKi pin is input to "H" level.
- The S I/Oi circuit keeps on with the shift operation as long as the synchronous clock is entered in it, so stop the synchronous clock at the instant when it counts to eight. The internal clock, if selected, automatically stops.

Note 3: If the internal clock is used for the synchronous clock, the transfer clock signal stops at the "H" state.

■ Functions for setting an SOUTi initial value

In carrying out transmission, the output level of the SOUTi pin as it is before transmitting 1-bit data can be set either to "H" or to "L". Fig.GA-19 shows the timing chart for setting an SOUTi initial value and how to set it.

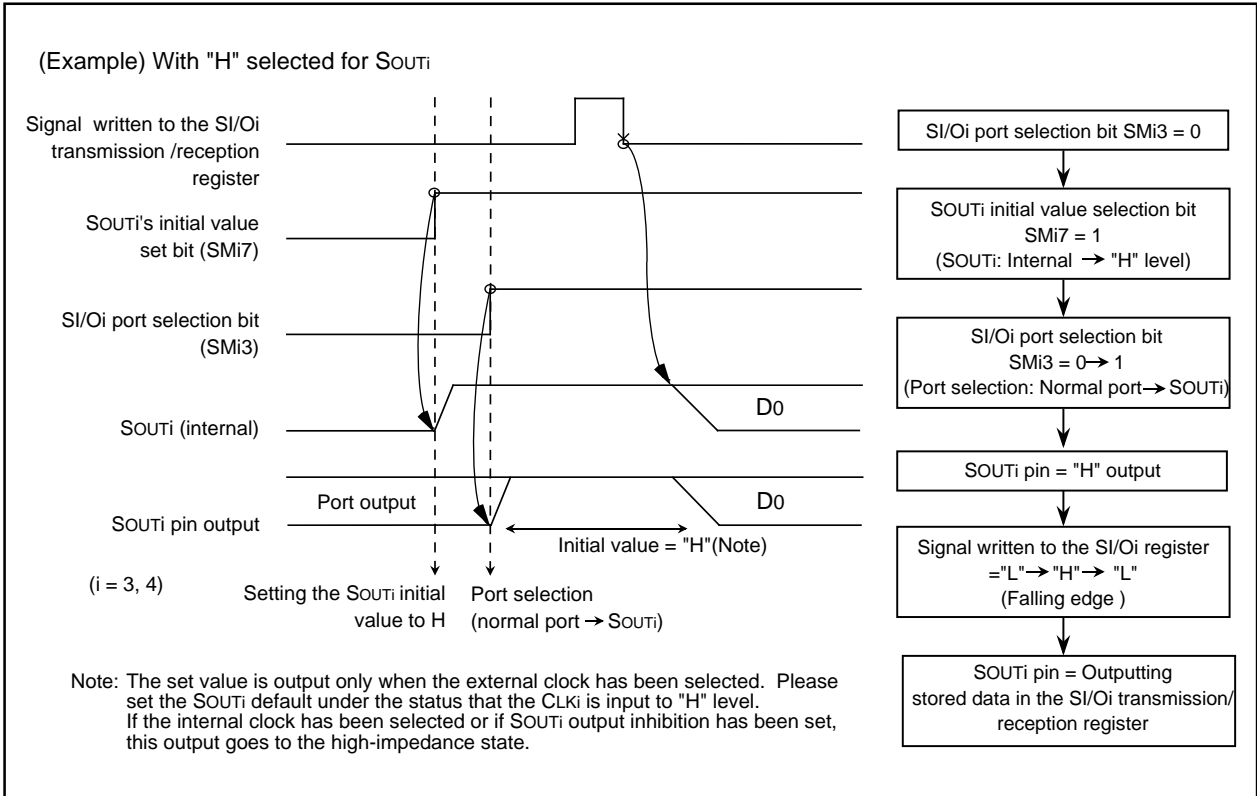


Fig.GA-19 Timing chart for setting SOUTi's initial value and how to set it

■ S I/Oi operation timing

Fig.GA-20 shows the S I/Oi operation timing

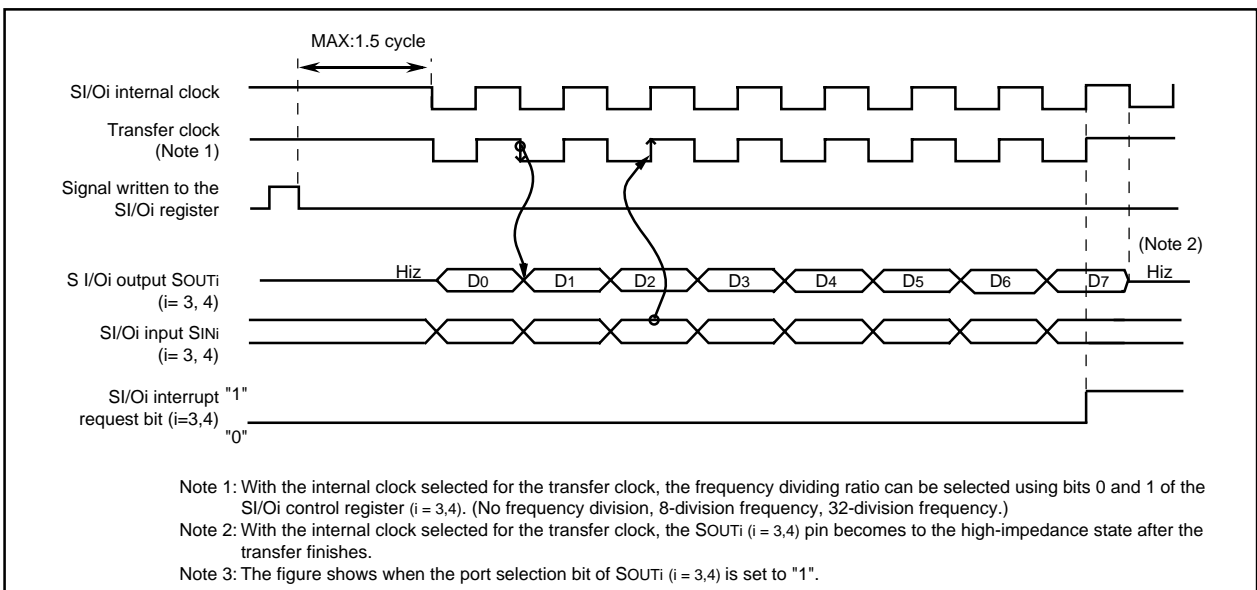


Fig.GA-20 S I/Oi operation timing chart

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P100 to P107, P95, and P96 also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 03D716) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of 03D716 to connect VREF.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table.JA-1 shows the performance of the A-D converter. Fig.JA-1 shows the block diagram of the A-D converter, and Fig.JA-2 and JA-3 show the A-D converter-related registers.

Table.JA-1 Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage	0V to AVCC (VCC)
Operating clock ϕ_{AD} (Note1)	fAD/divide-by-2 of fAD/divide-by-4 of fAD, fAD=f(XIN)
Resolution	8-bit or 10-bit (selectable)
Absolute precision	<ul style="list-style-type: none"> • 8-bit resolution ±2LSB • 10-bit resolution ±6LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8pins (AN0 to AN7) + 2pins (ANEX0 and ANEX1)
A-D conversion start condition	<ul style="list-style-type: none"> • Software trigger A-D conversion starts when the A-D conversion start flag changes to "1" • External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{ADTRG}/P97$ input changes from "H" to "L"
Conversion speed per pin	<ul style="list-style-type: none"> • Without sample and hold function 8-bit resolution : 49 ϕ_{AD} cycles 10-bit resolution : 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution : 28 ϕ_{AD} cycles 10-bit resolution : 33 ϕ_{AD} cycles

Note 1: The frequency ϕ_{AD} should be set to 250kHz min and 8MHz max.

Without sample and hold function, set the ϕ_{AD} frequency to 250kHz min.

With the sample and hold function, set the ϕ_{AD} frequency to 1MHz min.

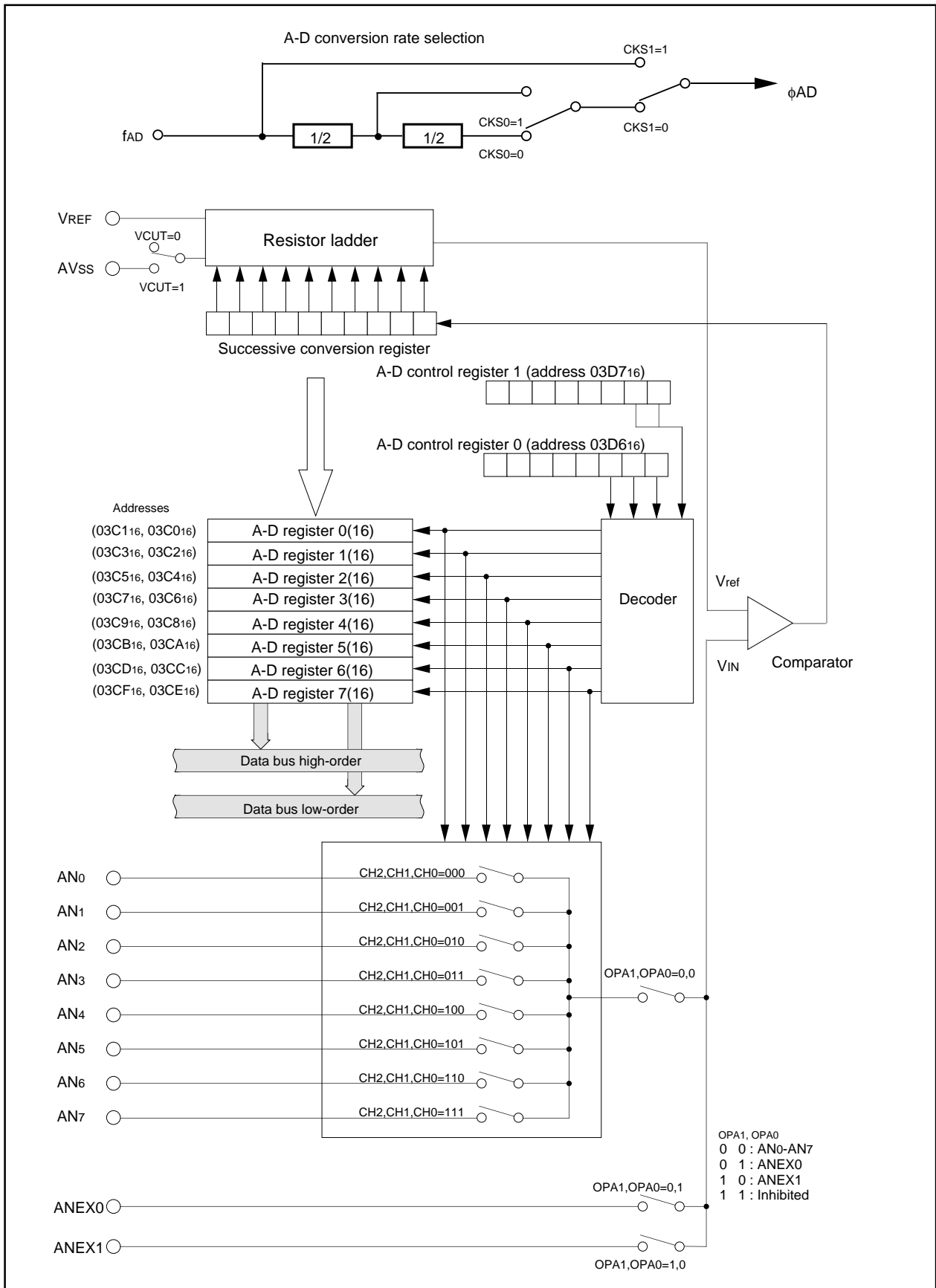


Fig.JA-1 Block diagram of A-D converter

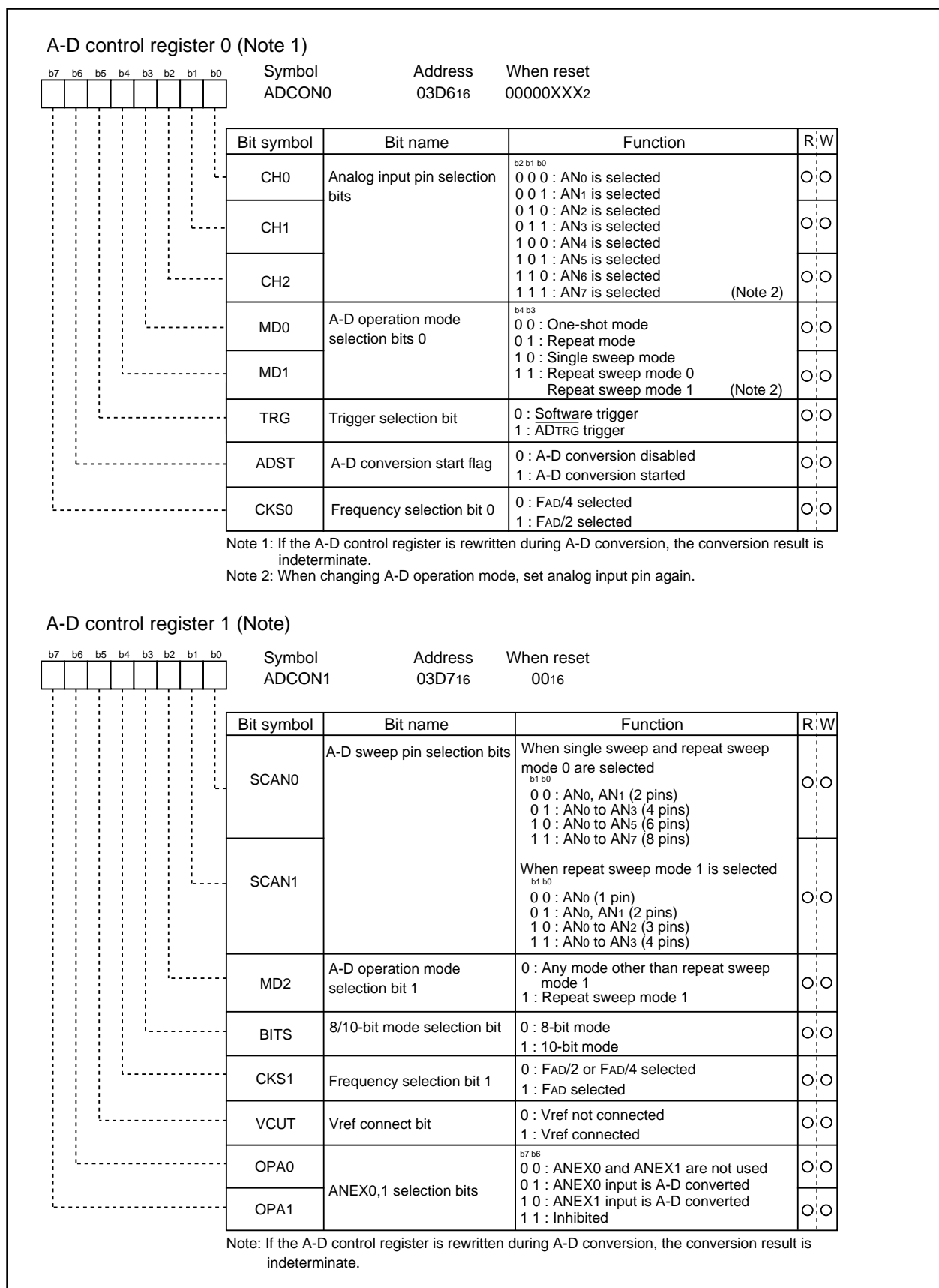


Fig.JA-2 A-D converter-related registers (1)

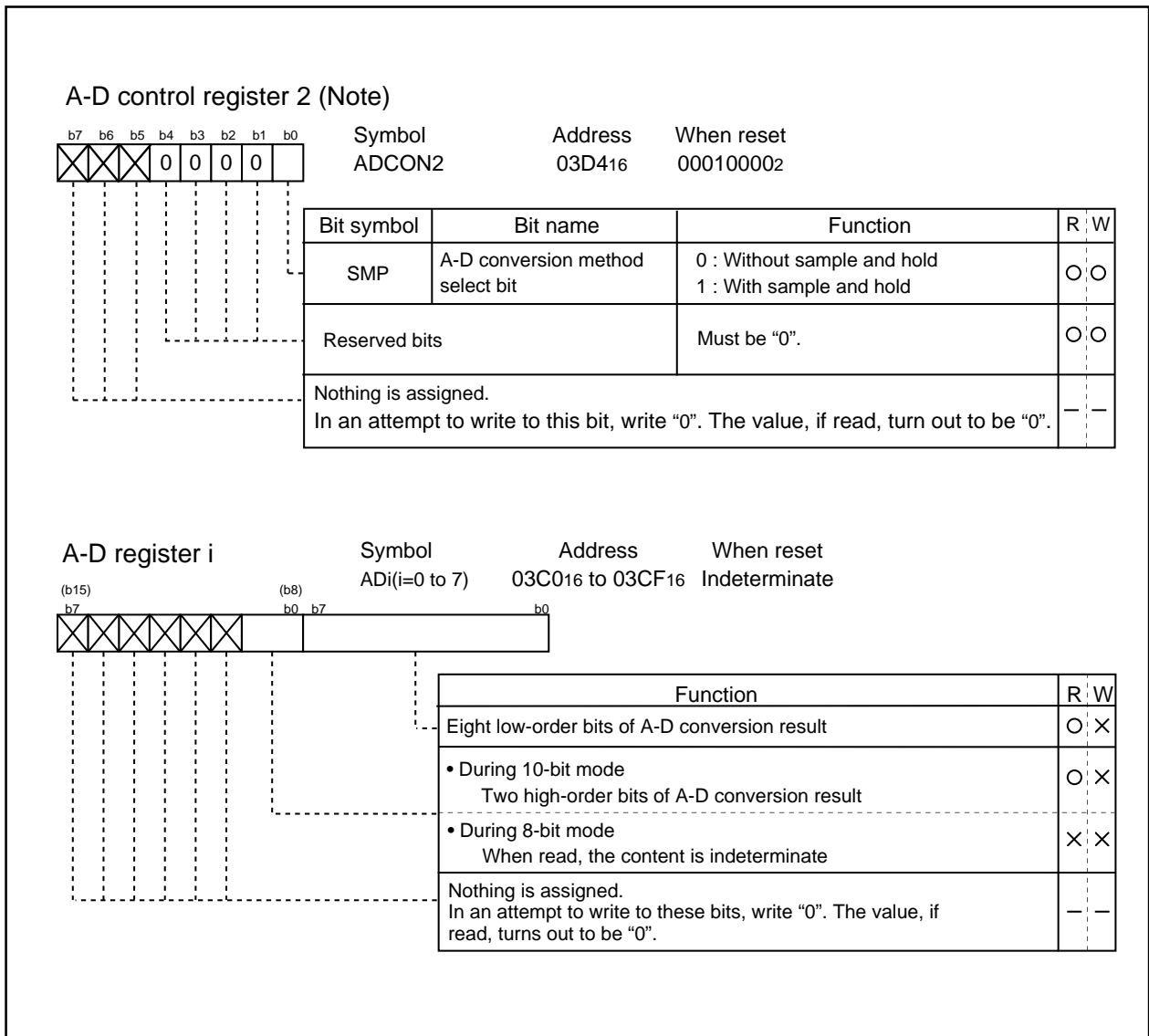


Fig.JA-3 A-D converter-related registers (2)

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin selection bits is used for one-shot A-D conversion. Table.JA-2 shows the specifications of one-shot mode. Fig.JA-4 shows the A-D control register in one-shot mode.

Table.JA-2 One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin selection bits is used for one A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	<ul style="list-style-type: none"> End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected) Writing "0" to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN ₀ to AN ₇ , as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

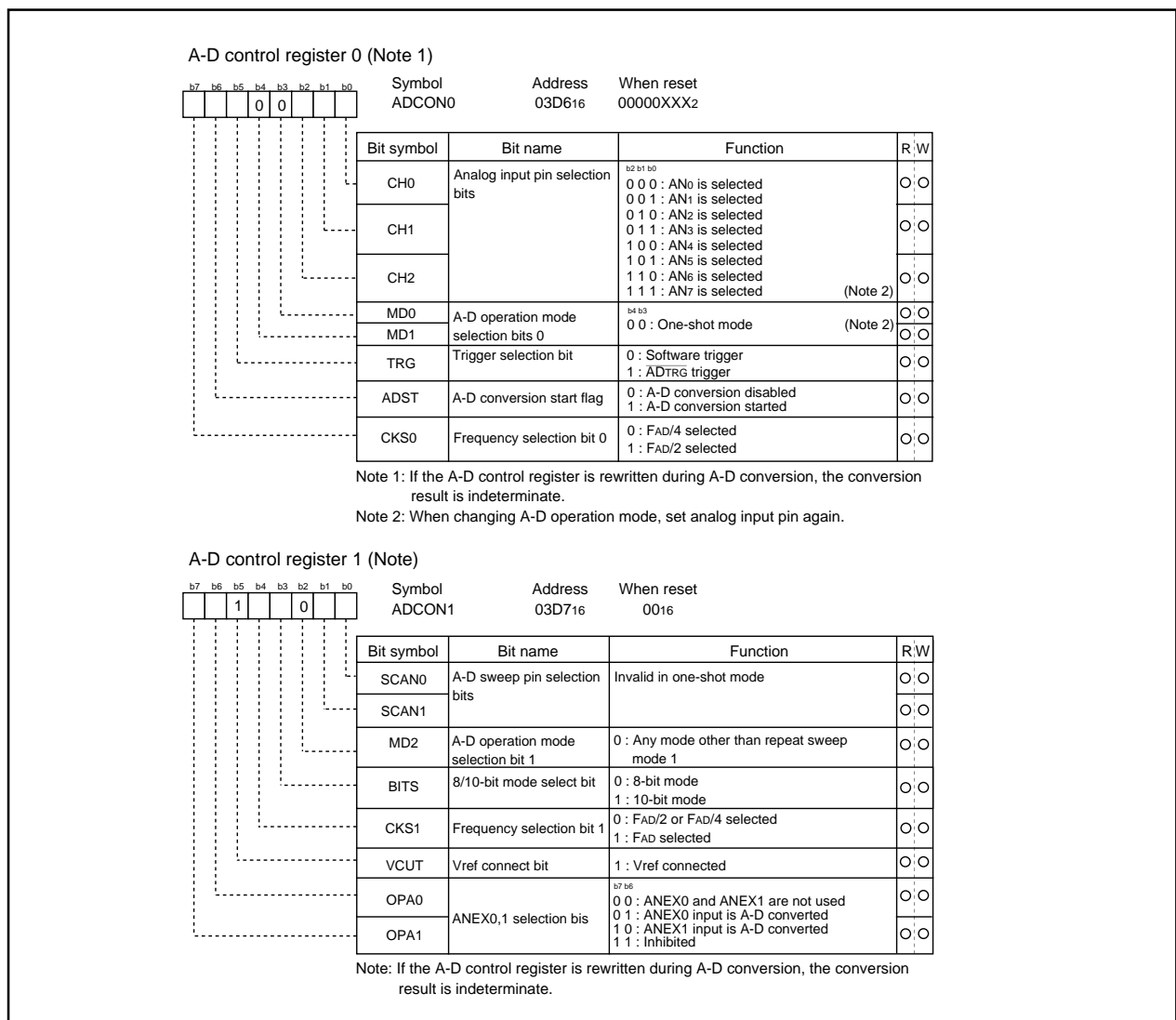


Fig.JA-4 A-D conversion register in one-shot mode

(2) Repeat mode

In repeat mode, the pin selected using the analog input pin selection bits is used for repeated A-D conversion. Table.JA-3 shows the specifications of repeat mode. Fig.JA-5 shows the A-D control register in repeat mode.

Table.JA-3 Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin selection bits is used for repeated A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	Not generated
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

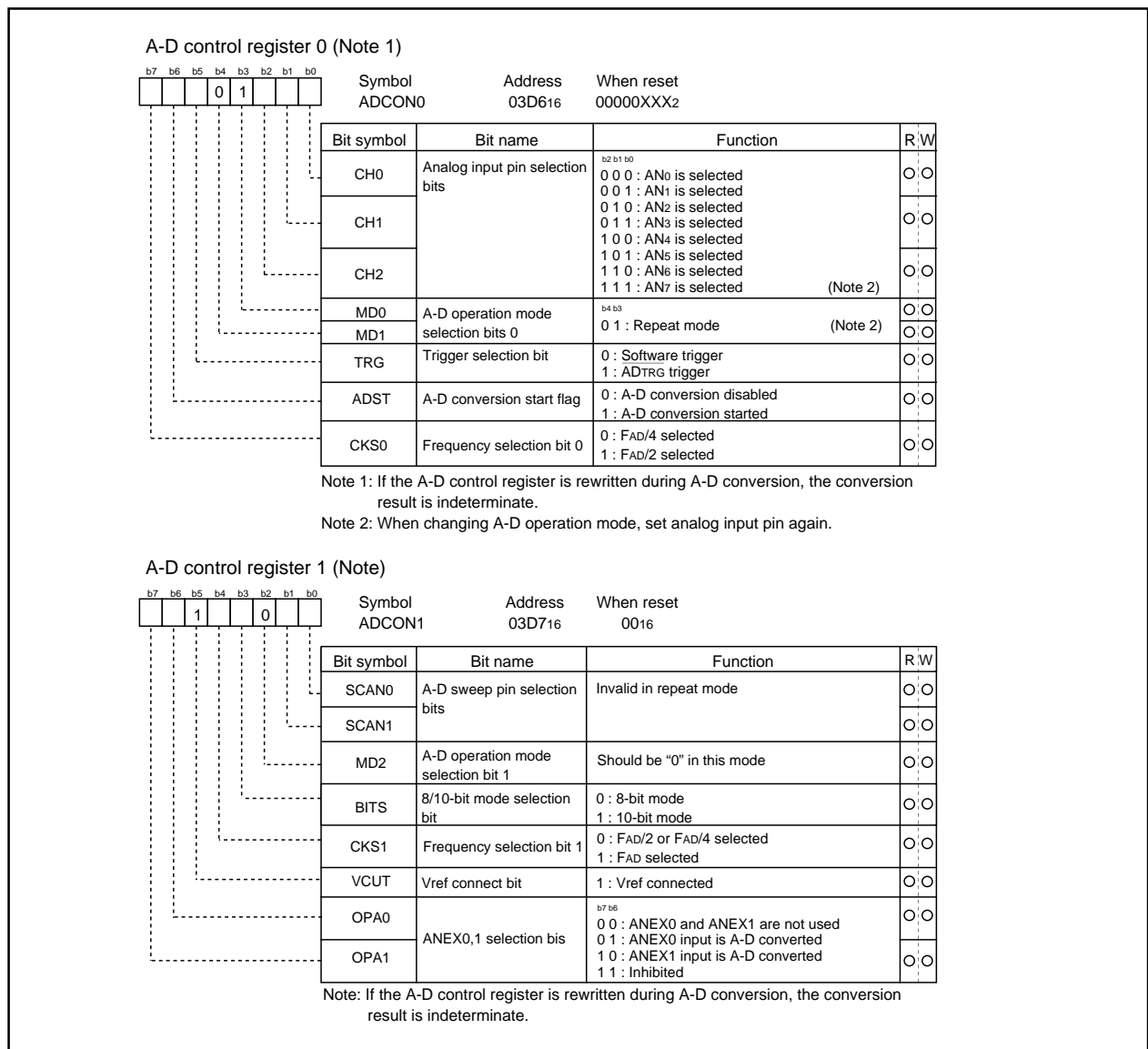


Fig.JA-5 A-D conversion register in repeat mode

(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin selection bits are used for one-by-one A-D conversion. Table.JA-4 shows the specifications of single sweep mode. Fig.JA-6 shows the A-D control register in single sweep mode.

Table.JA-4 Single sweep mode specifications

Item	Specification
Function	The pins selected by the A-D sweep pin selection bits are used for one-by-one A-D conversion
Start condition	Writing "1" to A-D converter start flag
Stop condition	<ul style="list-style-type: none"> End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected) Writing "0" to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN ₀ and AN ₁ (2 pins), AN ₀ to AN ₃ (4 pins), AN ₀ to AN ₅ (6 pins), or AN ₀ to AN ₇ (8 pins)
Reading of result of A-D converter	Read A-D registers corresponding to selected pins

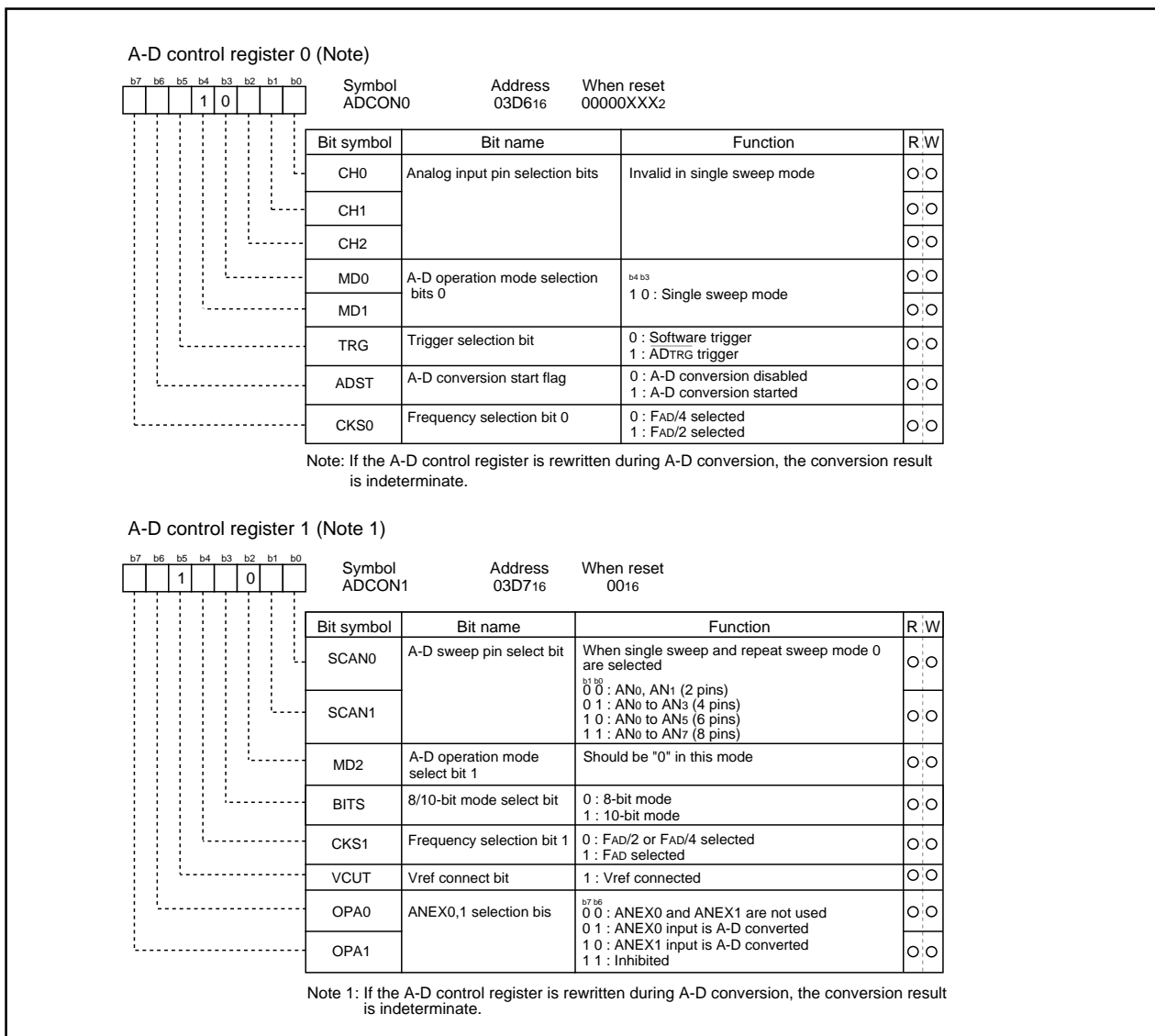


Fig.JA-6 A-D conversion register in single sweep mode

(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin selection bits are used for repeat sweep A-D conversion. Table.JA-5 shows the specifications of repeat sweep mode 0. Fig.JA-7 shows the A-D control register in repeat sweep mode 0.

Table.JA-5 Repeat sweep mode 0 specifications

Item	Specification
Function	The pins selected by the A-D sweep pin selection bits are used for repeat sweep A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	Not generated
Input pin	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins)
Reading of result of A-D converter	Read A-D registers corresponding to selected pins (at any time)

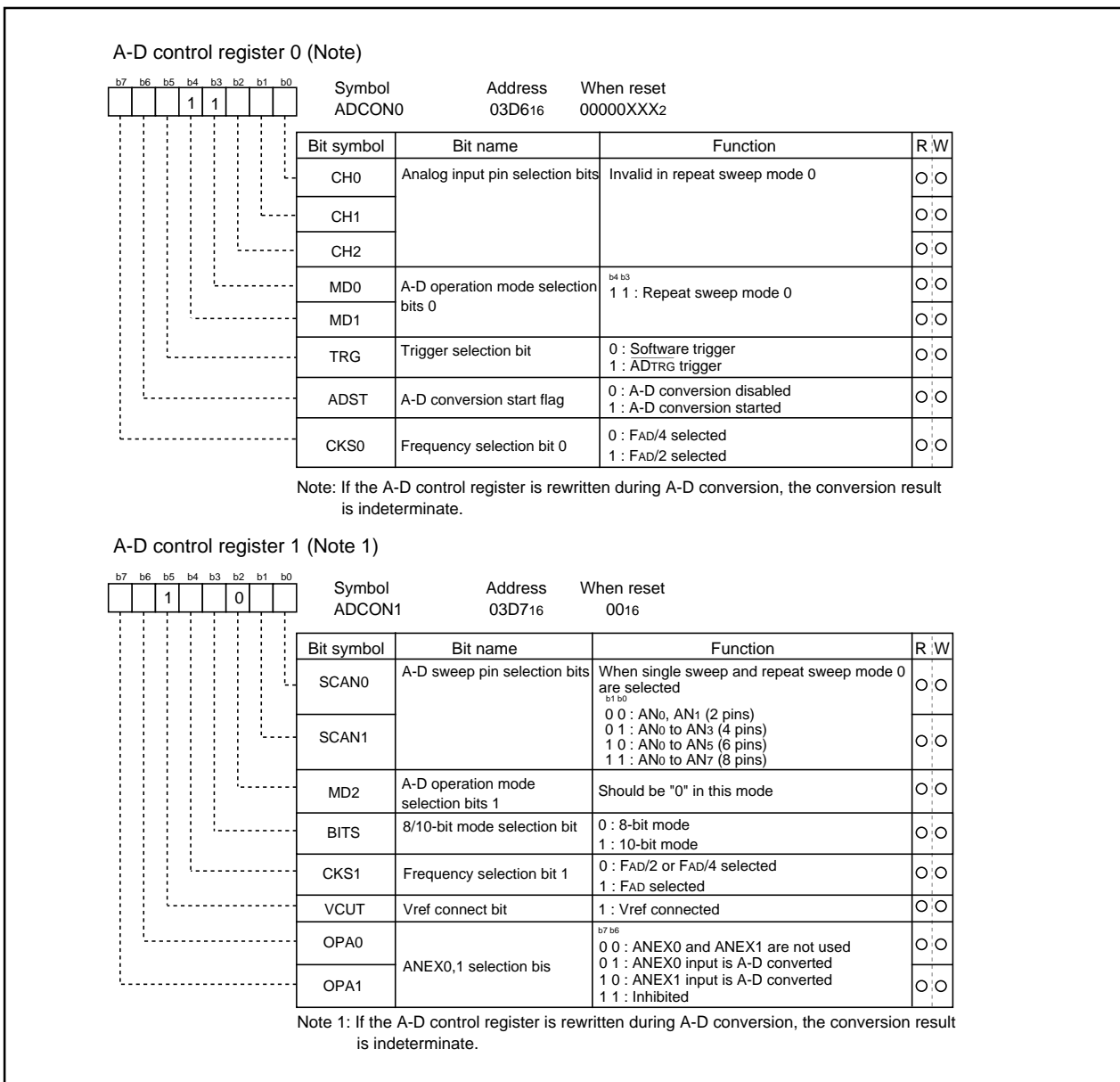


Fig.JA-7 A-D conversion register in repeat sweep mode 0

(5) Repeat sweep mode 1

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin selection bits. Table.JA-6 shows the specifications of repeat sweep mode 1. Fig.JA-8 shows the A-D control register in repeat sweep mode 1.

Table.JA-6 Repeat sweep mode 1 specifications

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin selection bits Example : AN ₀ selected AN ₀ → AN ₁ → AN ₀ → AN ₂ → AN ₀ → AN ₃ , etc
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	Not generated
Input pin	AN ₀ (1 pin), AN ₀ and AN ₁ (2 pins), AN ₀ to AN ₂ (3 pins), AN ₀ to AN ₃ (4 pins)
Reading of result of A-D converter	Read A-D registers corresponding to selected pins (at any time)

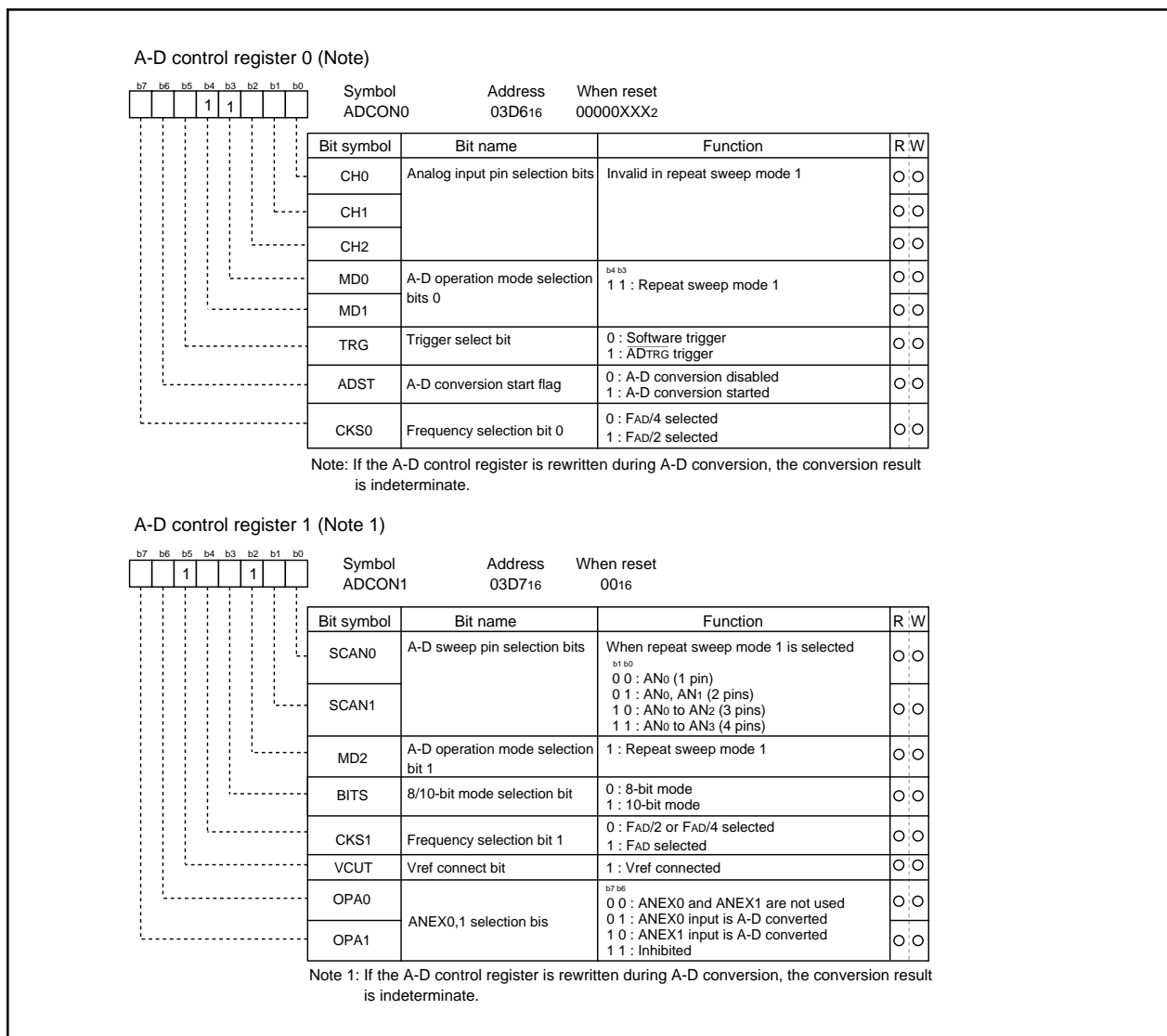


Fig.JA-8 A-D conversion register in repeat sweep mode 1

(a) Extended analog input pins

In one-shot mode and repeat mode, the input via the extended analog input pins ANEX0 and ANEX1 can also be converted from analog to digital.

When bit 6 of the A-D control register 1 (address 03D716) is "1" and bit 7 is "0", input via ANEX0 is converted from analog to digital. The result of conversion is stored in A-D register 0.

When bit 6 of the A-D control register 1 (address 03D716) is "0" and bit 7 is "1", input via ANEX1 is converted from analog to digital. The result of conversion is stored in A-D register 1.

(b) Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D416) to "1". When sample and hold is selected, the range of conversion of each pin increases. As a result, a $28\phi_{AD}$ cycle is achieved with 8-bit resolution and $33\phi_{AD}$ cycle with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

PWM output circuit (PWM : Pulse Width Modulation)

There are 6 PWM output circuits, PWM0 to PWM5, with 8-bit resolution and operating independently. The input clock $f(PC)$ for PWM is based on X_{IN} , 2 division of X_{IN} or 4 division of X_{IN} .

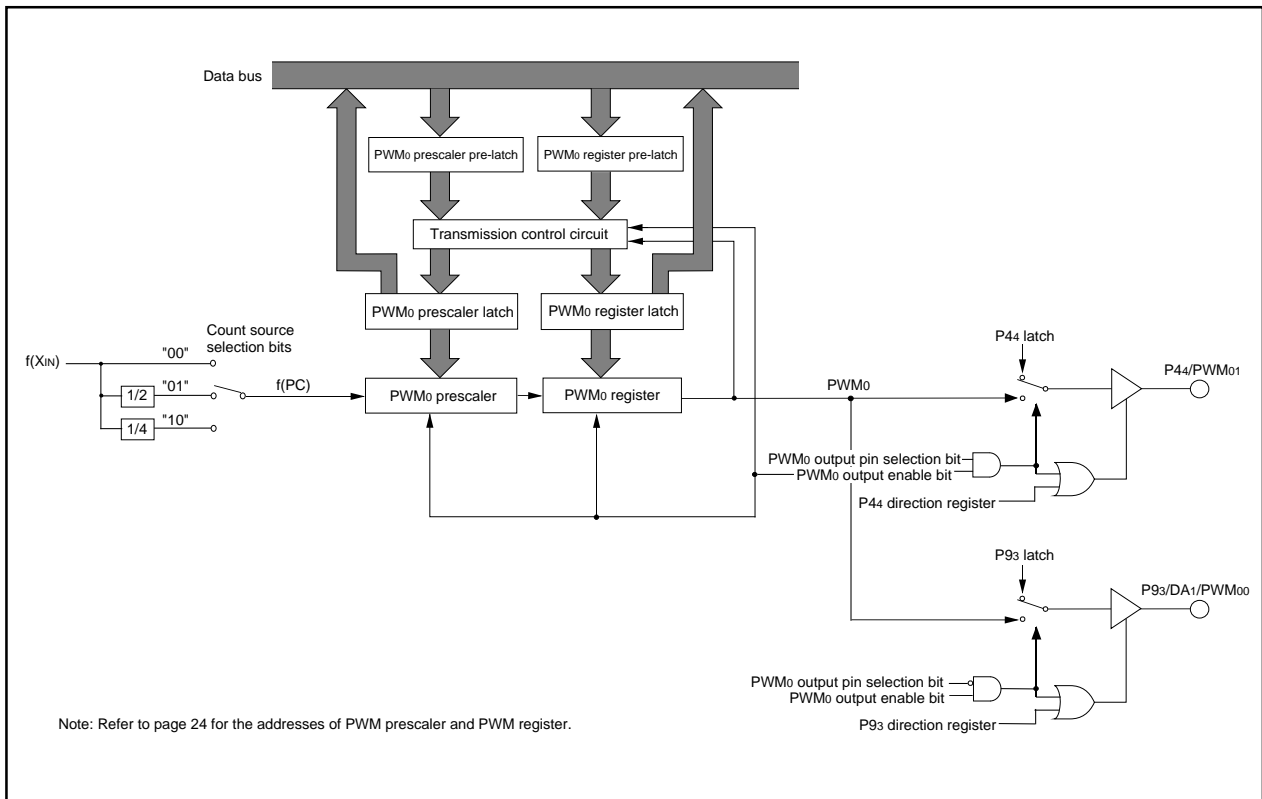


Fig.LA-1 PWM circuit (PWM0)

Data Setup (PWM0)

PWM0 output pin shares with P93 or P44. PWM0 output pin is selected from either P93/PWM00 or P44/PWM01 by bit 0 of PWM control register 0 (address 030C16). PWM0 output is enabled and starts to operate by setting bit 0 of PWM control register 1 (address 030D16) to "1".

The period of PWM is set by PWM0 prescaler (address 030016), The "H" width of output pulse is set by PWM0 register (address 030116).

The following are the calculations if the prescaler value is n and PWM0 register value is m.

(n = 0 to 255, m = 0 to 255)

$$\text{PWM period} = \frac{255 \times (n+1)}{f(X_{IN})} = 31.875 \times (n+1) \mu\text{s}$$

(In the case of $f(X_{IN})=8\text{MHz}$, PWM counter source selection bits="002")

$$\text{"H" width of output pulse} = \frac{\text{PWM period} \times m}{255} = 0.125 \times (n+1) \times m \mu\text{s}$$

(In the case of $f(X_{IN})=8\text{MHz}$, PWM counter source selection bits="002")

The setting of PWM1 to PWM5 are the same.

PWM Operation

By setting bit 0 (PWM0 output enable bit) of PWM control register 1 to "1", the PWM output circuit starts to operate from default state with "H" pulse output.

If the values of PWM0 register and PWM0 prescaler are modified during the PWM output operation, the corresponding pulse will be output from the next period after the modification.

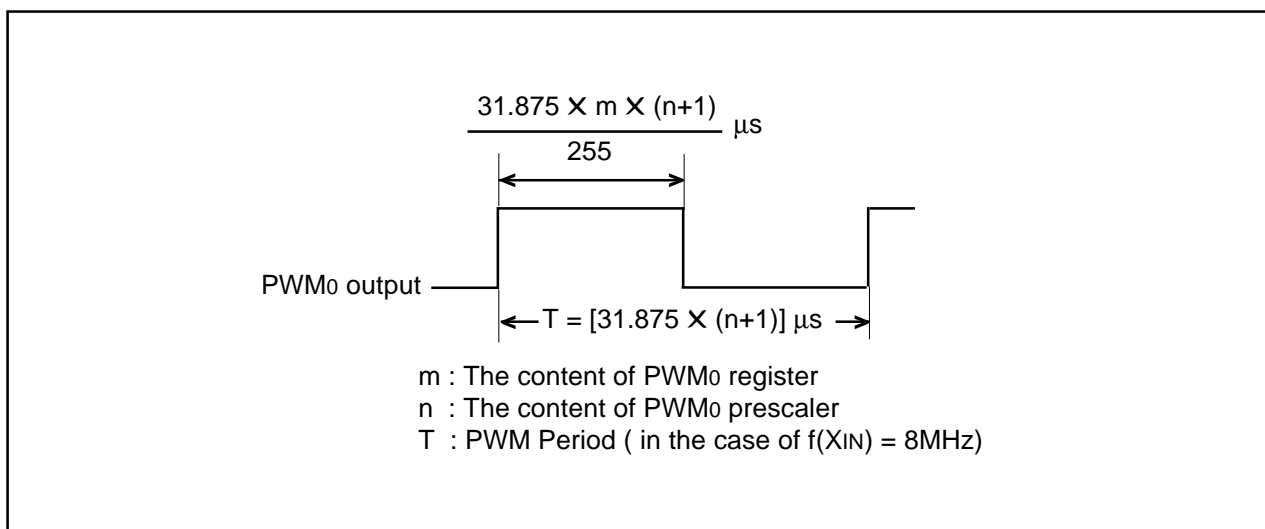


Fig.LA-2 The timing of PWM period (PWM0)

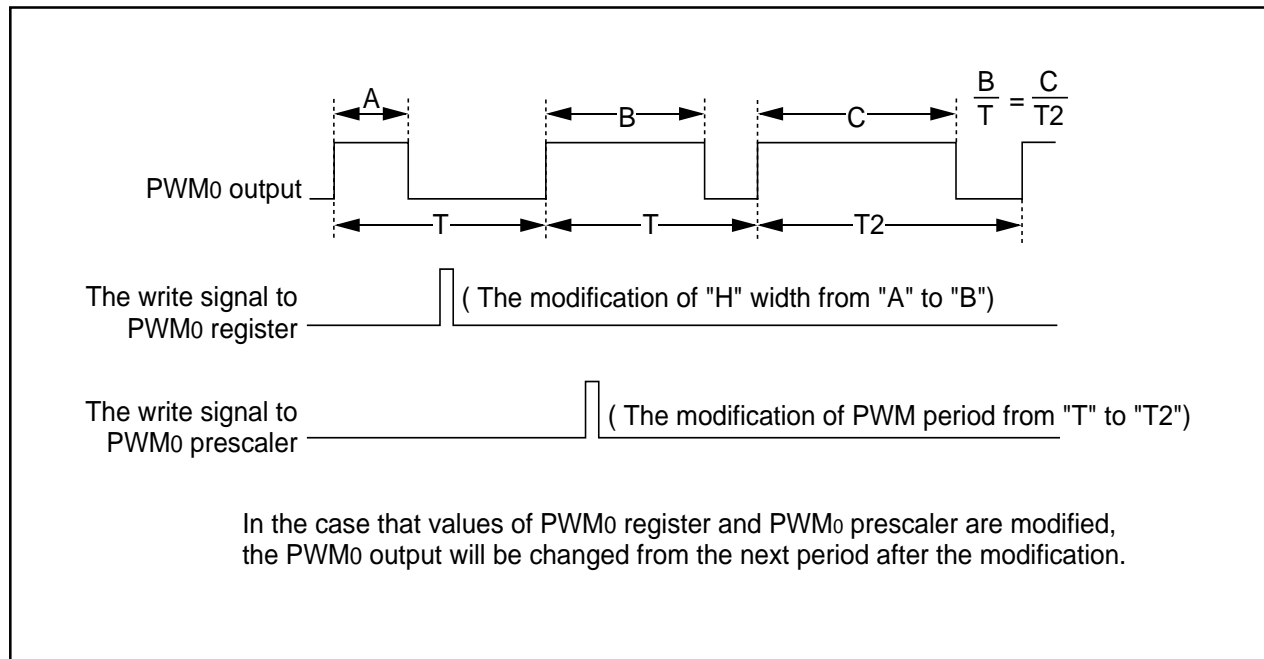


Fig.LA-3 The PWM output timing in the case of modification of PWM register and PWM prescaler (PWM0)

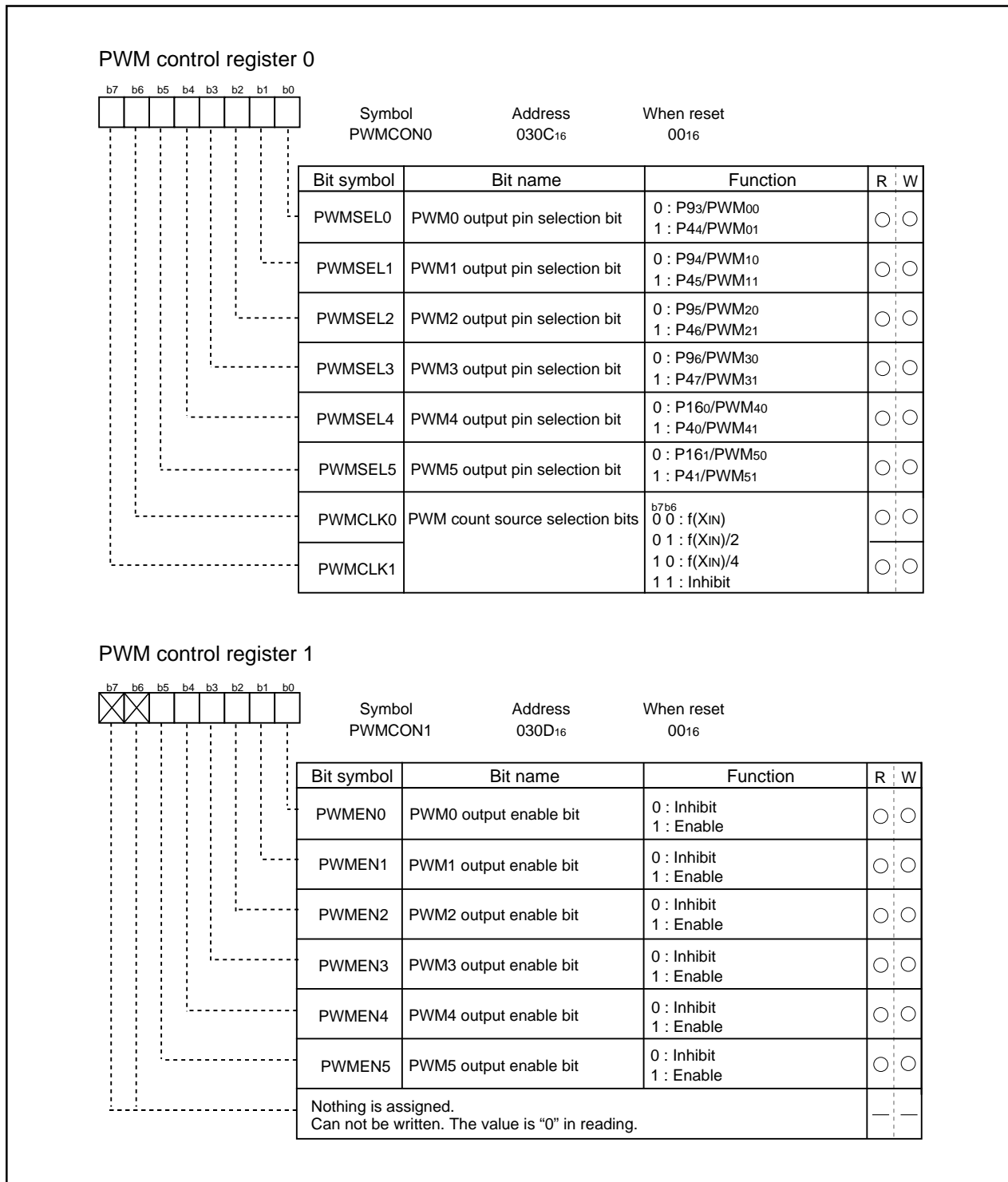


Fig.LA-4 PWM control registers

LPC Bus Interface

LPC bus interface is based on Intel Low Pin Count (LPC) Interface Specification, Revision 1.0. It is I/O cycle data transfer format of serial communication. 4 channels are built in. The function of data bus buffer and data bus buffer status are almost the same as that of MELPS8-41 series. It can be written in or read out (as slave mode) by the control signals from host CPU side. The LPC bus interface functionality block diagram is shown in Figure GF-2. LPC data bus buffer functional Input / Output ports (P30-P36) are shared with GPIO port.

The setting of bit3 (LPC bus buffer enable bit) of LPC control register (address 02D6₁₆) is as below:

0: General purpose Input / Output port

1: LPC bus buffer functional Input / Output port

The enabling of channel of LPC bus buffer is controlled by bits 4-7 (LPC bus buffer 0-3 enable bits) of LPC control register (address 02D6₁₆). The slave address (16 bits) of LPC bus buffer channel 0 is fixed on 0060h, 0064h. The slave addresses (16 bits) of LPC bus buffer channel 1-3 are definable by setting LPC 1-3 address register H, L (address 02D0₁₆ to 02D5₁₆). The setting value of bit2 of LPC1-3 address register (A2) L will not be decoded. The bit is "0" when read from slave CPU. The A2 status of slave address is latched to XA2 flag when written by host CPU. The input buffer full interrupt is generated when written in the data by host CPU. The Output buffer empty interrupt is generated when read out the data by host CPU.

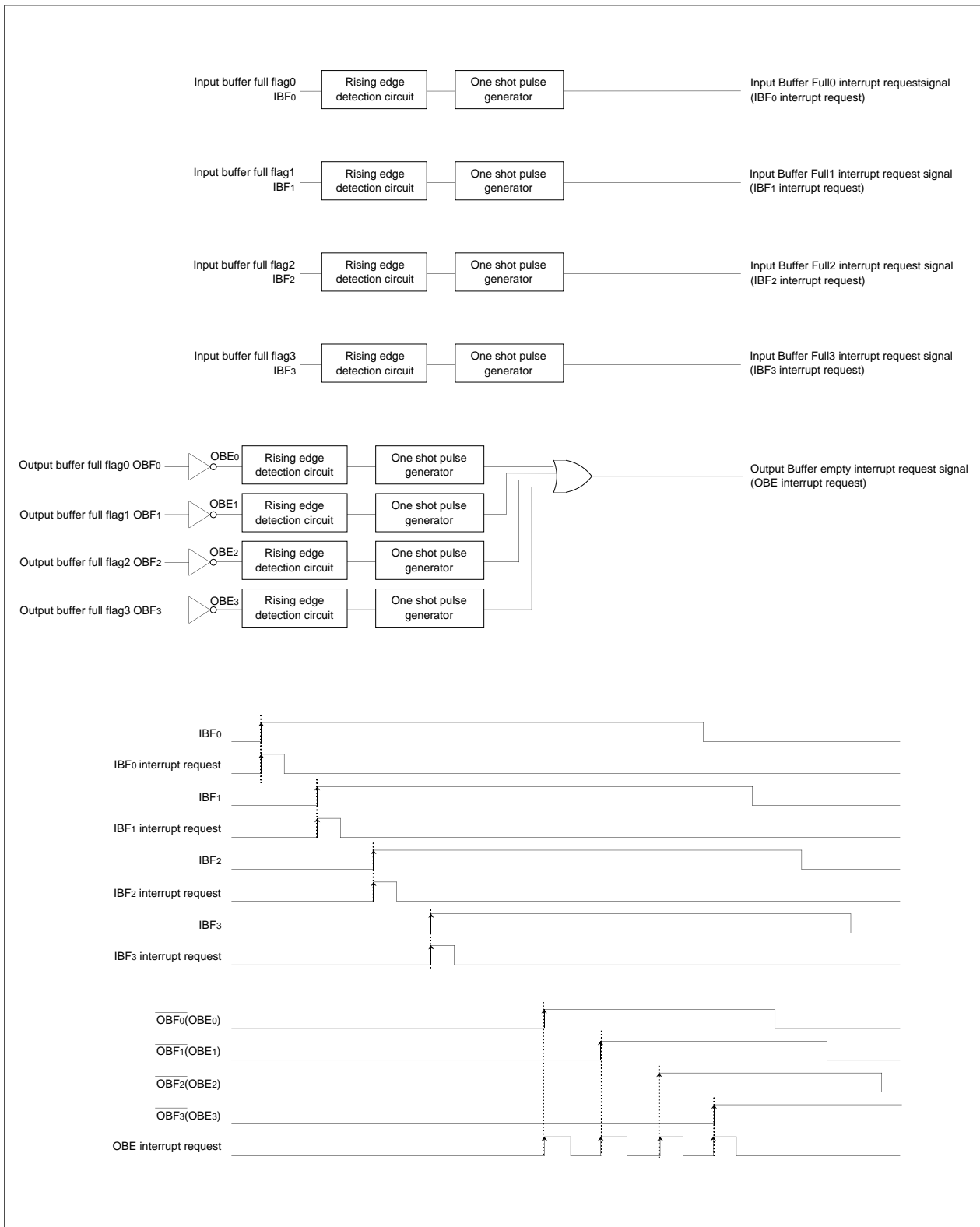


Fig.GF-1 Interrupt, request, circuit of Data Bus Buffer

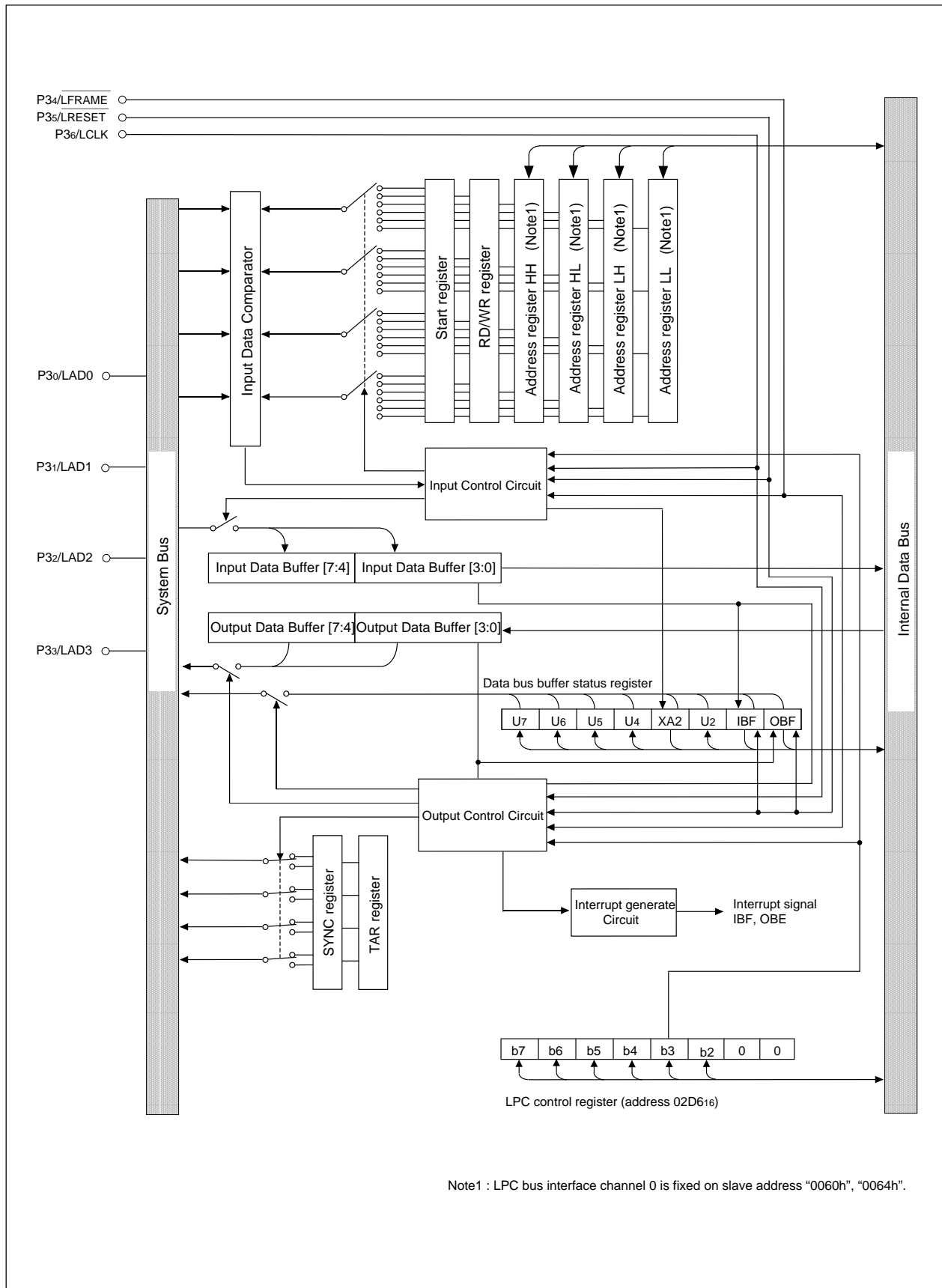


Fig.GF-2 LPC bus interface function block diagram (LPC1)

Figure GF-3: Data bus buffer control registers

Figure GF-4: Data bus buffer status register

Figure GF-5, 6: LPC related registers

Data bus buffer status register (DBBSTS0-DBBSTS3)

This is 8-bit register.

The bit 0, 1, 3 are read only bits and indicate the status of data bus buffer.

Bit 2, 4, 5, 6, 7 are user definable and flags which can be read and written by software. The data bus buffer status register can be read out by host CPU when the slave address (16 bit) bit2 (A2) is high.

- Output buffer full flag (OBF)

The bit will be set to "1" when a data is written into output data bus buffer and will be cleared to "0" when host CPU read out the data from output data bus buffer.

- Input buffer full flag (IBF)

The bit will be set to "1" while a data is written into input data bus buffer by host CPU and will be cleared to "0" when the data is read out from input data bus buffer by slave CPU.

- XA2 flag (XA2)

The bit 2 of slave address (16 bits) is latched while a data is written into data bus buffer.

Input data bus buffer register (DBBIN0-DBBIN3)

When there is a write request from host CPU, the data on the data bus will be latched to DBBIN0-3. The data of DBBIN0-3 can be read out from data bus buffer registers (Address: 02C0₁₆, 02C2₁₆, 02C4₁₆, 02C6₁₆) in SFR field.

Output data bus buffer register (DBBOUT0-DBBOUT3)

When writing data to data bus buffer registers (Address: 02C0₁₆, 02C2₁₆, 02C4₁₆, 02C6₁₆), the data will be transferred to DBBOUT0-3 automatically. The data of DBBOUT0-3 will be output to the data bus when there is a read request from host CPU and the status of bit2 (A2) of slave address (16 bits) is low.

LPCi address register H/L (LPC1ADH-LPC3ADH / LPC1ADL-LPC3ADL)

The slave address (16 bits) of LPC bus buffer channel 0 is fixed on 0060h, 0064h.

The slave addresses (16 bits) of LPC bus buffer channel 1-3 are definable by setting LPC1-3 address registers H/L (02D0₁₆ to 02D5₁₆). The settings are for slave address upper 8 bits and lower 8 bits. And these registers can be set and cleared in any time.

The bit 2 of LPC 1-3 address L is not decoded regardless of the setting value. When slave CPU reads LPC1-3 address registers, the bit2 (A2) of address low byte will be fixed to "0". The bit2 (A2) status of slave address is latched to XA2 flag when written by host CPU. The slave addresses that are already set in these registers will be used for comparing with the addresses to be received.

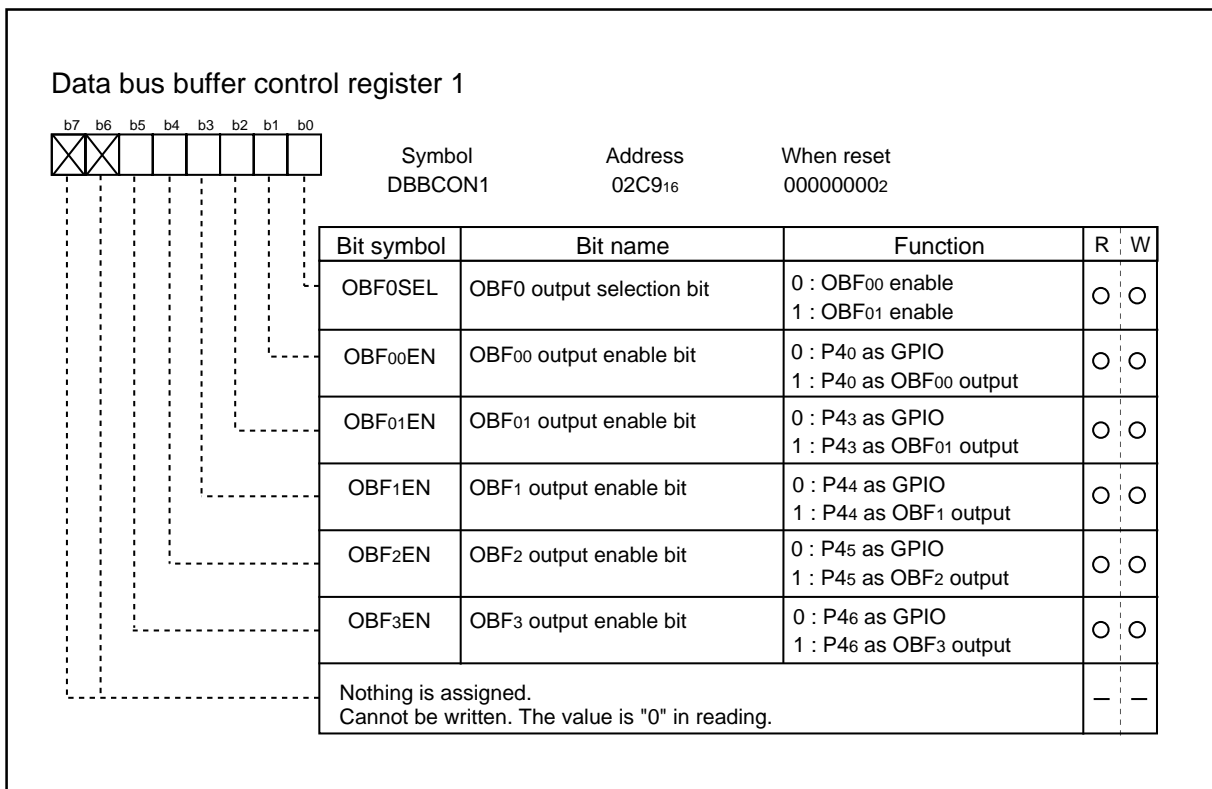


Fig.GF-3 Data bus buffer control registers

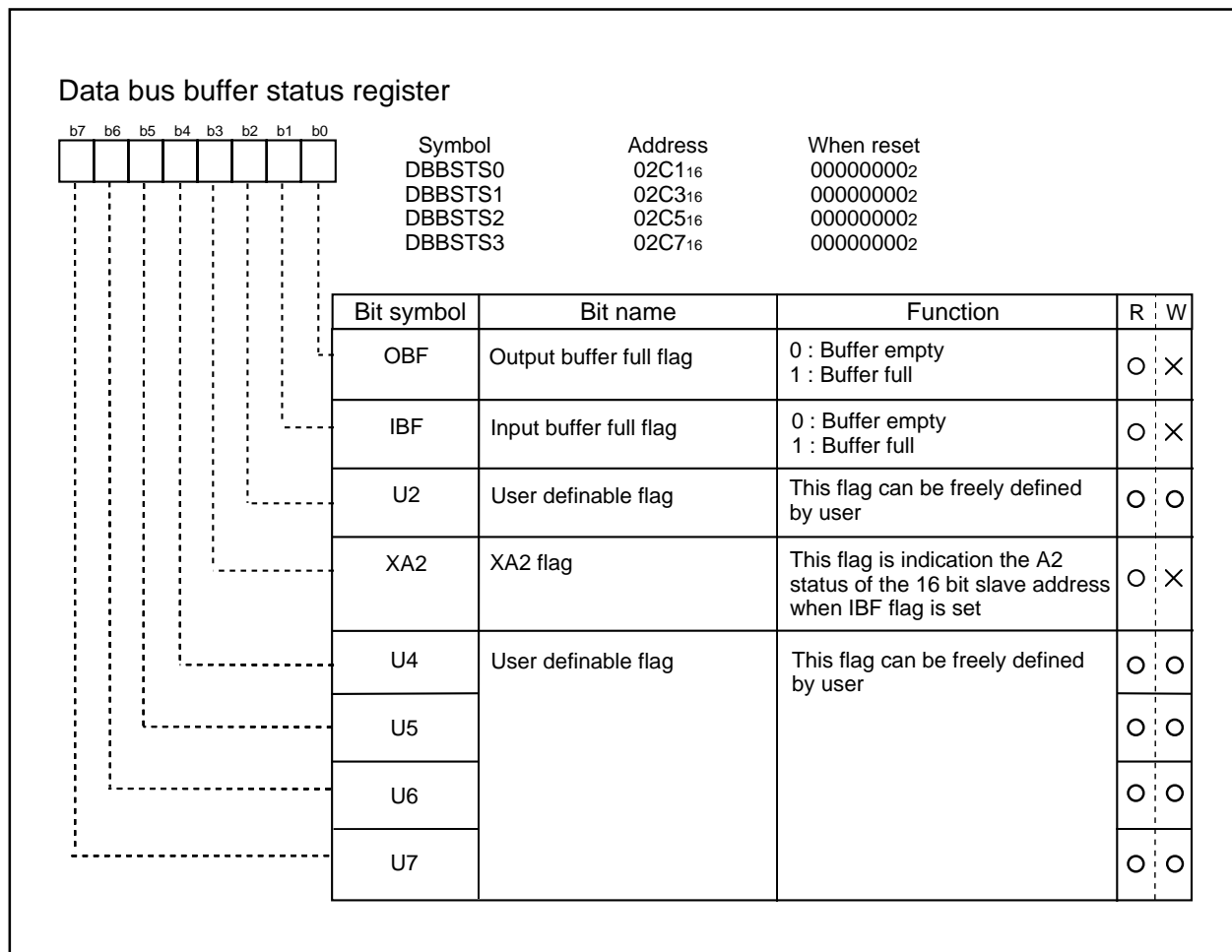


Fig.GF-4 Data bus buffer status register

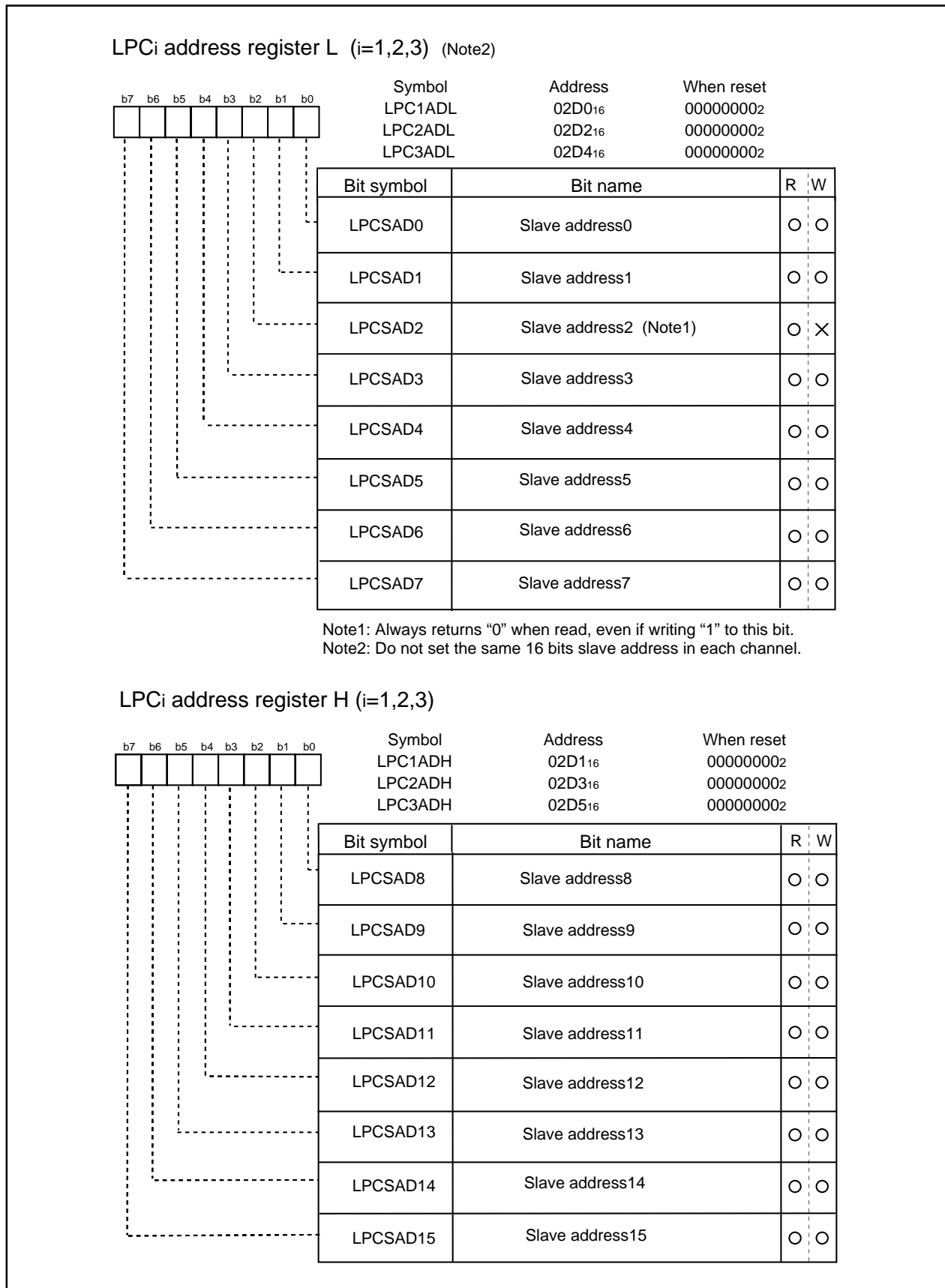


Fig.GF-5 LPC related registers

LPC control register (LPCCON)

- LPC bus interface enable bit (LPCBEN)
 - "0": P30 -P36 use as GPIO
 - "1": P30 -P36 use as LPC bus interface
- LPC bus buffer 0 enable bit (LPCEN0)
 - "0": LPC bus buffer0 disable
 - "1": LPC bus buffer0 enable
- LPC bus buffer 1 enable bit (LPCEN1)
 - "0": LPC bus buffer1 disable
 - "1": LPC bus buffer1 enable
- LPC bus buffer 2 enable bit (LPCEN2)
 - "0": LPC bus buffer2 disable
 - "1": LPC bus buffer2 enable
- LPC bus buffer 3 enable bit (LPCEN3)
 - "0": LPC bus buffer3 disable
 - "1": LPC bus buffer3 enable

- LPC software reset bit (LPCSR)

By setting the bit to "1", LPC interface is reset by the same status as $\overline{\text{LRESET}}=\text{"L"}$. After 1.5 cycles of BCLK at writing "1", reset is released and the bit becomes "0".

Nothing happens if "0" is set.

- SYNC output selection bits (SYNCSEL0, SYNCSEL1)

The content of SYNC output can be selected by bit 0,1 (SYNC output selection bits) of LPC control register.

Fig.GF-6 shows the configuration of LPC control register, Table.GF-1 shows the content of SYNC output selected by SYNC output selection bits.

Table GF-1 SYNC output

SYNCSEL1	SYNCSEL0	SYNC cycle	SYNC output			
			1st cycle	2nd cycle	3rd cycle	4th cycle
0	0	1	00002	—	—	—
0	1	4	01102	01102	01102	00002
1	0	1	10102	—	—	—
1	1	4	01102	01102	01102	10102

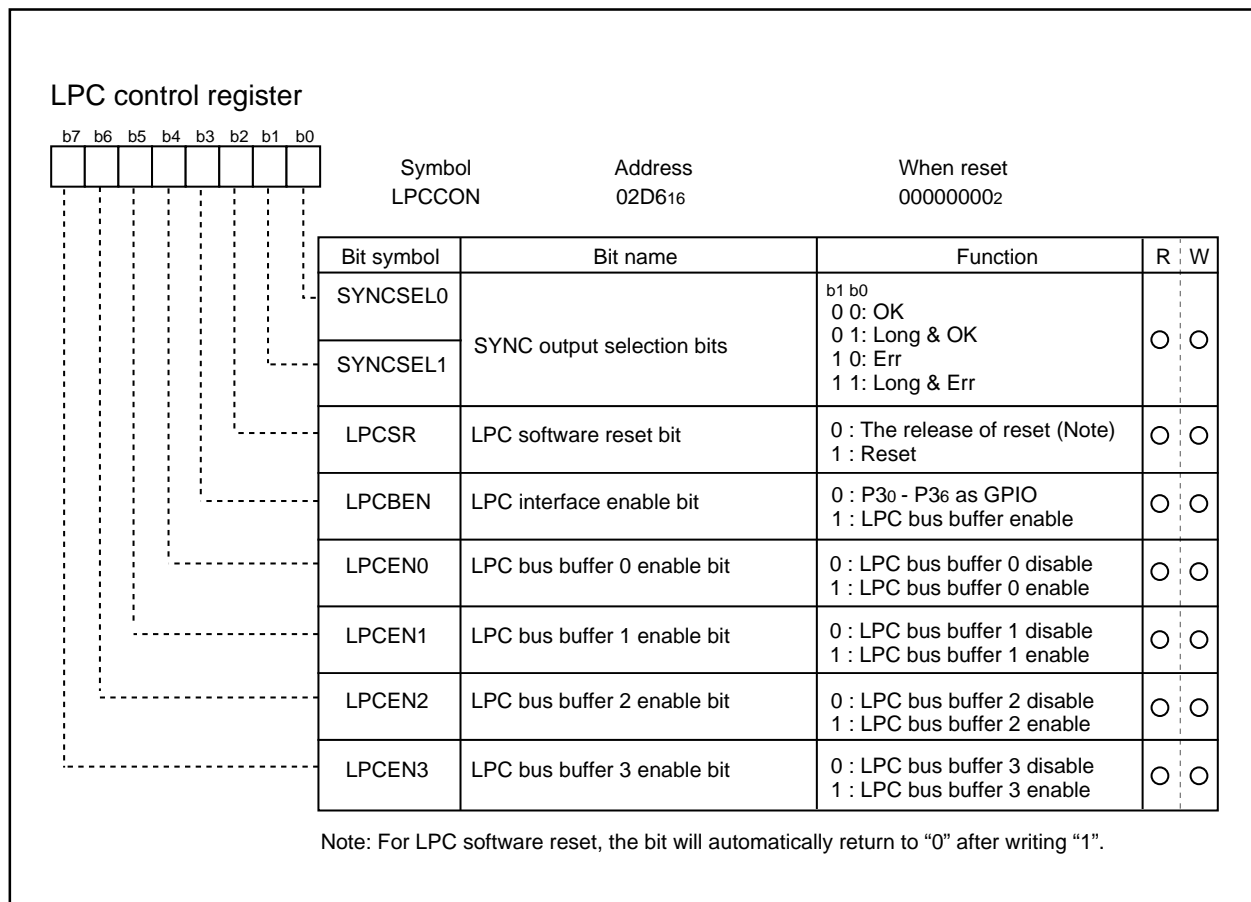


Fig.GF-6 LPC control register

Basic operation of LPC bus interface

The status transition of LPC bus interface is shown in Figure GF-7.

Setting steps for using LPC bus interface is explained below.

- Setting bit3 (LPC interface enable bit) of LPC control register (address 02D6₁₆) to "1"
- Choosing which LPC bus buffer channel will be used
- Setting "1" to bits 4-7 (LPC bus buffer 0-3 enable bit) of LPC control register (address 02D6₁₆).
- The 16-bit slave address of LPC bus buffer channel is defined by writing 16-bit slave address to LPC 1-3 address registers (address 02D0₁₆ to 02D5₁₆). If channel 1-3 LPC bus buffer is chosen, set the address to the corresponding address register.
- Selecting IBF/ OBE interrupt in data bus buffer control register0 (address 02C8₁₆)
- Selecting OBF output port in data bus buffer control register1 (address 02C9₁₆)

<1> Example of I/O writing cycle from HOST

Writing timing is shown in Figure GF-8.

The basic communication cycles of LPC I/O protocol are 13 cycles. The data of LAD[3:0] will be read by the rising edge of LCLK. Communication will start from $\overline{\text{LFRAME}}$ falling edge.

- 1st cycle : When $\overline{\text{LFRAME}}$ is "Low", sending "00002 " to LAD[3:0] for communication start frame detecting.
- 2nd cycle : When $\overline{\text{LFRAME}}$ is "High", sending "001X2 " to LAD[3:0] for write frame detecting.
- From 3rd cycle to 6th cycle: These four cycles are detecting for 16 bits slave address.
 - 3rd cycle: The slave address which is from host is written to slave address register [15:12] through LAD[3:0]
 - 4th cycle: The slave address which is from host is written to slave address register [11:8] through LAD[3:0]
 - 5th cycle: The slave address which is from host is written to slave address register [7:4] through LAD[3:0]
 - 6th cycle: The slave address which is from host is written to slave address register [3:0] through LAD[3:0]
- 7th and 8th cycles are used for one data byte transfer.
 - 7th cycle: The data which is from host is written to input data buffer[3:0] through LAD[3:0]
 - 8th cycle: The data which is from host is written to input data buffer[7:4] through LAD[3:0]
- 9th and 10th cycles are for changing the communication direction from host→slave to slave→host
 - 9th cycle: Host outputs "11112 " to LAD[3:0]
 - 10th cycle: The LAD[3:0] will be set to Hi-Z by HOST to switch the communication direction.
- 11th cycle: The "00002 " (SYNC OK) is output to LAD[3:0] for acknowledge.
- 12th cycle: The "11112 " is output to LAD[3:0]. The XA2 and IBF flag are set. IBF interrupt signal is generated.
- 13th cycle: The LAD[3:0] will be set to Hi-Z by slave to switch the communication direction.

During the host write period, the bit2 (A2) status of 16 bits slave address will be latched to XA2 flag. When 8 bits data from input data buffer are read out by slave CPU, the IBF flag will be cleared simultaneously.

<2> Example for I/O reading cycle from HOST

Reading timing is shown in Figure GF-9.

The basic communication cycles of LPC I/O protocol are 13 cycles. The data of LAD[3:0] will be read by the rising edge of LCLK. Communication will start from $\overline{\text{LFRAME}}$ falling edge.

- 1st cycle: When $\overline{\text{LFRAME}}$ is "Low", sending "00002 " to LAD[3:0] for communication start detecting.
- 2nd cycle: When $\overline{\text{LFRAME}}$ is "High", the host send "000X2 " on LAD[3:0] to inform the cycle type as I/O read.
- From 3rd cycle to 6th cycle: These four cycles are detecting for 16 bits slave address.
 - 3rd cycle: The slave address which is from host is written to slave address register [15:12] through LAD[3:0]
 - 4th cycle: The slave address which is from host is written to slave address register [11:8] through LAD[3:0]
 - 5th cycle: The slave address which is from host is written to slave address register [7:4] through LAD[3:0]
 - 6th cycle: The slave address which is from host is written to slave address register [3:0] through LAD[3:0]
- 7th and 8th cycles are used for changing the communication direction from host→slave to slave→host
- 7th cycle: Host is output "11112 " to LAD[3:0]
- 8th cycle: The LAD[3:0] will be set to Hi-Z by HOST to switch the communication direction.
- 9th cycle : The "00002 " (SYNC OK) is output to LAD[3:0] for acknowledge.
- 10th and 11th cycles are for output 8 bits data from output data buffer or output 8 bits data from status register.
 - 10th cycle: Sending output data buffer [3:0] to LAD[3:0] or sending data of status register [3:0] to LAD[3:0]
 - 11th cycle: Sending output data buffer [7:4] to LAD[3:0] or sending data of status register [7:4] to LAD[3:0].
- 12th cycle: The "11112 " is output to LAD[3:0]. The OBF flag is cleared and OBE interrupt signal is generated.
- 13th cycle: The LAD[3:0] will be set to Hi-Z by slave to switch the communication direction. OBF flag will be set when 8 bits data are written to output data buffer by slave CPU.

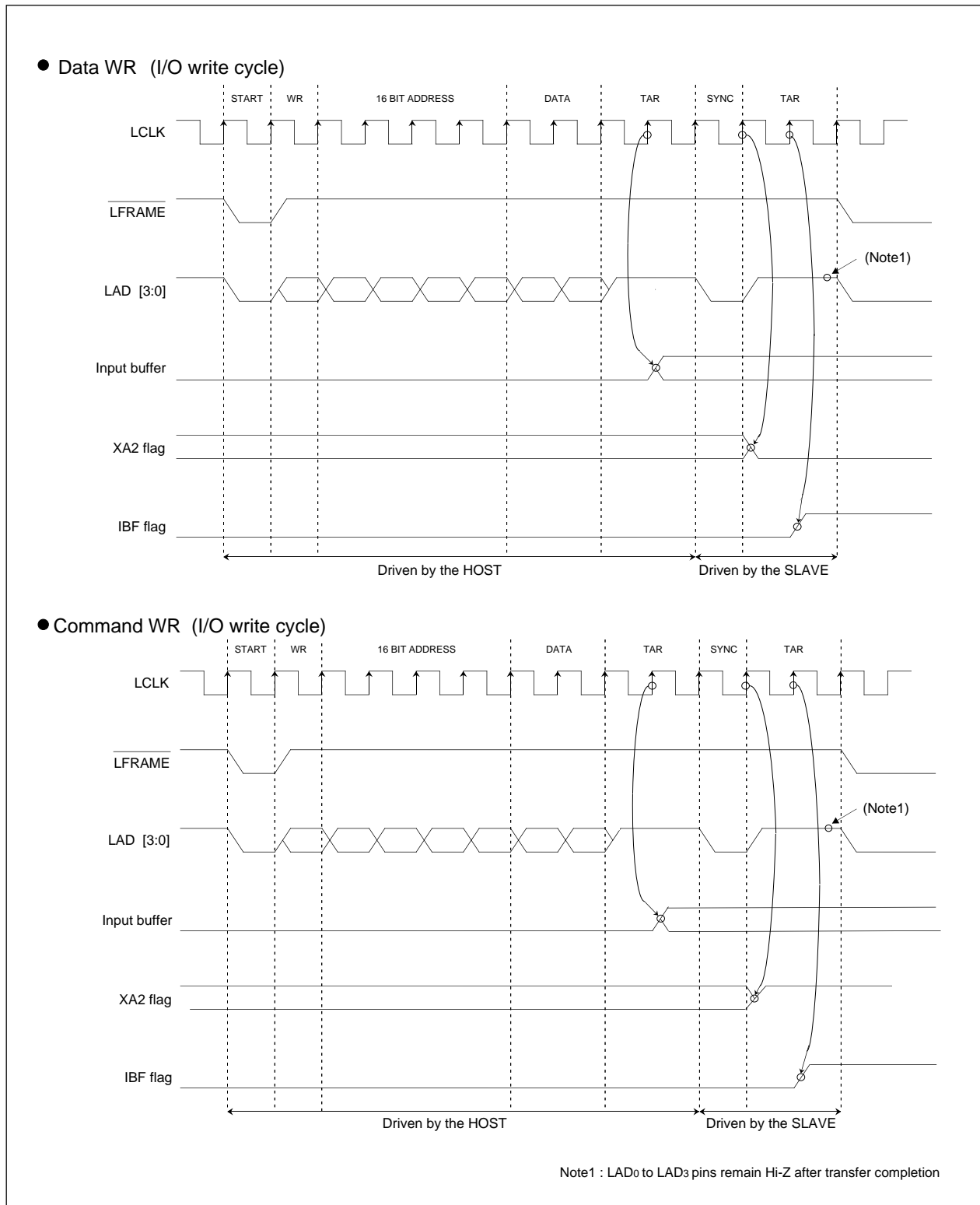


Fig.GF-7 Data and Command write timing figure

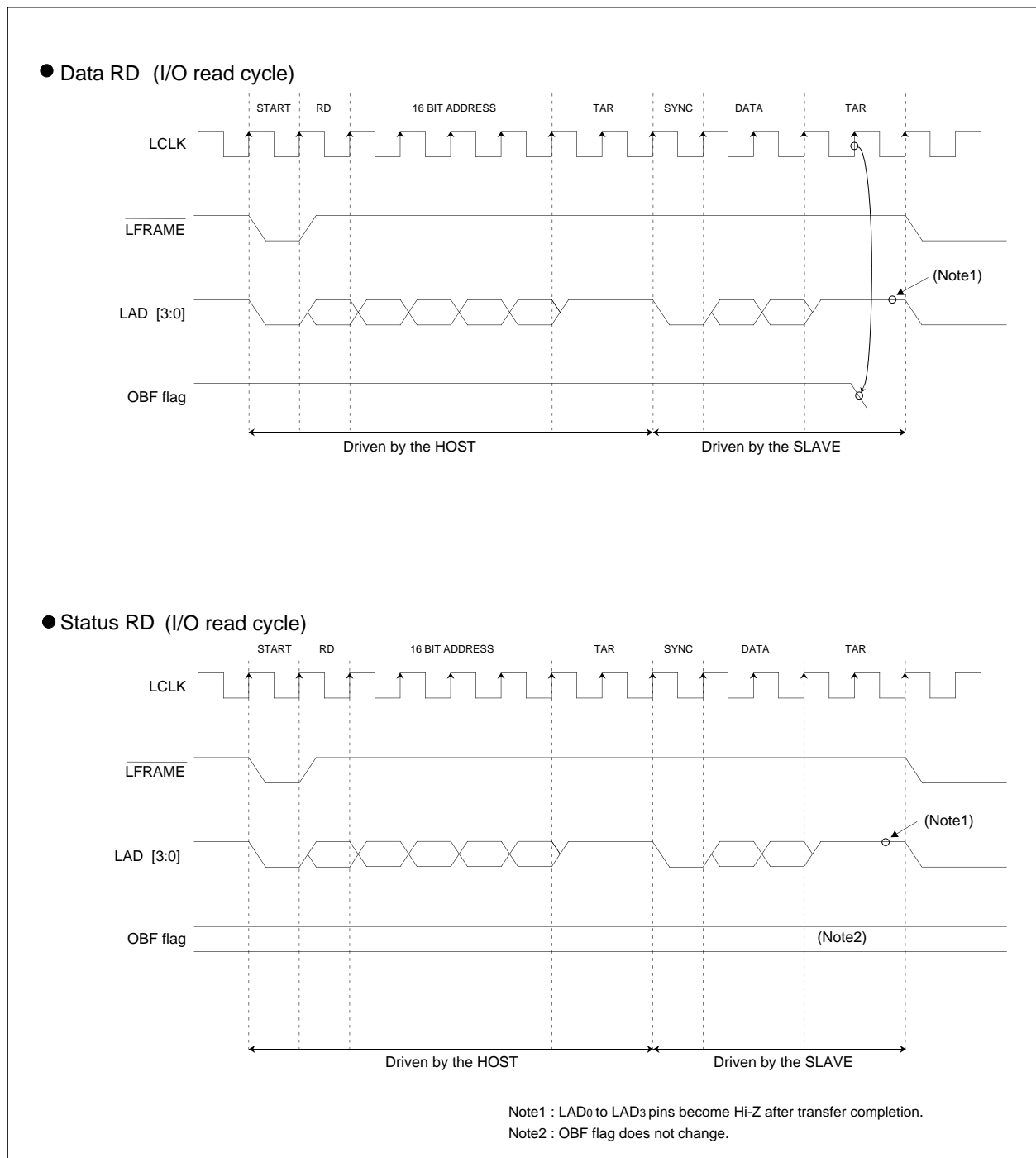


Fig.GF-8 Data and Status read timing figure

Table GF-2 Function explanation of the control input and output pins in LPC bus interface function

Pin name	Name	LPC interface enable bit 02D6 ₁₆ Bit 3	OBFO output enable bit 02C9 ₁₆ Bit 0	OBFO0 output enable bit 02C9 ₁₆ Bit 1	OBFO1 output enable bit 02C9 ₁₆ Bit 2	OBFO1 output enable bit 02C9 ₁₆ Bit 3	OBFO2 output enable bit 02C9 ₁₆ Bit 4	OBFO3 output enable bit 02C9 ₁₆ Bit 5	HOSTEN control bit 02C9 ₁₆ Bit 6	Input/Output	Function
P30/LAD0	LAD0	1	—	—	—	—	—	—	—	I/O	LPC bus used for transmitting and receiving address, command and data between Host CPU and peripheral devices.
P31/LAD1	LAD1	1	—	—	—	—	—	—	—	I/O	
P32/LAD2	LAD2	1	—	—	—	—	—	—	—	I/O	
P33/LAD3	LAD3	1	—	—	—	—	—	—	—	I/O	
P34/LFRAME	LFRAME	1	—	—	—	—	—	—	—	I	It is used for indicating the start of LPC cycle and termination of abnormal communication cycle.
P35/LRESET	LRESET	1	—	—	—	—	—	—	—	I	LPC reset signal. LPC bus interface function is reset.
P36/LCLK	LCLK	1	—	—	—	—	—	—	—	I	LPC synchronous clock signal.
P40/OBF00	OBF00	—	0	1	0	—	—	—	—	O	Status output signal. OBF00 output.
P43/OBF01	OBF01	—	1	0	1	—	—	—	—	O	Status output signal. OBF01 output.
P44/OBF1	OBF1	—	—	—	—	1	—	—	—	O	Status output signal. OBF1 output.
P45/OBF2	OBF2	—	—	—	—	—	1	—	—	O	Status output signal. OBF2 output.
P46/OBF3	OBF3	—	—	—	—	—	—	1	—	O	Status output signal. OBF3 output.

Table GF-3 Conditions of LPC bus interface function induced by $\overline{\text{LRESET}}$ input

	Pin name / Internal register	$\overline{\text{LRESET}}=\text{"H"}$	$\overline{\text{LRESET}}=\text{"L"}$	Note
Pin	P30/LAD0	LPC bus interface function(function is select)	I/O port	
	P31/LAD1			
	P32/LAD2			
	P33/LAD3			
	P34/LFRAME $\overline{\text{ }}$		I/O port	
	P35/LRESET		LPC bus interface function	
	P36/LCLK			
	P40/OBF00		I/O port	OBF output is enable until $\overline{\text{LRESET}}=\text{"L"}$. A spike pulse may be output to the port when the port is already set to L output port and OBF signal is output to the port just before $\overline{\text{LRESET}}$ is set to L.
	P43/OBF01			
	P44/OBF1			
	P45/OBF2			
	P46/OBF3			
	P42/GateA20			
	Internal register		Input data bus buffer	
Output data bus buffer			It can't be written by slave side.	
U flag 7,6,5,4,2			It can be written and read by slave side.	Initialization to "0" only for DBBSTS0.
XA2 flag			Initialization to "0"	
IBF flag			Initialization to "0"	There is possibility to generate IBF interrupt request.
OBF flag			Initialization to "0"	There is possibility to generate OBE interrupt request.
LPCADH/L			It can be written and read by slave side.	
LPCCON			It can be written and read by slave side.	
GA20 circuit			Initialization	

GateA20 output function

The GateA20 pin (port P42) can be controlled by LPC interface function channel 0 in hardware. Hardware GateA20 is sharing with P42 pin. Setting "1" to bit 0 of GateA20 control register enables the hardware GateA20 function. The default value of hardware GateA20 is "1".

The GateA20 control register is shown in Fig.GF-9.

When the host CPU writes "D1" command to address 0064₁₆, and then writes data to address 0060₁₆ in succession, the value of bit 1 of the data will be output to GateA20 pin. The timing is shown in Fig.GF-10.

The GateA20 operation sequences are shown in Fig.GF-11, Fig.GF-12. As shown in the figures, there is no change in input buffer full flag (IBF0) and no input buffer full (IBF) interrupt request, but the input data bus buffer and XA2 flag are changed in these sequences.

The value of the GateA20 output pin will be held till the data next to D1 command is written in. P42 becomes I/O port and the the value of GateA20 becomes "0" when $\overline{\text{LRESET}}$ input is "L". GateA20 will be initialized even if the sequence is executed. However, the GateA20 enable bit will not be changed and GateA20 output pin will be resumed after the $\overline{\text{LRESET}}$ input becomes "H".

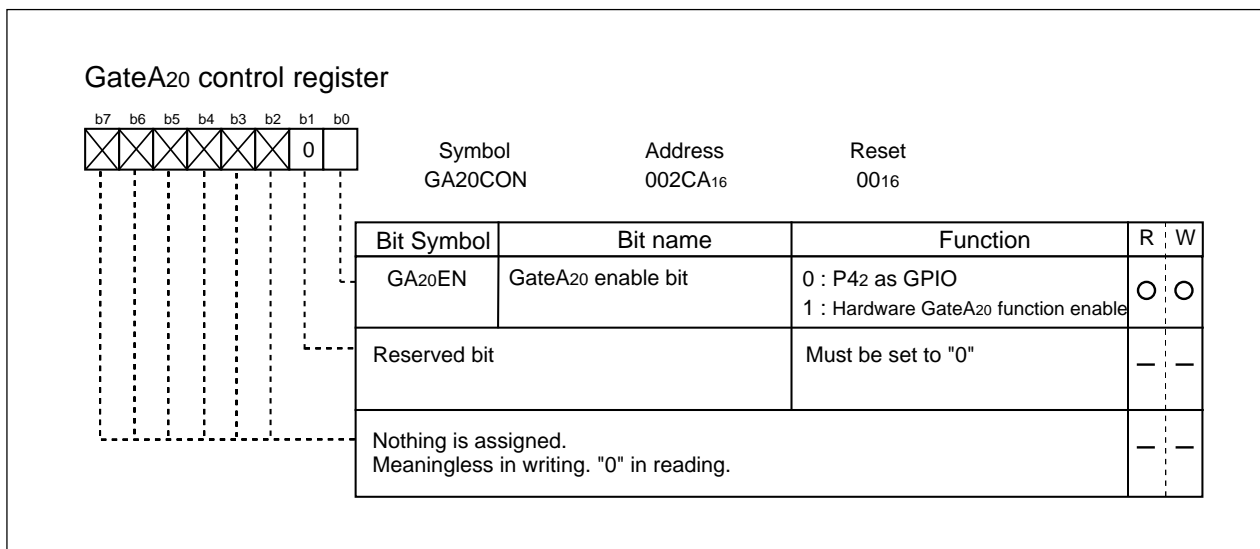


Fig.GF-9 GateA20 control register

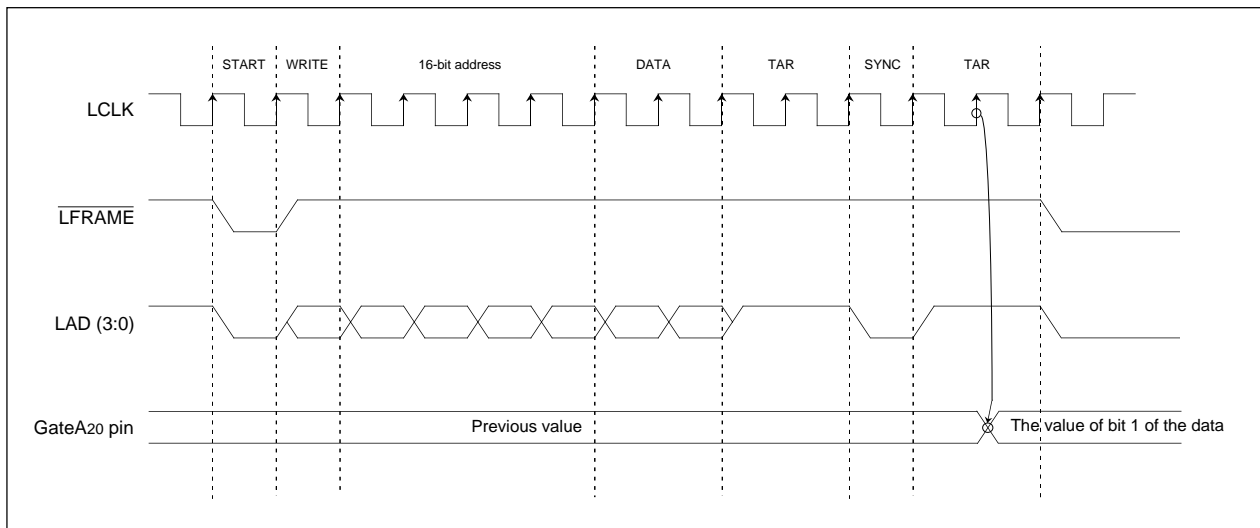


Fig.GF-10 GateA20 output timing

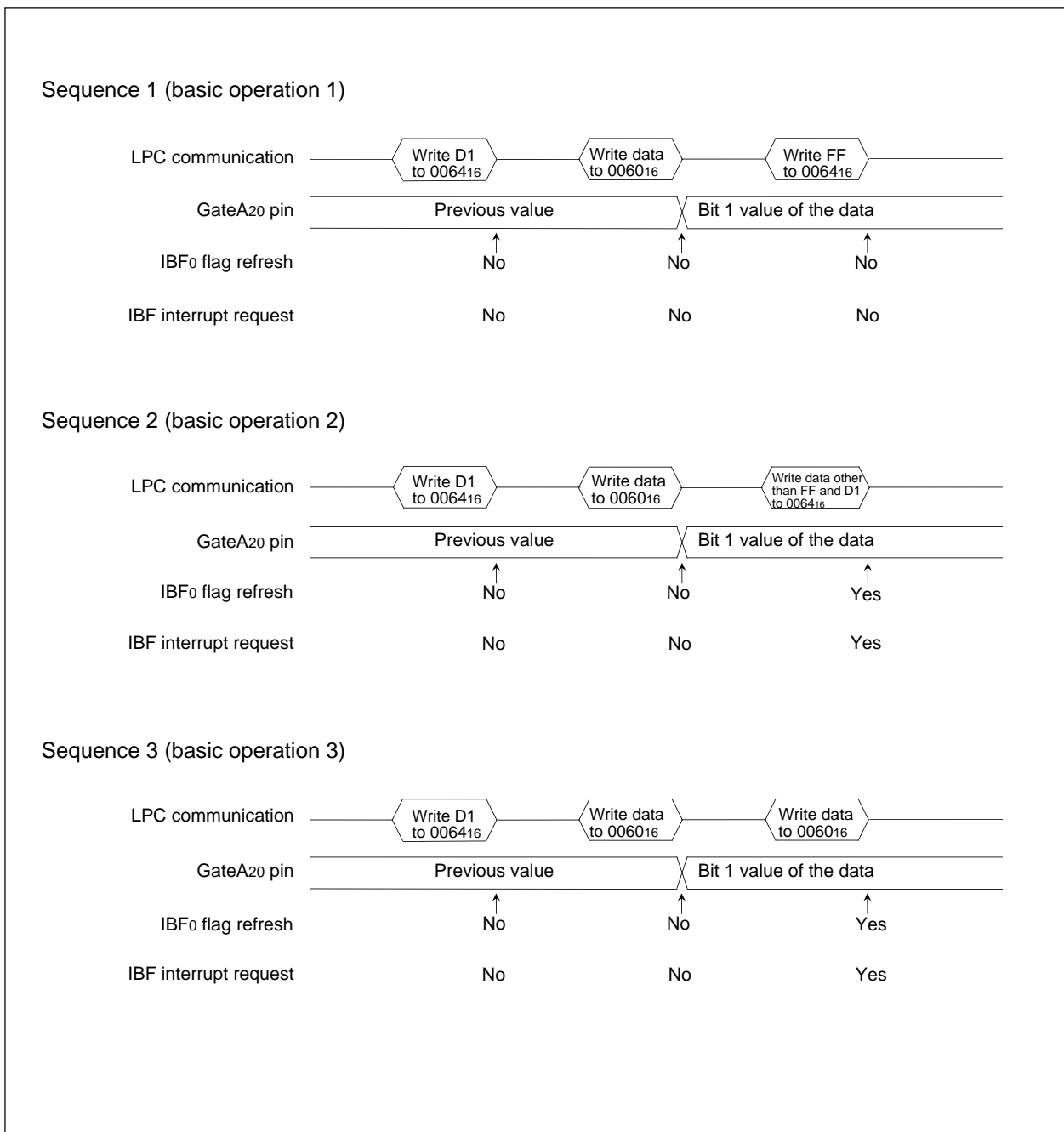


Fig.GF-11 GateA20 operation sequence (1)

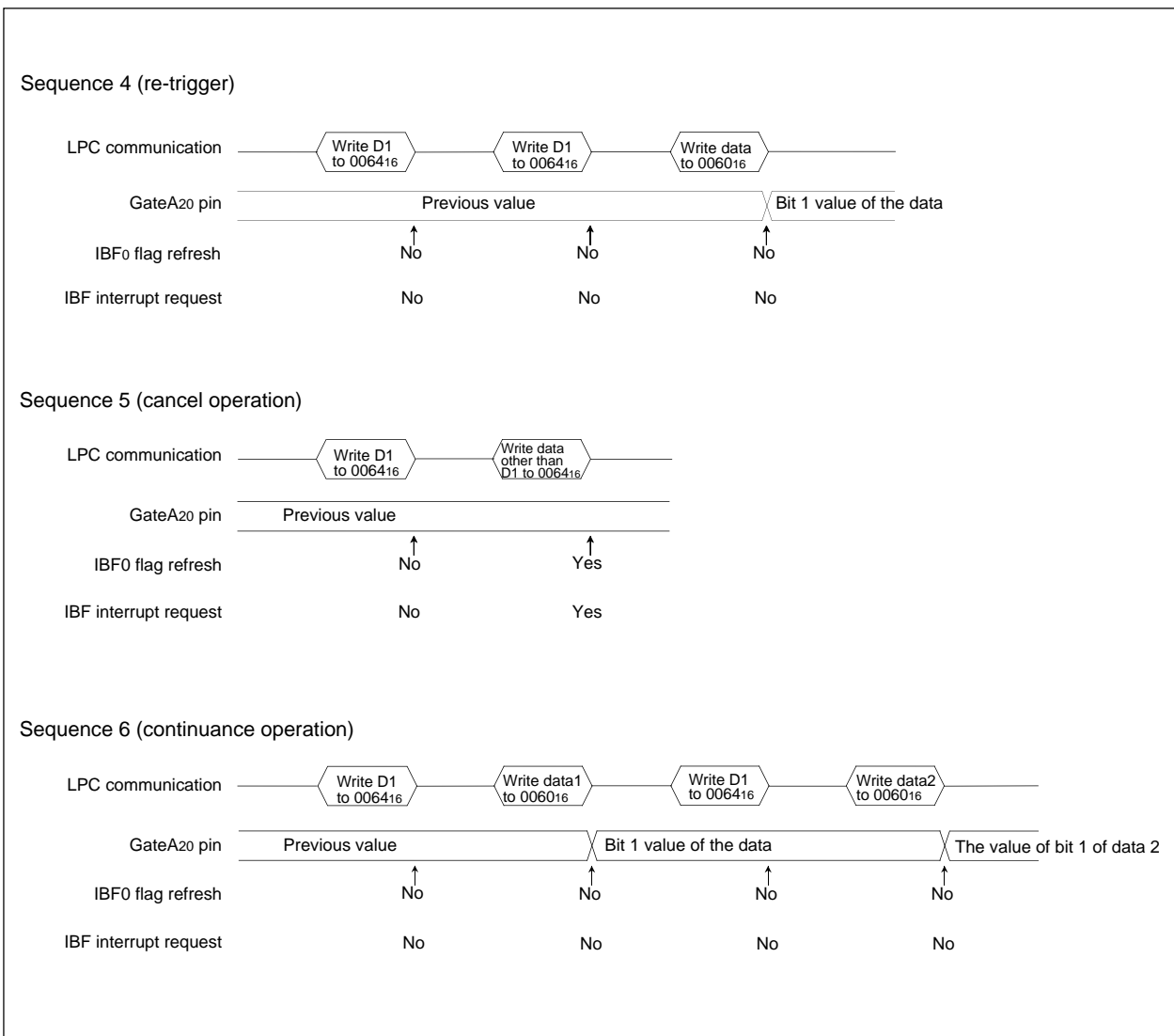


Fig.GF-12 GateA20 operation sequence (2)

Serial Interrupt Output

The serial interrupt output is the circuit that outputs the interrupt request to the host with serial interrupt data format.

Tab.SI-1 shows the specification of serial interrupt output.

Table.SI-1 Specifications of serial interrupt output

Item	Specification
The factors of serial interrupt	<p>The numbers of serial interrupt requests (numbers of channels) that can output simultaneously are 5 factors. Each interrupt factor of each channel is explained as follows.</p> <ul style="list-style-type: none"> • Channel 0 <ol style="list-style-type: none"> (1) By setting "1" to IRQ_i request bit (bit 5, 6 i=1,12) of IRQ request register 0, the interrupt request can be generated. (2) Synchronized with OBF00 and OBF01 that are the host bus interface internal signals, the serial interrupt request can be generated. • Channel 1-3 <ol style="list-style-type: none"> (1) By setting "1" to IRQ request bit (bit 5) of IRQ request register 1-3, the interrupt request can be generated. (2) Synchronized with OBF1-3 that are the host bus interface internal signals, the serial interrupt request can be generated. • Channel 4 <p>By setting "1" to IRQ request bit (bit 5) of IRQ request register 4, the interrupt request can be generated.</p>
The number of frame	<ul style="list-style-type: none"> • Channel 0 <ol style="list-style-type: none"> (1) Setting the IRQ1 request bit (bit 5) of IRQ request register0 to "1" or detecting OBF00, which is the host bus interface internal signal, selects Frame 1. (2) Setting the IRQ12 request bit (bit 6) of IRQ request register0 to "1" or detecting OBF01, which is the host bus interface internal signal, selects Frame 12. • Channel 1-4 <p>Selecting the frame select bit (bit 0-4) of IRQ request register1-4 selects Frame 1-15 or extend Frame 0-10.</p>
Operation clock	The operation synchronized with LCLK (Max. 33MHz). (Note)
Clock restart	Setting the clock restart enable bit (bit 6) of serial interrupt control register0 to "1" requests the clock restart if the clock has stopped or slowed down in serial interrupt output.
Clock stop inhibition	Setting the clock stop inhibition bit (bit 5) of serial interrupt control register0 to "1" requests the inhibition of clock stop if the clock tends to stop or slow down in serial interrupt output.
OBF sync enable	Setting the OBF00, OBF01, OBF1-3 sync enable bit (bit 0-4) of serial interrupt control register0 to "1" enables the OBF synchronization.

Note: To enable LCLK, it is necessary to enable the LPC bus interface function.

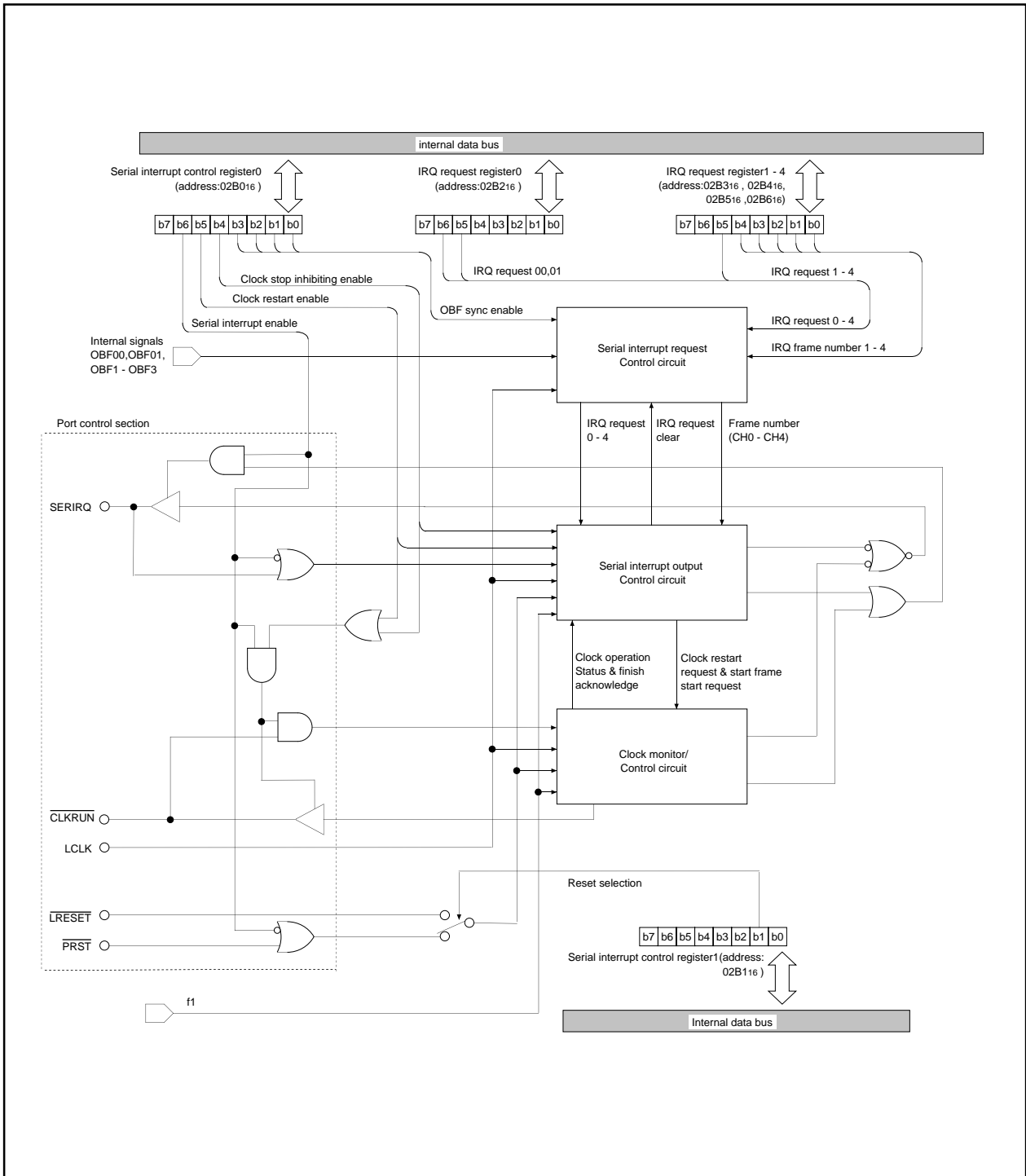


Fig.SI-1 Serial interrupt block chart

(1) Register explanation

Fig.SI-2 shows the configuration of IRQ request register0, Fig.SI-3 shows the configuration of IRQ request register1-4, Fig.SI-4, SI-5 show the configurations of serial interrupt control register0,1 respectively.

● IRQ request register0 IRQR0

The serial interrupt request of Channel 0 is set by software.

•IRQ1 request bit IR0

Setting the bit to “1” generates the serial interrupt request (Frame 1).

By setting the OBF00 sync enable bit (bit 0) of the serial interrupt control register0 to “1”, the value of IR0 is the same as that of OBF00, which is the host bus interface internal signal. When the internal signal OBF00 is “1”, the serial interrupt is generated.

IR0 is cleared to “0” by writing “0” in software.

IR0 can not be cleared to “0” by software when the internal signal OBF00 is “1” if OBF00 sync enable bit is set to “1”.

•IRQ12 request bit IR1

Setting the bit to “1” generates the serial interrupt request (Frame 12).

By setting the OBF01 sync enable bit (bit 1) of the serial interrupt control register0 to “1”, the value of IR1 is the same with that of OBF01, which is the host bus interface internal signal. When the internal signal OBF01 is “1”, the serial interrupt is generated.

IR1 is cleared to “0” by writing “0” in software.

IR1 can not be cleared to “0” by software when the internal signal OBF01 is “1” if OBF01 sync enable bit is set to “1”.

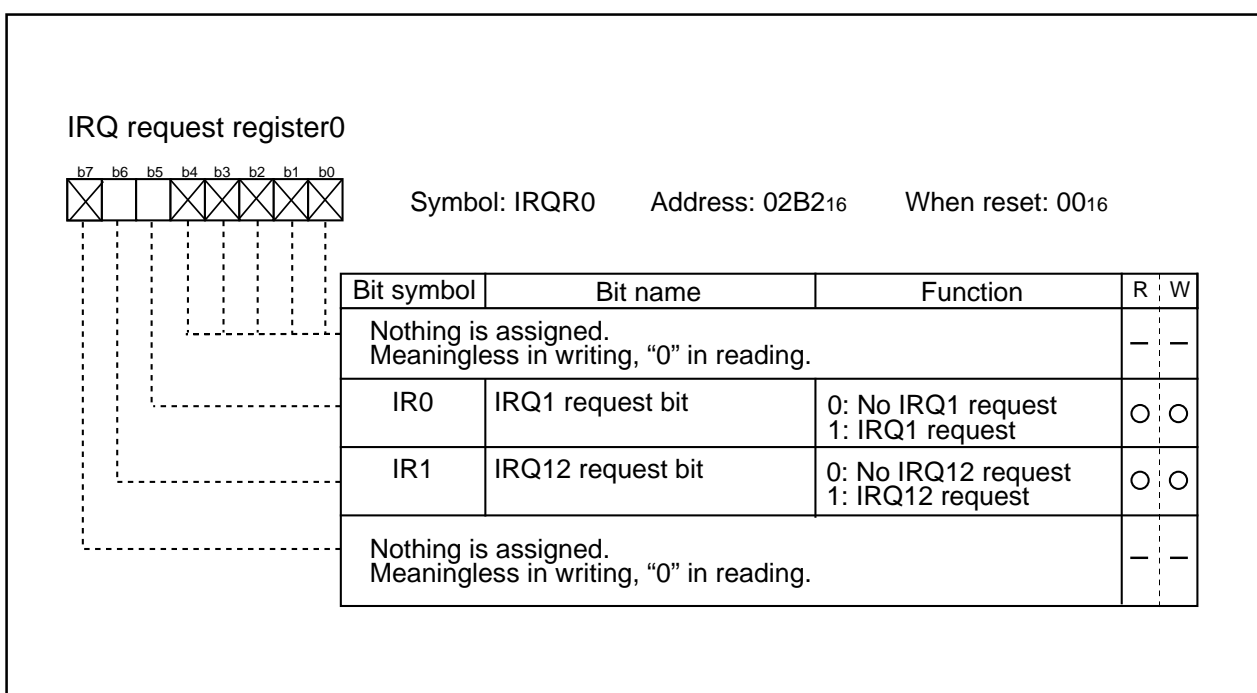


Fig.SI-2 Configuration of IRQ request register0

● **IRQ request register i IRQRi (i=1-4)**

The serial interrupt request of Channel 1-4 is set by software, or asserting frame is selected.

● **IRQ request bit IR**

Setting the bit to “1” generates the serial interrupt request.

By setting the OBFj sync enable bit (bit2-4, j=1-3) of the serial interrupt control register0 to “1”, the value of IR is the same as that of OBFj, which is the host bus interface internal signal. When the internal signal OBFj is “1”, the serial interrupt is generated.

IR is cleared to “0” by writing “0” in software.

IR can not be cleared to “0” by software when the internal signal OBFj is “1” if OBFj sync enable bit is set to “1”.

● **IRQ select bit IS0-4**

The asserting frame is selected.

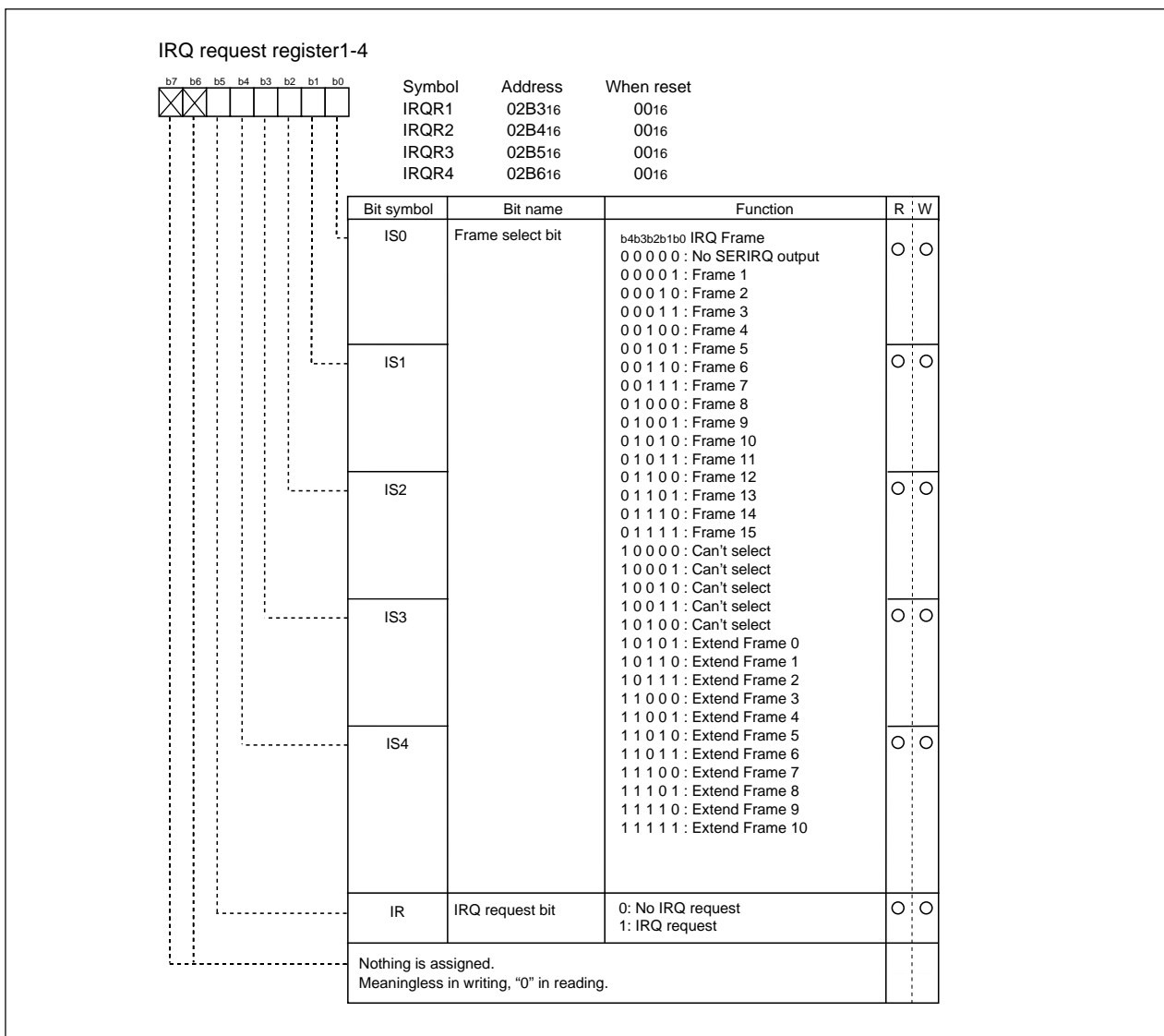


Fig.SI-3 Configuration of IRQ request register1-4

● Serial interrupt control register0 SERCON0

The operation condition of serial interrupt is set.

● OBFi sync enable bit SENi (i=00,01,1-3)

By setting the bit to "1", sync with the OBFi of host interface, the serialized interrupt can be generated.

● Clock stop inhibition bit SUPEN

Setting the bit to "1" will request the inhibition of clock if the clock tends to stop or slow down in serial interrupt request.

● Clock restart enable bit RUNEN

Setting the bit to "1" requests the clock restart during the clock stop or clock slow down in serial interrupt request.

● Serial interrupt enable bit IRQEN

0: SERIRQ, PRST, CLKRUN are I/O ports.

1: SERIRQ, PRST, CLKRUN are serial interrupt function ports.

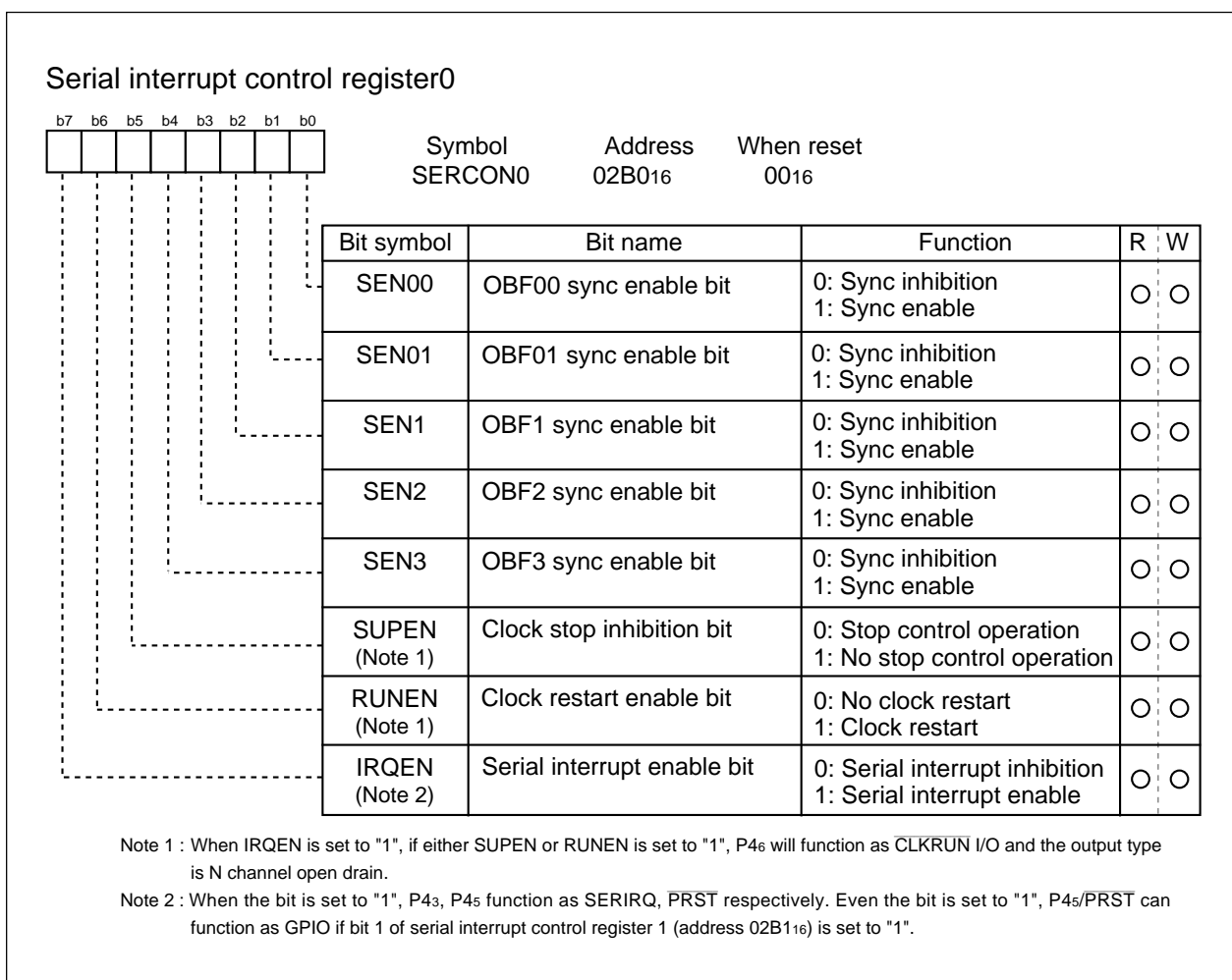


Fig.SI-4 Configuration of serial interrupt control register0

● Serial interrupt control register1 SERCON1

The register is for setting the pins of serial interrupt.

●Reset selection bit RSEL

0: The input of $\overline{\text{PRST}}$ is the reset signal.

1: The input of $\overline{\text{LRESET}}$ is the reset signal. (Note1)

Note 1: The $\overline{\text{PRST}}$ pin becomes I/O port if setting the bit to "1".

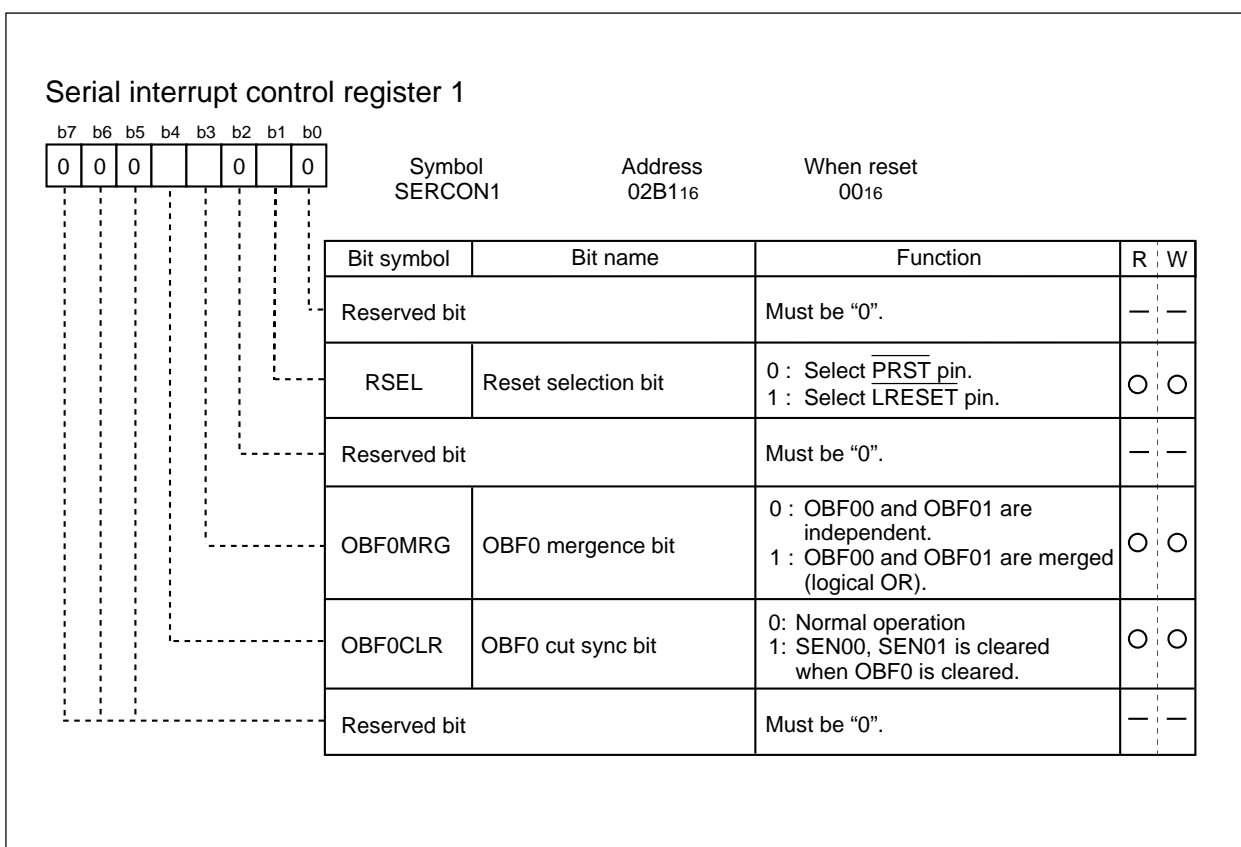


Fig.SI-5 Configuration of serial interrupt control register1

● Serial interrupt control register2 SERCON2

The polarity of serial interrupt output can be selected by bit 0 to bit 5 of serial interrupt control register 2.

When the bit is set to “0”:

If there is a request, Hiz-Hiz-Hiz

If there is no request, L-H-Hiz

When the bit is set to “1”:

If there is a request, L-H-Hiz

If there is no request, Hiz-Hiz-Hiz

Only the default value of bit 4 (serial interrupt polarity bit 3) of serial interrupt control register 2 after reset is “1”.

Fig.SI-7 shows the configuration of serial interrupt control register 2.

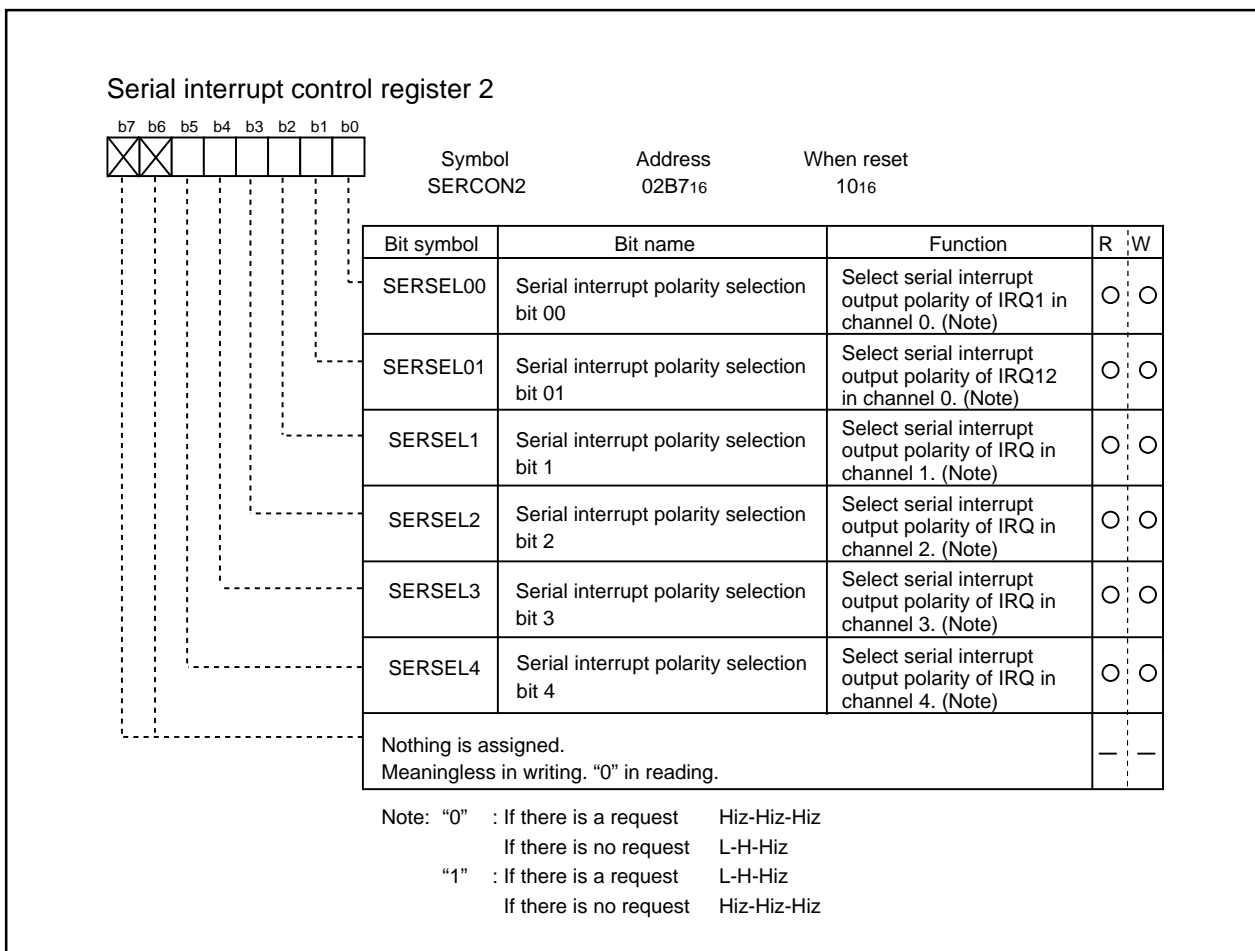


Fig.SI-7 The configuration of serial interrupt control register 2

(2) The operation of serial interrupt

A cycle operation of serial interrupt starts with start frame and finishes with stop frame. There are 2 kinds of operation mode: continuous mode and quiet mode. The next operation mode is judged by monitoring the length of stop frame sent from host side.

● The timing of serial interrupt cycle

Fig.SI-8 shows an example of basic timing of serial interrupt cycle.

① Start frame

The start frame will be detected if the SERIRQ remains “L” in 4-8 clock cycles.

② IRQ data frame

Each IRQ data frame is 3 clock cycles.

- Channel 0-2,4: If the IRQ request bit is “0”, then the SERIRQ is driven to “L” during the 1st clock cycle of the corresponding data frame, to “H” during the 2nd clock cycle, to high impedance during the 3rd clock cycle. If the IRQ request bit is “1”, then the SERIRQ is high impedance during all of the 3 clock cycles.

- Channel 3: If the IRQ request bit is “0”, then the SERIRQ is high impedance during all of the 3 clock cycles. If the IRQ request bit is “1”, then the SERIRQ is driven to “L” during the 1st clock cycle of the corresponding data frame, to “H” during the 2nd clock cycle, to high impedance during the 3rd clock cycle.

Serial Interrupt output polarity of each channel can be changed by Serial Interrupt polarity selection bit i ($i=00, 01, 1, 2, 3, 4$) of Serial Interrupt control register 2.

③ Stop frame

The stop frame will be detected if the SERIRQ remains “L” in 2 or 3 clock cycles. The next operation mode is quiet mode if the length of “L” is 2 clock cycles, the continuous mode if the length is 3 clock cycles.

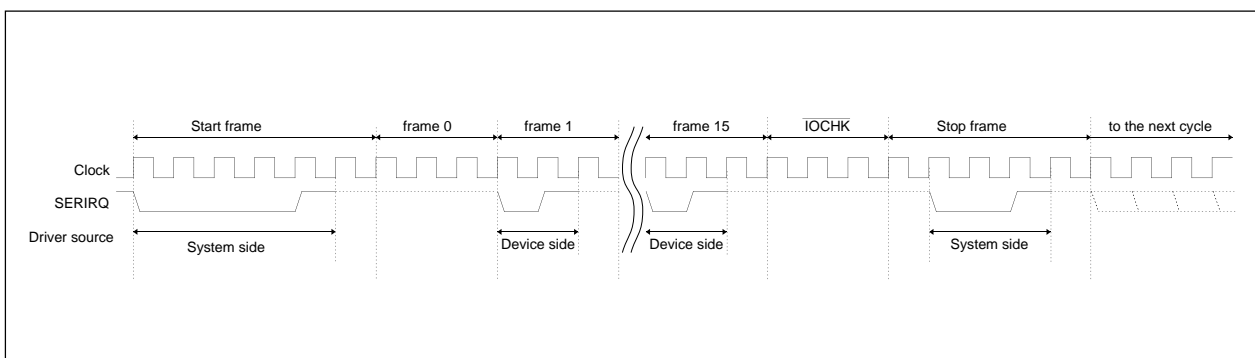


Fig.SI-8 Basic timing of serial interrupt cycle

● Operation mode

Fig.SI-9 shows an example of timing of continuous mode, Fig.SI-10 shows that of quiet mode.

① Continuous mode

After reset, at the rising edge of \overline{PRST} (or \overline{LRESET}) or the length of the last stop frame of serial interrupt cycle being 3 clock cycles, it will be the continuous mode.

After receiving the start frame (Note 1), the Frame 1, Frame 12 or frames selected in each channel will be asserted.

Note 1: If the length of "L" is less than 4 clock cycles or more than 9 clock cycles, the start frame will not be detected and the next start (the falling edge of SERIRQ) is waited.

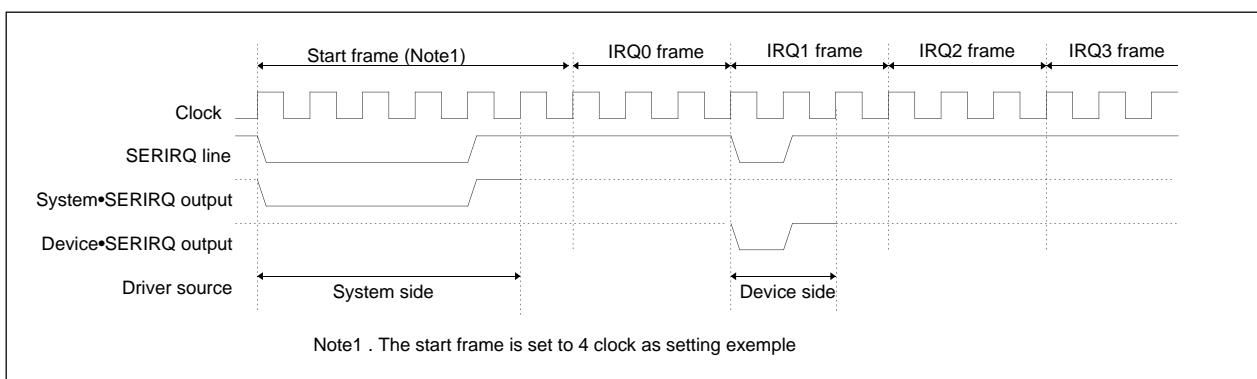


Fig.SI-9 Timing diagram of continuous mode

② Quiet mode

At clock stop or clock slow down, or the length of the last stop frame of serial interrupt cycle being 2 clock cycles, it will be the quiet mode.

In this mode the SERIRQ is driven to "L" in the 1st clock cycle by device and after the receiving of the rest start frame (Note 1) from host, the IRQ1 Frame , IRQ12 Frame or frames selected in each channel will be asserted.

Note 1: If the sum of length of "L" that is driven by the device in the 1st clock cycle and by the host in the rest clock cycles is within 4-8 clock cycles, the start frame will be detected.

If the sum of length of "L" is less than 4 clock cycles or more than 9 clock cycles, the start frame will not be detected and the next start (the falling edge of SERIRQ) is waited.

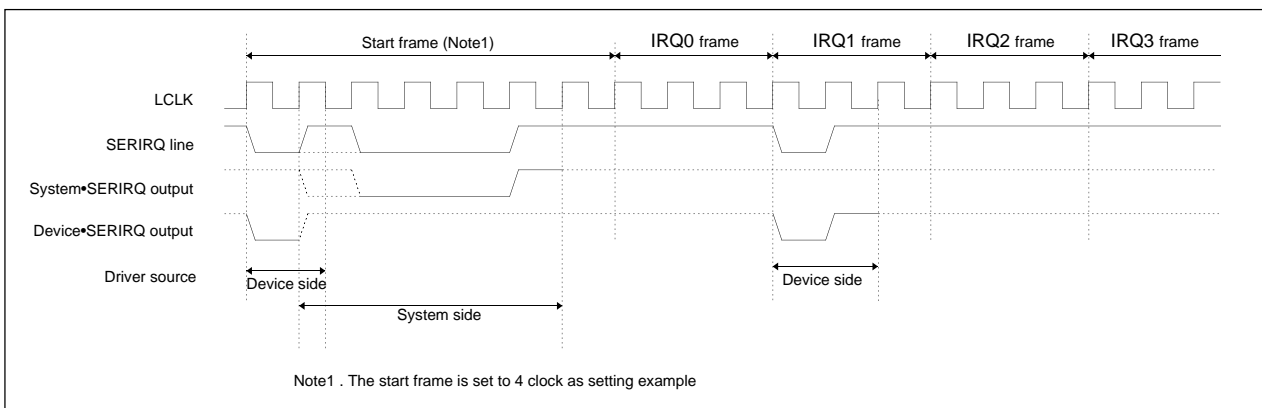


Fig.SI-10 Timing diagram of quiet mode

(3) Clock restart/ stop inhibition request

Asserting the $\overline{\text{CLKRUN}}$ signal can request to restart or maintain the clock which stops or slows down or request the host to tend to stop or slow down.

Fig.SI-11 shows an example of timing of clock restart request, Fig.SI-12 shows an example of timing of clock stop inhibition request.

① Clock restart operation

Setting the clock restart bit of serial interrupt control register0 to "1" will request the clock restart if the clock has slowed down or stopped at serial interrupt request.

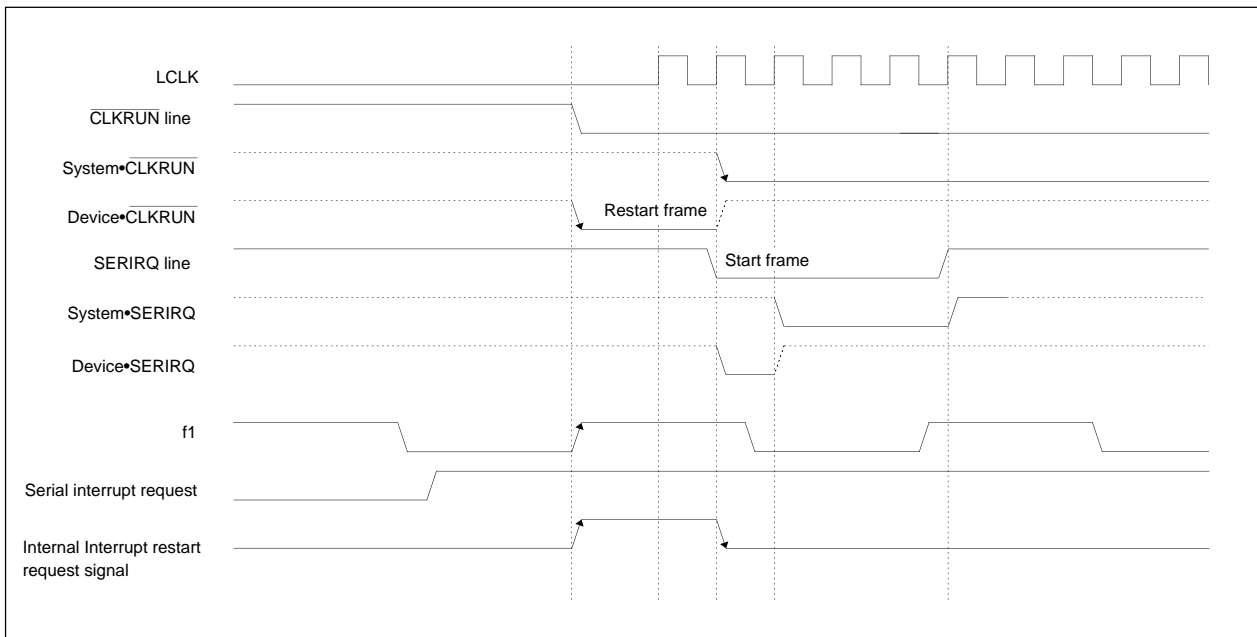


Fig.SI-11 Timing diagram of clock restart request

② Clock stop inhibition request

Setting the clock stop inhibition bit of serial interrupt control register0 to "1" will request the inhibition of clock stop if the clock tends to stop or slow down during all the period of serial interrupt output.

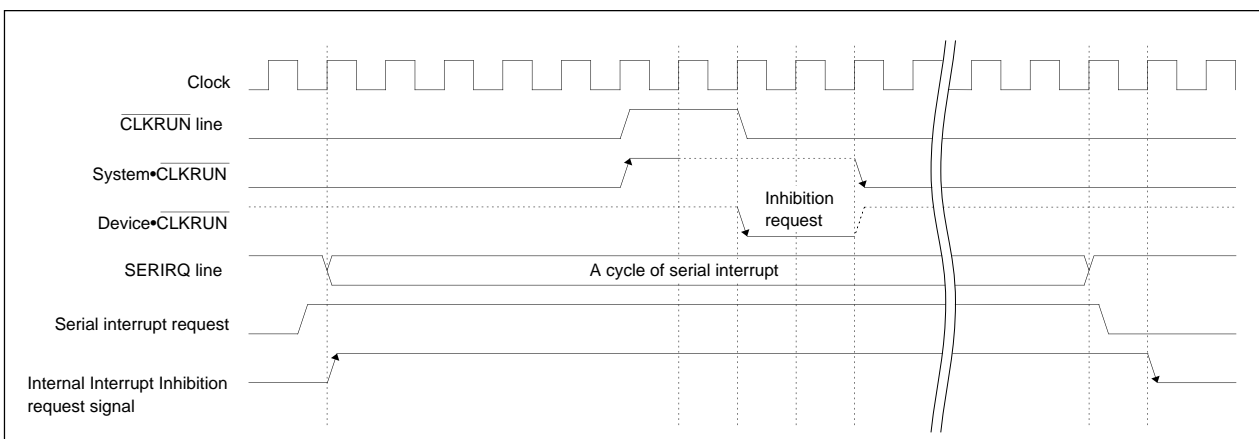


Fig.SI-12 Example of timing of clock stop inhibition request

MULTI-MASTER I²C-BUS INTERFACE

The multi-master I²C-BUS interface is a serial communication circuit based on Philips I²C-BUS data transfer format. 3 independent channels, with both arbitration lost detection and a synchronous functions, are built in for the multi-master serial communication. Fig.GC-1 shows a block diagram of the multi-master I²C-BUS interface and Table.GC-1 lists the multi-master I²C-BUS interface functions. The multi-master I²C-BUS interface consists of the I²C address register, the I²C data shift register, the I²C clock control register, the I²C control register 1, I²C control register 2, the I²C status register, the I²C start/stop condition control register and other control circuits.

Table.GC-1 Multi-master I²C-BUS interface functions

Item	Function
Format	Based on Philips I ² C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	Based on Philips I ² C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1kHz to 400kHz (at V _{IIC} = 4MHz)

*V_{IIC}=I²C system clock

* : Purchase of Renesas Technology Corporation's I²C components conveys a license under the Philips I²C Patent Rights to use these components an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

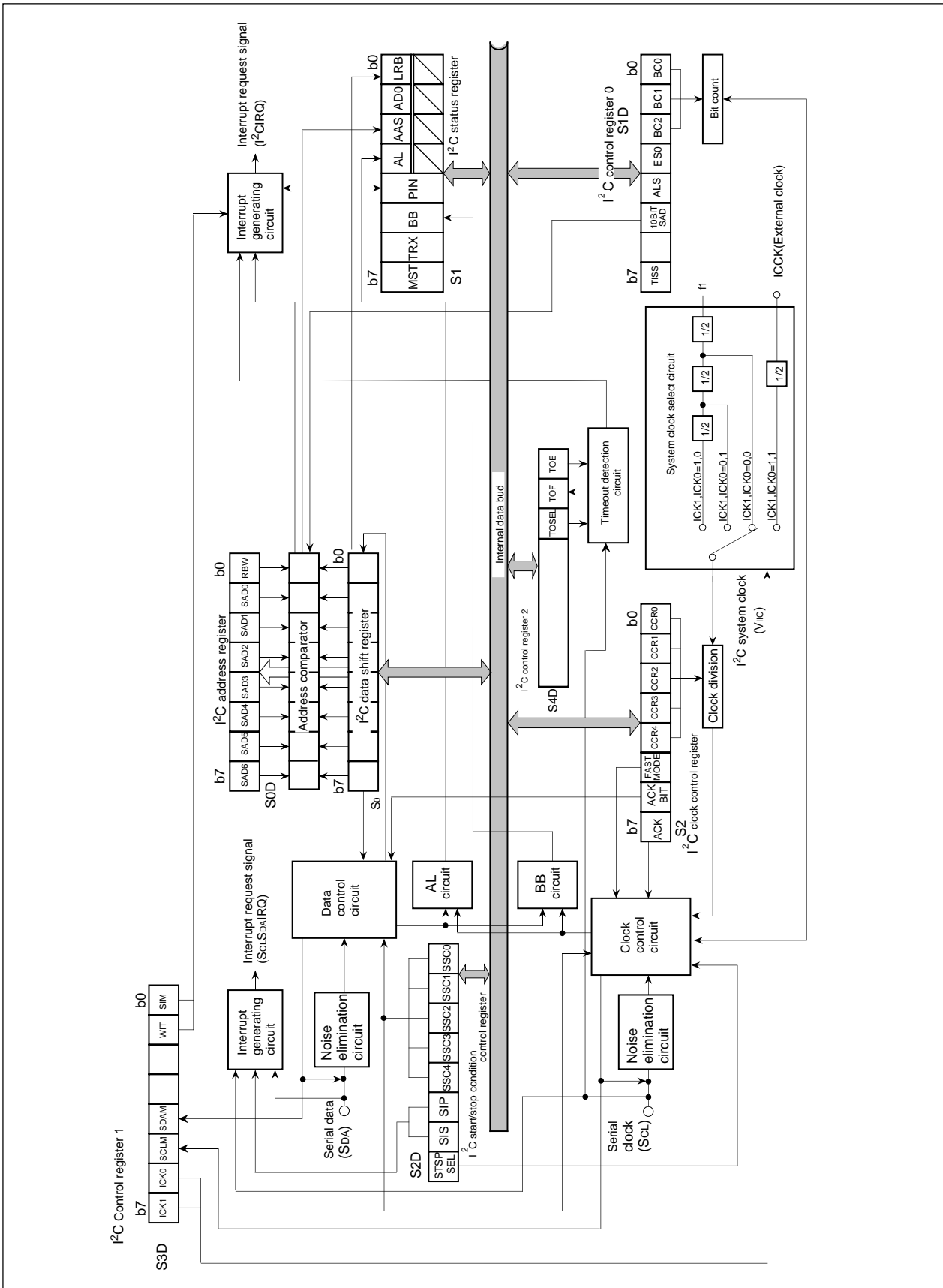


Fig.GC-1 Block diagram of multi-master I²C-BUS interface

I²C Data Shift Register

The I²C data shift register (address 0320₁₆, 0330₁₆, 0310₁₆) is an 8-bit shift register to store receiving data and write transmission data. When transmit data is written into this register, it is transferred to the outside from bit 7 in synchronization with the SCL clock, and each time one-bit data is output, the data of this register are shifted by one bit to the left. When data is received, it is input to this register from bit 0 in synchronization with the SCL clock, and each time one-bit data is input, the data of this register are shifted by one bit to the left. The timing of storing received data to this register is shown in figure GC-3. The I²C data shift register is in a write enable status only when the I²C-BUS interface enable bit (ES0 bit : bit 3 of address 0323₁₆, 0333₁₆, 0313₁₆) of the I²C control register 0 is "1". The bit counter is reset by a write instruction to the I²C data shift register. When both the ES0 bit and the MST bit of the I²C status register (address 0328₁₆, 0338₁₆, 0318₁₆) are "1", the SCL is output by a write instruction to the I²C data shift register. Reading data from the I²C data shift register is always enabled regardless of the value of ES0 bit.

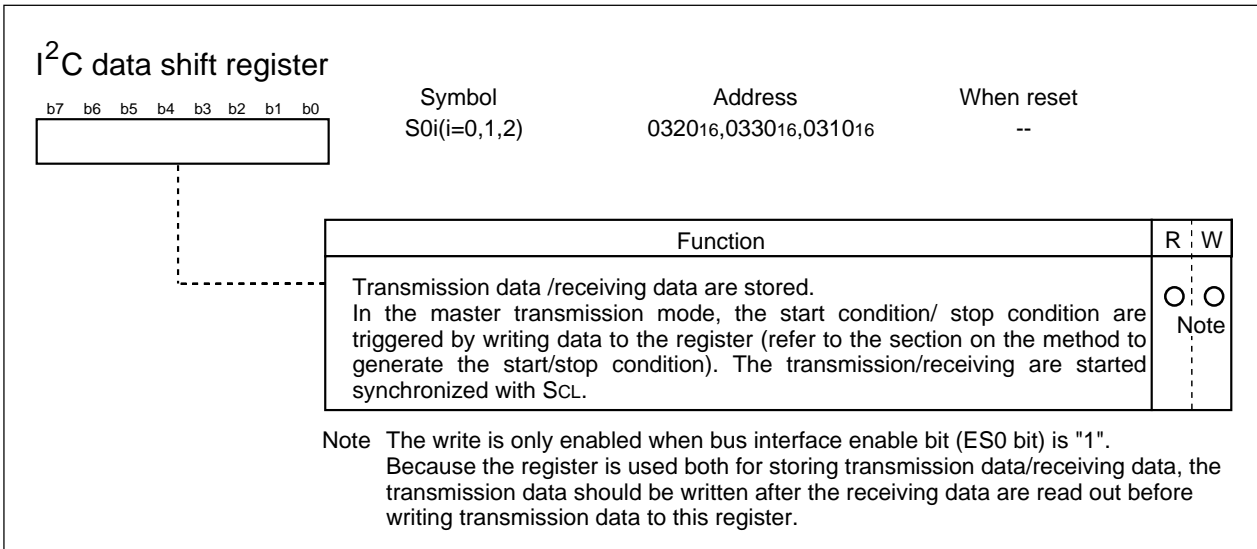


Fig.GC-2 I²C data shift register

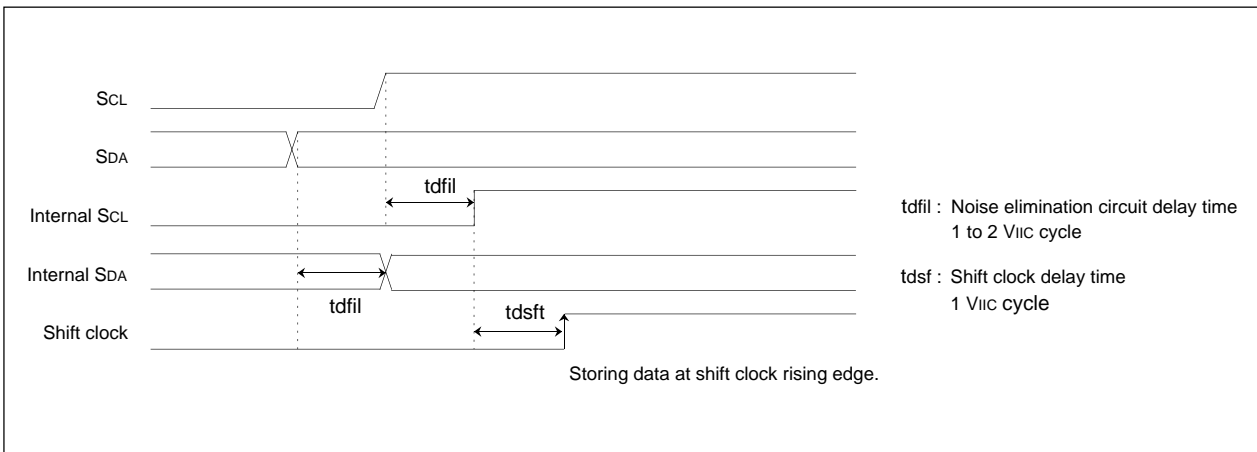


Fig.GC-3 The timing of receiving data stored to I²C data shift register

I²C Address Register

The I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆) consists of a 7-bit slave address and a read/write bit. In the addressing mode, the slave address written in this register is compared with the address data to be received immediately after the START condition is detected.

•Bit 0: Read/write bit (RBW)

This is not used in the 7-bit addressing mode. In the 10-bit addressing mode, the first byte address data to be received are compared with the contents (SAD6 to SAD0 + RBW) of the I²C address register.

The RBW bit is cleared to "0" automatically when the stop condition is detected.

•Bits 1 to 7: Slave address (SAD0–SAD6)

These bits store slave addresses. Regardless of the 7-bit addressing mode or the 10-bit addressing mode, the address data transmitted from the master is compared with the contents of these bits.

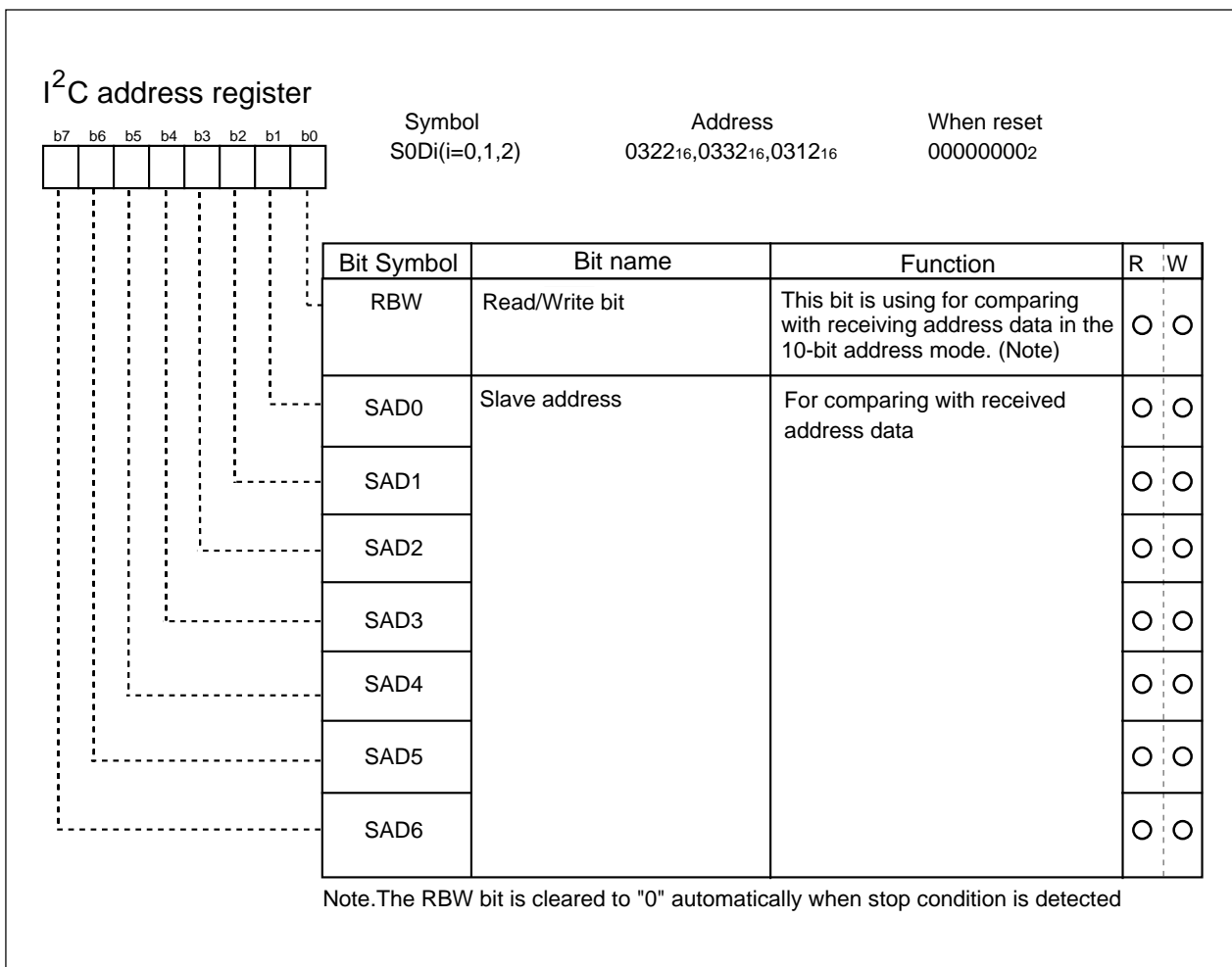


Fig.GC-4 I²C address register

I²C Clock Control Register

The I²C clock control register 0,1 (address 0324₁₆, 0334₁₆, 0314₁₆) is used to set ACK control, SCL mode and SCL frequency.

•Bits 0 to 4: SCL frequency control bits (CCR0–CCR4)

These bits control the SCL frequency. Refer to Table GC-2.

•Bit 5: SCL mode specification bit (FAST MODE)

This bit specifies the SCL mode. When this bit is set to “0”, the standard clock mode is selected. When the bit is set to “1”, the high-speed clock mode is selected. When connecting to the bus with the high-speed mode I²C-BUS standard (maximum 400 kbits/s), set 4 MHz or more to I²C system clock(V_{IIC}).

•Bit 6: ACK bit (ACK BIT)

This bit sets the SDA status when an ACK clock is generated. When this bit is set to “0”, the ACK return mode is selected and SDA goes to “L” at the occurrence of an ACK clock. When the bit is set to “1”, the ACK nonreturn mode is selected. The SDA is held in the “H” status at the occurrence of an ACK clock. However, when the slave address agrees with the address data in the reception of address data at ACK BIT = “0”, the SDA is automatically made “L” (ACK is returned). If there is a disagreement between the slave address and the address data, the SDA is automatically made “H” (ACK is not returned).

*ACK clock: Clock for acknowledgment

•Bit 7: ACK clock bit (ACK)

This bit specifies the mode of acknowledgment which responses to the data transferring. When this bit is set to “0”, the no ACK clock mode is selected. In this case, no ACK clock occurs after data transmission. When the bit is set to “1”, the ACK clock mode is selected and the master generates an ACK clock at the completion of each 1-byte data transfer. The device for transmitting address data and control data releases the SDA at the occurrence of an ACK clock (makes SDA “H”) and receives the ACK bit generated by the data receiving device.

Note: Except for ACK bit (ACKBIT), do not write data into the I²C clock control register during transfer.

If data is written during transfer, the I²C clock generator is reset, so that data cannot be transferred normally.

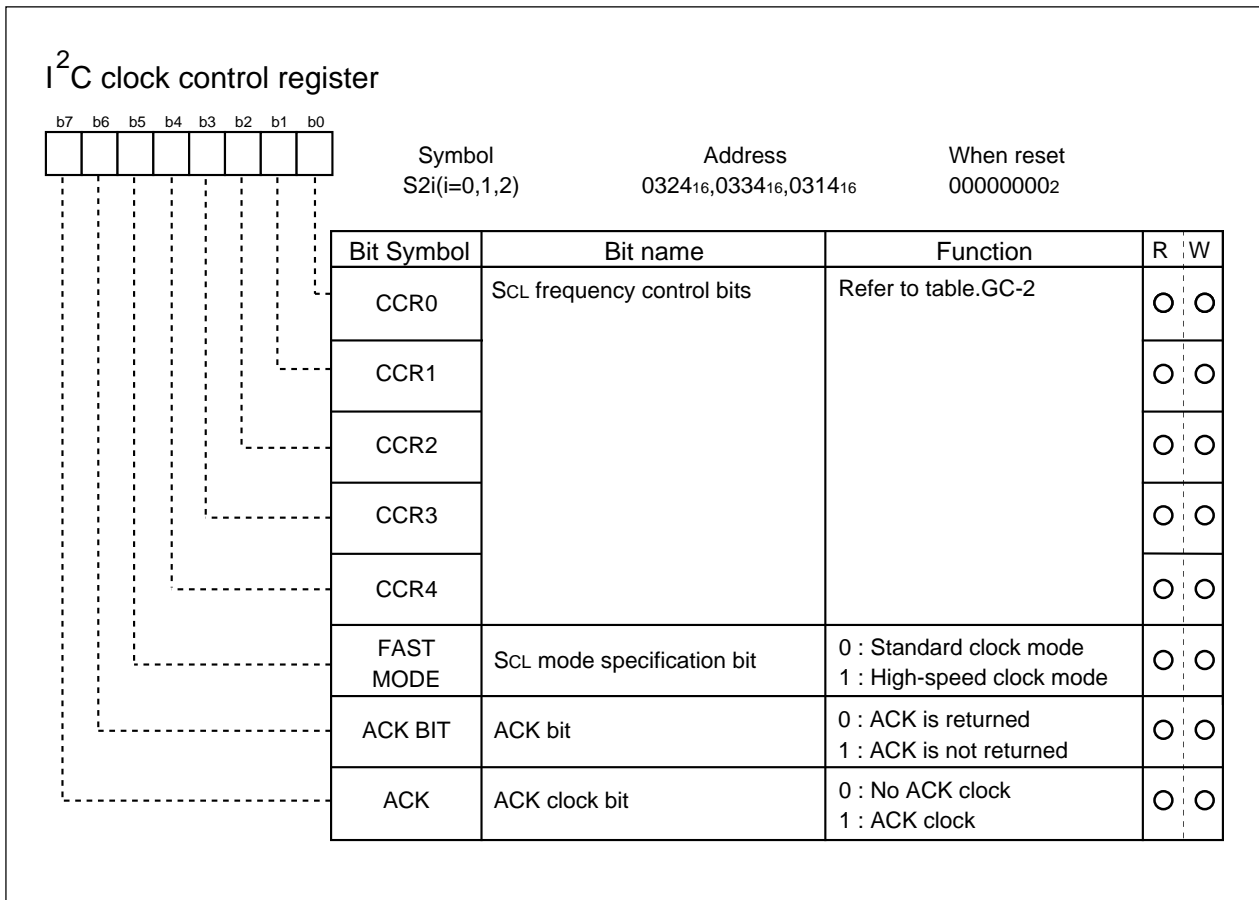


Fig.GC-5 I²C clock register

Table.GC-2 Set values of I²C clock control register and SCL frequency

Setting value of CCR4 to CCR0					SCL frequency (at V _{IIC} =4MHz, unit : kHz) (Note1)	
CCR4	CCR3	CCR2	CCR1	CCR0	Standard clock mode	High-speed clock mode
0	0	0	0	0	Setting disabled	Setting disabled
0	0	0	0	1	Setting disabled	Setting disabled
0	0	0	1	0	Setting disabled	Setting disabled
0	0	0	1	1	- (Note2)	333
0	0	1	0	0	- (Note2)	250
0	0	1	0	1	100	400 (Note3)
0	0	1	1	0	83.3	166
↓	↓	↓	↓	↓	500 / CCR value (Note3)	1000 / CCR value (Note3)
1	1	1	0	1	17.2	34.5
1	1	1	1	0	16.6	33.3
1	1	1	1	1	16.1	32.3

Notes 1:Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at V_{IIC} = 4 MHz). “H” duration of the clock fluctuates from –4 to +2 machine cycles in the standard clock mode, and fluctuates from –2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because “L” duration is extended instead of “H” duration reduction. These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

2:Each value of SCL frequency exceeds the limit at V_{IIC} = 4 MHz or more. When using these setting value, use V_{IIC} = 4 MHz or less. Refer to I²C system clock selection bits (bit 6, 7 of I²C control register 1) on V_{IIC}.

3:The data formula of SCL frequency is described below:

V_{IIC}/(8 X CCR value) Standard clock mode

V_{IIC}/(4 X CCR value) High-speed clock mode (CCR value ≠ 5)

V_{IIC}/(2 X CCR value) High-speed clock mode (CCR value = 5)

Do not set 0 to 2 as CCR value regardless of V_{IIC} frequency.

Set 100 kHz (max.) in the standard clock mode and 400 kHz (max.) in the high-speed clock mode to the SCL frequency by setting the SCL frequency control bits CCR4 to CCR0.

I²C Control Register 0

The I²C control register 0 (address 0323₁₆, 0333₁₆, 0313₁₆) of channel 0, 1 controls data communication format.

•Bits 0 to 2: Bit counter (BC0–BC2)

These bits decide the number of bits for the next 1-byte data to be transmitted. The I²C interrupt request signal occurs immediately after the number of count specified with these bits (ACK clock is added to the number of count when ACK clock is selected by ACK bit (bit 7 of address 0324₁₆, 0334₁₆, 0314₁₆)) have been transferred, and BC0 to BC2 are returned to "0002".

Also when a START condition is detected, these bits become "0002" and the address data is always transmitted and received in 8 bits.

•Bit 3: I²C interface enable bit (ES0)

This bit enables to use the multi-master I²C-BUS interface. When this bit is set to "0", the interface is disabled and the SDA and the SCL become high-impedance. When the bit is set to "1", the interface is enabled.

When ES0 = "0", the following is performed.

- 1)Set MST = "0", TRX = "0", PIN = "1", BB = "0", AL = "0", AAS = "0", and AD0 = "0", of I²C status register (Address : 0328₁₆, 0338₁₆, 0318₁₆)
- 2)Writing data to I²C data shift register (Address : 0320₁₆, 0330₁₆, 0310₁₆) is inhibited.
- 3)The TOF bit of I²C control register (Address : 0327₁₆, 0337₁₆, 0317₁₆) is cleared to "0"
- 4)I²C system clock (V_{IIC}) is stopped and the interval counter, flags are initialized.

•Bit 4: Data format selection bit (ALS)

This bit decides if the recognition of slave address should be processed. When this bit is set to "0", the addressing format is selected, so that address data will be recognized. The transfer will be processed only when a comparison is matched between the slave address and the address data or a general call is received (refer to the item of bit 1 of I²C status register: general call detection flag). When this bit is set to "1", the free data format is selected, so that slave address will not be recognized.

•Bit 5: Addressing format selection bit (DBIT SAD)

This bit selects a slave address specification format. When this bit is set to "0", the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆) are compared with address data. When this bit is set to "1", the 10-bit addressing format is selected, and all the bits of the I²C address register are compared with address data.

•Bit 6: I²C-BUS interface reset bit (IHR)

The bit is used to reset I²C-BUS interface circuit in the case that the abnormal communication occurs.

When the ES0 bit is "1" (I²C-BUS interface is enabled), writing "1" to the IHR bit makes a H/W reset.

Flags are processed as follows:

- 1)Set MST = "0", TRX = "0", PIN = "1", BB = "0", AL = "0", AAS = "0", and AD0 = "0", of I²C status register (Address : 0328₁₆, 0338₁₆, 0318₁₆)
- 2)The TOF bit of I²C control register (Address : 0327₁₆, 0337₁₆, 0317₁₆) is cleared to "0"
- 3)The interval counter, flags are initialized.

After writing "1" to IHR bit, the circuit reset processing will be finished in Max. 2.5 V_{IIC} cycles and IHR bit will be automatically cleared to "0". Fig.GC-6 shows the reset timing.

•Bit 7: I²C-BUS interface pin input level selection bit

This bit selects the input level of the SCL and SDA pins of the multi-master I²C-BUS interface. When this bit is set to "1" the P60, P61/P62, P63/P76, P77 will become SMBus input level.

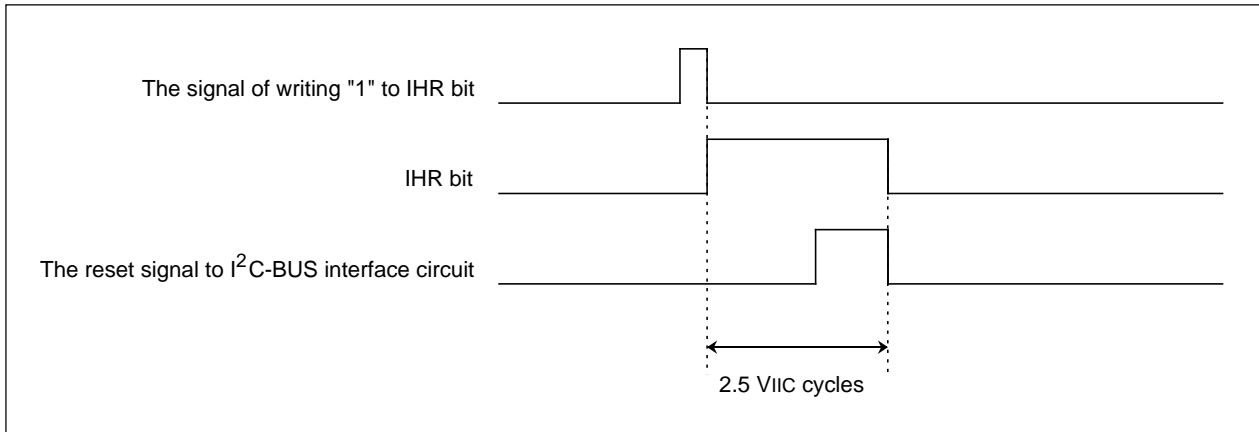


Fig.GC-6 The timing of reset to the I²C-BUS interface circuit

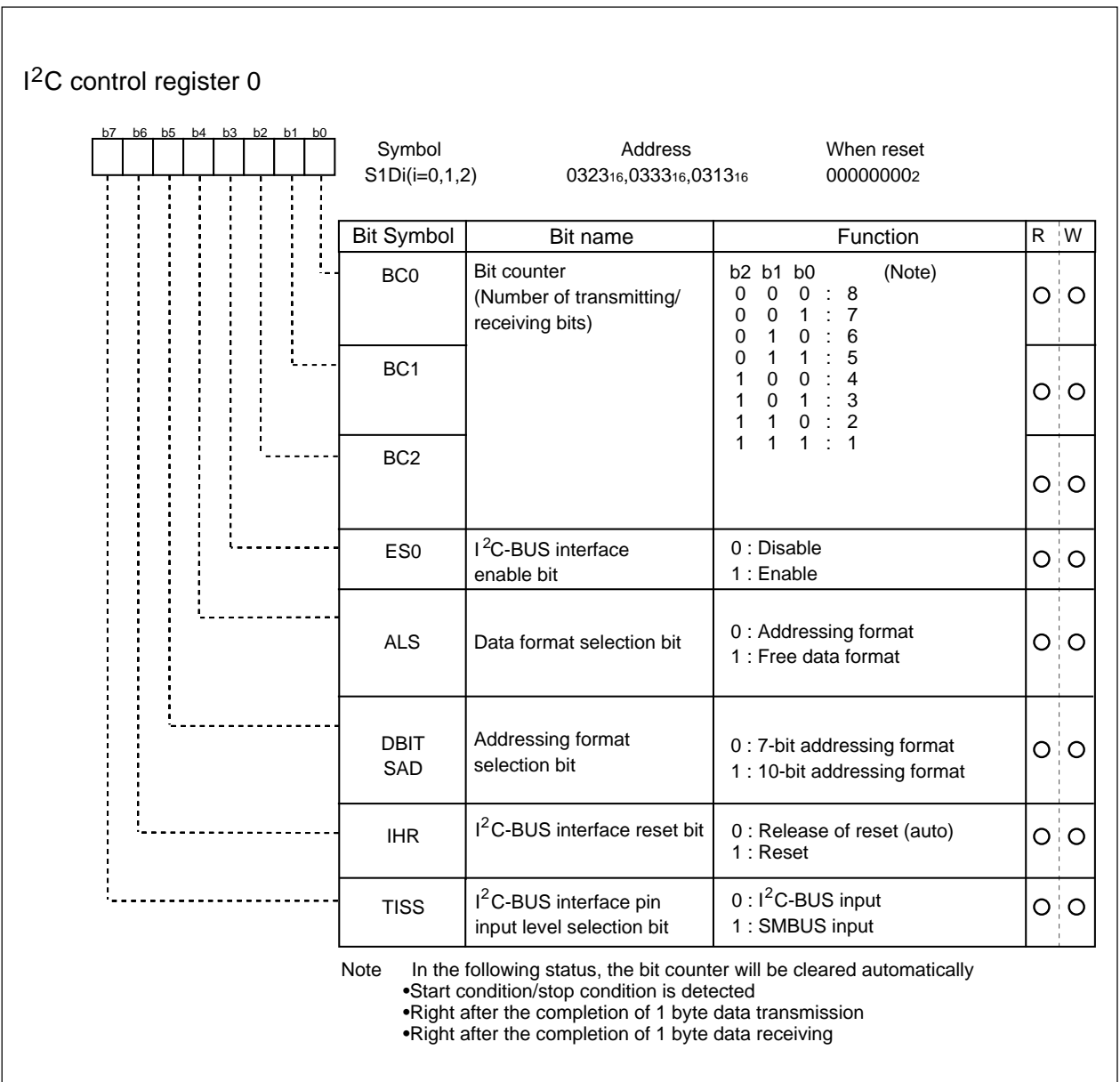


Fig.GC-7 I²C control register

I²C Status Register

The I²C status register (address 0328₁₆, 0338₁₆, 0318₁₆) controls the I²C-BUS interface status. The low-order 6 bits are read-only if it is used for status check. The high-order 2 bits can be both read and written. Regarding to the function of writing to the low-order 6 bits, refer to the method of start condition/stop condition generation described later.

•Bit 0: Last receive bit (LRB)

This bit stores the last bit value of received data and can also be used for ACK receive confirmation. If ACK is returned when an ACK clock occurs, the LRB bit is set to "0". If ACK is not returned, this bit is set to "1". Except in the ACK mode, the last bit value of received data is input. The bit will be "0" by executing a write instruction to the I²C data shift register (address 0320₁₆, 0330₁₆, 0310₁₆).

•Bit 1: General call detecting flag (AD0)

When the ALS bit is "0", this bit is set to "1" when a general call* whose address data is all "0" is received in the slave mode. By a general call of the master device, every slave device receives control data after the general call. The AD0 bit is set to "0" by detecting the STOP condition, START condition, or ES0 is "0", or reset.

*General call: The master transmits the general call address "00₁₆" to all slaves.

•Bit 2: Slave address comparison flag (AAS)

This flag indicates a comparison result of address data when the ALS bit is "0".

1) In the slave receive mode, when the 7-bit addressing format is selected, this bit is set to "1" in one of the following conditions:

- The address data, which following the start conduction, is same with upper bits data of I²C address register (Address 0322₁₆, 0332₁₆, 0312₁₆)
- A general call is received.

2) In the slave reception mode, when the 10-bit addressing format is selected, this bit is set to "1" with the following condition:

- When the address data is compared with the I²C address register (8 bits consisting of slave address and RBW bit), the first bytes agree.

3) This bit is set to "0" by executing a write instruction to the I²C data shift register (address 0320₁₆, 0330₁₆, 0310₁₆) when ES0 is set to "1". The bit is also set to "0" when ES0 is set to "0" or when reset.

•Bit 3: Arbitration lost* detecting flag (AL)

In the master transmission mode, when the SDA is made "L" by any other device, arbitration is judged to have been lost, so that this bit is set to "1". At the same time, the TRX bit is set to "0". Immediately after transmission of the byte whose arbitration was lost is completed, the MST bit is set to "0". The arbitration lost can be detected only in the master transmission mode. When arbitration is lost during slave address transmission, the TRX bit is set to "0" and the reception mode is set. Consequently, it becomes possible to detect the agreement between its own slave address and address data transmitted by another master device. The bit is cleared to "0" if writing to I²C data shift register (address 0320₁₆, 0330₁₆, 0310₁₆) when ES0 is "1".

The bit is also cleared to "0" when ES0 is set to "0" or when reset.

*Arbitration lost: The status in which communication as a master is disabled.

•Bit 4: I²C-BUS interface interrupt request bit (PIN)

This bit generates an interrupt request signal. After each byte data is transmitted, the PIN bit changes from “1” to “0”. At the same time, an I²C interrupt request signal occurs to the CPU. The PIN bit is set to “0” synchronized with the falling edge of the last internal transmitting clock (including the ACK clock) and an interrupt request signal occurs synchronized with the falling edge of the PIN bit. When the PIN bit is “0”, the SCL is kept in the “0” state and clock generation is disabled. In the ACK clock enable mode, if WIT bit (bit 1 of I²C control register 1) is set to “1”, synchronized with the falling edge of last bit clock and ACK clock, PIN bit becomes to “0” and I²C interrupt request is generated (Refer to the description on bit 1 of I²C control register 1: the data reception completion interrupt enable bit). Fig.GC-9 shows the timing of I²C interrupt request generation. The bit is read-only, the value should be “0” in writing.

The PIN bit is set to “0” in one of the following condition:

- Executing a write instruction to the I²C data shift register (address 0320₁₆, 0330₁₆, 0310₁₆).
- Executing a write instruction to the I²C clock control register (Address : 0324₁₆, 0334₁₆, 0314₁₆) (only when WIT is “1” and internal WAIT flag is “1”)
- When the ES0 bit is “0”
- At reset

The PIN bit is set to “0” in one of the following condition:

- Immediately after the completion of 1-byte data transmission (including arbitration lost is detected)
- Immediately after the completion of 1-byte data reception
- In the slave reception mode, with ALS = “0” and immediately after the completion of slave address agreement or general call address reception
- In the slave reception mode, with ALS = “1” and immediately after the completion of address data reception

•Bit 5: Bus busy flag (BB)

This bit indicates the in-use status the bus system. When this bit is set to “0”, bus system is not busy and a START condition can be generated. The BB flag is set/reset by the SCL, SDA pins input signal regardless of master/slave. This flag is set to “1” by detecting the start condition, and is set to “0” by detecting the stop condition. The condition of the detecting is set by the start/stop condition setting bits (SSC4–SSC0) of the I²C start/stop condition control register (address 0325₁₆, 0335₁₆, 0315₁₆). When the ES0 bit (bit 3) of the I²C control register (address 0323₁₆, 0333₁₆, 0313₁₆) is “0” or reset, the BB flag is set to “0”. For the writing function to the BB flag, refer to the sections “START Condition Generating Method” and “STOP Condition Generating Method” described later.

•Bit 6: Communication mode specification bit (transfer direction specification bit: TRX)

This bit decides a direction of transfer for data communication. When this bit is “0”, the reception mode is selected and the data from a transmitting device is received. When the bit is “1”, the transmission mode is selected and address data and control data are output onto the SDA synchronized with the clock generated on the SCL. This bit can be set/reset by software or hardware. This bit is set to “1” by hardware in the following condition:

In slave mode with ALS = “0”, if the AAS flag is set to “1” after the address data reception and the received $\overline{R/W}$ bit is “1”.

This bit is set to “0” by hardware in one of the following conditions:

- When arbitration lost is detected.
- When a STOP condition is detected.
- When a start condition is prevented by the start condition duplication preventing function (**Note**).
- When a start condition is detected with MST = “0”.
- When ACK non-return is detected with MST = “0”.
- When ES0 = “0”.
- At reset

•**Bit 7: Communication mode specification bit (master/slave specification bit: MST)**

This bit is used for master/slave specification for data communication. When this bit is “0”, the slave is specified, so that a START condition and a STOP condition generated by the master are received. The data communication is performed synchronized with the clock generated by the master. When this bit is “1”, the master is specified and a START condition and a STOP condition are generated.

Additionally, the clocks required for data communication are generated on the SCL.

This bit is set to “0” by hardware in one of the following conditions.

- Immediately after the completion of 1-byte data transfer when arbitration lost is detected.
- When a STOP condition is detected.
- Writing a start condition is prevented by the start condition duplication preventing function (**Note**).
- At reset

Note: START condition duplication preventing function The MST, TRX, and BB bits is set to “1” at the same time after confirming that the BB flag is “0” in the procedure of a START condition occurrence.

However, when a START condition by an other master device occurs and the BB flag is set to “1” immediately after the contents of the BB flag is confirmed, the START condition duplication preventing function makes the writing to the MST and TRX bits invalid. The duplication preventing function becomes valid from the rising of the BB flag to reception completion of slave address. Refer to the method on the start condition generation in detail.

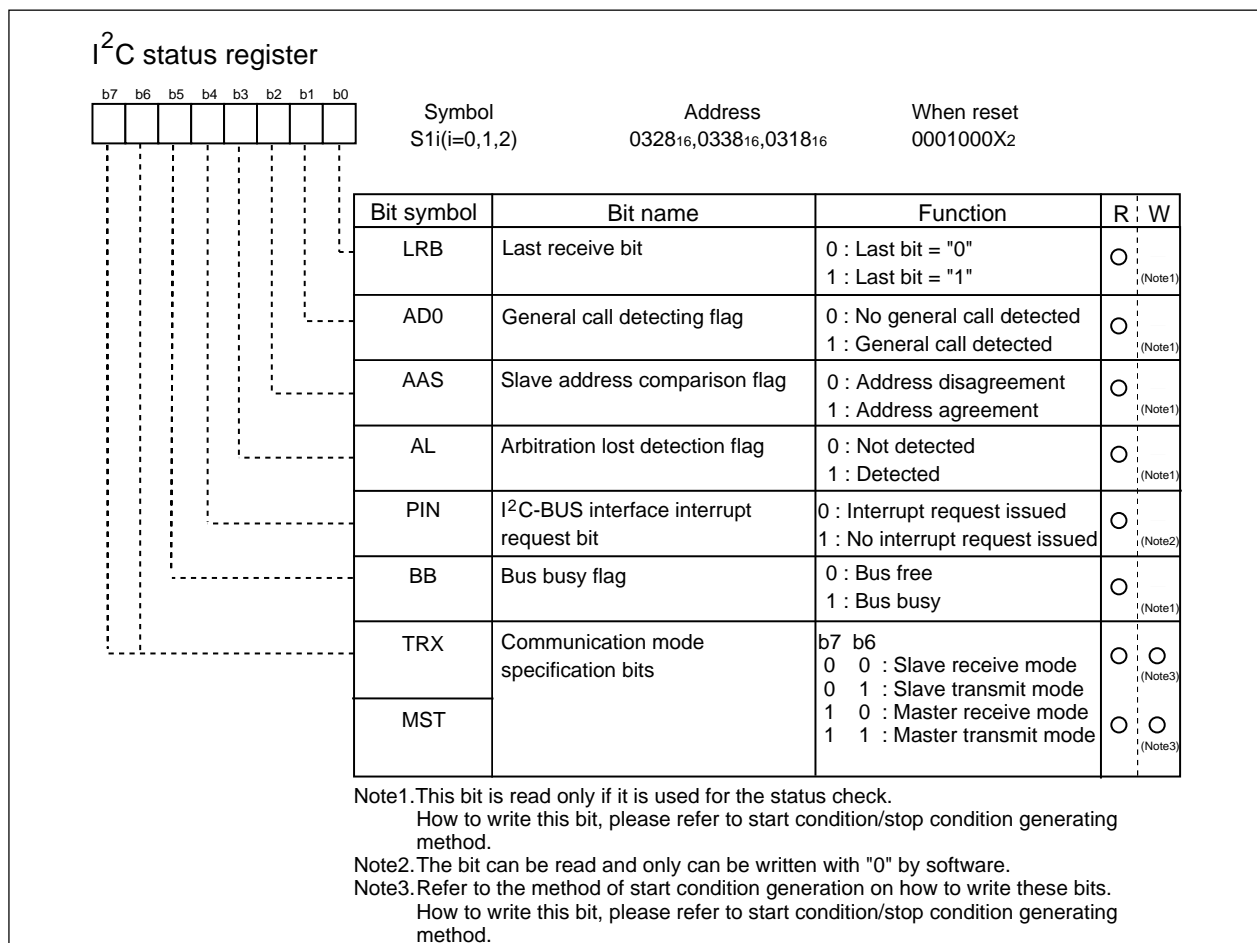


Fig.GC-8 I²C status register

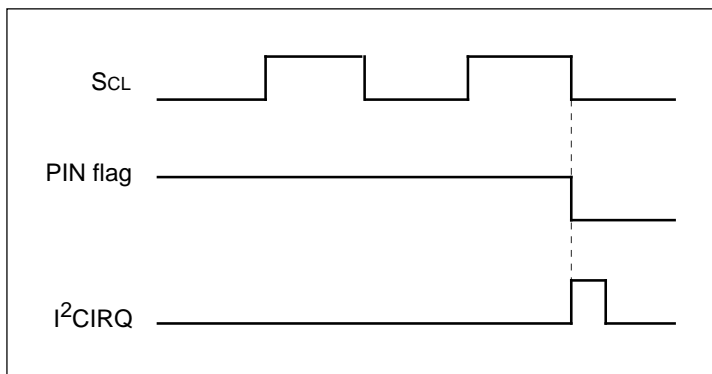


Fig.GC-9 Interrupt request signal generating timing

I²C0, I²C1 control register 1

I²C control register 10, 11, 12 (address 0326₁₆, 0336₁₆, 0316₁₆) controls I²C-BUS interface circuit.

•Bit 0 : Interrupt enable bit by STOP condition (SIM)

It is possible for I²C-BUS interface to request an interrupt by detecting a STOP condition. If the bit set to “1”, an interrupt from I²C-BUS interface occurs by detecting a STOP condition (There is no change for PIN flag)

•Bit 1: Interrupt enable bit at the completion of data receiving (WIT)

When with-ACK mode (ACK bit = “1”) is specified, by enabling the interrupt at the completion of data receiving (WIT bit = “1”), the I²C interrupt request occurs and PIN bit becomes “0” synchronized with the falling edge of last data bit clock. SCL is fixed “L” and the generation of ACK clock is suppressed.

Table GC-3 and Fig.GC-10 show the I²C interrupt request timing and the method of communication restart. After the communication restart, synchronized with the falling edge of ACK clock, PIN bit becomes to “0” and I²C interrupt request occurs.

Table.GC-3 Timing of interrupt generation in data receiving

The timing of I ² C interrupt generation	The method of communication restart
(1) Synchronized with the falling edge of the last data bit clock	The execution of writing to ACKBIT of I ² C clock control register. (Do not write to I ² C data shift register. The processing of ACK clock would be incorrect.)
(2) Synchronized with the falling edge of the ACK clock	The execution of writing to I ² C data shift register

The state of internal WAIT flag can be read out by reading the WIT bit. The internal WAIT flag is set after writing to I²C data shift register, and it is reset after writing to I²C clock control register. Consequently, which of the timing 1) and 2) of interrupt request occurring can be understood. (See Fig.GC-10)In the cases of transmission and address data reception immediately after the START condition, the interrupt request only occurs at the falling edge of ACK clock regardless of the value of WIT bit and the WAIT flag remains the reset state. Write “0” to WIT bit when in NACK is specified. (ACK bit = “0”)

•**Bits 2, 3 : Port function selection bits PED, PEC**

When ES0 bit of I²C control register 0 is set to “1”, P61/P63/P77 and P60/P62/P76 function as SCL and SDA respectively. However, if PED is set to “1”, SDA functions as output port so as to SCL if PEC is set to “1”. In this case, if “0” or “1” is written to the port register, the data can be output on to the I²C-BUS regardless of the internal SCL/SDA output signals. The functions of SCL/SDA are returned back by setting PED/PEC to “1” again.

If the ports are set in input mode, the values on the I²C-BUS can be known by reading the port register regardless of the values of PED and PEC. Table GC-4 shows the port specification.

Table.GC-4 Ports specifications

Pin name	ES0 bit	PED bit	P6/P7 port direction register	Function
P60/P62/P76	0	-	0/1	Port I/O function
	1	0	-	SDA I/O function
	1	1	-	SDA input function, port output function
P61/P63/P77	ES0 bit	PEC bit	P6/P7 port direction register	Function
	0	-	0/1	Port I/O function
	1	0	-	SCL I/O function
	1	1	-	SCL input function, port output function

•**Bits 4, 5 : SDA/SCL logic output value monitor bits SDAM /SCLM**

It is possible to monitor the logic value of the SDA and SCL output signals from I²C-BUS interface circuit. SDAM can monitor the output logic value of SDA. SCLM can monitor the output logic value of SCL. The bits are read-only. Write “0” if in writing (Writing “1” is reserved)

•**Bits 6, 7 : I²C system clock selection bits ICK0, ICK1**

These bits select the basic operation clock of I²C-BUS interface circuit. It is possible to select I²C system clock V_{IIC} among 1/2, 1/4 and 1/8 of main clock f(X_{IN}) and 1/2 of external I²C clock (ICCK)

Table.GC-5 I²C system clock selecting bits

ICK1	ICK0	I ² C system clock
0	0	V _{IIC} = 1 / 2f ₁
0	1	V _{IIC} = 1 / 4f ₁
1	0	V _{IIC} = 1 / 8f ₁
1	1	V _{IIC} = 1 / 2ICCK

Note: f₁ = f(X_{IN})

ICCK = External I²C clock

•The address reception in STOP mode /WAIT mode

It is possible for I²C-BUS interface to receive address data even in STOP mode or in WAIT mode. However the I²C system clock V_{IC} should be supplied. Table.GC-6 shows the setting list.

Table.GC-6 Clock setting to the I²C system in different operation mode.

Mode	The setting content
STOP mode	The external clock is selected as the I ² C system clock (ICK1 = 1, ICK0 = 1) and the external I ² C clock is supplied by ICCK.
WAIT mode	The external clock is selected as the I ² C system clock (ICK1 = 1, ICK0 = 1) and the external I ² C clock is supplied by ICCK. Select the peripheral function clock stop bit CMO2 (bit 2 of the system clock control register 0, address : 000616) to the state of not stopping f1,f8,f32 (CMO2 = 0) when in WAIT mode, and then execute the WAIT command.
Low power consumption mode	The external clock is selected as the I ² C system clock (ICK1 = 1, ICK0 = 1) and the external I ² C clock is supplied by ICCK.

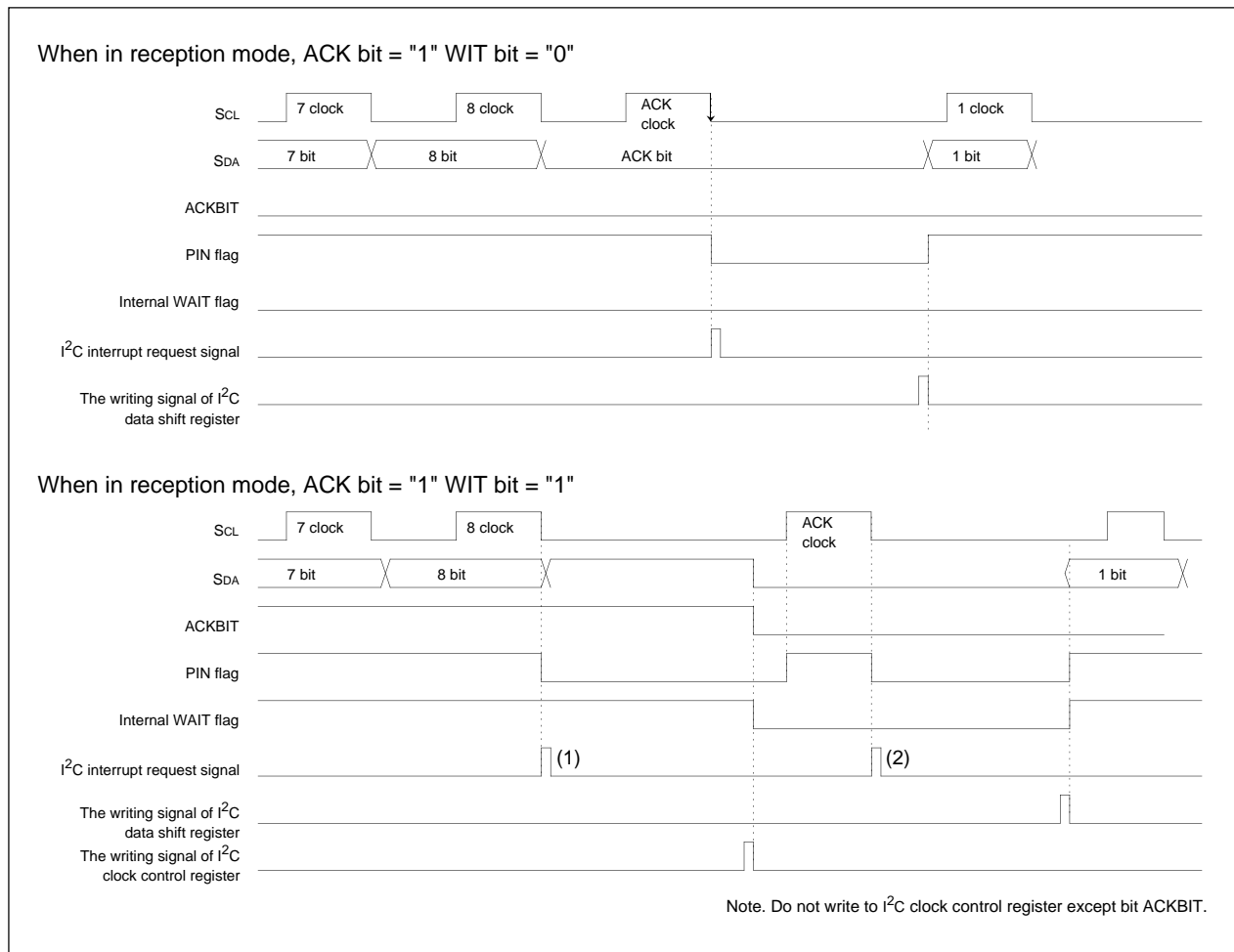


Fig.GC-10 The timing of the interrupt generation at the completion of data reception

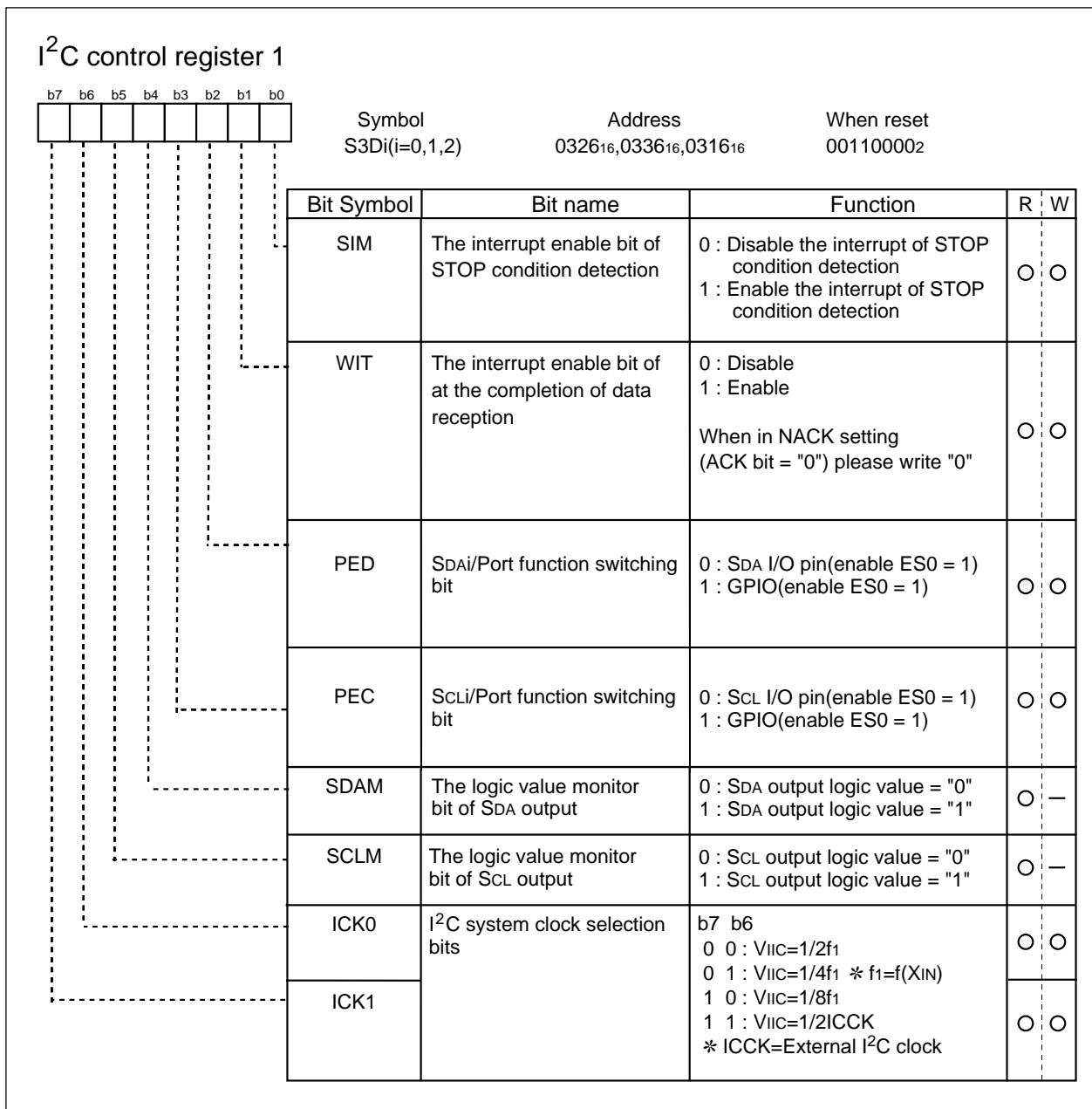


Fig.GC-11 I²C control register 1

I²C control register 2

I²C0, 1 control register 2 (address: 0327₁₆, 0337₁₆, 0317₁₆) control the detection of communication abnormality. In I²C-BUS communication, the data transfer is controlled by the SCL clock signal. The devices will stop in the communication state if SCL stops during transfer. So if the SCL clock stops in "H" state for a period of time, the I²C-BUS interface circuit can detect the time out and request an I²C interrupt. Please see Fig.GC-12.

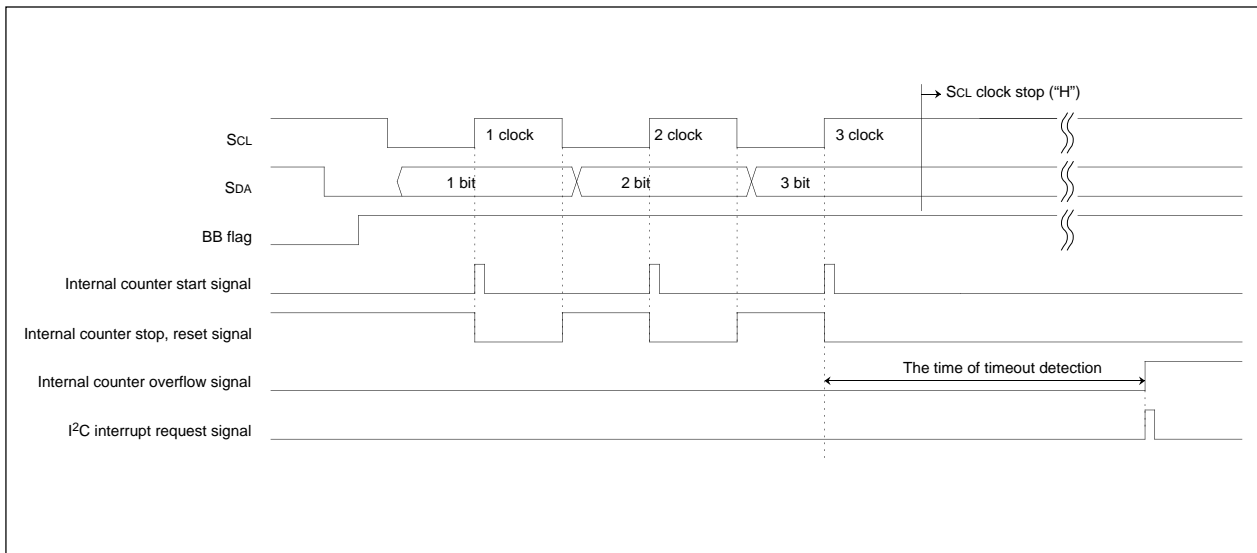


Fig.GC-12 The timing of timeout detection

•Bit0: Time out detection function enable bit (TOE)

The bit enables timeout detection function. By setting this bit to "1", the I²C interrupt request signal will be generated if the SCL clock stops in "H" state for a period of time during bus busy (BB flag = "1").

The time of time out detection which is selected by timeout detection time selection bit (TOSEL) with long time mode or short time mode will be calculated by internal counter. When time out is detected, please set "0" to I²C-BUS interface enable bit (ES0) and then process initialization.

•Bit1: Time out detection flag (TOF)

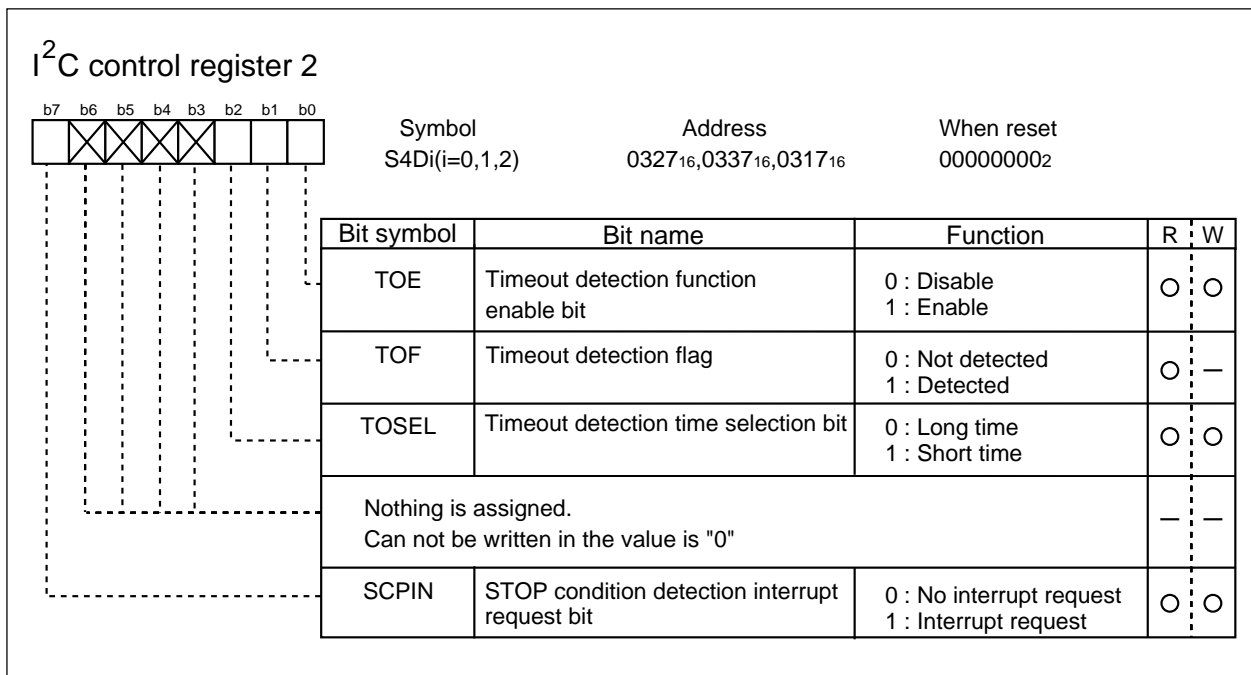
The bit is the flag showing timeout detection status. If the time which is calculated by the internal counter overflows, the time out detection flag (TOF) becomes to "1", and at the same time the I²C interrupt request signal is generated.

•Bit2: timeout detection time selection bit (TOSEL)

The bit selects timeout detection time from long time and short time mode. If TOSEL = "0", the long time mode; TOSEL = "1", the short mode is selected respectively. The long time is up counted by 16 bits counter and the short time is up counted by 14 bits counter based on I²C system clock (V_{IIC}). Table GC-7 shows examples of the timeout detection time.

Table.GC-7 Examples of timeout detection time (Unit: ms)

V _{IIC} (MHz)	Long time mode	Short time mode
4	16.4	4.1
2	32.8	8.2
1	65.6	16.4

Fig.GC-13 I²C control register 2

•**Bit7: STOP condition detection interrupt request bit (SCPIN)**

The bit monitors the stop condition detection interrupt. The bit becomes to "1" when I²C-BUS interface interrupt is generated by the detecting of STOP condition. Writing "0" clears the bit and "1" can not be written.

I²C START/STOP condition control register

I²C START/STOP condition register(address 032516, 033516, 031516) controls the detection of START/STOP condition.

•Bit0-Bit4: START/STOP condition setting bits (SSC4-SSC0)

Because the release time, set up time and hold time of SCL is calculated on the base of I²C system clock(V_{IIC}). The detecting condition changes depending on the oscillation frequency and I²C system clock selecting bits. It is necessary to set the suitable value of START/STOP condition setting bits (SSC4-SSC0) so that obtain the release time, set up time and hold time corresponding to the system clock frequency. Refer to Table GC-11. Do not set odd number or "000002" to START/STOP condition setting bits. The recommended setting value to START/STOP condition setting bits (SSC4-SSC0) at each oscillation frequency under standard clock mode is shown in Table. GC-8. The detection of START/STOP condition starts immediately after the setting of ES0=1.

•Bit5: SCL/SDA interrupt pin polarity selection bit (SIP)

The interrupt can be generated by detecting the rising edge or the falling edge of SCL pin or SDA pin. SCL/SDA interrupt pin polarity selection bit selects the polarity of SCL pin or SDA pin for interrupt.

•Bit6 : SCL/SDA interrupt pin selection bit (SIS)

SCL/SDA interrupt pin selection bit selects either SCL pin or SDA pin as SCL/SDA interrupt enable pin.

Note: The SCL/SDA interrupt request may be set when the setting of I²C-BUS interface enable bit ES0 changes.

Thus set the interrupt disable before the setting of SCL/SDA interrupt pin polarity selection bit (SIP) and SCL/SDA interrupt selection bit(SIS). After that reset "0" to the interrupt request bit before enabling the interrupt.

•Bit7: START/STOP condition generation selecting bit (STSPSEL)

The bit selects the length of set up/hold time when START/STOP condition occurs. The length of set up/hold time is based on the I²C system clock cycles. Refer to Table GC-9. Set the bit to "1" if I²C system clock frequency is over 4MHz.

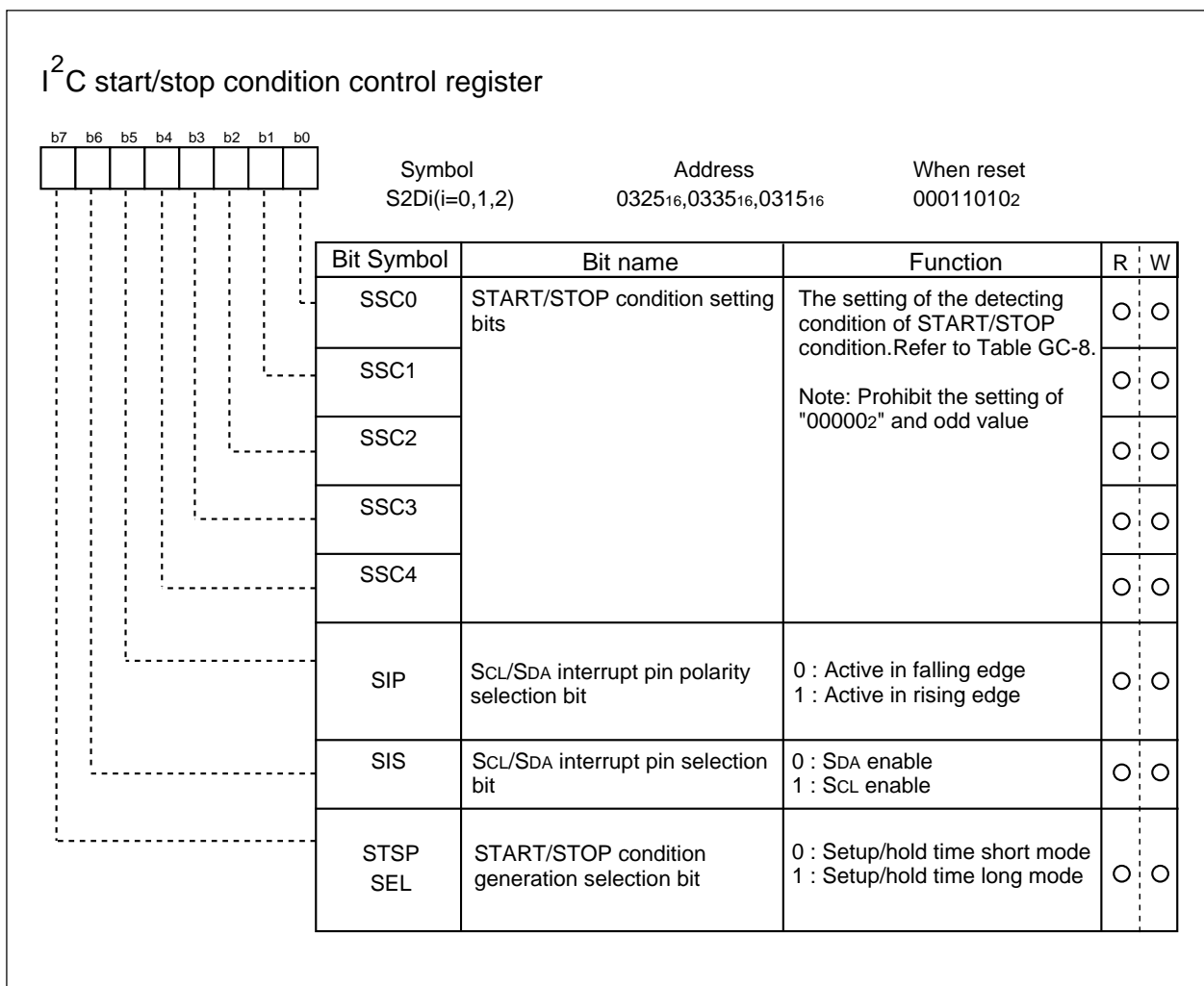


Fig.GC-14 I²C start/stop condition control register

Table.GC-8 Recommended setting value (SSC4 - SSC0) start/stop condition at each oscillation frequency

Oscillation f(XIN) (MHz)	I ² C system clock selection	I ² C system clock(MHz)	SSC4-SSC0	SCL release time(cycle)	Setup time (cycle)	Hold time (cycle)
10	1 / 2f ₁	5	XXX11110	6.2μs (31)	3.2μs (16)	3.0μs (15)
8	1 / 2f ₁	4	XXX11010	6.75μs(27)	3.5μs (14)	3.25μs(13)
			XXX11000	6.25μs(25)	3.25μs(13)	3.0μs (12)
8	1 / 8f ₁	1	XXX00100	5.0μs (5)	3.0μs (3)	2.0μs (2)
4	1 / 2f ₁	2	XXX01100	6.5μs (13)	3.5μs (7)	3.0μs (6)
			XXX01010	5.5μs (11)	3.0μs (6)	2.5μs (5)
2	1 / 2f ₁	1	XXX00100	5.0μs (5)	3.0μs (3)	2.0μs (2)

Note: Do not set odd value or "000002" to START/STOP condition setting bits.

START Condition Generation Method

When ES0 bit of the I²C control register is "1" and the BB flag of I²C status register is "0", writing "1" to the MST, TRX, and BB bits and "0" to the PIN and low-order 4 bits of the I²C status register (address 032816, 033816, 031816) simultaneously enters the standby status to generate the start condition. The start condition is generated after writing slave address data to the I²C data shift register. After that, the bit counter becomes "0002" and 1 byte SCL are output. The START condition generation timing is different in the standard clock mode and the high-speed clock mode. Refer to Fig.GC-17 the START condition generation timing diagram, and Table GC-9 the START condition generation timing table.

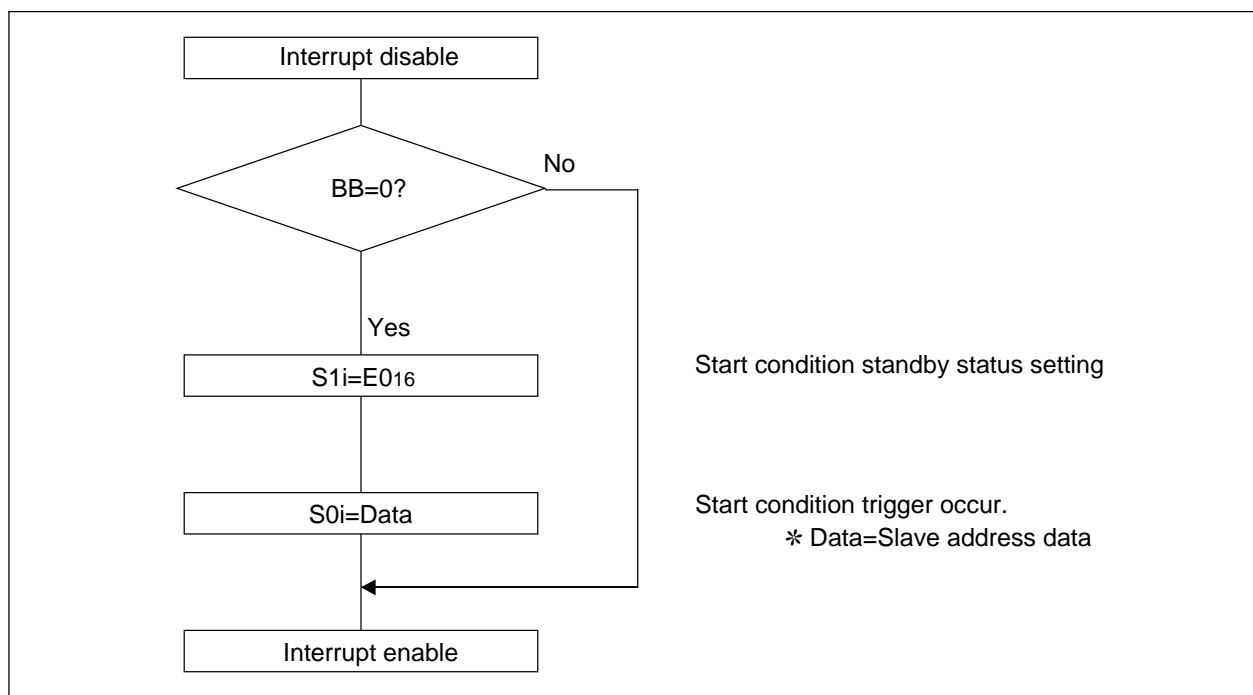


Fig.GC-15 Start condition generation flow chart

Function of protection of duplicate START condition

It is necessary to verify that the bus is not in use via BB flag before setting up a START condition. However, there is a possibility that right after the verification of BB flag, the BB flag becomes to "1" because a START condition is generated by another master device. In this case, the function to interrupt the start condition is built in. When the function starts, it works as follows:

- The prohibition of setting up START condition standby

If the START condition standby has been set up, releases it and resets the bits of MST and TRX.

- The prohibition of writing to the I²C data shift register (The prohibition of generating a START condition trigger)
- If the generation of start condition is interrupted, sets the AL flag.

The function of protection of duplicate START condition is valid from the falling edge of SDA of START condition to the completion of slave reception. Fig.GC-16 shows the valid period of the function of protection of duplicate START condition.

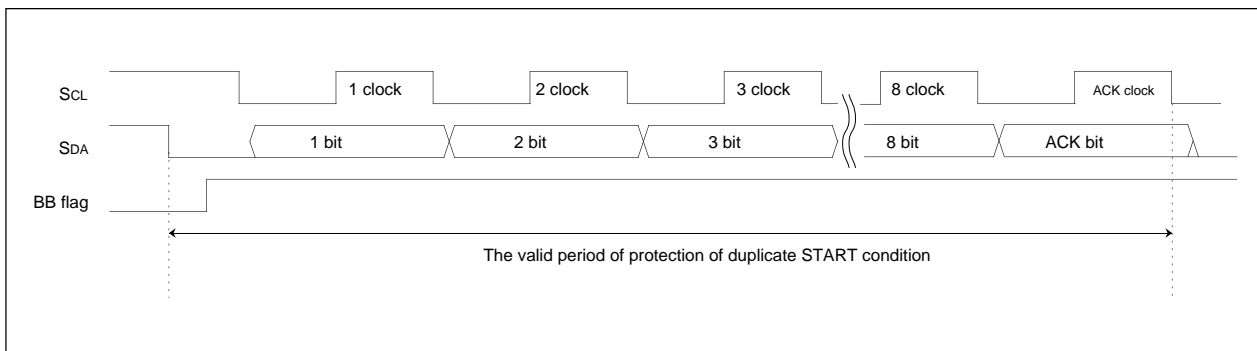


Fig.GC-16 The valid period of the function of protection of duplicate START condition

STOP Condition Generation Method

When the ES0 bit of the I²C control register is “1”, writing “1” to the MST and TRX bits, and “0” to the BB, PIN and low-order bits of the I²C status register simultaneously enters the standby status to generate the stop condition. The stop condition is generated after writing dummy data to the I²C data shift register. The STOP condition generation timing is different in the standard clock mode and the high-speed clock mode. Refer to Fig.GC-18, the STOP condition generation timing diagram, and Table GC-9, the STOP condition generation timing table. Do not write data to I²C status register and I²C data shift register, before BB flag becomes to “0” after the instruction to generate the stop condition to avoid the influence on generating STOP condition waveform.

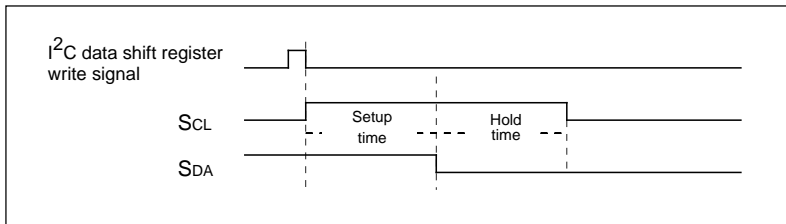


Fig.GC-17 Start condition generation timing diagram

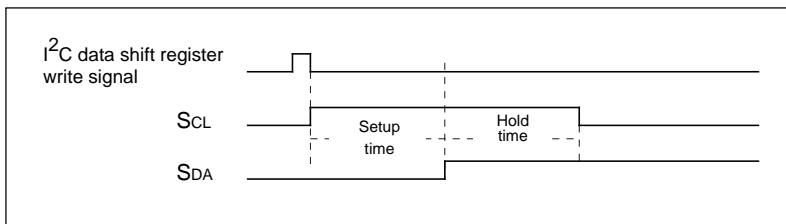


Fig.GC-18 Stop condition generation timing diagram

Table.GC-9 Start/Stop generation timing table

Item	Start/Stop condition generation selection bit	Standard clock mode	High-speed clock mode
Setup time	“0”	5.0μs (20 cycle)	2.5μs (10 cycle)
	“1”	13.0μs (52 cycle)	6.5μs (26 cycle)
hold time	“0”	5.0μs (20 cycle)	2.5μs (10 cycle)
	“1”	13.0μs (52 cycle)	6.5μs (26 cycle)

Note: V_{IIC} = 4MHz

As mentioned above, Writing “1” to MST and TRX bits.

Writing “1” or “0” to BB bit, writing “0” to PIN and low-order 4 bits, simultaneously sets up the START or STOP condition standby. It releases SDA in START condition standby, makes SDA to “L” in STOP condition standby. The signal of writing to data shift register triggers the generation of START/STOP condition. In the case of setting MST, and TRX to “1” but do not want to generate a START/STOP condition. Write “1” to the low-order 4 bits simultaneously. Fig.GC-10 illustrates the function of writing to status register.

Table.GC-10 The function of writing to status register

The value of the data writing to status register								Function
MST	TRX	BB	PIN	AL	AAS	AS0	LRB	
1	1	1	0	0	0	0	0	Setting up the START condition stand by in master transmission mode
1	1	0	0	0	0	0	0	Setting up the STOP condition stand by in master transmission mode
0/1	0/1	-	0	1	1	1	1	Setting up the communication mode (refer to the description on I ² C status register)

START/STOP Condition Detecting Operation

The START/STOP condition detection operations are shown in Fig.GC-19, GC-20 and Table.GC-11. The START/STOP condition is set by the START/STOP condition set bit. The START/STOP condition can be detected only when the input signal of the SCL and SDA pins satisfy with three conditions: SCL release time, setup time, and hold time (see Table.GC-11). The BB flag is set to “1” by detecting the START condition and is reset to “0” by detecting the STOP condition. The BB flag set/reset timing is different in the standard clock mode and the high-speed clock mode. Refer to Table GC-11, the BB flag set/reset time.

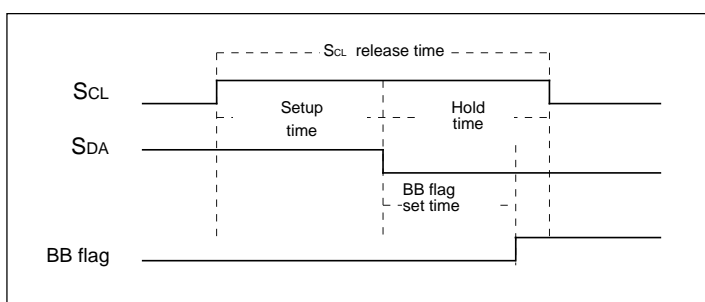


Fig.GC-19 Start condition detection timing diagram

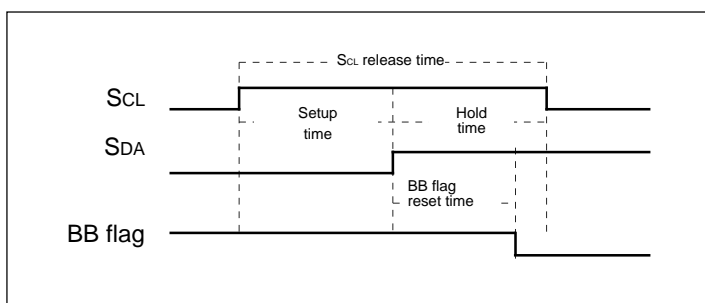


Fig.GC-20 Stop condition detection timing diagram

Table.GC-11 Start/Stop generation timing table

	Standard clock mode	High-speed clock mode
SCL release time	SSC value + 1 cycle (6.25μs)	4 cycle (1.0μs)
Setup time	$\frac{\text{SSC value}}{2} + 1 \text{ cycle} < 4.0\mu\text{s}$ (3.25μs)	2 cycle (0.5μs)
Hold time	$\frac{\text{SSC value}}{2} \text{ cycle} < 4.0\mu\text{s}$ (3.0μs)	2 cycle (0.5μs)
BB flag set/reset time	$\frac{\text{SSC value} - 1}{2} + 2 \text{ cycle}$ (3.375μs)	3.5 cycle (0.875μs)

Note: Unit : Cycle number of system clock V_{IIC}

SSC value is the decimal notation value of the START/STOP condition set bits SSC4 to SSC0.

Do not set “0” or an odd number to SSC value. The value in parentheses is an example when the I²C START/STOP condition control register is set to “1816” at V_{IIC} = 4 MHz.

Address Data Communication

There are two address data communication formats, namely, 7-bit addressing format and 10-bit addressing format. The respective address communication formats are described below.

(1) 7-bit addressing format

To adapt the 7-bit addressing format, set the DBIT SAD bit of the I²C control register 0 (address 0323₁₆, 0333₁₆, 0313₁₆) to "0". The first 7-bit address data transmitted from the master is compared with the high-order 7-bit slave address stored in the I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆). At the time of this comparison, address comparison of the RBW bit of the I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆) is not performed. For the data transmission format when the 7-bit addressing format is selected, refer to Fig.GC-21 (1) and (2).

(2) 10-bit addressing format

To adapt the 10-bit addressing format, set the DBIT SAD bit of the I²C control register 0 (address 0323₁₆, 0333₁₆, 0313₁₆) to "1". Also set the WIT bit of I²C control register 1 to "1". An address comparison is performed between the first-byte address data transmitted from the master and the 8-bit slave address stored in the I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆). At the time of this comparison, an address comparison between the RBW bit of the I²C address register (address 0320₁₆, 0330₁₆, 0310₁₆) and the R/\bar{W} bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the RBW bit which is the last bit of the address data not only specifies the direction of communication for control data, but also is processed as an address data bit.

When the first-byte address data agree with the slave address, the AAS bit of the I²C status register (address 0328₁₆, 0338₁₆, 0318₁₆) is set to "1". After the second-byte address data is stored into the I²C data shift register (address 320₁₆, 330₁₆, 0310₁₆), perform an address comparison between the second-byte data and the slave address by software. When the address data of the 2 bytes agree with the slave address, write "0" to the ACKBIT to I²C clock control register, to return an ACK. When the address data of the 2 bytes do not agree with the slave address, it does not return an ACK so that makes the finish of the communication by writing "1" to the ACKBIT. If the address data agree with each other, set the RBW bit of the I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆) to "1" by software. This processing can make the 7-bit slave address and R/\bar{W} data agree, which are received after a RESTART condition is detected, with the value of the I²C address register (address 0322₁₆, 0332₁₆, 0312₁₆). For the data transmission format when the 10-bit addressing format is selected, refer to Fig.GC-21(3) and (4).

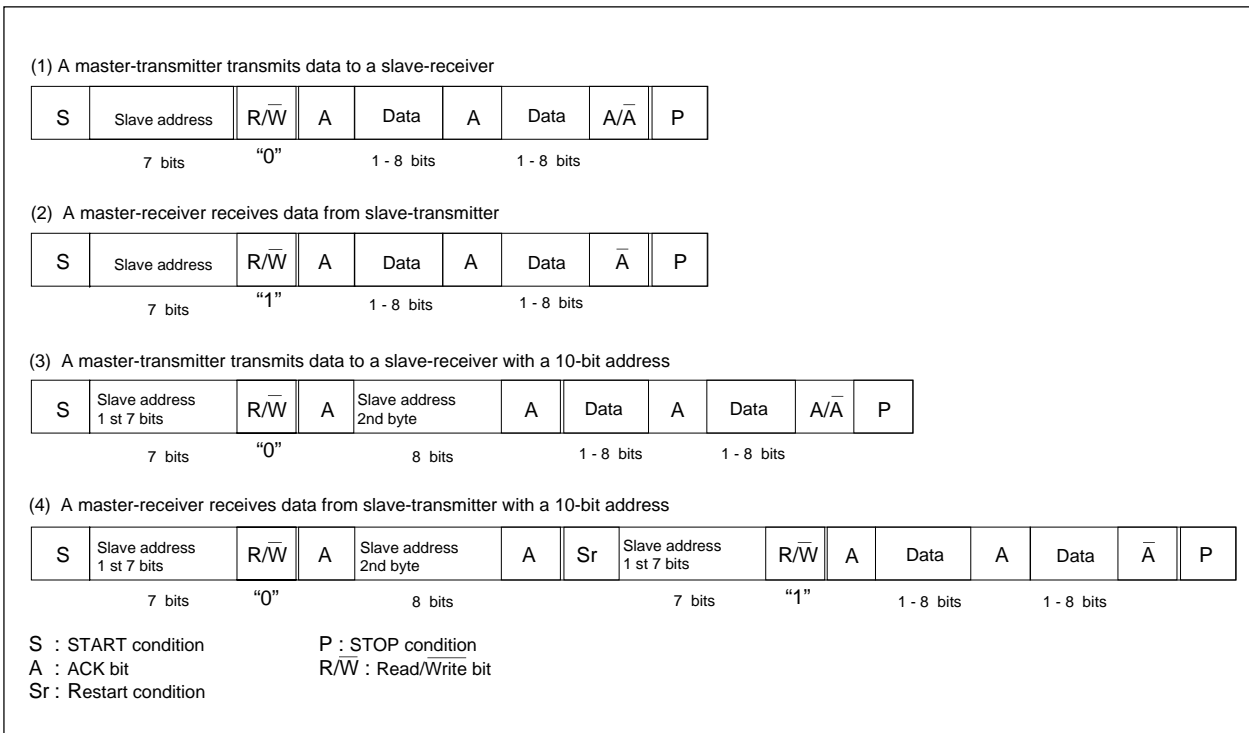


Fig.GC-21 Address data communication format

Example of Master Transmission

An example of master transmission in the standard clock mode, at the SCL frequency of 100 kHz and in the ACK return mode is shown below.

- 1) Set a slave address in the high-order 7 bits of the I²C address register and "0" into the RBW bit.
- 2) Set the ACK return mode and SCL = 100 kHz by setting "00₁₆" in the I²C control register 1 and "85₁₆" in the I²C clock control register respectively. (f(XIN)=8MHz)
- 3) Set "00₁₆" in the I²C status register so that transmission/reception mode is initialized.
- 4) Set a communication enable status by setting "08₁₆" in the I²C control register 0.
- 5) Confirm the bus free condition by the BB flag of the I²C status register.
- 6) Set "E0₁₆" in the I²C status register to setup a standby of START condition.
- 7) Set the destination address data for transmission in high-order 7 bit of I²C data shift register and set "0" in the least significant bit. And then a START condition occurs. At this time, SCL for 1 byte and an ACK clock automatically generate.
- 8) Set transmission data in the I²C data shift register. At this time, an SCL and an ACK clock automatically generate.
- 9) When transmitting control data of more than 1 byte, repeat step 8).
- 10) Set "C0₁₆" in the I²C status register to setup a STOP condition if ACK is not returned from slave reception side or transmission ends.
- 11) A STOP condition occurs when writing dummy data to I²C data shift register.

Example of Slave Reception

An example of slave reception in the high-speed clock mode, at the SCL frequency of 400 kHz, in the ACK return mode and using the addressing format is shown below.

- 1) Set a slave address in the high-order 7 bits of the I²C address register and "0" in the RBW bit.
- 2) Set the ACK clock mode and SCL = 400 kHz by setting "00₁₆" in the I²C control register 1 and "A5₁₆" in the I²C clock control register respectively. (f(XIN)=8MHz)
- 3) Set "00₁₆" in the I²C status register so that transmission/reception mode is initialized.
- 4) Set a communication enable status by setting "08₁₆" in the I²C control register 0.
- 5) When a START condition is received, an address comparison is performed.
- 6) •When all transmitted addresses are "0" (general call):
 - AD0 of the I²C status register is set to "1" and an interrupt request signal occurs.
- When the transmitted addresses agree with the address set in 1):
 - ASS of the I²C status register is set to "1" and an interrupt request signal occurs.
- In the cases other than the above AD0 and AAS of the I²C status register are set to "0" and no interrupt request signal occurs.
- 7) Set dummy data in the I²C data shift register.
- 8) After receiving 1 byte data, it returns an ACK automatically and an interrupt request signal occurs.
- 9) In the case of whether returning an ACK or not by the content of the received control data, set the WIT bit of I²C control register 1 to "1", and after writing dummy data to I²C data shift register, receives the control data.
- 10) After receiving 1 byte data, an interrupt request signal occurs, set the ACKBIT to "1" or "0" by reading the content of the data shift register, and then returns or does not return an ACK.
- 11) When receiving control data of more than 1 byte, repeat step 7) 8) or 7) 10).
- 12) When a STOP condition is detected, the communication ends.

Usage precautions

(1) Access to the registers of I²C-BUS interface circuit

The precaution of read/write to the control registers of I²C-BUS circuit is as follows.

- I²C data shift register (S0i : 0320₁₆, 0330₁₆, 0310₁₆)

Do not write the register during transfer. The transfer bit counter will be reset and makes data communication incorrect.

- I²C address register (S0Di : address 0322₁₆, 0332₁₆, 0312₁₆)

After the detection of a STOP condition, RBW is reset by H/W. Do not read/write the register at the time, because data may become undetermined. Fig.GC-22 shows the RBW bit H/W reset timing.

- I²C control register 0 (S1Di : address 0323₁₆, 0333₁₆, 0313₁₆).

After the detection of a START condition or the completion of 1 byte transfer, bit counter (bits BC0 - BC2) is reset by H/W. Do not read/write the register at the time, because data may become undetermined. Fig.GC-23, GC-24 show the bit counter H/W reset timing.

- I²C clock control register (S2i : address 0324₁₆, 0334₁₆, 0314₁₆)

Do not write to this register except ACKBIT during transfer. The I²C clock generator will be reset and makes transfer incorrect.

- I²C control register 1 (S3Di : address 0326₁₆, 0336₁₆, 0316₁₆)

Write I²C system clock selection bits when I²C-BUS interface enable bit (ES0) is in disable state. By reading the data reception completion interrupt enable bit (WIT), the internal WAIT flag will be read. Thus, do not use bit manipulation (read-modify-write instruction) to access the register.

- I²C status register (S1i : address 0328₁₆, 0338₁₆, 0318₁₆)

Do not use bit manipulation (read-modify-write instruction) to access the register because all bits of this register are changed by H/W. Do not read/write during the timing when communication mode setting bits MST and TRX are changed by H/W. Data may become undetermined. Fig.GC-22, GC-23, and GC-24 show the change timing of MST and TRX bits by H/W.

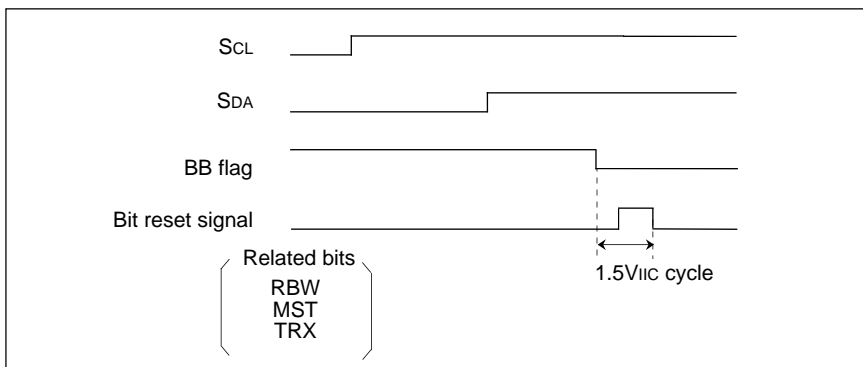


Fig.GC-22 The timing of bit reset (The detection of STOP condition)

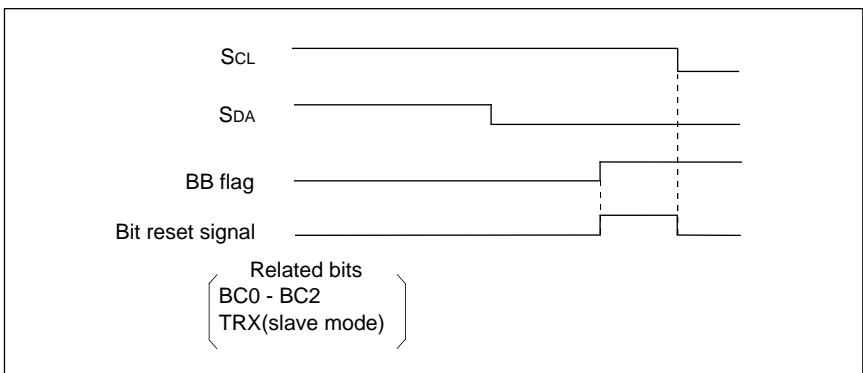


Fig.GC-23 The timing of bit reset (The detection of START condition)

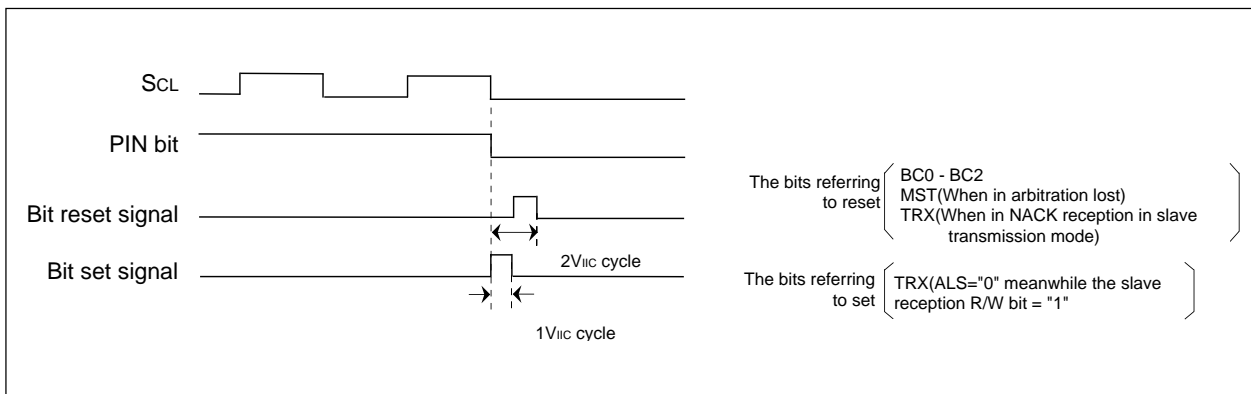


Fig.GC-24 Bit set/reset timing (at the completion of data transfer)

(2) Generation of RESTART condition

After 1 byte data transfer, a RESTART condition standby can be set up by writing "E016" to I²C status register and the SDA pin will be released. Wait in software until SDA become "H" stable and then owing to writing to I²C data shift register a START condition trigger will be generated. Fig.GC-25 shows the restart condition generation timing.

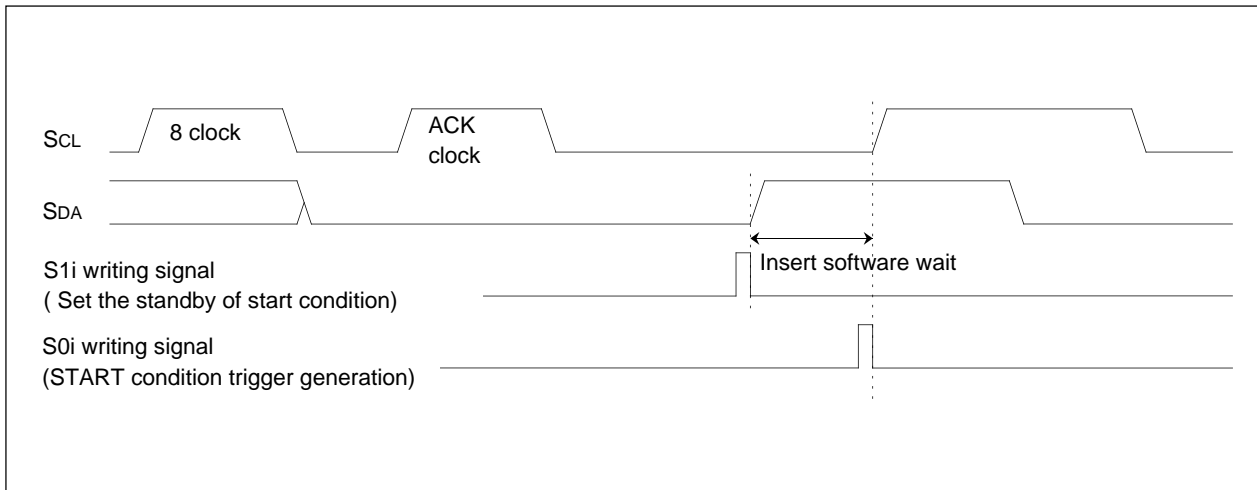


Fig.GC-25 The time of generation of RESTART condition

PS2 Interface

PS2 interface is supported by 3 channels of serial transmission/reception circuit which is based on PS2 standard specifications.

There are two signal lines, used by PS2 interface : PS2 data(DAT) and PS2 clock(CLK).

The DAT and CLK signal lines are bidirection and should be connected to positive power supply via external pull-up resistors. These two pins are N-channel open drain output. While bus is released, the states of DAT and CLK is "High". Fig.GK-1 shows the system configuration.

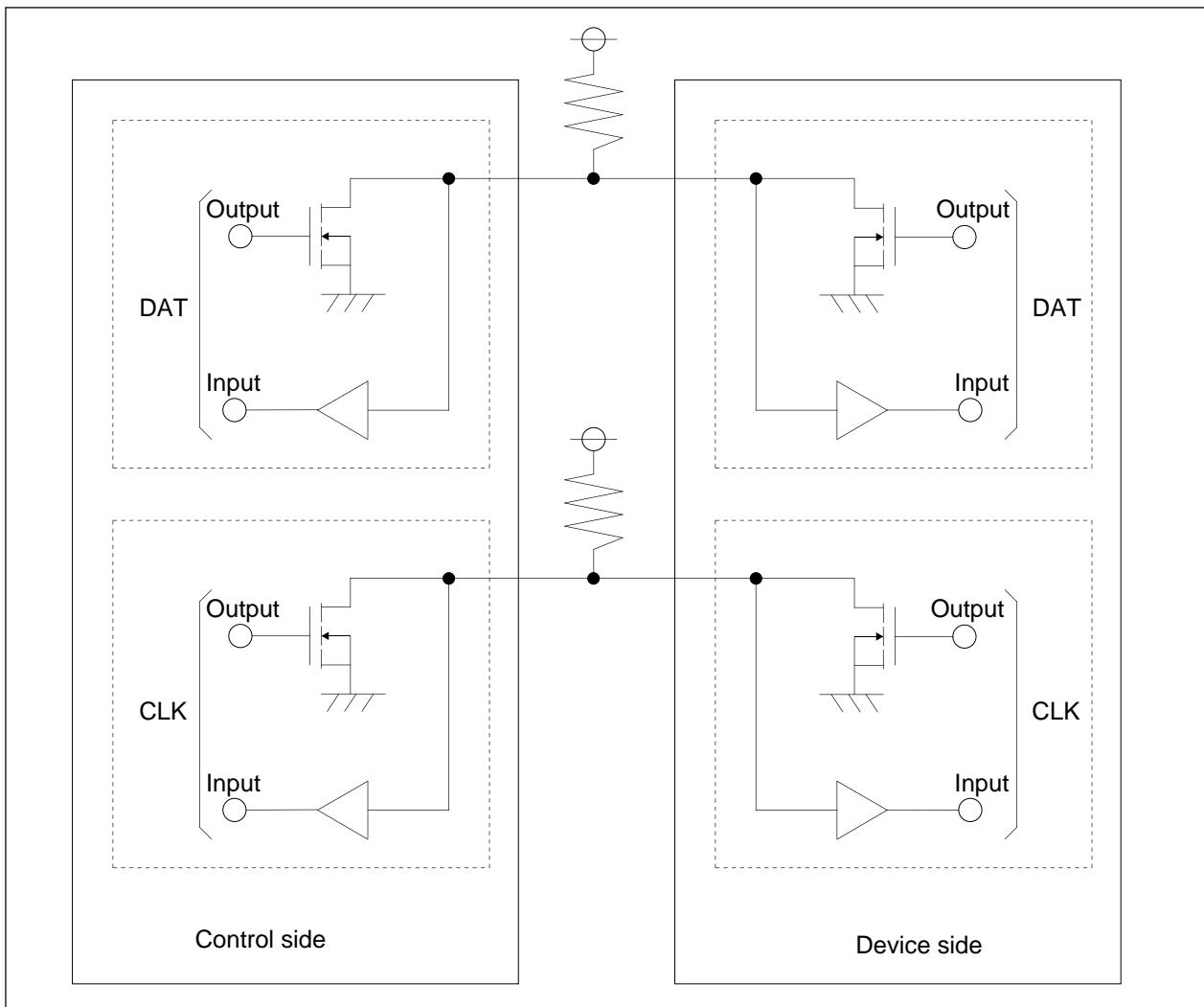


Fig.GK-1 System configuration

The PS2 interface performs 1 byte data transfer with the format shown in Fig.GK-2.

Table GK-1 shows the communication specification.

Table GK-1 Communication specification

Item	Specification
Data transfer format	*Start bit : 1 bit *Data bit : 8 bits (LSB first) *Parity bit : 1 bit (Odd) *Stop bit : 1 bit *Acknowledge : 1 bit (Transmission only)
Transfer clock	*Using the clock, which is synchronized with the sampling clock of PS2 clock (CLK)
Reception start condition	*The following conditions should be met for reception start 1) Setting reception enable bit to "1" 2) The detection of "L" on both PS2 clock (CLK) and PS2 data (DAT) lines
Transmission start condition	*The following conditions should be met for transmission start 1) Setting transmission data to PS2i shift register 2) Setting transmission enable bit to "1"
Transfer abort	*The following conditions should be met for transfer abort 1) Setting transfer interruption bit to "1" 2) The transfer completion flag becomes "1"
Interrupt request generation timing	*In reception: At the completion of stop bit reception *In transmission: At the completion of ACK bit reception. *In transfer interruption: At the completion of transfer interruption
Error detection	*Parity error (In reception) It occurs when there is a parity error in data reception *Framing error (In reception) It occurs when the detection of stop bit of reception data fails. *Abnormal acknowledge reception (In transmission) It occurs when NAK is received from a device side after the data transmission
Selection function	*Sampling clock selection Selecting the clock which samples the PS2 clock (CLK) and PS2 data (DAT)

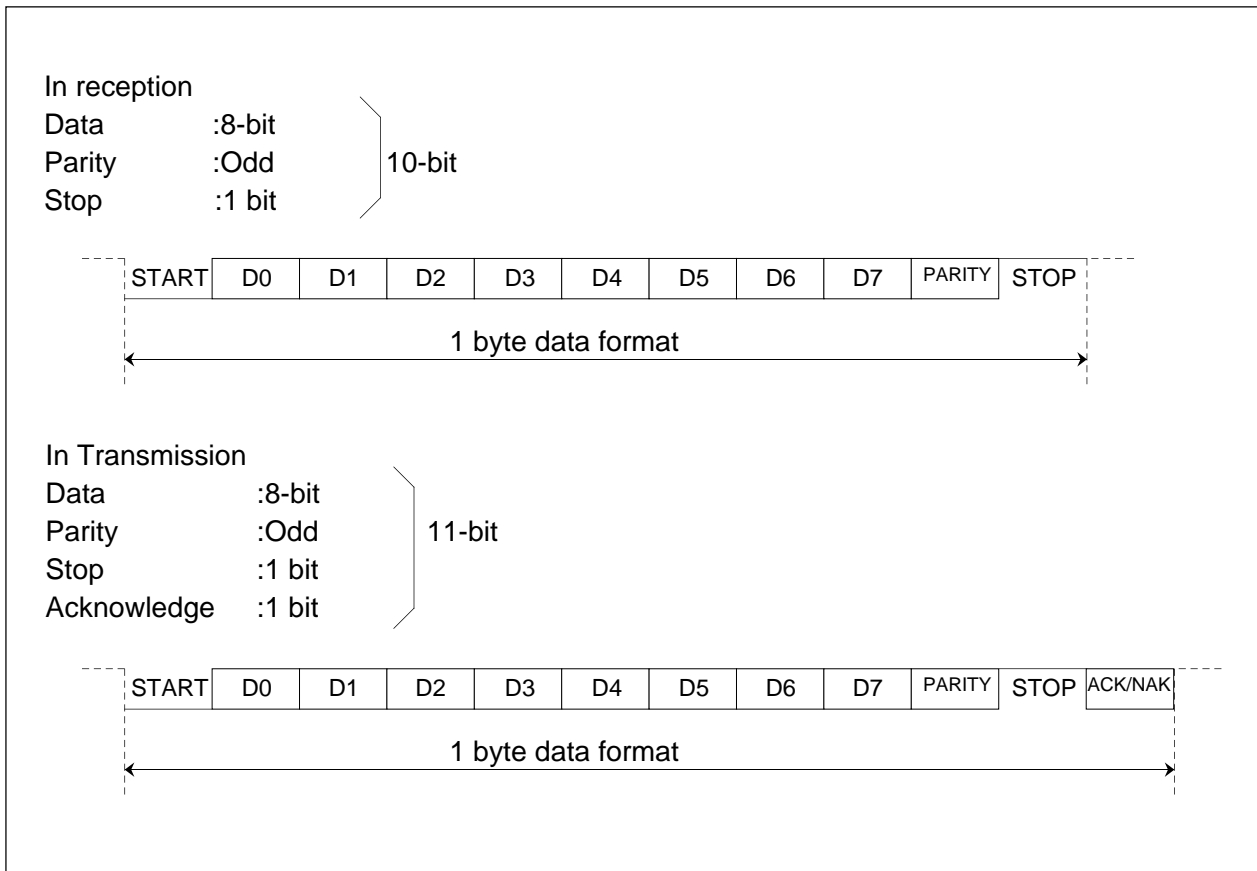
**Fig.GK-2 1 byte data format**

Fig.GK-3 shows the PS2 interface overall block diagram.
 Fig.GK-4 shows the transmission/reception block diagram.

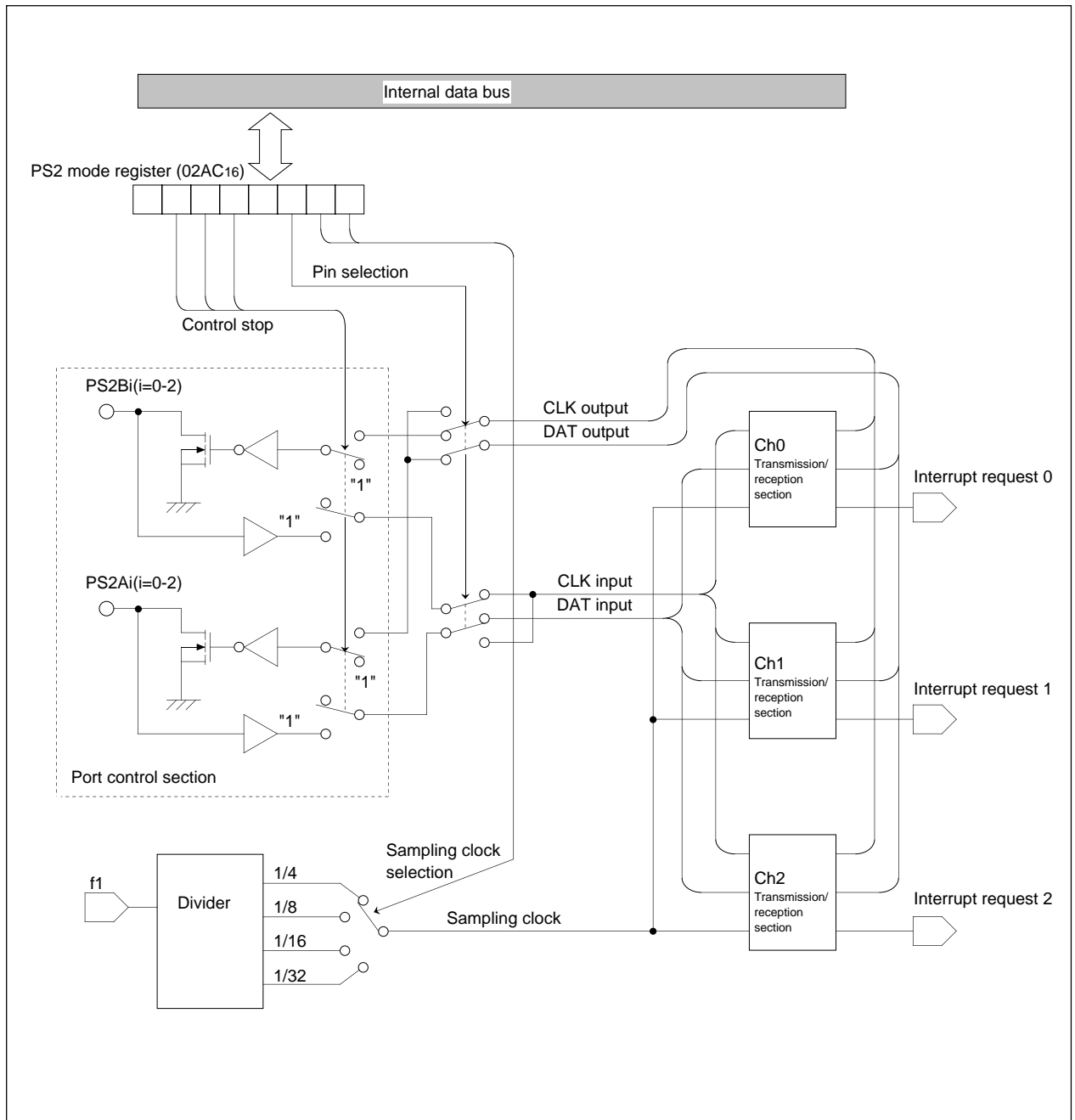


Fig.GK-3 PS2 interface block diagram

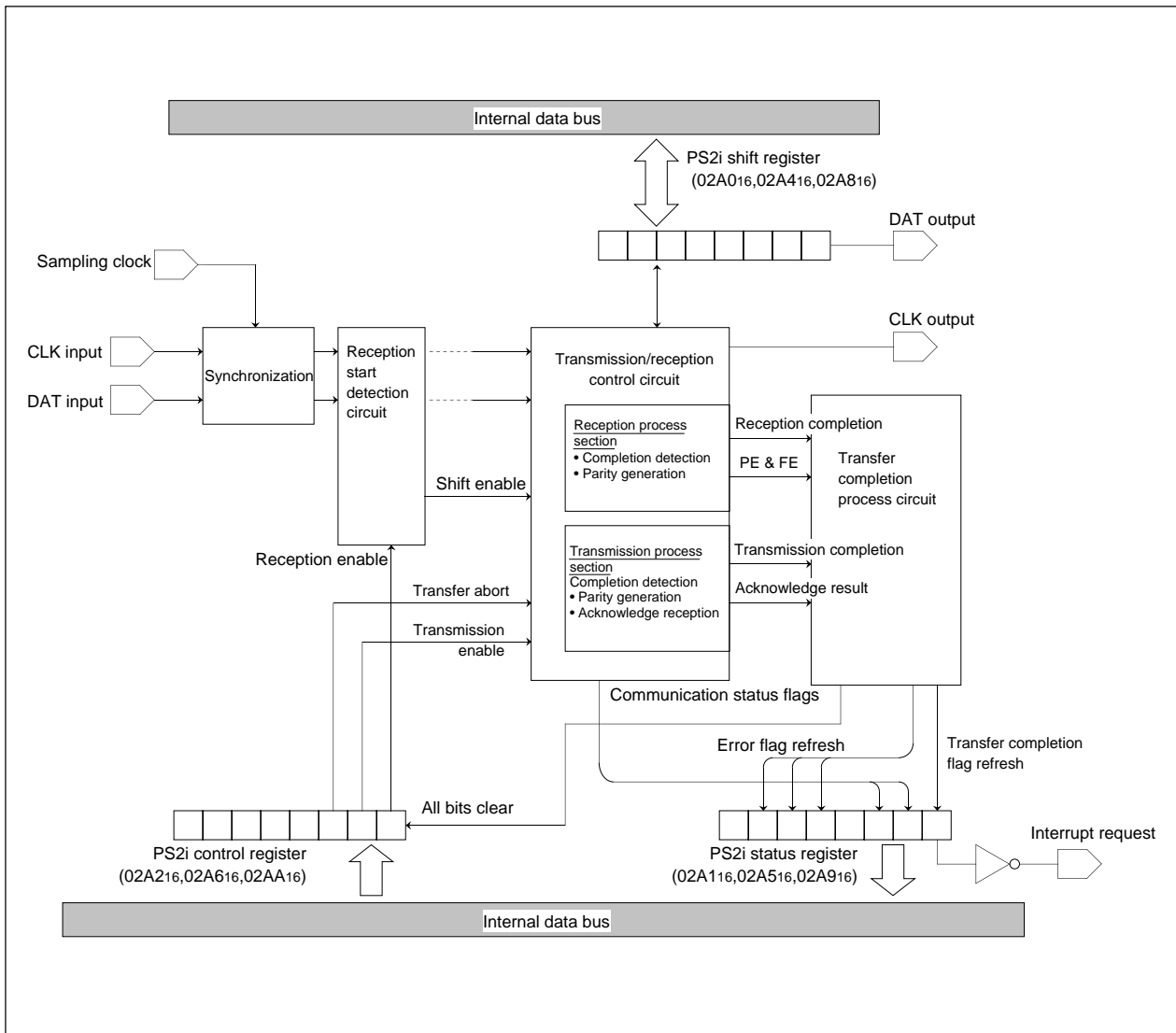


Fig.GK-4 Transmission/reception section block diagram (One channel)

(1) Register description

● PS2i shift register

• Transmission/reception data

(1) Data reception

The reception data are stored.

(2) Data transmission

By writing the transmitted data to the register, data transmission is ready to start. PS2 data (DAT) will become "L" automatically (transmission start).

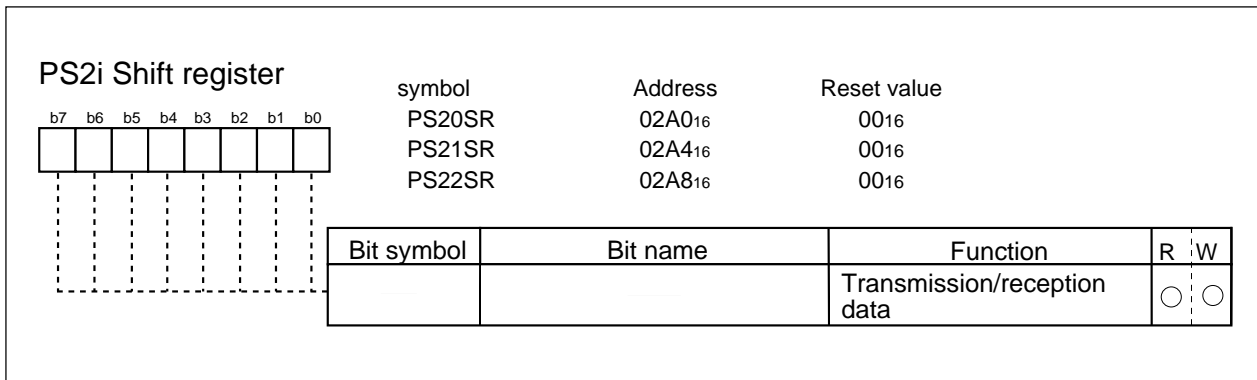


Fig.GK-5 PS2i shift register

● PS2i Control register

• Reception enable bit (REN)

The data reception is allowed when this bit is set to "1". The PS2 clock (CLK) will become to "H" (reception enable status) automatically.

This bit will be cleared to "0" automatically after the completion of data reception and PS2 clock (CLK) will become "L" (reception disable status) .

If this bit is needed wants to be cleared after setting it to "1" but before the transfer completion flag is set, set reception enable bit = "0", transfer abort request bit = "0" and process the transfer abort simultaneously.

• Transmission enable bit (TEN)

After writing transmission data to the PS2i shift register, setting the bit to "1" makes data transmission enabled and PS2 clock (CLK) will become "H" automatically (transmission enable status).

This bit will be cleared to "0" automatically after the completion of data transmission and PS2 clock (CLK) will become "L" (transmission disable status).

If this bit is needed to be cleared after setting it to "1" but before the transfer completion flag is set, set reception enable bit = "0", transfer abort request bit = "0" and process the transfer abort simultaneously.

• Transfer abort request bit (RSTOP)

This bit is used to abort the data transfer procession.

At the completion of transfer abort procession, the transfer completion flag and transfer abort flag of PS2i status register are set to "1", the bit is cleared to "0" automatically and PS2 clock (CLK) will become "L" (reception disable status).

After "L" is output to the PS2 clock (CLK), do not execute the following transmission/reception before the device recognizes the transfer abort request.

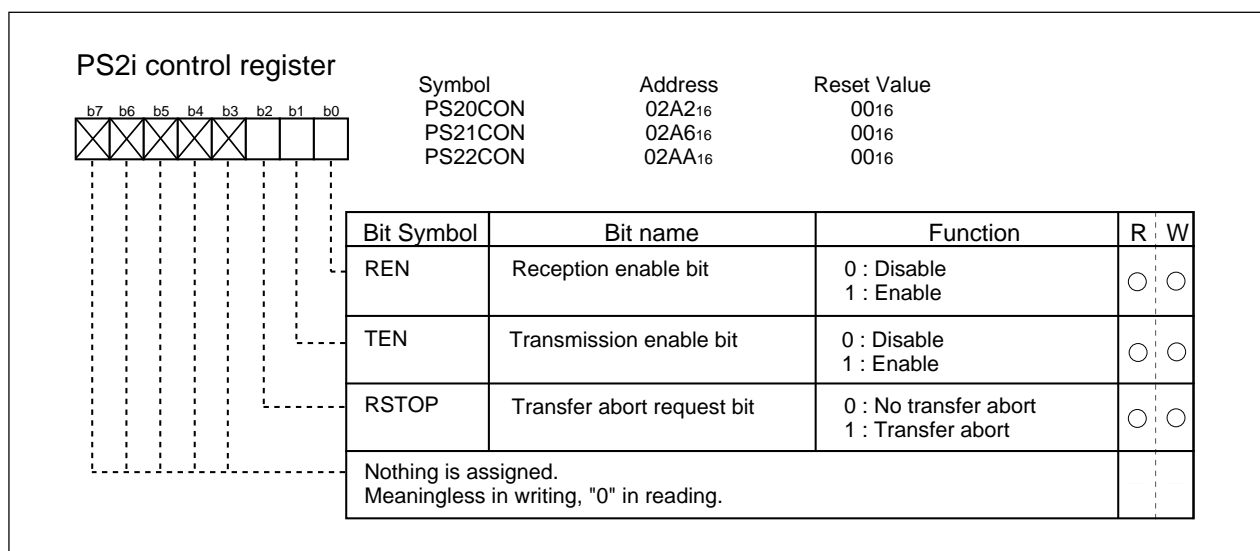


Fig.GK-6 PS2i control register

●PS2i status register**• Transfer completion flag (TI)**

The flag is set to “1” at the completion of transmission/reception and the completion of transfer abort.

The flag is cleared at read out from PS2i shift register or when the reception enable bit is changed from “0” to “1”.

• Receiving flag (RF)

The flag is set to “1” during the data reception.

The flag is cleared automatically after the data reception or after the transfer abort.

• Reception abort incognizable flag (CD)

The flag is set in the case that device side can not recognize the abort even if the reception abort is requested. (The flag is set in the period between the completion of data bit 6 reception and the completion of stop bit reception.) The flag is cleared automatically after the completion of data reception or after the completion of transfer abort.

* Note that during the period when the flag is set, the device side can not recognize the reception abort request even if the transfer abort is executed. Thus the data that the transfer abort is requested will not be resent from device side.

• Transfer status flag (TS)

The flag is set to “1” at the completion of data reception.

The flag is cleared at read out from PS2i shift register or when the reception enable bit is changed from “0” to “1”.

• Parity error flag (PE)

This bit is set to “1” when parity error occurs in received data.

The flag is cleared at read out from PS2i shift register or when the reception enable bit is changed from “0” to “1”.

• Framing error / NACK reception flag (FE)

At the completion of reception: The flag is set when the detection of stop bit of reception data fails.

At the completion of transmission: The flag is set when NAK is received from the device side.

The flag is cleared at read out from PS2i shift register, the reception enable bit or the transmission bit is changed from “0” to “1”.

• Transfer abort completion flag (CC)

This bit is set to “1” when transfer abort procession is completed.

The flag is cleared at read out from PS2i shift register, or when the reception enable bit or the transmission bit is changed from “0” to “1”.

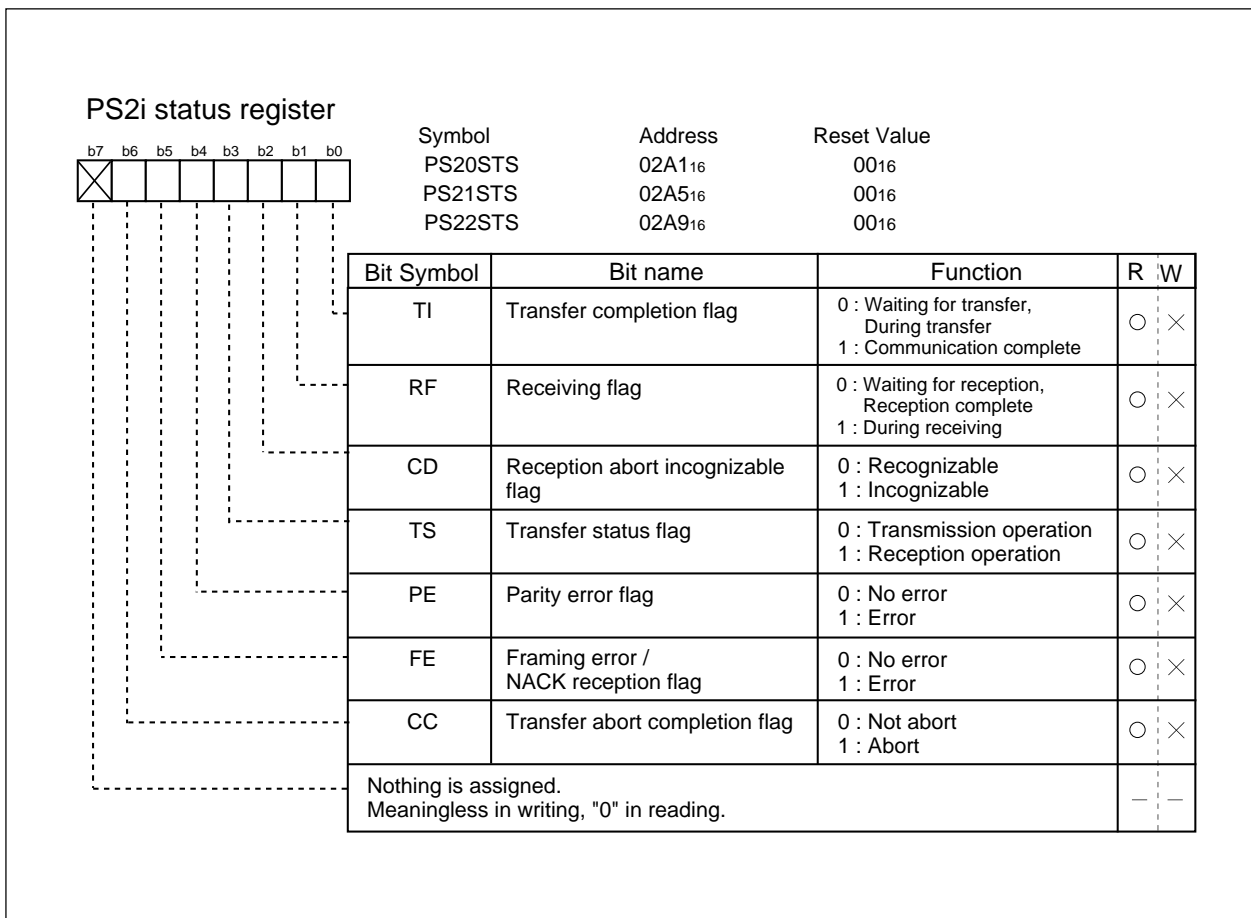


Fig.GK-7 PS2i status register

- PS2 mode register

- Sampling clock selection bits (SCK0,1)

These two bits select clock frequency for sampling PS2 clock (CLK) and PS2 data (DAT).

The relation between main clock (XIN) and sampling cycle is shown in table below.

XIN \ Setting value	1/4	1/8	1/16	1/32
8MHz	0.5 μ	1.0 μ	2.0 μ	4.0 μ
5MHz	0.8 μ	1.6 μ	3.2 μ	6.4 μ

*Sampling clock, which samples each line periodically, is used for avoiding the reflection from each line. The sampling clock will be delayed by internal circuit around 1 cycle. Thus, set the sampling clock as fast as possible.

- Pin selection bit (PSEL)

This bit is for selecting PS2 clock (CLK) or PS2 data (DAT) to connect to PS2Bi (i=0 to 2). PS2Bi are external interrupt input pins. The bit setting definition is shown in table below.

Pin selection bit	PS2Bi (i= 0 to 2)
"0"	PS2 clock (CLK)
"1"	PS2 data (DAT)

- PS2 interface enable bit (PSEN)

The PS2Ai (i= 0 to 2) and PS2Bi (i= 0 to 2) will be disconnected to hardware PS2 control section and become GPIO port when the bit is "0".

The PS2Ai and PS2Bi will be connected to hardware PS2 control section when this bit is "1".

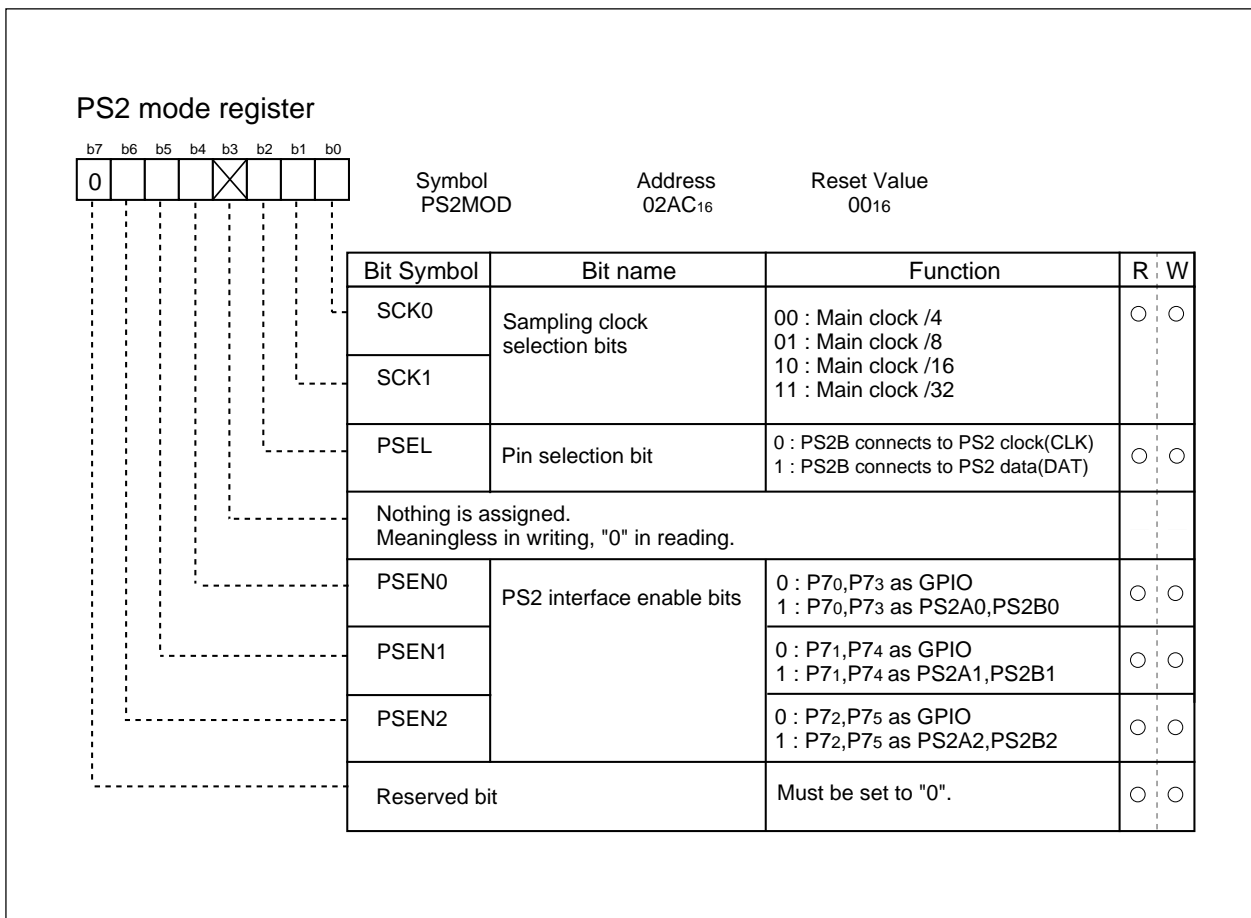


Fig.GK-8 PS2 mode register

(2) Operation description

● Basic setting

The following items should be set for PS2 mode register (address 02AC16) set when PS2 interface is used.

- PS2 interface enable bit

Set the PS2 interface enable bits (bit4 to 6 of PS2 mode register) to "1" to enable the PS2 channels to be used. At this time PS2 clock goes "Low" (Receiving disable).

- Sampling clock selection bit

Sampling clock cycle (1/4,1/8,1/16,1/32 of main clock) is selected by setting sampling clock bit (bits 0,1).

- External interrupt function support pin (PS2B) selection

This bit is used to select PS2 clock (CLK) or PS2 data (DAT) for the external interrupt function support pin (PS2B).

● Reception operation

Fig.GK-9 shows the reception operation timing.

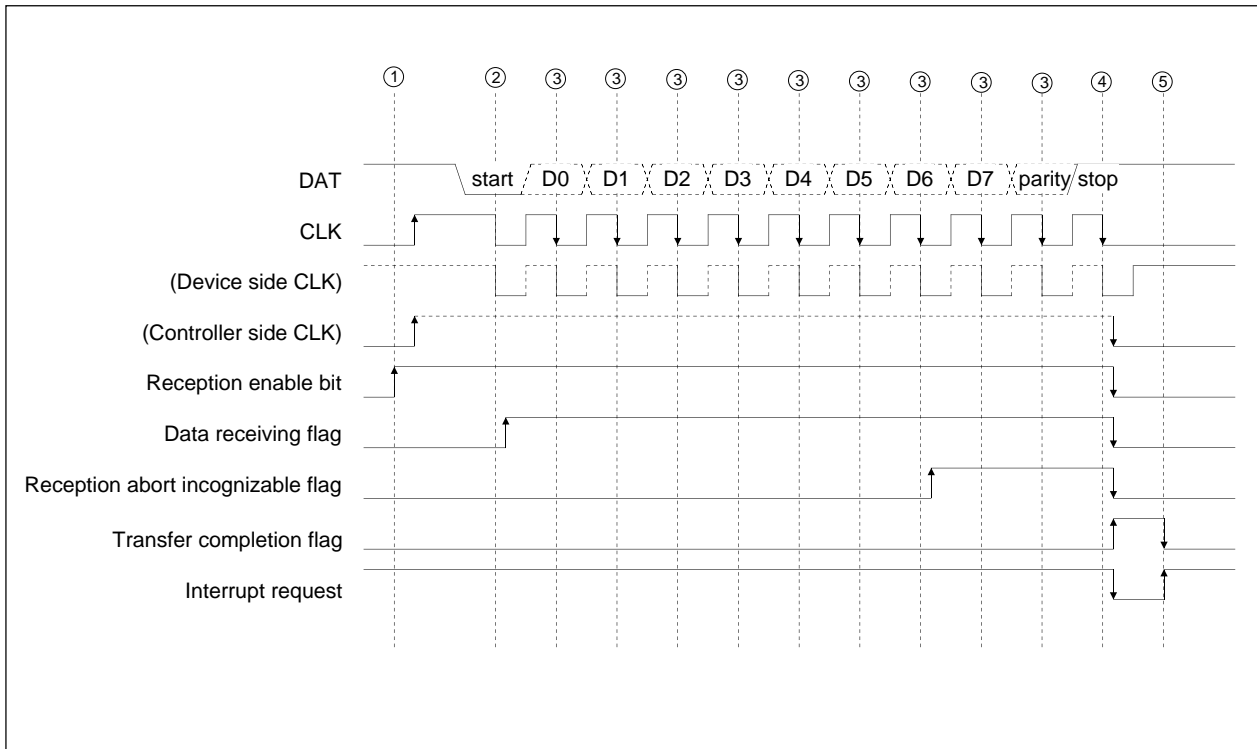


Fig.GK-9 Reception operation timing

(1) Reception enable

The reception operation is enabled by writing 01₁₆ (reception enable bit = "1") to PS2i control register (address : 02A2₁₆, 02A6₁₆, 02AA₁₆). The PS2 clock (CLK) will become "H".

(2) Reception start

The reception operation starts when both PS2 clock (CLK) and PS2 data (DAT) are detected with "L".

(3) Data reception (The reception of data and parity bits)

The content PS2 data (DAT) is read into PS2i shift register (address : 02A0₁₆, 02A4₁₆, 02A8₁₆) sequentially by the falling edge of PS2 clock (CLK). The data transfer sequence is data bit (D0 -D7) then parity bit.

(4) Reception completion (Stop bit Reception completion)

By detecting the falling edge of PS2 clock (CLK), the transfer completion flag (bit 0 of PS2i status register) is set to "1" after the update of error flag (bit 4 - 6 of PS2i status register) and the reception enable bit (bit 0 of PS2i control register) is cleared to "0". The PS2 clock (CLK) becomes "L" (reception disable status) and interrupt request occurs.

(5) Data read out

Read out data from PS2i shift register (address : 02A0₁₆, 02A4₁₆, 02A8₁₆). At this time, the error flags (Bit4 to 6) of and transfer completion flag (bit 0) of PS2i status register (address: 02A1₁₆, 02A5₁₆, 02A9₁₆) will be cleared to "0".

● Transmission operation

Fig.GK-10 shows the transmission operation timing.

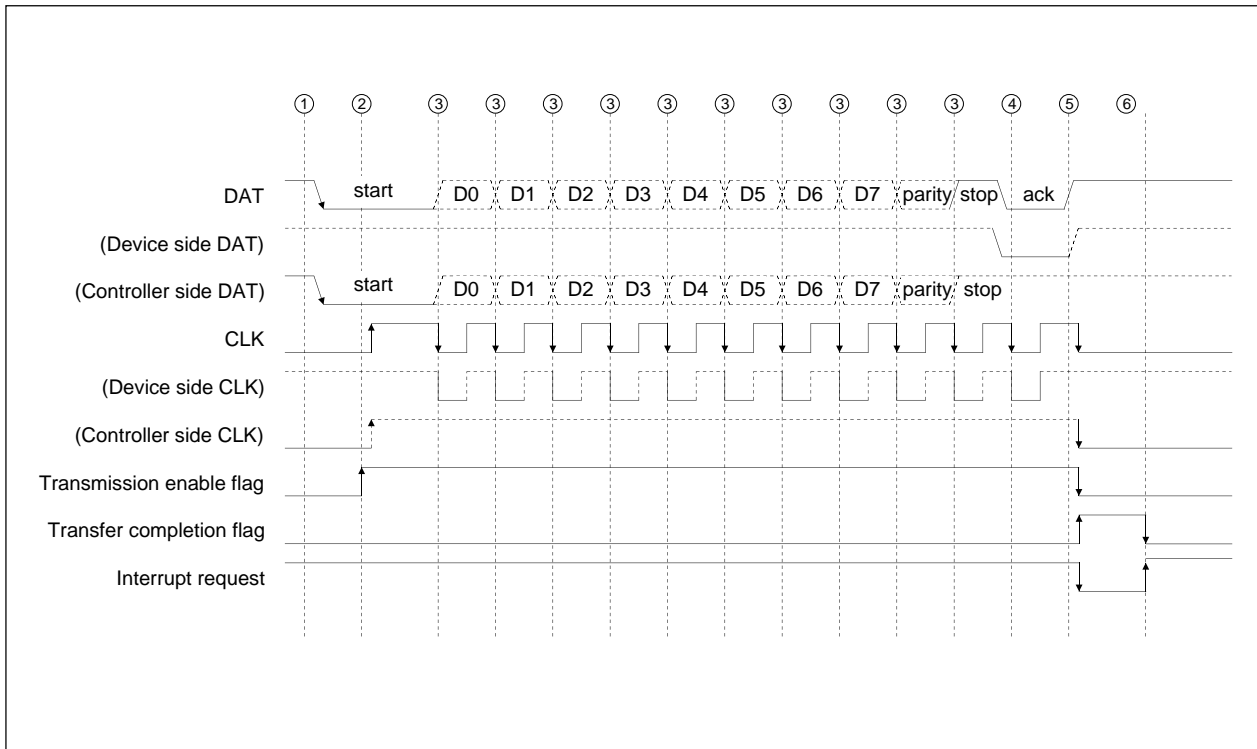


Fig.GK-10 Transmission operation timing

(1) Data writing

Write transmission data to PS2i shift register (address : 02A0₁₆, 02A4₁₆, 02A8₁₆). At this time, PS2 data will become "L" (transmission start).

(2) Transmission enable

Set 02₁₆ (Transmission enable bit = "1") to PS2i control register (address : 02A2₁₆, 02A6₁₆, 02AA₁₆) for enabling transmission operation. At this time, PS2 clock (CLK) will become "H".

(3) Data transmission (The transmission of data, parity and stop bits)

The content of PS2i shift register (address : 02A0₁₆, 02A4₁₆, 02A8₁₆) will be output to the PS2 data (DAT) sequentially by the falling edge of PS2 clock (CLK). The sequence of data transfer is data bits (D0 to D7), Parity bit, and stop bit.

(4) Acknowledge reception

The content of acknowledge bit will be read by the falling edge of PS2 clock (CLK).

(5) Communication completion

The communication operation is completed by detecting "H" on both PS2 clock (CLK) and PS2 data (DAT). After the update of error flag (bit 4 - 6 of PS2i status register), the transfer completion flag (bit 0 of PS2i status register) is set to "1" and the reception enable bit (bit 0 of PS2i control register) is cleared to "0". At this time, PS2 clock (CLK) becomes "L" (reception disable status) and the interrupt request occurs.

(6) Status clear

Read out the data from PS2i shift register (address : 02A0₁₆, 02A4₁₆, 02A8₁₆). At this time, the error flags (bits 4 to 6) and transfer completion flag (Bit0) of PS2i status register (address : 02A1₁₆, 02A5₁₆, 02A9₁₆) will be cleared to "0".

● Transfer abort operation

Fig.GK-11 shows the transfer abort operation timing.

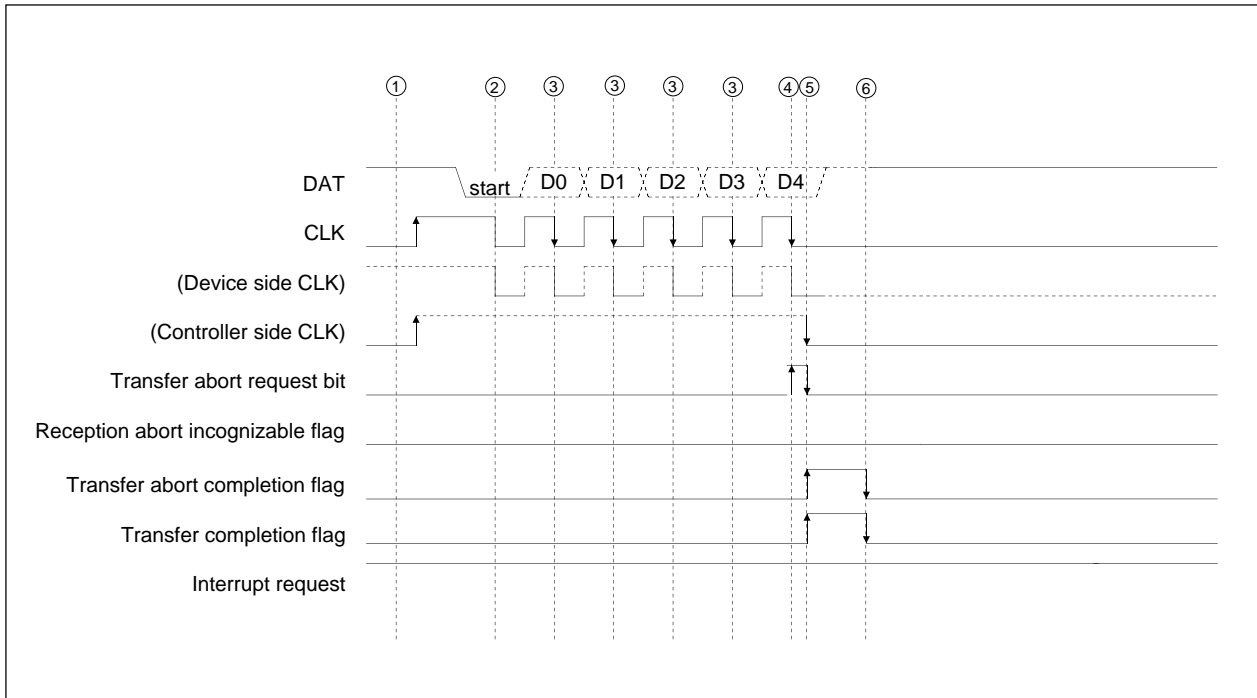


Fig.GK-11 Transfer abort operation timing (reception)

(1) - (3) Data reception operation

(4) Transfer abort request

Set 04₁₆ (transfer abort request bit = "1") to PS2i control register (address : 02A2₁₆, 02A6₁₆, 02AA₁₆).

(5) Transfer abort completion

The transfer abort completion flag (bit 6) and transfer completion flag (bit 0) of PS2i status register (address : 02A1₁₆, 02A5₁₆, 02A9₁₆) are set to "1", transfer abort request bit (bit 2) of PS2i control register (address : 02A2₁₆, 02A6₁₆, 02AA₁₆) is cleared to "0". At this time, PS2 clock (CLK) becomes "L" (reception disable status) and interrupt request occurs.

(6) Status clear

By a pseudo read of PS2i shift register (address : 02A0₁₆, 02A4₁₆, 02A8₁₆), the transfer abort completion flag (bit 6) and transfer completion flag (bit 0) of PS2i status register (address : 02A1₁₆, 02A5₁₆, 02A9₁₆) are cleared to "0".

Note: Do not execute the following transmission/reception during the period between the "L" output from PS2 clock (CLK) and the transfer abort request recognition of the device.

Programmable I/O Ports

There are 129 programmable I/O ports: P0 to P16 (excluding P85). Each port can be set independently for input or output using the direction register. Pull-up resistances can be set in 4-port unit. (except for P10 and P14). The N channel open drain ports P60 to P63, P70 to P77, P80 to P84, P130 to P137 and P85 (input only port) do not build internal pull-up resistance.

Fig.UA-1 to UA-6 show the configurations of programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices, they function as outputs regardless of the contents of the direction registers. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

Fig.UA-7 shows the configurations of direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

Note: There is no direction register bit for P85.

(2) Port registers

Fig.UA-8 shows the configurations of port registers.

These registers are used to write and read data for input and output to and from exterior. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

(3) Pull-up control registers

Fig.UA-9 and UA-10, UA-11 shows the configurations of pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports (except for P10 and P14). When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

Pull-up resistance can be set to each pin of P10 and P14.

(4) Port control register

Fig.UA-12 shows the configurations of port control register 0, 1. Fig.UA-13 shows the configurations of port control register 2, 3. The bit 0 of port control register 0 is used to read port P1 as follows:

0 : When port P1 is input port, port input level is read.

When port P1 is output port, the contents of port P1 register is read.

1 : The contents of port P1 register is read always. (neither input port nor output port)

The P0, P1, P40 to P46, P11 and P14 output type, CMOS or N channel open drain, are set by bit 0 to 6 of port control register 1 and bit 0 to 2 of port control register 3.

0 : CMOS output

1 : N channel open drain output

Exception: P42 output type is N channel open drain if either bit 4 of port control register 1 or bit 2 of port control register 3 is set to "1".

Bit 7 of port control register1 functions as below

0 : P40/P43 output is cleared by software only

1 : P40/P43 output is cleared by software or when output buffer 0 is read by host side.

The driving ability of N channel output transistors for P140 to P143 can be selected by bit 6 of port control register 2 controls as below:

0 : Driving ability of N channel open drain output transistor is LOW

1 : Driving ability of N channel open drain output transistor is HIGH

(5) Port P4/P7 input register

Fig.UA-14 shows the configurations of P4 and P7 input register.

By reading the registers, the input level of the corresponding pins can be known regardless the input/output mode. These two registers can be read regardless port direction setting. And the ports level will be read out.

Port4 : Bit 0 to bit6's level will be read out. And bit7 is always "0".

Port7 : Bit 0 to bit5's level will be read out. And bit6,7 is always "0".

(6) Port function selection register 0,1,2

Fig.UA-15, UA-16 shows the configurations of port function selection register 0,1. The port functions of UART1 input/output, TimerA0 to TimerA2 output, TimerB3,B4 input, external interrupt $\overline{INT6}$ to $\overline{INT12}$ input or I²C1,2 input/output can be switched by setting these two registers.

By setting bit0,1 of port function selection register 2, the same frequency clock with f(XIN) can be output from P110 and P111.

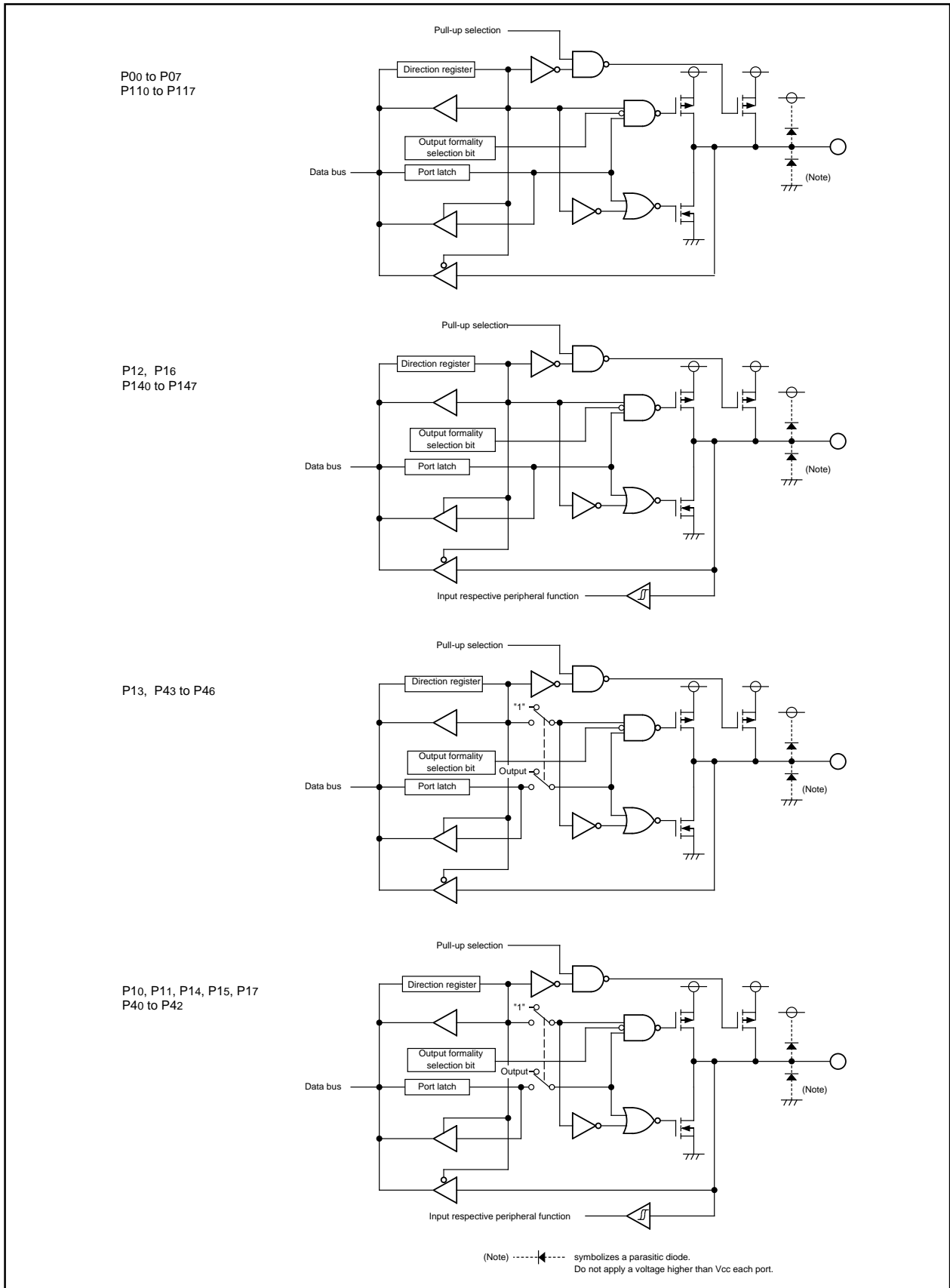


Fig.UA-1 Programmable I/O ports (1)

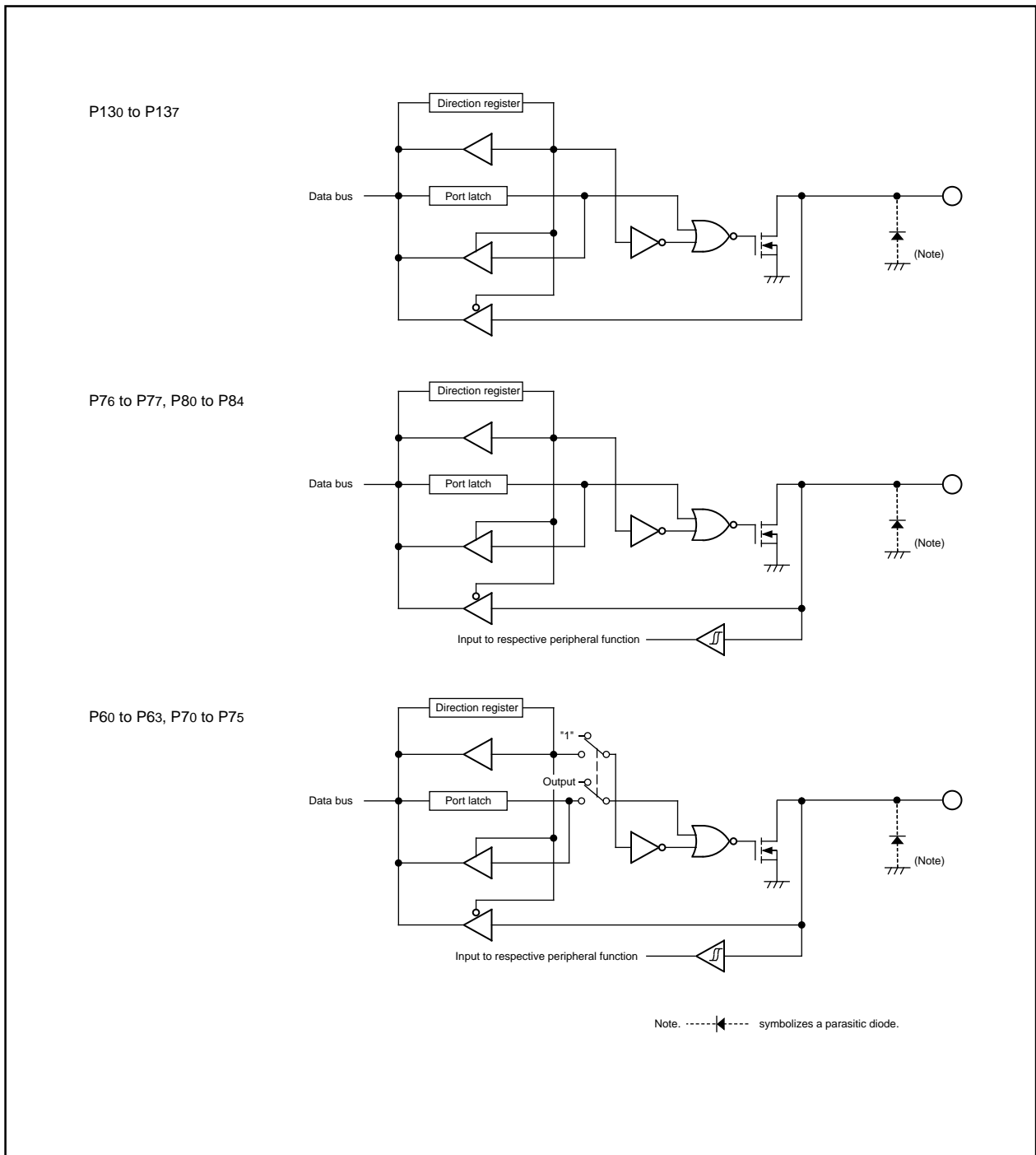


Fig.UA-3 Programmable I/O ports (3)

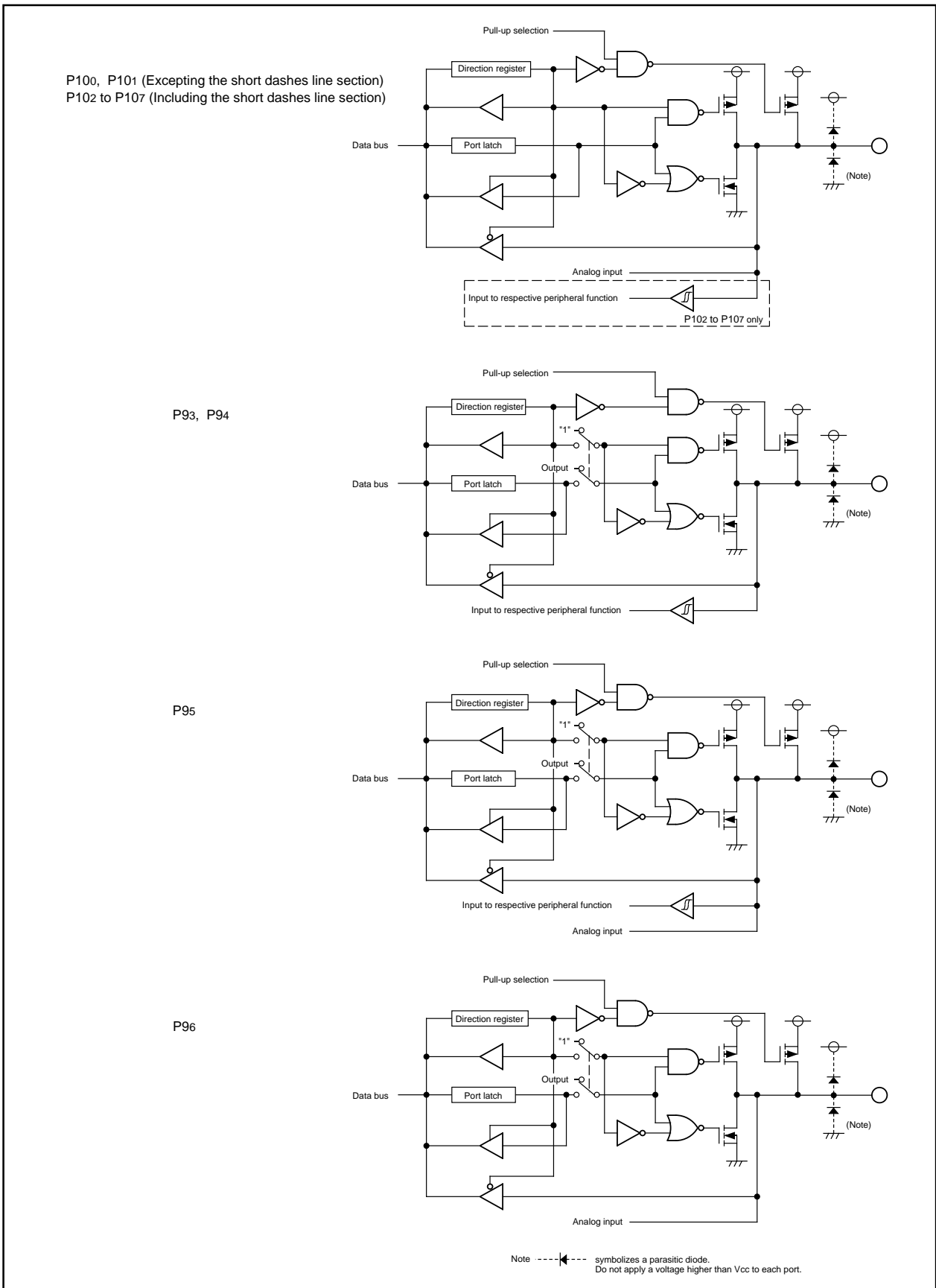


Fig.UA-4 Programmable I/O ports (4)

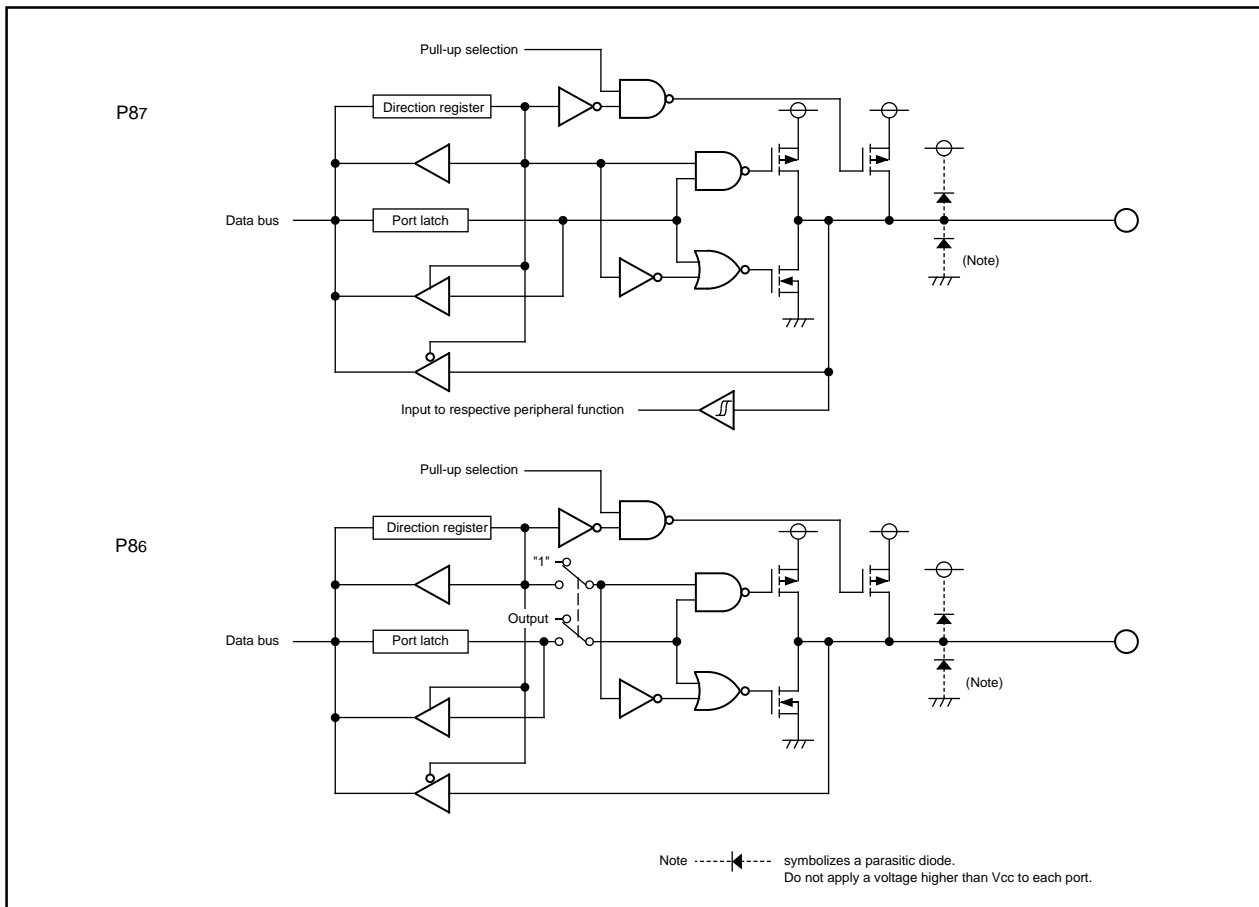


Fig.UA-5 Programmable I/O ports (5)

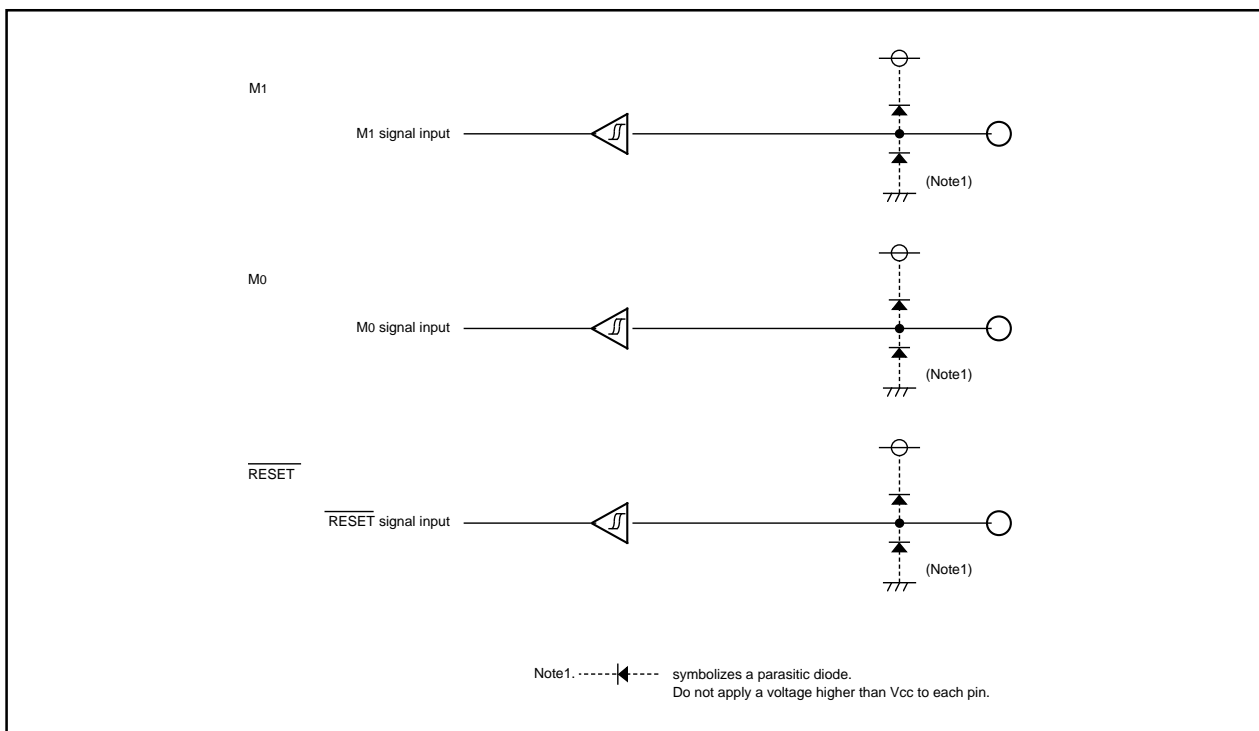


Fig.UA-6 I/O pins

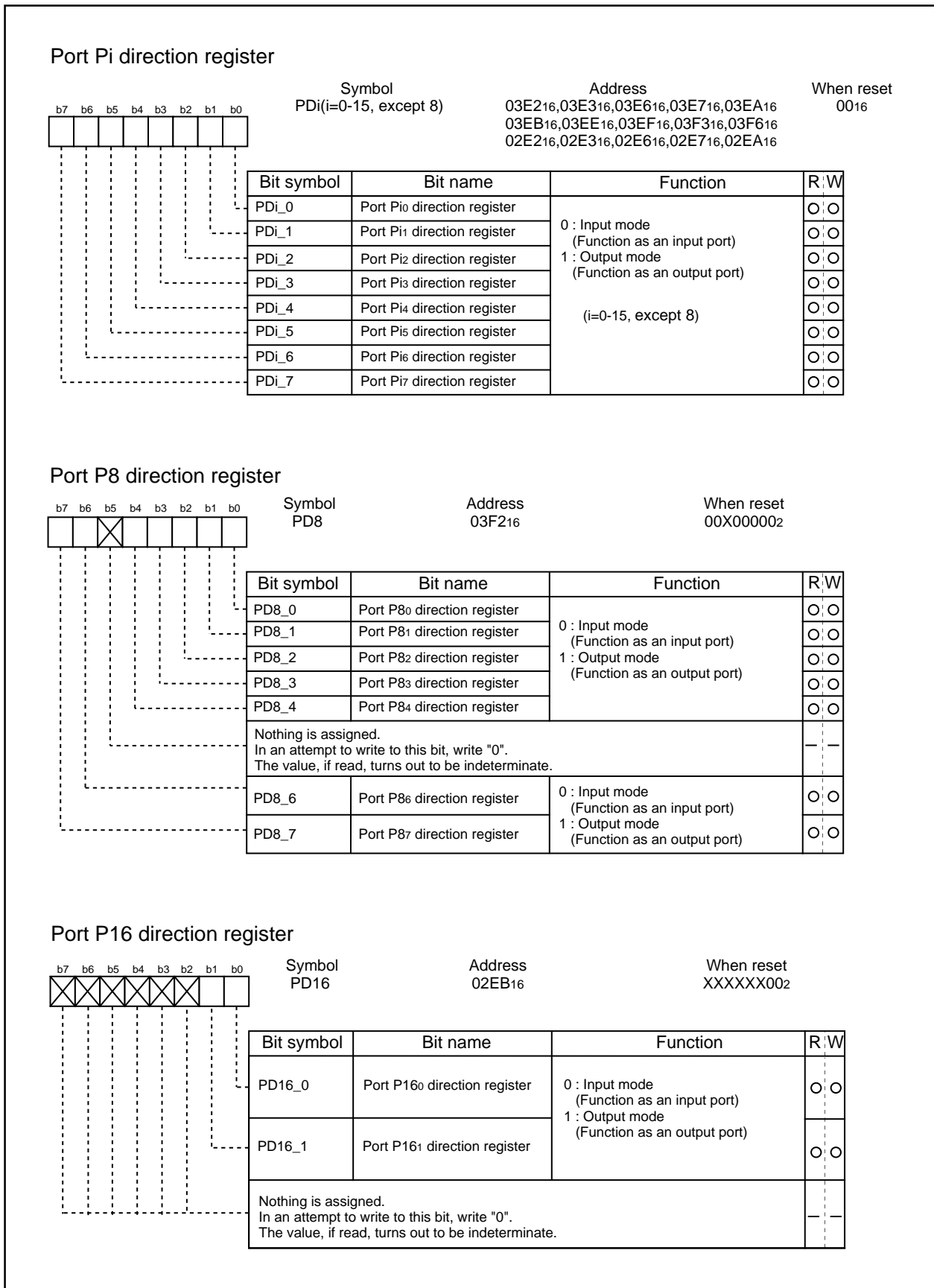


Fig.UA-7 Direction register

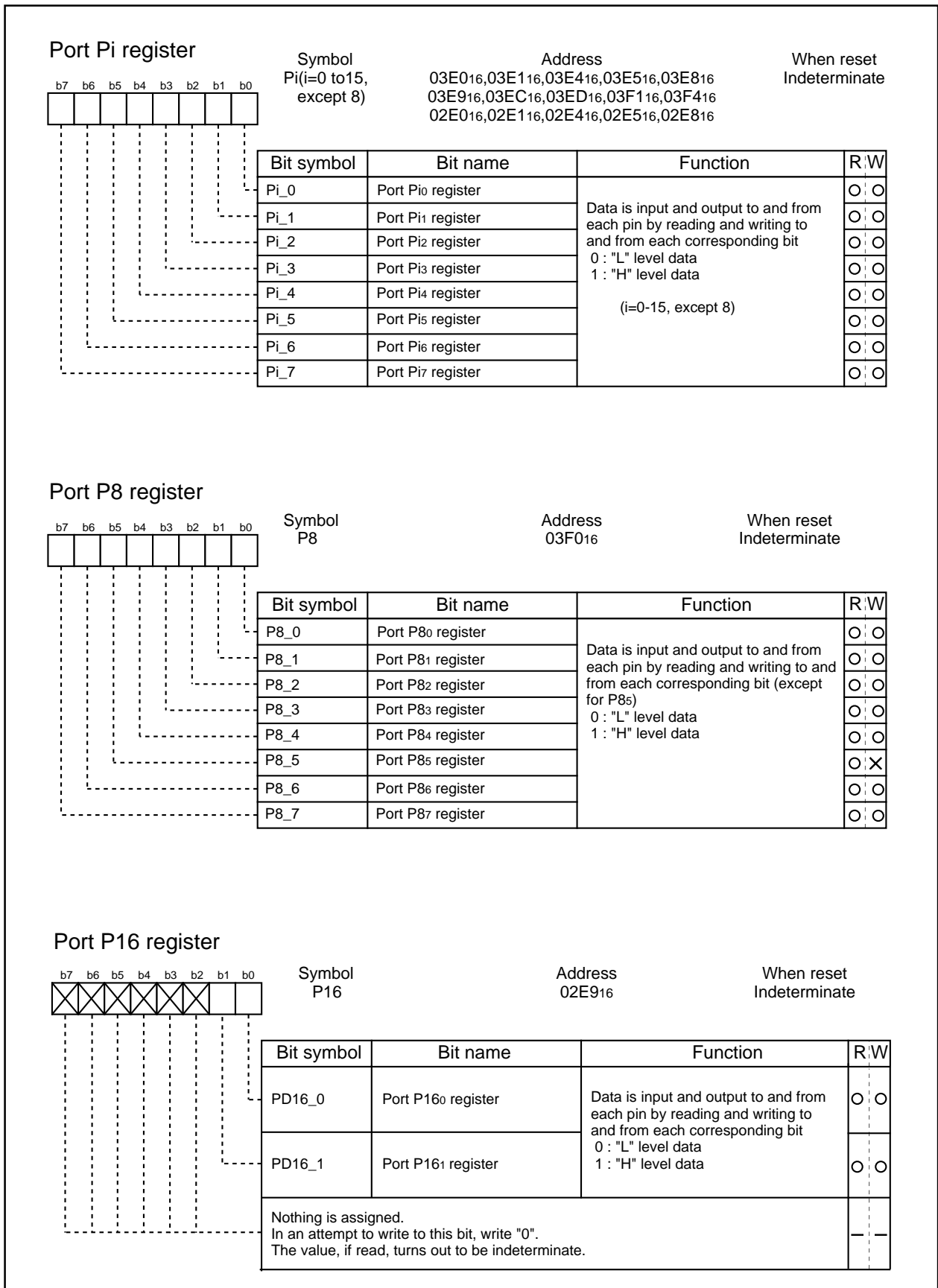


Fig.UA-8 Port register

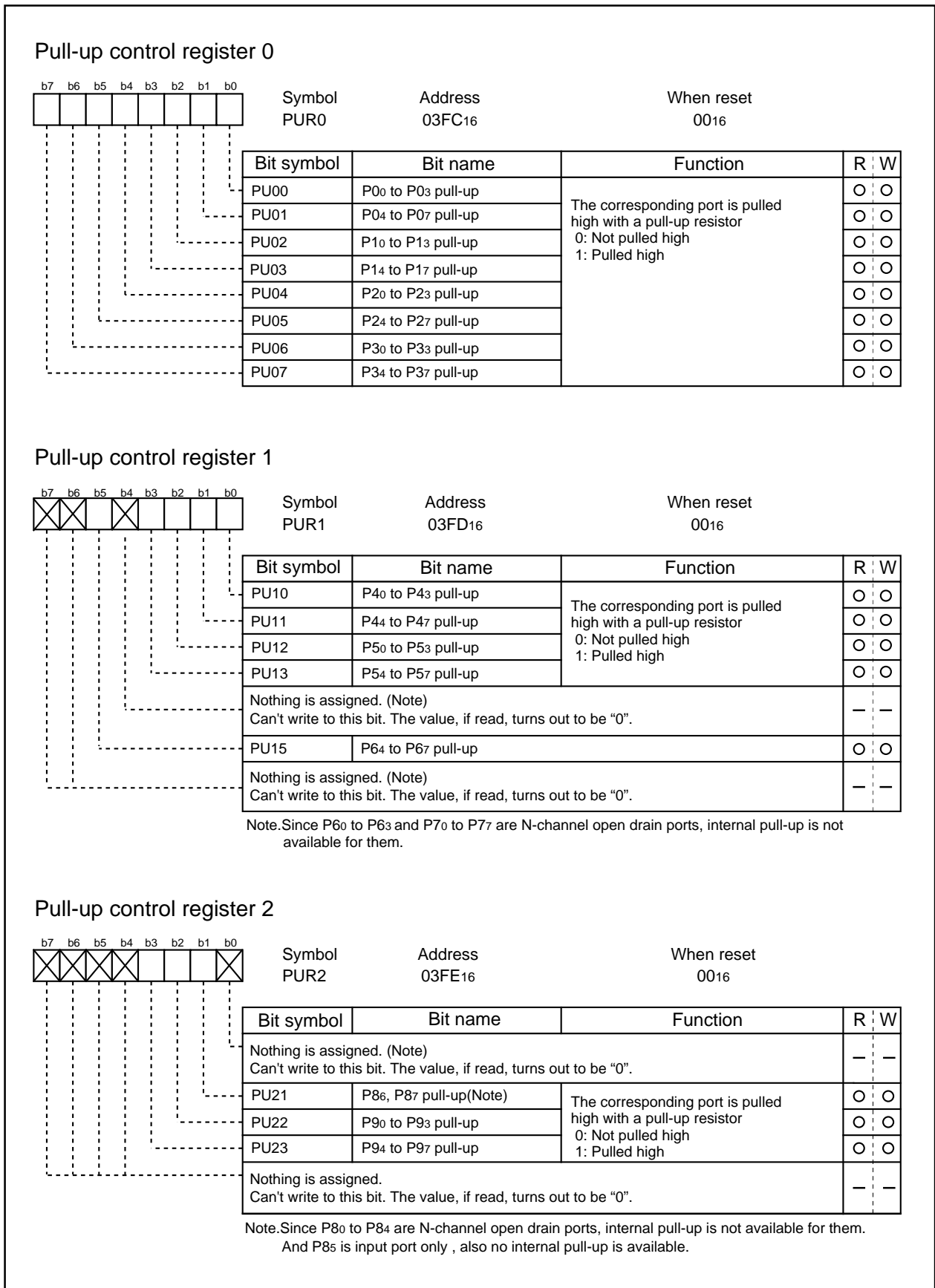


Fig.UA-9 Pull-up control register(1)

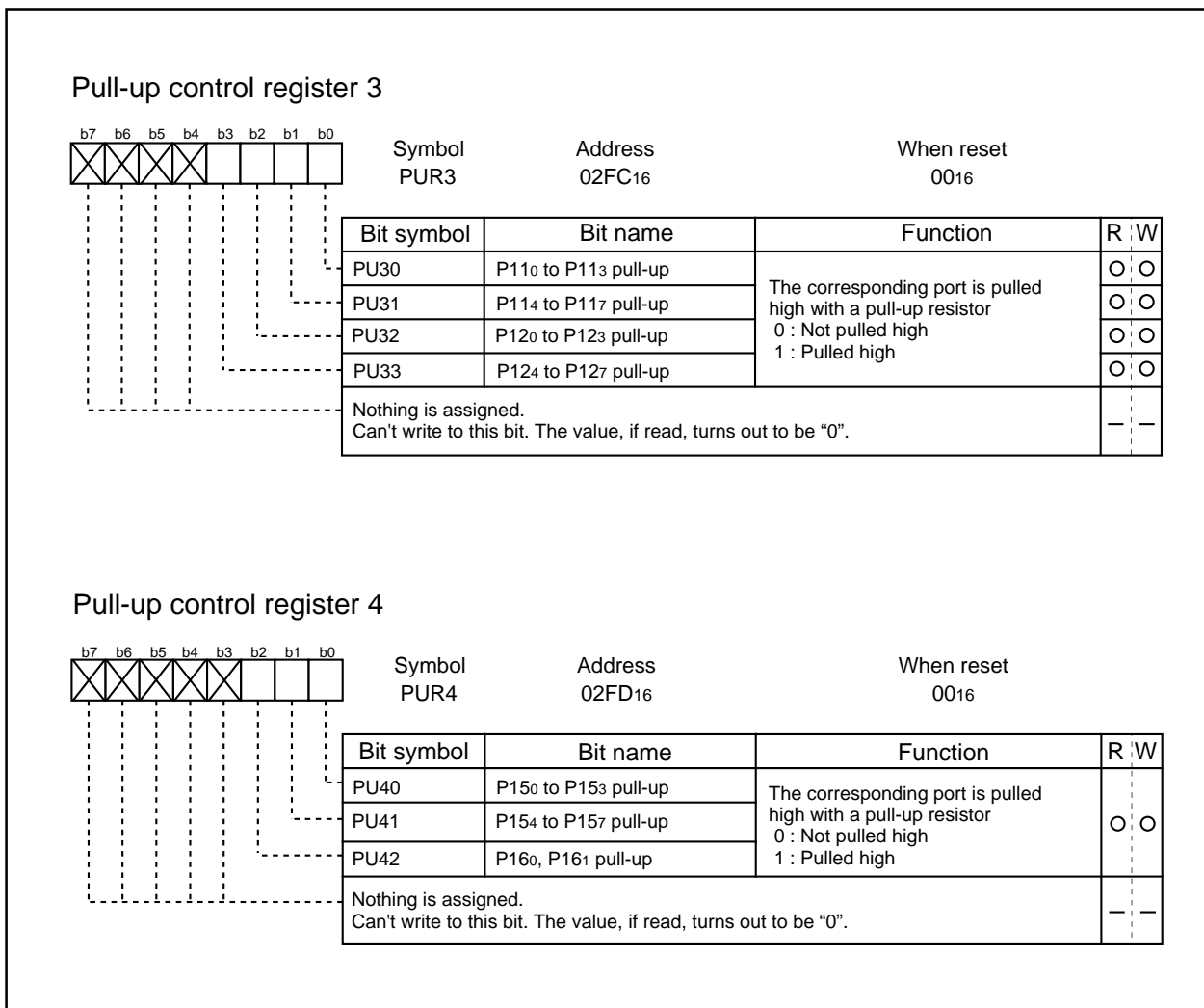


Fig.UA-10 Pull-up control register(2)

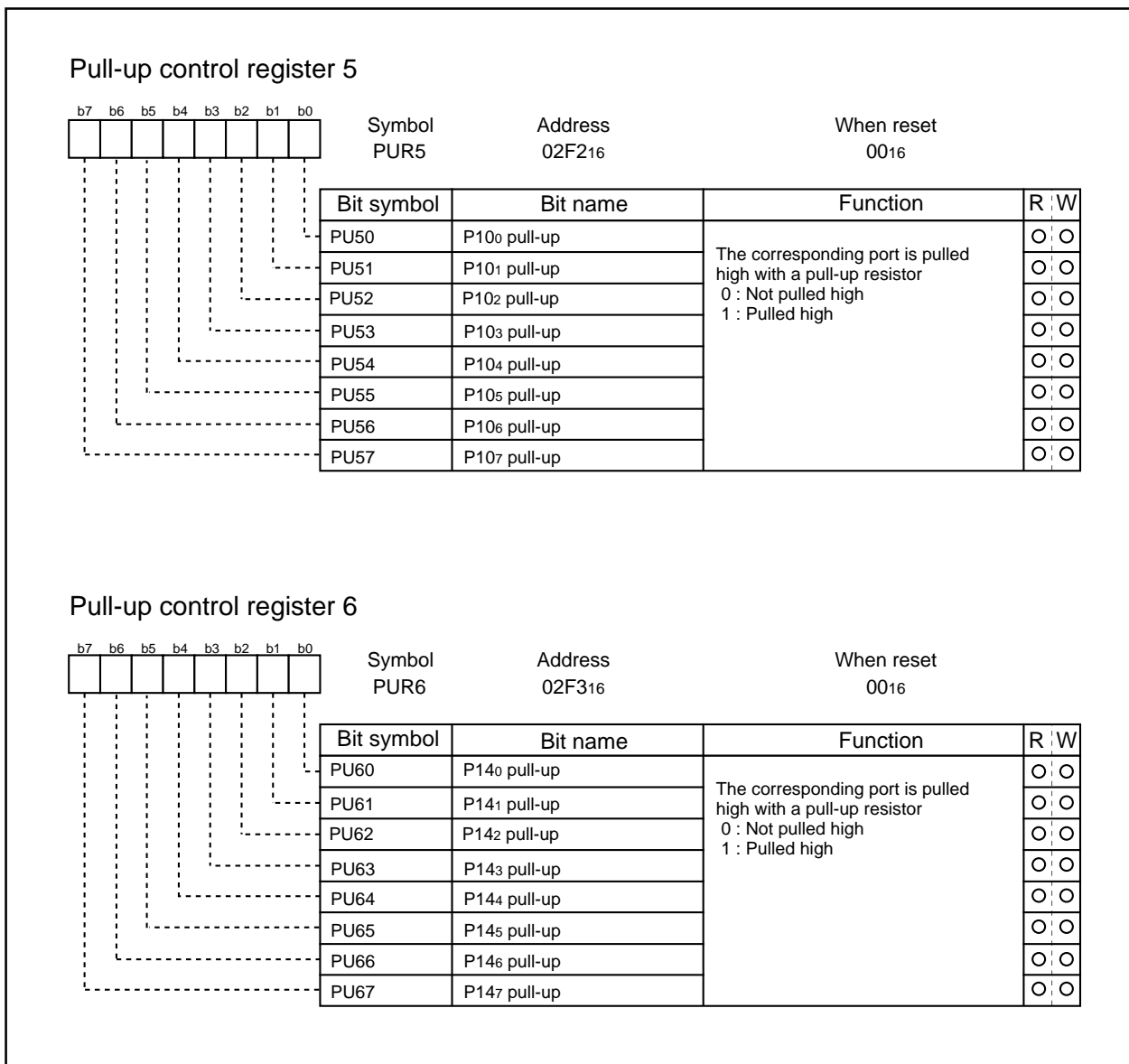


Fig.UA-11 Pull-up control register(3)

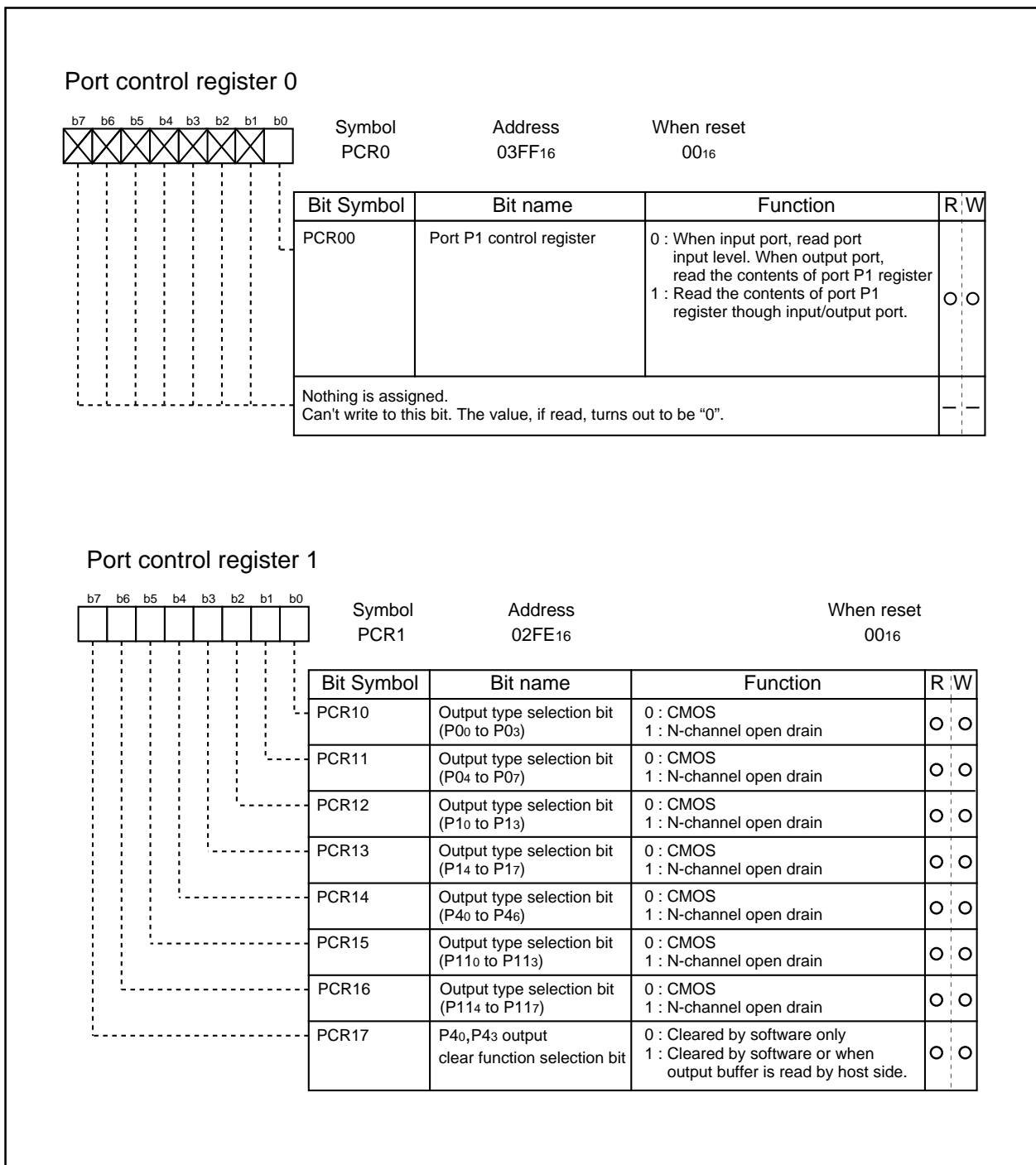


Fig.UA-12 Port control register 0, 1

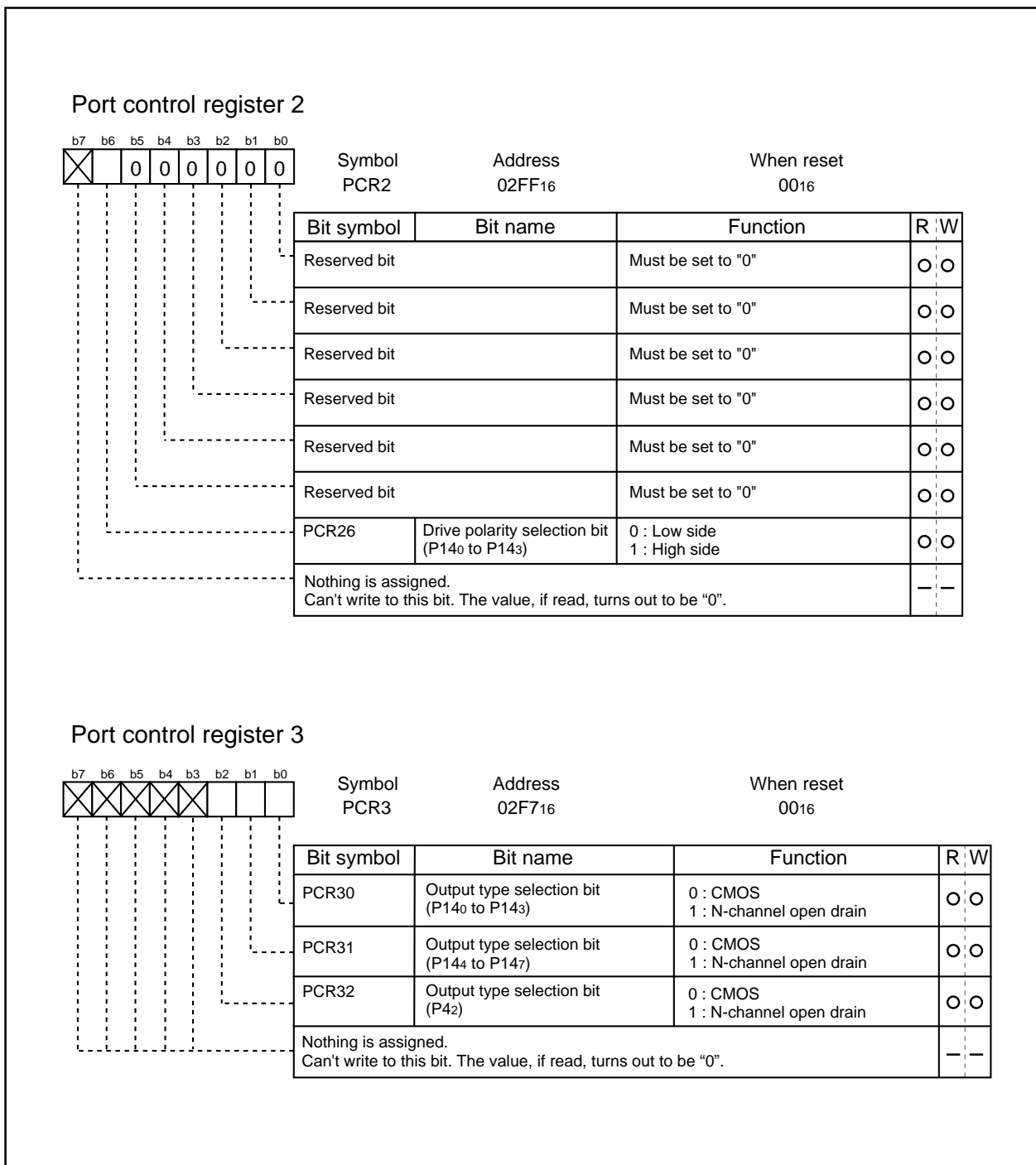


Fig.UA-13 Port control register 2, 3

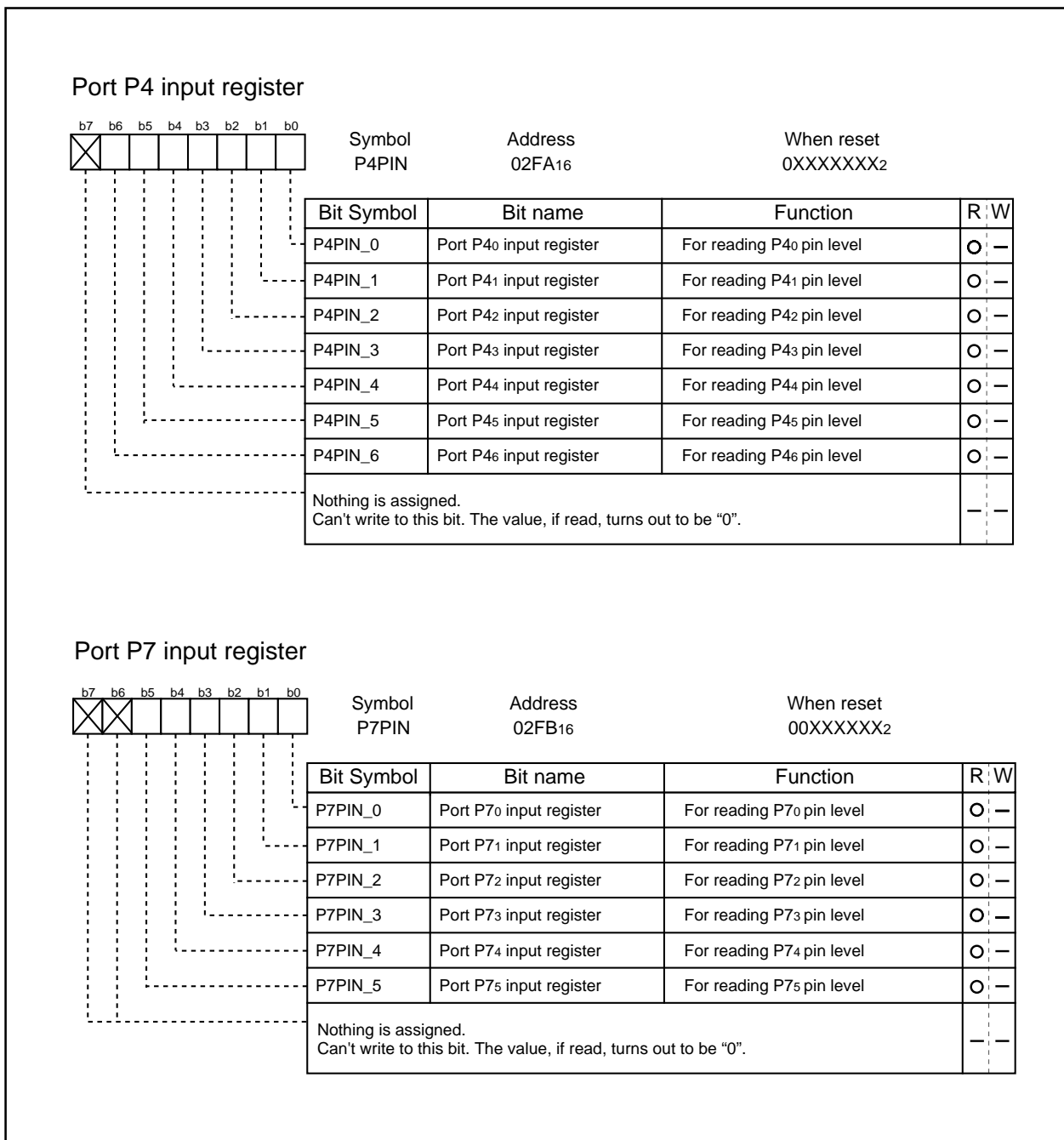


Fig.UA-14 Port P4,P7 input register

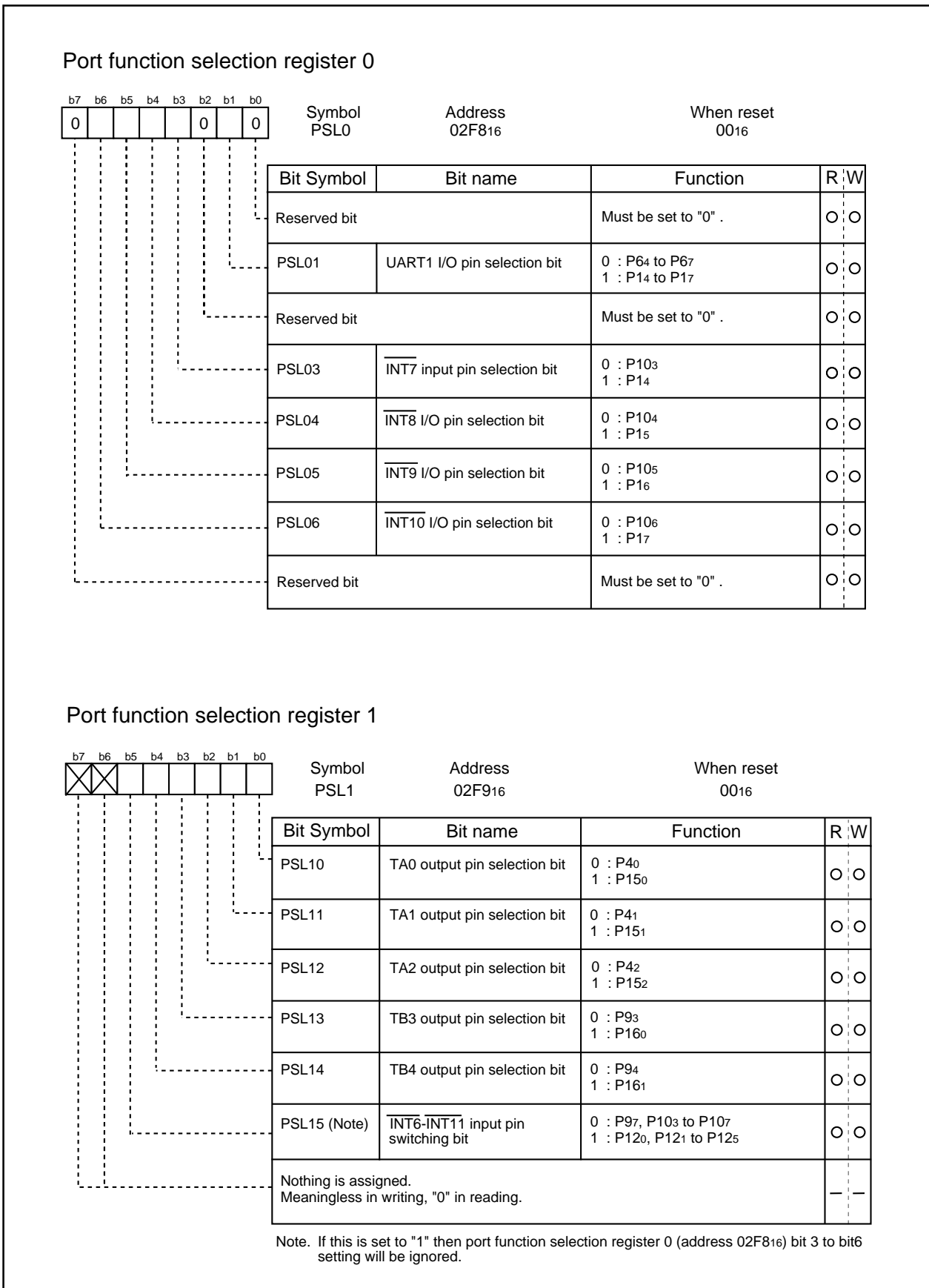


Fig.UA-15 Port function selection register 0,1

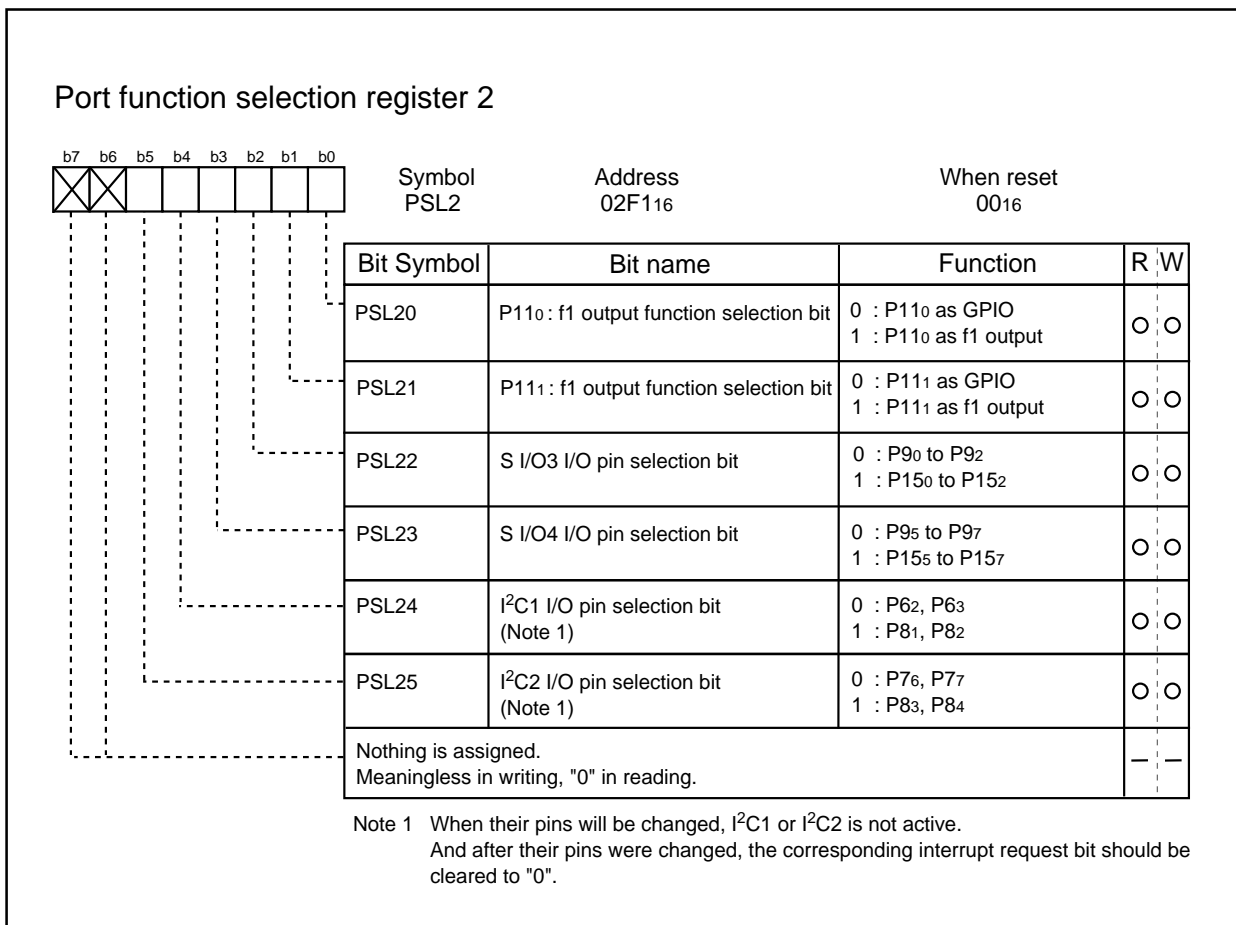
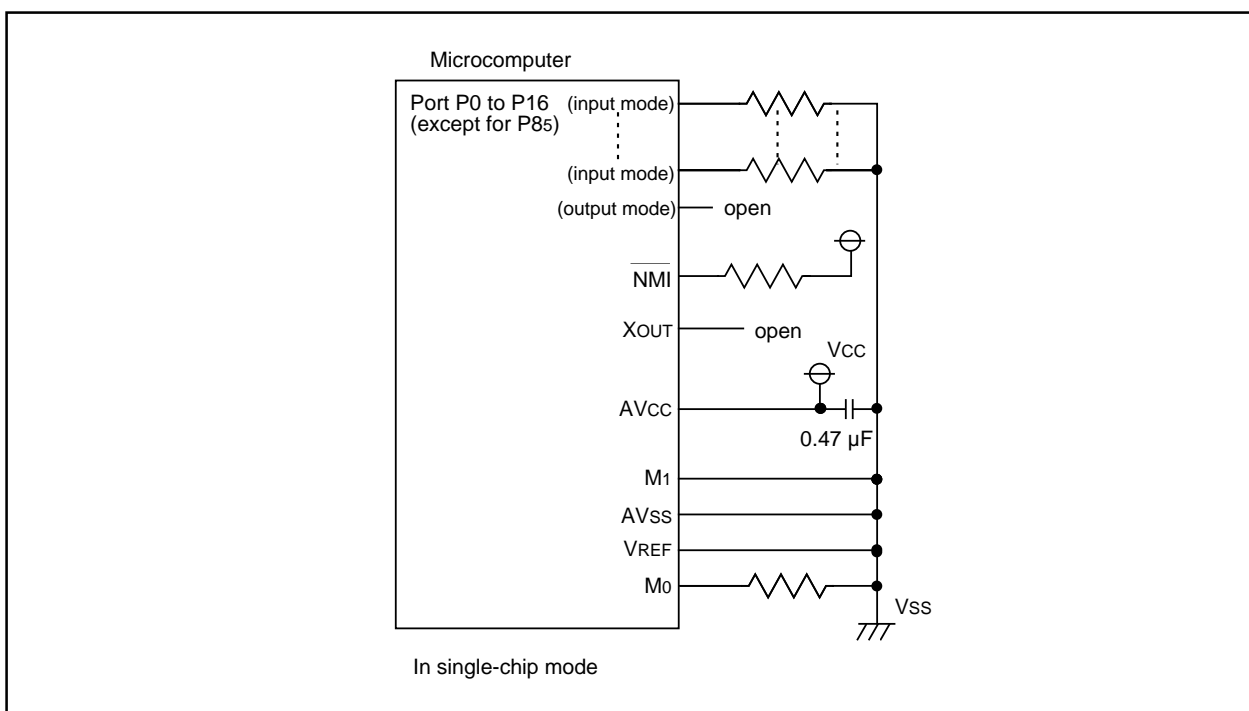


Fig.UA-16 Port function selection register 2

Table.UA-1 Example connection of unused pins in single-chip mode

Pin name	Connection
Ports P0 to P10 (excluding P85)	After setting for input mode, connect every pin to Vss or Vcc via a resistor; or after setting for output mode, leave these pins open.
Ports P11 to P16	After setting for input mode, connect every pin to Vss or Vcc via a resistor; or after setting for output mode, leave these pins open.
XOUT (Note)	Open
NMI	Connect via resistor to Vcc (pull-up)
AVcc	Connect to Vcc
AVss, VREF, M1	Connect to Vss

Note: With external clock input to XIN pin.

**Fig.UA-17 Example connection of unused pins**

Usage Precaution

Timer A (timer mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. But, reading the timer Ai register with the reload timing gets "FFFF16". Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a setting value to the timer.

Timer A (event counter mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. But, reading the timer Ai register with the reload timing gets "FFFF16" by underflow or "000016" by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a setting value to the timer.
- (2) When stop counting in free run type, set timer again.

Timer A (one-shot timer mode)

- (1) Setting the count start flag to "0" while a count is in progress causes as follows:
 - The counter stops counting and a content of reload register is reloaded.
 - The TAIOUT pin outputs "L" level.
 - The interrupt request generated and the timer Ai interrupt request bit goes to "1".
- (2) The timer Ai interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:
 - Selecting one-shot timer mode after reset.
 - Changing operation mode from timer mode to one-shot timer mode.
 - Changing operation mode from event counter mode to one-shot timer mode.Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

Timer A (pulse width modulation mode)

- (1) The timer Ai interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:
 - Selecting PWM mode after reset.
 - Changing operation mode from timer mode to PWM mode.
 - Changing operation mode from event counter mode to PWM mode.Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.
- (2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TAIOUT pin is outputting an "H" level in this instance, the output level goes to "L", and the timer Ai interrupt request bit goes to "1". If the TAIOUT pin is outputting an "L" level in this instance, the level does not change, and the timer Ai interrupt request bit does not becomes "1".

Timer B (timer mode, event counter mode)

- (1) Reading the timer Bi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. But, reading the timer Bi register with the reload timing gets "FFFF16". Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a setting value to the timer.

Timer B (pulse period/pulse width measurement mode)

- (1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to "1".
- (2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

A-D Converter

- (1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).
In particular, when the Vref connection bit is changed from "0" to "1", start A-D conversion after an elapse of 1 μ s or longer.
- (2) When changing A-D operation mode, select analog input pin again.
- (3) Using one-shot mode or single sweep mode
Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)
- (4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1
Use the undivided main clock as the internal CPU clock.

Stop Mode and Wait Mode

- (1) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to "L" level until main clock oscillation is stabilized.
- (2) When switching to either wait mode or stop mode, instructions occupying four bytes either from the WAIT instruction or from the instruction that sets the every-clock stop bit to "1" within the instruction queue are prefetched and then the program stops. So put at least four NOPs in succession either to the WAIT instruction or to the instruction that sets the every-clock stop bit to "1".

Interrupts

- (1) Reading address 00000₁₆
 - When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
The interrupt request bit of the certain interrupt written in address 00000₁₆ will then be set to "0".
Reading address 00000₁₆ by software sets enabled highest priority interrupt source request bit to "0".
Though the interrupt is generated, the interrupt routine may not be executed.
Do not read address 00000₁₆ by software.
- (2) Setting the stack pointer
 - The value of the stack pointer immediately after reset is initialized to 0000₁₆. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.
When using the $\overline{\text{NMI}}$ interrupt, initialize the stack point at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.
- (3) The $\overline{\text{NMI}}$ interrupt
 - As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistor (pull-up) if unused. Be sure to work on it.
 - Do not get into stop mode with the $\overline{\text{NMI}}$ pin set to "L".

(4) External interrupt

- When the polarity of the $\overline{\text{INT0}}$ to $\overline{\text{INT11}}$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0".

(5) Rewrite the interrupt control register

- To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```

INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP                               ; Four NOP instructions are required when using HOLD function.
  NOP
  FSET  I           ; Enable interrupts.

```

Example 2:

```

INT_SWITCH2:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  MOV.W MEM, R0     ; Dummy read.
  FSET  I           ; Enable interrupts.

```

Example 3:

```

INT_SWITCH3:
  PUSHC FLG         ; Push Flag register onto stack
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  POPC  FLG         ; Enable interrupts.

```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Noise

- (1) Insert the by-pass condenser to the Vcc-Vss line for preventing a noise and latch-up. Connect the by -pass condencer (about 0.1 μ F) between Vcc pin and Vss pin. It is distance must be shortest rather sicker line.

Table.ZA-1 Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
V _{cc}	Supply voltage		V _{cc} =AV _{cc}	-0.3 to 4.6	V
AV _{cc}	Analog Supply voltage		V _{cc} =AV _{cc}	-0.3 to 4.6	V
V _i	Input voltage	RESET, M0, M1, V _{REF} , X _{IN} , P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₄ to P6 ₇ , P8 ₆ , P8 ₇ , P9 ₀ to P9 ₇ , P10 ₀ to P10 ₇ , P11 ₀ to P11 ₇ , P12 ₀ to P12 ₇ , P14 ₀ to P14 ₇ , P15 ₀ to P15 ₇ , P16 ₀ , P16 ₁		-0.3 to V _{cc} +0.3	V
		P6 ₀ to P6 ₃ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₄ , P8 ₅ , P13 ₀ to P13 ₇		-0.3 to 5.8	V
V _o	output voltage	P0 ₀ to P0 ₇ , P1 ₀ to P1 ₇ , P2 ₀ to P2 ₇ , P3 ₀ to P3 ₇ , P4 ₀ to P4 ₇ , P5 ₀ to P5 ₇ , P6 ₄ to P6 ₇ , P8 ₆ , P8 ₇ , P9 ₀ to P9 ₇ , P10 ₀ to P10 ₇ , P11 ₀ to P11 ₇ , P12 ₀ to P12 ₇ , P14 ₀ to P14 ₇ , P15 ₀ to P15 ₇ , P16 ₀ , P16 ₁ , X _{OUT}		-0.3 to V _{cc} +0.3	V
		P6 ₀ to P6 ₃ , P7 ₀ to P7 ₇ , P8 ₀ to P8 ₄ , P13 ₀ to P13 ₇		-0.3 to 5.8	V
P _d	Power dissipation		T _a =25 °C	300	mW
T _{opr}	Operating ambient temperature			-20 to 85	°C
T _{stg}	Storage temperature			-40 to 125	°C

Table.ZA-2 Recommended operating conditions (referenced to Vcc=3.0V to 3.6V, Ta= -20 to 85°C)

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
Vcc	Supply voltage		3.0	3.3	3.6	V
AVcc	Analog Supply voltage			Vcc		V
Vss	Supply voltage			0		V
AVss	Analog Supply voltage			0		V
VIH	"H" input voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161, XIN, RESET, M0, M1	0.8Vcc		Vcc	V
		P60 to P63, P70 to P77, P80 to P84, P85, P130 to P137, PSA0 to PSA2, PSB0 to PSB2	0.8Vcc		5.5	V
		LAD0 to LAD3, LFRAME, LCLK, SERIRQ, CLKRUN	0.6Vcc		Vcc	V
		SDA0, SCL0, SDA1, SCL1, SDA2, SCL2, P60 to P63, P76, P77, P81 to P84	i ² C-BUS input level selected SMBUS input level selected	0.7Vcc 1.4		5.5 5.5
VIL	"L" input voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161, XIN, RESET, M0, M1	0		0.2Vcc	V
		P60 to P63, P70 to P77, P80 to P84, P85, P130 to P137, PSA0 to PSA2, PSB0 to PSB2	0		0.2Vcc	V
		LAD0 to LAD3, LFRAME, LCLK, SERIRQ, CLKRUN	0		0.2Vcc	V
		SDA0, SCL0, SDA1, SCL1, SDA2, SCL2, P60 to P63, P76, P77, P81 to P84	i ² C-BUS input level selected SMBUS input level selected	0 0		0.3Vcc 0.6

Table.ZA-3 Recommended operating conditions (referenced to Vcc=3.0V to 3.6V, Ta= -20 to 85°C)

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
I _{OH} (peak)	"H" peak output current P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161			-10.0	mA
I _{OL} (peak)	"L" peak output current P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P76 to P77, P80 to P84, P86 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P144 to P147, P150 to P157, P160, P161			10.0	mA
		P20 to P27		20.0	mA
I _{OL} (peak)	"L" peak output current P140 to P143	Driven ability : High		20.0	mA
		Driven ability : Low		10.0	mA
I _{OH} (avg)	"H" average output current P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P86 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161			-5.0	mA
I _{OL} (avg)	"L" average output current P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P76 to P77, P80 to P84, P86 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161			5.0	mA
		P20 to P27		15.0	mA
I _{OL} (avg)	"L" average output current P140 to P143	Driven ability : High		15.0	mA
		Driven ability : Low		5.0	mA
f (XIN)	Main clock input oscillation frequency	0		8	MHz

Note1 : The average output current is the average value during the 100ms period limited current.

Note2 : The value are as follow:

The sum of I_{OL} (peak) of P0, P1, P2, P86 to P87, P9, P10, P11, P120 to P126, P153 to P157 P16 should be under 80mA.

The sum of I_{OH} (peak) of P0, P1, P2, P86 to P87, P9, P10, P11, P120 to P126, P153 to P157 P16 should be under 80mA.

The sum of I_{OL} (peak) of P3, P4, P5, P6, P7, P80 to P84, P13, P14, P150 to P152 should be under 80mA.

The sum of I_{OH} (peak) of P3, P4, P5, P64 to P67, P14, P150 to P152 should be under 80mA.

Table.ZA-4 Electrical characteristics
(referenced to Vcc=3.0V, Vss=0V, Ta=25°C, f(XIN)=16MHz with 0 wait unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
VOH	High output voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161	IOH=-1mA	2.5			V
VOH	High output voltage	XOUT	HIGH POWER	IOH=-0.1mA	2.5		V
			LOWPOWER	IOH=-50μA	2.5		
VOL	Low output voltage	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161	IOL=1mA			0.5	V
		P20 to P27	Vcc=3V, IOl=3mA			0.5	
VOL	Low output voltage	P140 to P143	HIGH POWER	IOl=3mA		0.5	V
			LOWPOWER	IOl=1mA		0.5	
VOL	Low output voltage	XOUT	HIGH POWER	IOH=0.1mA		0.5	V
			LOWPOWER	IOH=50μA		0.5	
VT+VT-	Hysteresis	TA0IN to TA4IN, TB0IN to TB5IN, INT0 to INT11, ADTRG, CTS1, CLK1, CLK3, CLK4, SIN3, SIN4, RxD1, ICCK, NMI, Kl00 to Kl07		0.2		0.8	V
VT+VT-	Hysteresis	RESET		0.2		1.8	V
IiH	HIGH input current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161, XIN, RESET, M0, M1	Vi=3V			4.0	μA
IiL	Low input current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161, XIN, RESET, M0, M1	Vi=0V			-4.0	μA
R PULLUP	Pull-up resistance	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161	Vi=0V	66.0	120.0	250.0	kΩ
R fXIN	Feedback resistance	XIN			3.0		MΩ
V RAM	RAM retention voltage		When clock is stopped	2.0			V
I CC	Power supply current	When reset, the output pins are opened, the other pins are connected to Vss.	f(XIN)=16MHz, Square wave without division		16.0	24.0	mA
			Ta=25°C			100.0	μA
			Ta=85°C			300.0	μA

Table.ZA-5 A-D conversion characteristics**(referenced to $V_{CC}=AV_{CC}=V_{REF}=3V, V_{SS}=AV_{SS}=0V$ at $T_a=25^\circ C$ unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
—	Resolution		$V_{REF}=V_{CC}$			10	Bits
—	Absolute accuracy	8 bit	$V_{REF}=V_{CC}=3V, \emptyset_{AD}=f_{AD}$			± 2	LSB
—		10 bit				± 6	LSB
R_{LADDER}	Ladder resistance		$V_{REF}=V_{CC}$	10		40	k Ω
t_{CONV}	Conversion time	8 bit		6.125			ms
		10 bit		7.375			ms
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Timing requirements (referenced to $V_{CC}=3V, V_{SS}=0V$ at $T_a=25^{\circ}C$ unless otherwise specified)

Table.ZA-6 External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	62.5		ns
$t_{w(H)}$	External clock input HIGH pulse width	25		ns
$t_{w(L)}$	External clock input LOW pulse width	25		ns
t_r	External clock rising time		9	ns
t_f	External clock falling time		9	ns

Timing requirements (referenced to $V_{CC}=3V, V_{SS}=0V$ at $T_a=25^\circ C$ unless otherwise specified)

Table.ZA-7 Timer A input (The count input of event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input "H" pulse width	60		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	60		ns

Table.ZA-8 Timer A input (The gating input of timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input "H" pulse width	300		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	300		ns

Table.ZA-9 Timer A input (The external trigger input of one shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input "H" pulse width	150		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	150		ns

Table.ZA-10 Timer A input (The external trigger input of pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input "H" pulse width	150		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	150		ns

Table.ZA-11 Timer A input (The up down input of event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input "H" pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input "L" pulse width	1500		ns
$t_{su(UP-TiN)}$	TAiOUT input setup time	200		ns
$t_{h(TiN-UP)}$	TAiOUT input hold time	200		ns

Timing requirements (referenced to $V_{CC}=3V, V_{SS}=0V$ at $T_a=25^\circ C$ unless otherwise specified)**Table.ZA-12 Timer B input (The count input of event counter mode)**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiN input cycle time (single edge count)	150		ns
$t_{w(TBH)}$	TBiN input "H" pulse width (single edge count)	60		ns
$t_{w(TBL)}$	TBiN input "L" pulse width (single edge count)	60		ns
$t_{c(TB)}$	TBiN input cycle time (double edge count)	300		ns
$t_{w(TBH)}$	TBiN input "H" pulse width (double edge count)	160		ns
$t_{w(TBL)}$	TBiN input "L" pulse width (double edge count)	160		ns

Table.ZA-13 Timer B input (Pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiN input cycle time	600		ns
$t_{w(TBH)}$	TBiN input "H" pulse width	300		ns
$t_{w(TBL)}$	TBiN input "L" pulse width	300		ns

Table.ZA-14 Timer B input (Pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiN input cycle time	600		ns
$t_{w(TBH)}$	TBiN input "H" pulse width	300		ns
$t_{w(TBL)}$	TBiN input "L" pulse width	300		ns

Table.ZA-15 A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	\overline{ADTRG} input cycle time (The Min. of trigger)	1500		ns
$t_{w(ADL)}$	\overline{ADTRG} input "L" pulse width	200		ns

Table.ZA-16 Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input "H" pulse width	150		ns
$t_{w(CKL)}$	CLKi input "L" pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	50		ns

Table.ZA-17 External interrupt \overline{INTi} input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	\overline{INTi} input "H" pulse width	380		ns
$t_{w(INL)}$	\overline{INTi} input "L" pulse width	380		ns

Timing requirements (referenced to Vcc = 3.0 to 3.6V, Vss = 0V, Ta =25 °C)

Table. ZA-18 Multi-master I²C-BUS line

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
tBUF	Bus free time	4.7		1.3		μ s
tHD;STA	The hold time in start condition	4.0		0.6		μ s
tLOW	The hold time in SCL clock "0" status	4.7		1.3		μ s
tR	SCL, SDA signals' rising time		1000	20+0.1Cb	300	ns
tHD;DAT	Data hold time	0		0	0.9	μ s
tHIGH	The hold time in SCL clock "1" status	4.0		0.6		μ s
tF	SCL, SDA signals' falling time		300	20+0.1Cb	300	ns
tsu;DAT	Data setup time	250		100		ns
tsu;STA	The setup time in restart condition	4.7		0.6		μ s
tsu;STO	Stop condition setup time	4.0		0.6		μ s

Table. ZA-19 PS2 interface (referenced to Vcc = 3.0 to 3.6V, Vss = 0V, Ta =25 °C)

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
twL	PS2 clock "L" pulse width	30		50	μ s
twH	PS2 clock "H" pulse width	30		50	μ s
tsu	PS2 data setup time	5			μ s
th	PS2 data hold time	0			ns
td	PS2 data delay time			twL-5	μ s
tv	PS2 data valid time	0		twL-5	μ s

Table. ZA-20 LPC bus interface/serial interrupt output

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
tC(CLK)	LCLK clock input cycle time	30		∞	ns
tWH(CLK)	LCLK clock input "H" pulse width	11			ns
tWL(CLK)	LCLK clock input "L" pulse width	11			ns
tsu(D-C)	LAD3-LAD0, SERIRQ, CLKRUN, LFRAME Input setup time	7			ns
th(C-D)	LAD3-LAD0, SERIRQ, CLKRUN, LFRAME input hold time	0			ns
tv(C-D)	LAD3-LAD0, SERIRQ, CLKRUN, LFRAME valid delay time	2		11	ns
toff(A-F)	LAD3-LAD0, SERIRQ, CLKRUN floating output delay time			28	ns

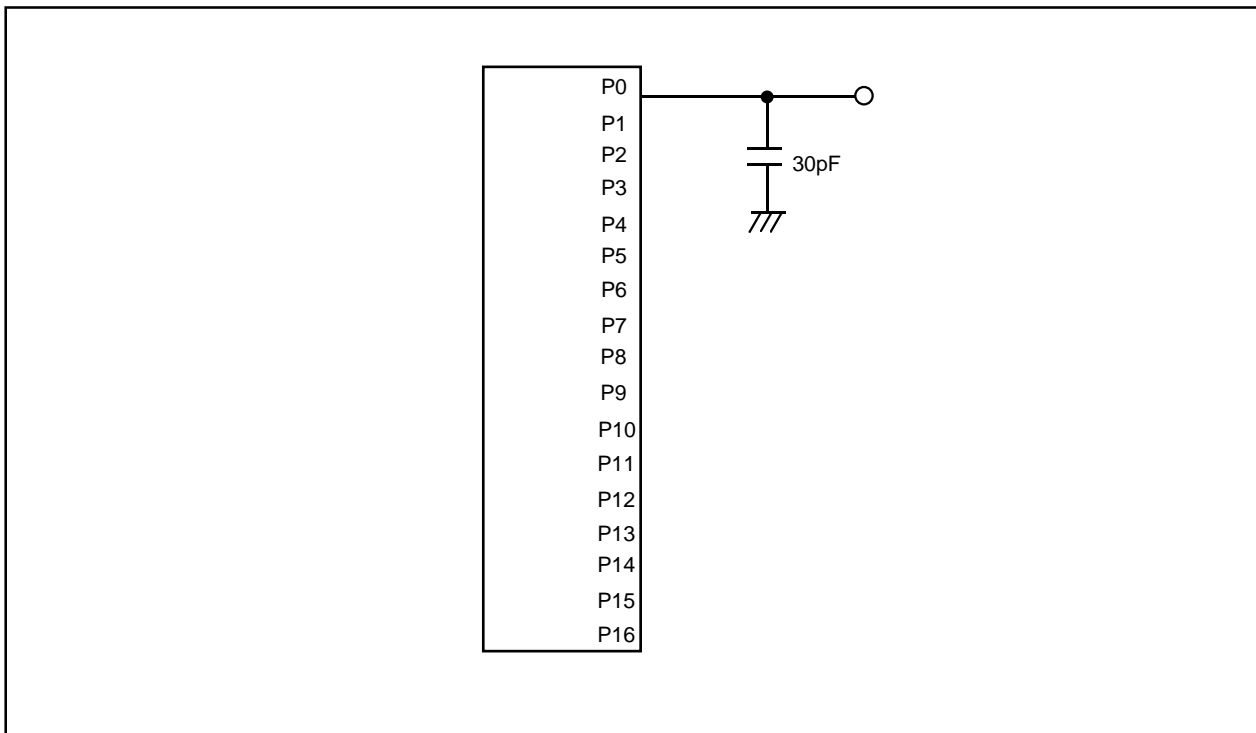


Fig.ZA-1 The measuring circuit for port 0 to port 16

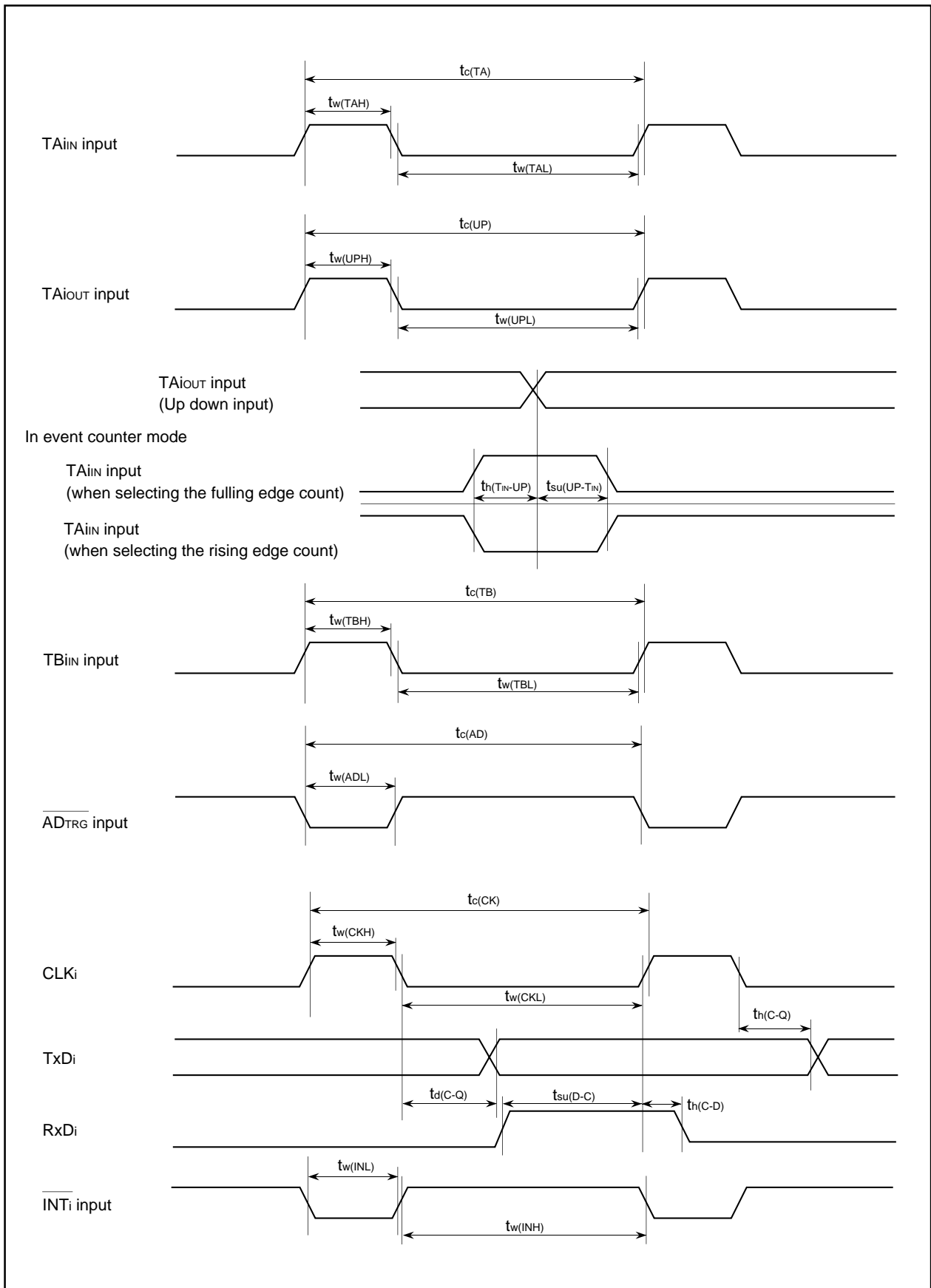


Fig.ZA-2 Timing diagram (1)

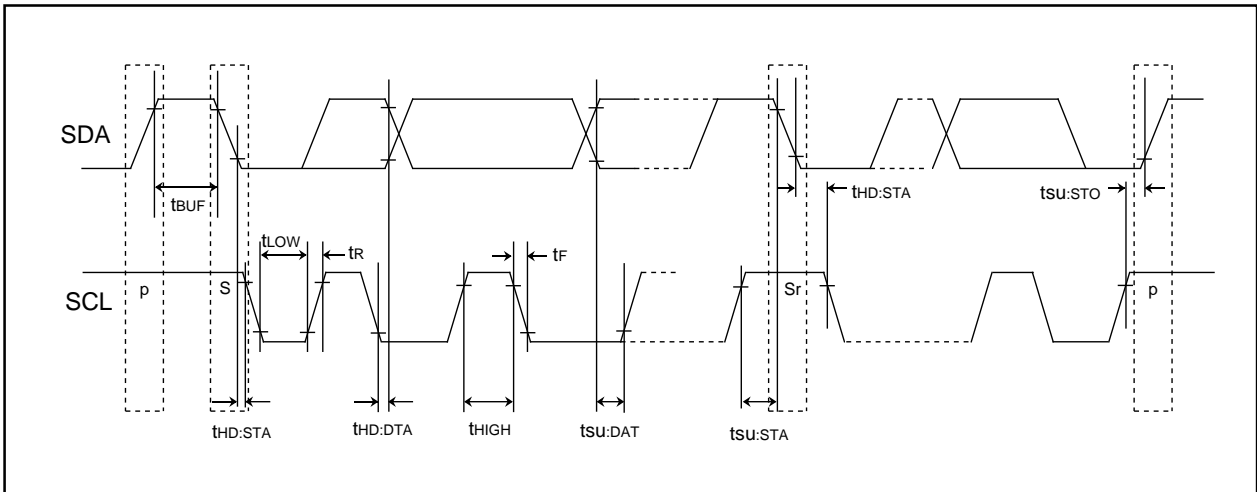


Fig.ZA-3 Timing diagram (2)

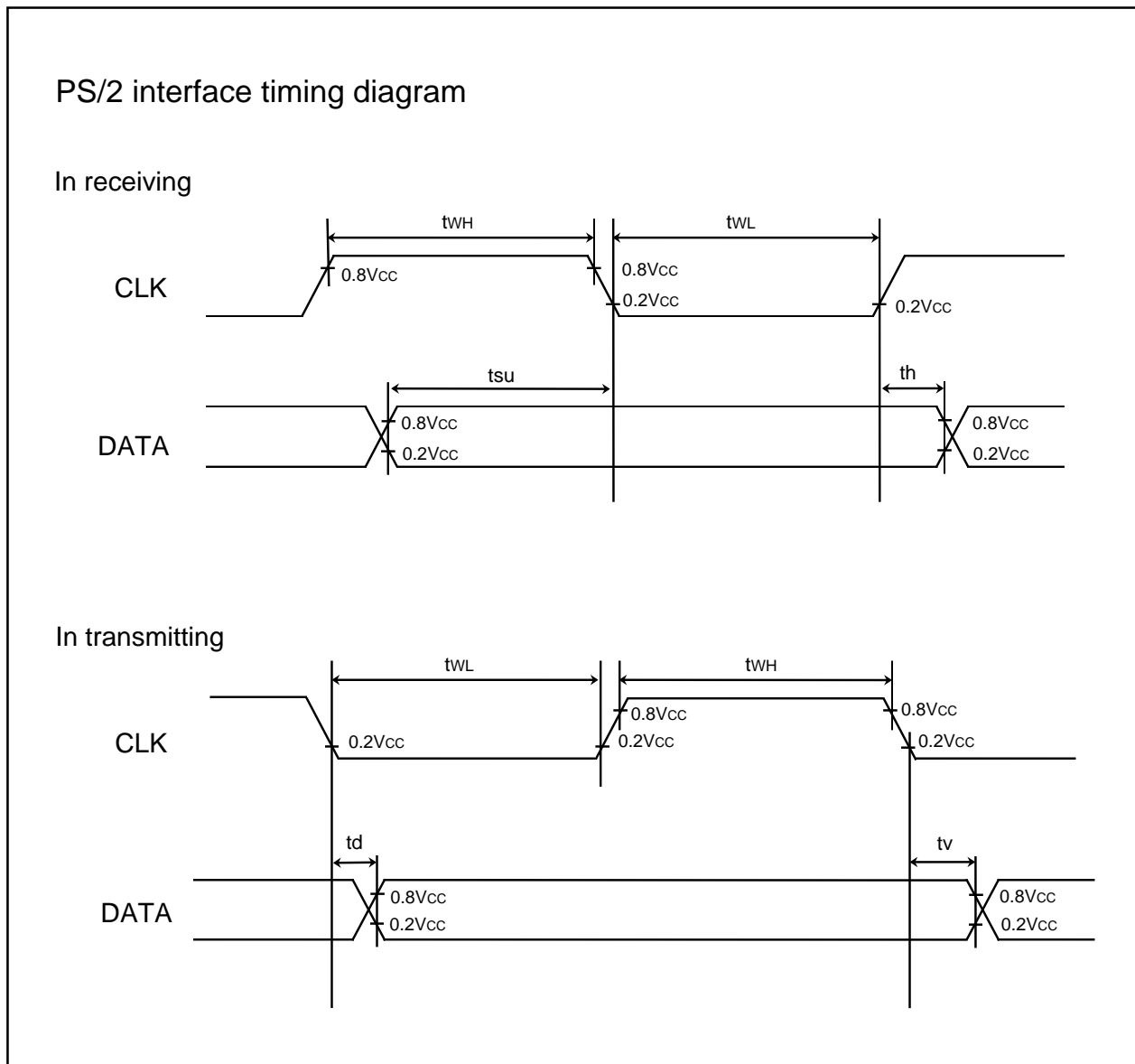


Fig.ZA-4 Timing diagram (3)

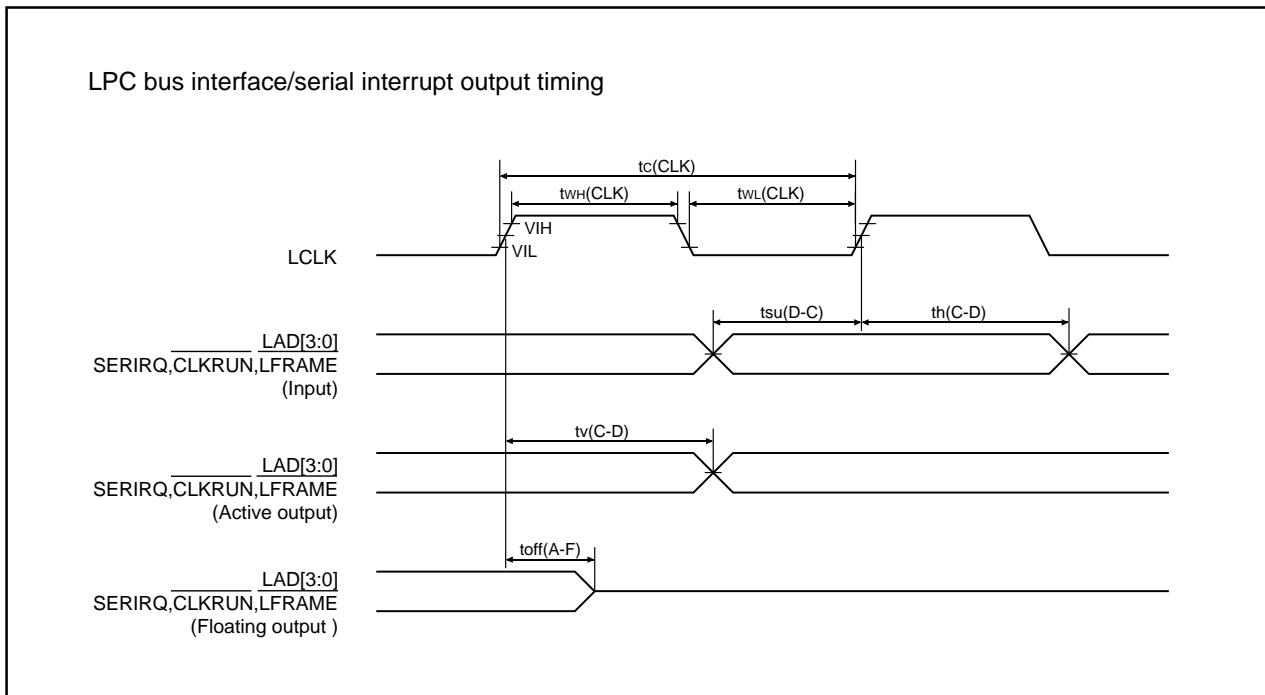


Fig.ZA-5 Timing diagram (4)

Feature Outline

Table AB-1 shows the feature outline of M16C/6KA (build-in NEW DINOR flash memory version).

Table.AB-1 Feature outline of M16C/6KA (build-in NEW DINOR flash memory version)

Item		Feature
Power supply voltage		3.0-3.6V (f(X _{IN})=16MHz, 0 wait)
Power supply voltage for program/erase		3.0-3.6V
Flash memory operation mode		3 modes (parallel I/O, standard serial I/O, CPU reprogram)
Erase block division	User ROM area	6 division (32K+32K+32K+16K+8K+8K bytes)
	Boot ROM area	1 division (4K bytes) (Note1)
Program method		2-byte unit
Erase method		Block erase
Program/ erase control method		Program/ erase controlled by s/w commands
Number of command		5 commands
Program/ erase count	User ROM area	100 times
	Boot ROM area	100 times
ROM code protect		Support for parallel I/O and standard serial I/O modes

Note1: The control program for standard serial I/O mode is stored in boot ROM area when shipping from factory.
The area can only be erased or programmed by parallel I/O mode.

Flash Memory

The M16C/6KA (build-in flash memory version) contains the NEW DINOR type flash memory, which is applied $V_{CC}=3.3V$ when using CPU reprogram or standard serial I/O mode. For the flash memory, 3 flash memory modes are available in which to read, program and erase. They are parallel I/O mode, standard serial I/O mode and CPU reprogram mode. For parallel I/O mode, a programmer is used. For standard serial I/O and CPU reprogram modes, the flash memory is manipulated by CPU. Each mode is detailed in the pages to follow.

Fig. AB-1 shows that flash memory is divided into several blocks. Erasing is in block unit.

In addition to the ordinary user ROM area there is a boot ROM area to store the control program for the CPU reprogram and standard serial I/O modes. The control program for standard serial I/O mode is stored in boot ROM area when shipping from factory. User can reprogram the program to suit its own application system. The area can only be erased or programmed by parallel I/O mode.

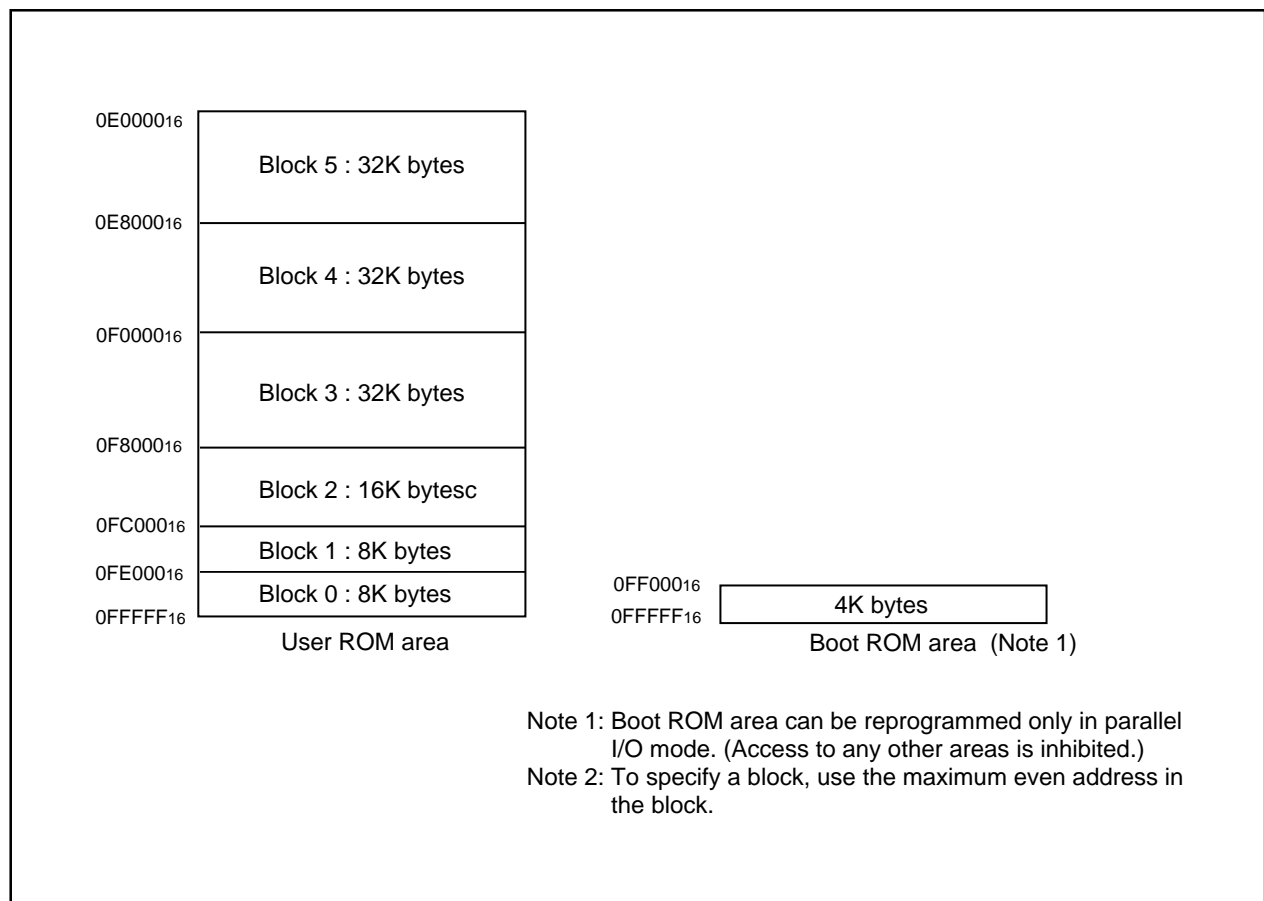


Fig.AB-1 Block diagram of flash memory version

CPU reprogram mode

In CPU reprogram mode, the on-chip flash memory can be operated on (read, program or erase) under the control of CPU.

In CPU reprogram mode, only the user ROM area shown in Fig.AB-1 can be reprogrammed. The boot ROM area cannot be reprogrammed. Make sure the program and block erase commands are issued only for each block of the user ROM area.

There are erase write mode 0 (EW0 mode) and erase write mode 1 (EW1 mode) in CPU reprogram mode. Table BB-1 shows the difference between EW0 mode and EW1 mode.

EW0 mode

The microcomputer is placed in CPU rewrite mode by setting the FMR0 register's FMR01 bit to "1" (CPU reprogram mode enabled), ready to accept commands. In this case, because the FMR1 register's FMR11 bit=0, EW0 mode is selected. The FMR01 bit can be set to "1" by writing "0" and then "1" in succession. Use software commands to control program and erase operations. Read the FMR0 register or status register to check the status of program or erase operation completion.

EW1 mode

EW1 mode is selected by setting FMR11 bit to "1" (by writing "0" and then "1" in succession) after setting the FMR01 bit to "1" (by writing "0" and then "1" in succession).

Read the FMR0 register to check the status of program or erase operation at completion. The status register can not be read during EW1 mode.

Microcomputer mode and Boot mode

The control program for CPU reprogram mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes disable.)

See Fig.AB-1 for details about the boot ROM area.

Normal microcomputer mode is entered when reset with pulling "L" of M0. In this case, the CPU starts operating the control program in user ROM area.

If the microcomputer is reset with M0 being "H" and M1 being "L", the CPU starts operating the control program in boot ROM area. This mode is called as "boot" mode.

Block address

Block address refers to the maximum even address of each block. The address is used in block erase command.

Table BB-1 The list of software command (CPU reprogram mode)

Item	EW0 mode	EW1 mode
Operation mode	<ul style="list-style-type: none"> •Single chip mode •Boot mode 	<ul style="list-style-type: none"> •Single chip mode
The area, which the reprogram control program can be allocated	<ul style="list-style-type: none"> •User ROM area •Boot ROM area 	<ul style="list-style-type: none"> •User ROM area
The area, which the reprogram control program can be executed	Should be transferred to RAM and then executed	<ul style="list-style-type: none"> •Can be executed in User ROM area
Reprogrammable area	<ul style="list-style-type: none"> •User ROM area 	<ul style="list-style-type: none"> •User ROM area However, the block, which the reprogram control program is allocated, is excluded.
The limitation of software command	No	<ul style="list-style-type: none"> •Program block area command Inhibited to the block, which reprogram control program is allocated •Read status register command Inhibited
The mode after program erase	Read status register mode	Read array mode
CPU status during auto programming auto erasing	Operate	Hold status (the status of I/O ports is hold the same before executing the command) (Note 1)
The check of the status of flash memory	*Check FMR00, FMR06, FMR07 bits of FMR0 register by S/W *Execute the read status register command, and then check SR7, SR5 and SR4 bit of status register	*Check FMR00, FMR06, FMR07 bits of FMR0 register by S/W

Note1: Do not use interrupts (exclude $\overline{\text{NMI}}$ and WDT).

Feature Outline (CPU reprogram mode)

In CPU reprogram mode, the writing and reading of the commands and data should be in even address ("0" for byte address A0) in 16-bit unit, so the 8-bit unit S/W commands should be written in even address. Commands are ignored with odd address.

Fig. BB-1 shows the flash recognition register, the flash control register 0 and the flash control register 1.

Bit 0 of flash control register 0 is the RY/BY status flag exclusively used to read the operating status of the flash memory. During programming and erasing operation, it is "0", otherwise it is "1".

Bit 1 of flash control register 0 is EW0 mode selection bit. When setting the bit to "1", EW0 mode is selected and the receiving of software command is possible. Keep $\overline{\text{NMI}}$ pin to "H" for the setting. For setting the bit to "1", it is necessary to set the bit to "0" and then to "1" in succession. For setting "0", only set the bit to "0".

Bit 3 of flash control register 0 is the flash memory reset bit used to reset the control circuit of the on-chip flash memory. The bit is used when exiting EW0 mode and when flash memory access has failed. When EW0 mode selection bit is "1", writing "1" to the bit resets the control circuit.

Bit 5 of the flash control register 0 is the user ROM selection bit. It is enabled only in boot mode. When the bit is set to "1", the accessed area is switched from boot ROM to user ROM. When CPU reprogram mode is entered in boot mode, set this bit to "1". The bit is disabled when program starts in user ROM. When in boot mode, the function of the bit is enabled regardless the CPU rewrite mode. Write the bit with the program that is not located in on-chip flash memory area.

Bit 6 of the flash control register 0 is a read only bit indicating the status of auto program operation. The bit is set to "1" when a program error occurs. Otherwise, it is cleared to "0".

Bit 7 of the flash control register 0 is a read only bit indicating the status of auto erase operation. The bit is set to "1" when a erase error occurs. Otherwise, it is cleared to "0".

Fig. BB-2 and Fig. BB-3 show the flow charts of the setting/resetting for EW0 mode and EW1 mode respectively. The operation specified in the flow chart should be followed.

Bit 1 of flash control register 1 is EW1 mode selection bit. When setting the bit to "1", EW1 mode is selected and the receiving of software command is possible. Keep $\overline{\text{NMI}}$ pin to "H" for the setting. For setting the bit to "1", the EW0 mode selection bit should be "1" and it is necessary to set the bit to "0" and then to "1" in succession. For setting to "0", only set the bit to "0". In the case that EW1 selection bit is set to "1" (both EW0 and EW1 mode selection bits are set to "1"), if writing "0" to EW0 mode selection bit, both EW0 and EW1 mode selection bits are cleared to "0".

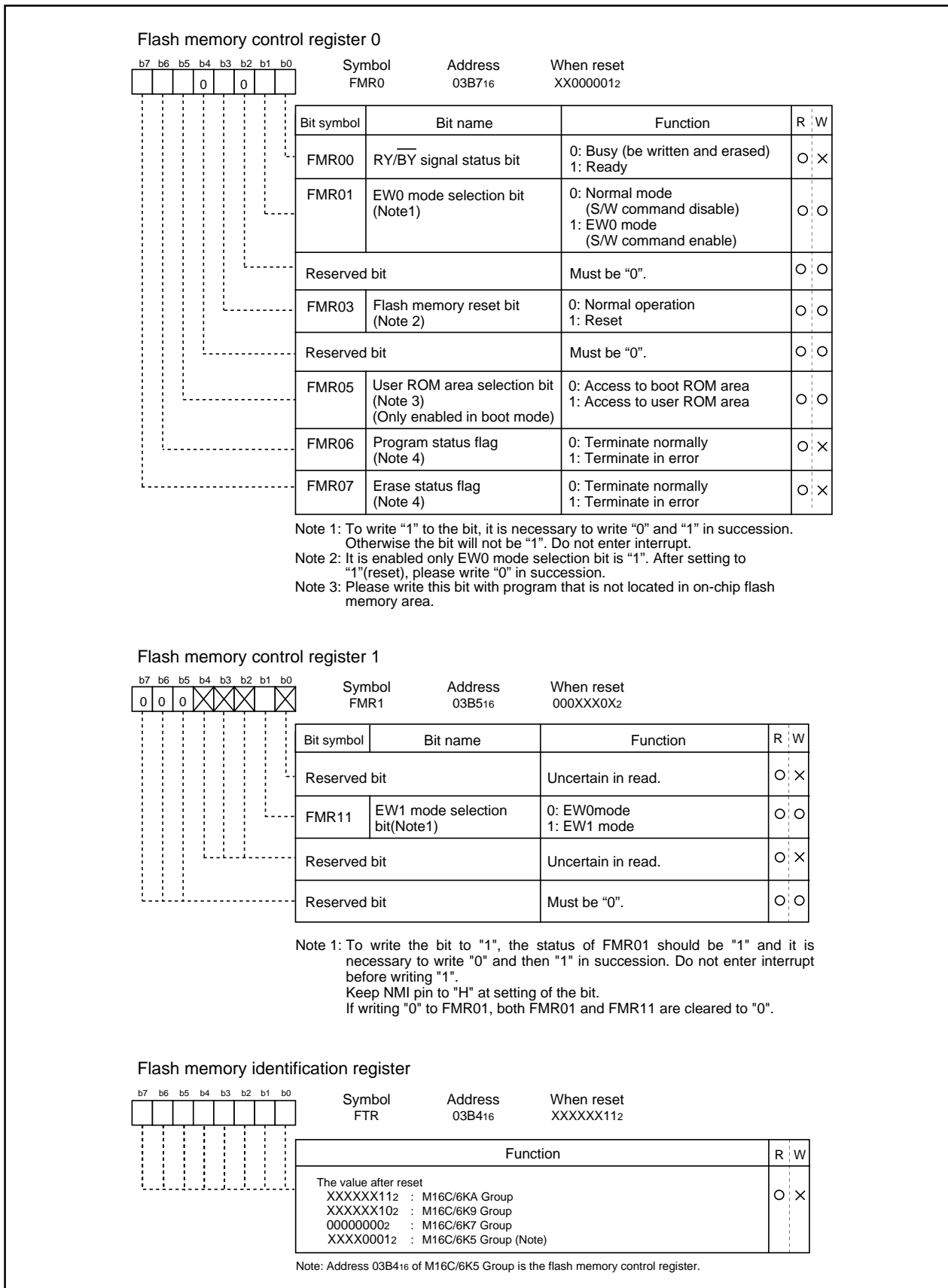


Fig.BB-1 The structure of flash memory control register and flash memory identification register

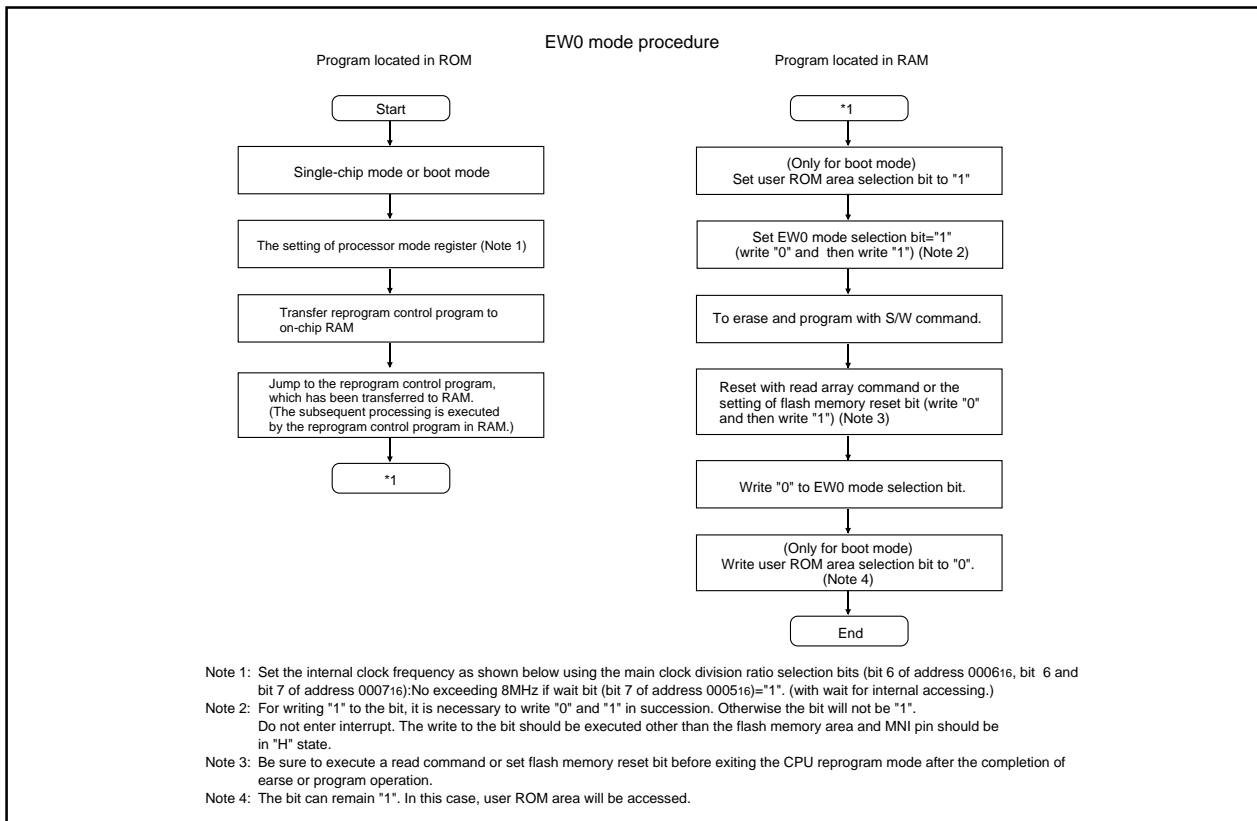


Fig.BB-2 CPU reprogram mode 0 set/reset flowchart

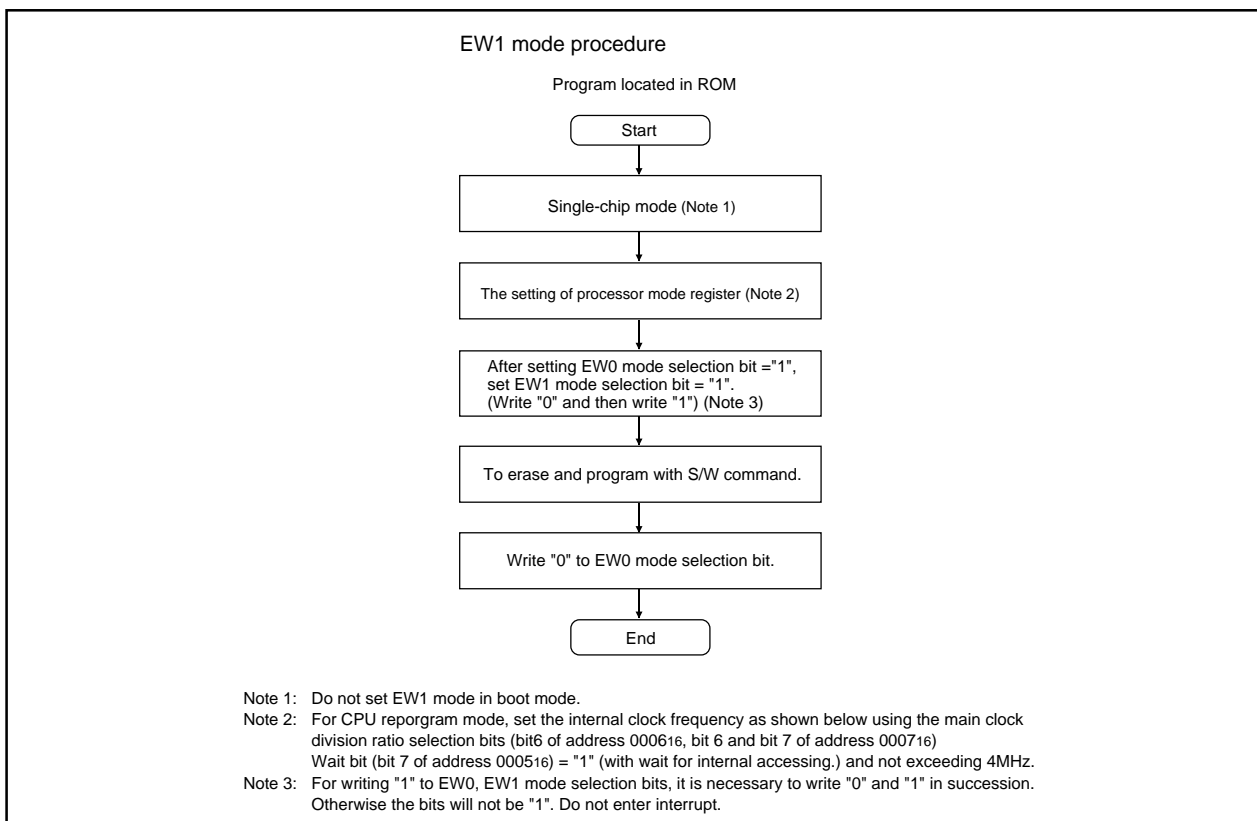


Fig.BB-3 CPU reprogram mode 1 set/reset flowchart

Precautions on CPU reprogram mode

Described below are the precautions to be observed in programming the flash memory in CPU reprogram mode.

(1) Operation speed

During CPU reprogram mode, set the internal clock frequency as shown below using the main clock divide ratio select bits (bit 6 at address 0006₁₆ and bit 6 and 7 at address 0007₁₆):

- EW0 mode

Not exceeding 8MHz if wait bit (bit 7 of address 0005₁₆) = "1". (with wait for internal accessing)

- EW1 mode

Not exceeding 4MHz if wait bit (bit 7 of address 0005₁₆) = "1". (with wait for internal accessing)

(2) Instructions inhibited against use

The instructions listed below cannot be used during CPU reprogram mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction and BRK instruction

(3) Interrupts inhibited against use

The $\overline{\text{NMI}}$, address match and WDC interrupts cannot be used during CPU reprogram mode because they refer to the internal data of the flash memory. If interrupts have their vectors in the variable vector table, they can be used by transferring the vector into the RAM area.

(4) Reset

The reset is always receivable.

(5) The reprogram in user ROM area

When CPU reprogram mode is entered and the block that the flash reprogram control program is located is being reprogramming, the block may not be reprogrammed correctly if the power supply is suddenly down. It is possible that the flash reprogram cannot be executed again in this case. Thus, it is recommended to use standard serial I/O mode and parallel I/O mode.

Software commands

Table BB-2 lists the S/W commands available.

In CPU reprogram mode, the S/W commands can be used to specify the erase or program operation. Note that when entering a S/W command, the upper byte (D15–D8) is ignored.

The content of each S/W command is explained below.

Table BB-2 List of software commands (CPU reprogram mode)

Command	Cycle number	The 1 st bus cycle			The 2 nd bus cycle		
		Mode	Address	Data (D15–D0)	Mode	Address	Data (D15–D0)
Read array	1	Write	X (Note 5)	FF ₁₆			
Read status register	2	Write	X	70 ₁₆	Read	X	SRD(Note 2)
Clear status register	1	Write	X	50 ₁₆			
Program	2	Write	X	40 ₁₆	Write	WA(Note 3)	WD(Note 3)
Block erase	2	Write	X	20 ₁₆	Write	BA(Note 4)	D0 ₁₆

Note 1: When a S/W command is input, the high-order byte of the data(D15–D8) is ignored.

Note 2: SRD = Status Register Data. The address should be even and within the user ROM area.

Note 3: WA = Write Address, WD = Write Data

Note 4: BA = Block Address (the maximum even address of the block)

Note 5: "X" can be any even address in user ROM area.

Read Array Command (FF₁₆)

Issuing the command code "FF₁₆" in the 1st bus cycle enters the read array mode. When an even address is issued in one of the bus cycle that follows, the content of the address is read out at the data bus (D15–D0), 16 bits at a time.

The read array mode is retained intact until another command is written.

Read Status Register Command (70₁₆)

When the command code "70₁₆" is issued in the 1st bus cycle, the content of the status register is read out at the data bus (D7–D0) by a read in the 2nd bus cycle. (The odd address in user ROM area should be specified.)

The status register is explained in the next section.

Clear Status Register Command (50₁₆)

The command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in error. To use this command, issue the command code "50₁₆" in the 1st bus cycle.

Program Command (4016)

Program operation starts when the command code "4016" is issued in the 1st bus cycle. If the address and data are issued in the 2nd bus cycle, program operation (data programming and verification) will start.

Whether the program operation is completed can be conformed by reading the status register or the $\overline{RY}/\overline{BY}$ status flag. When the program starts, the read status register mode is accessed automatically and the content of the status register can be read on the data bus (D7–D0). The status register bit 7 (SR7) is set to "0" at the same time when the program operation starts and is returned to "1" upon the completion of the program operation. In this case, the read status register mode remains active until the Read Array Command (FF16) is issued.

The $\overline{RY}/\overline{BY}$ status flag is "0" during program operation and "1" when the program operation is completed same as the status register bit 7.

After the program, reading the status register can check the result. Refer to the section where the status register is detailed.

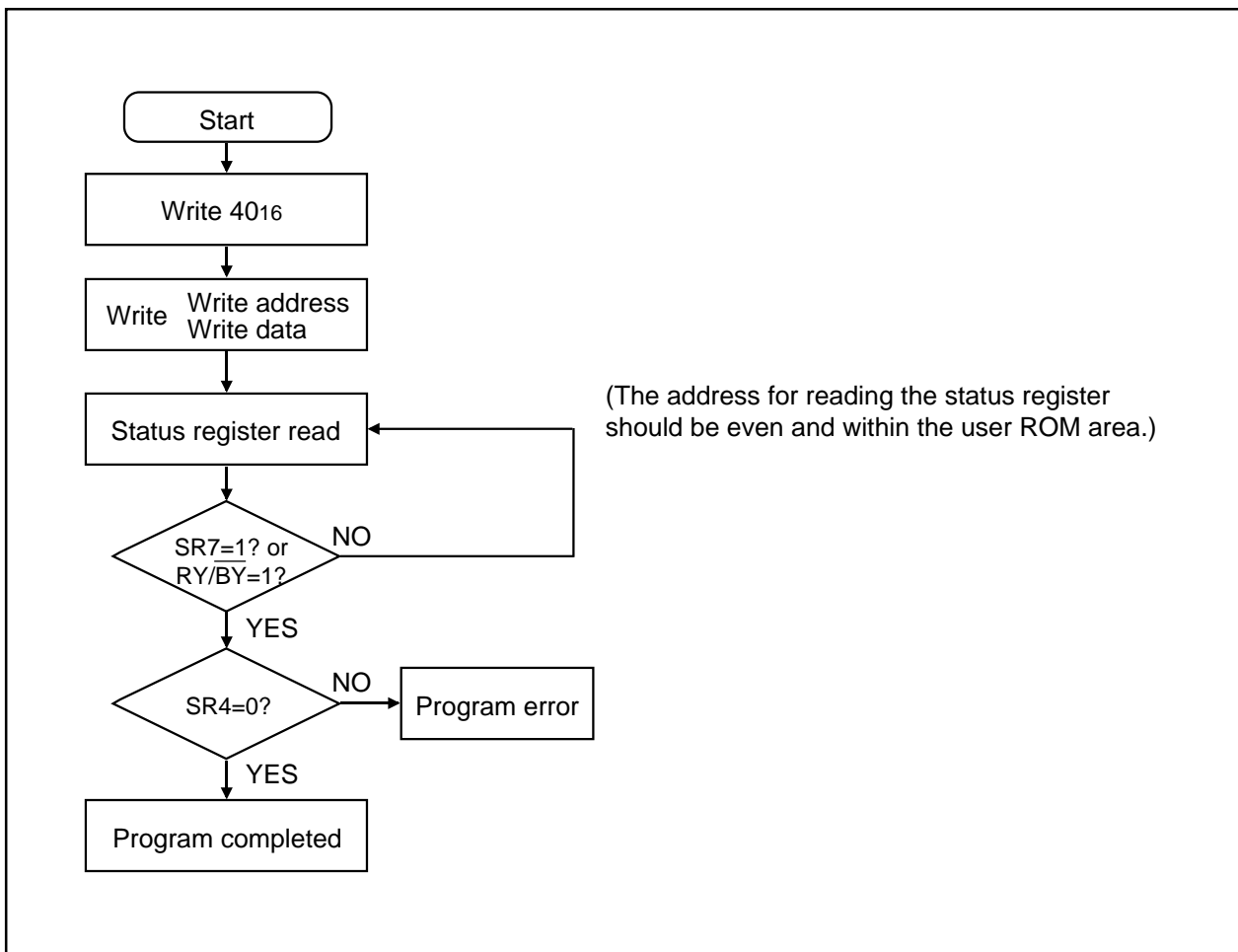


Fig.BB-4 Program flowchart

Block Erase Command (20₁₆/D0₁₆)

By issuing the command code "20₁₆" in the 1st bus cycle and the conformation command code "D0₁₆" and block address in the 2nd bus cycle, the erase operation specified by the block address starts (erase and erase verification).

Whether the block erase command is terminated can be conformed by reading the status register or the RY/ $\overline{\text{BY}}$ status flag. When the block erase operation starts, the read status register mode is accessed automatically and the content of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time when the erase operation starts and is returned to "1" upon the completion of the erase operation. In this case, the read status register mode remains active until the Read Array Command (FF₁₆) is written.

The RY/ $\overline{\text{BY}}$ status flag is "0" during erase operation and "1" when the erase operation is completed the same as the bit 7 of status register.

After the block erase, reading the status register can check the result. Refer to the section where the status register is detailed.

Block Erase Command is inhibited to the block, which reprogram control program is allocated.

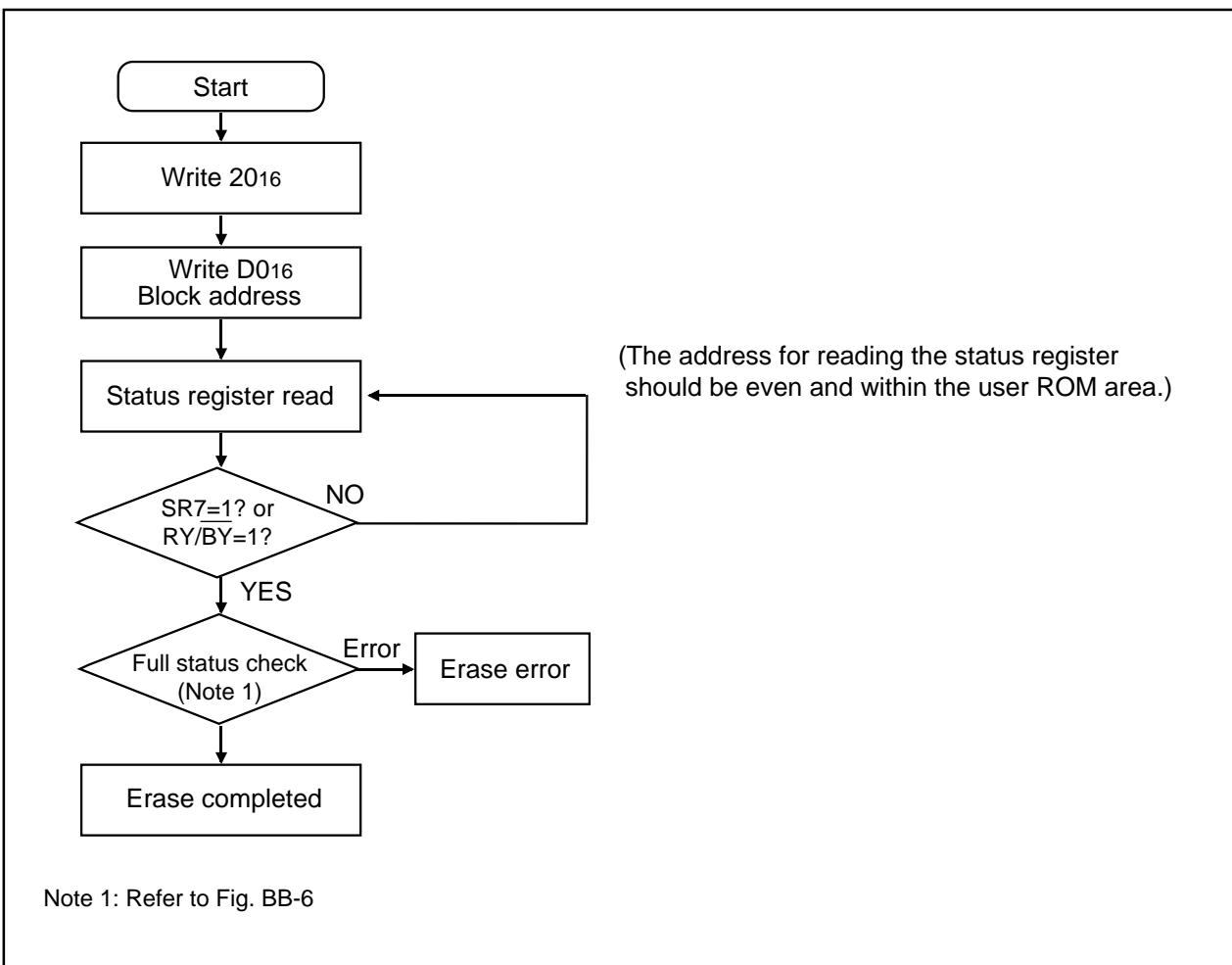


Fig.BB-5 Erase flowchart

Status register

The status register shows the operation status of the flash memory and whether program and erase operations end successfully or not. It can be read in the following conditions.

- (1) By reading an arbitrary address from the user ROM area after issuing the read status register command (70₁₆).
- (2) By reading an arbitrary address from the user ROM area in the period from the start of program or erase operation to the execution of read array command (FF₁₆).

Table BB-2 shows the status register.

The status register can be cleared in the following condition.

- (1) By issuing the clear status register command (50₁₆).
- (2) After reset, the status register is set to "80₁₆".

Each bit of the register is shown below.

Sequencer status (SR7)

After power-on, the sequencer status is set to "1" (ready).

The bit is set to "0" (busy) during program and erase operations and is set to "1" upon the completion of these operations.

Erase status (SR5)

Erase status indicates the status of erase operation. When erase error occurs, it is set to "1".

The bit becomes "0" when it is cleared.

Program status (SR4)

Program status indicates the status of program operation. When program error occurs, it is set to "1".

The bit becomes "0" when it is cleared.

If "1" is set to SR5 or SR4, the program and block erase operations are not accepted. Before execution of these commands, it is necessary to execute the clear status register command (50₁₆) to clear the status register.

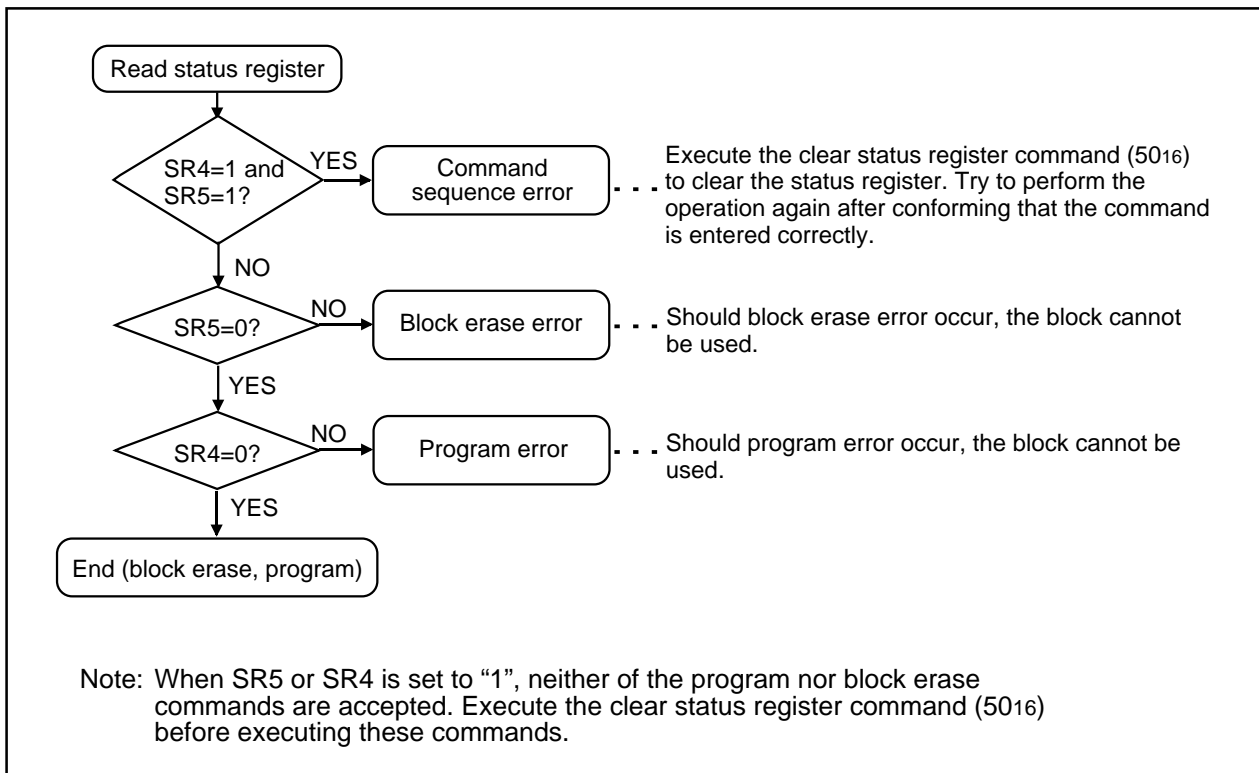
If any S/W commands are not correct, both the SR5 and SR4 are set to "1".

Table BB-3 Definition of each bit of status register

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Sequencer status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Full status check

By performing full status check, the execution result of erase and program operations can be known. Fig.BB-6 shows the full status check flowchart and the method to deal with the error.

**Fig.BB-6 Full status check flowchart and the method to deal with errors**

Functions to inhibit rewriting to the on-chip flash memory

To prevent flash memory from being miss-read or miss-written, ROM code protect function for parallel I/O mode and ID code check function for standard serial mode are introduced.

ROM code protect function

ROM code protect function can inhibit readout from or modification to the flash memory by setting the content in ROM code protect control address (0FFFFFF₁₆) for parallel I/O mode. Fig.BB-7 shows the content of ROM code protect control address (0FFFFFF₁₆). (The address exists in user ROM area.)

If one of the pair of ROM code protect bits is set to "0", ROM code protect is turned on, so that the flash memory is protected against the readout or modification. ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When both level 1 and level 2 are set, level 2 will be selected.

If both of the two ROM code protect reset bits are set to "00", ROM code protect is turned off, so that the flash memory can be read out or modified. Once ROM code protect is turned on, the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial mode or other to rewrite these two bits.

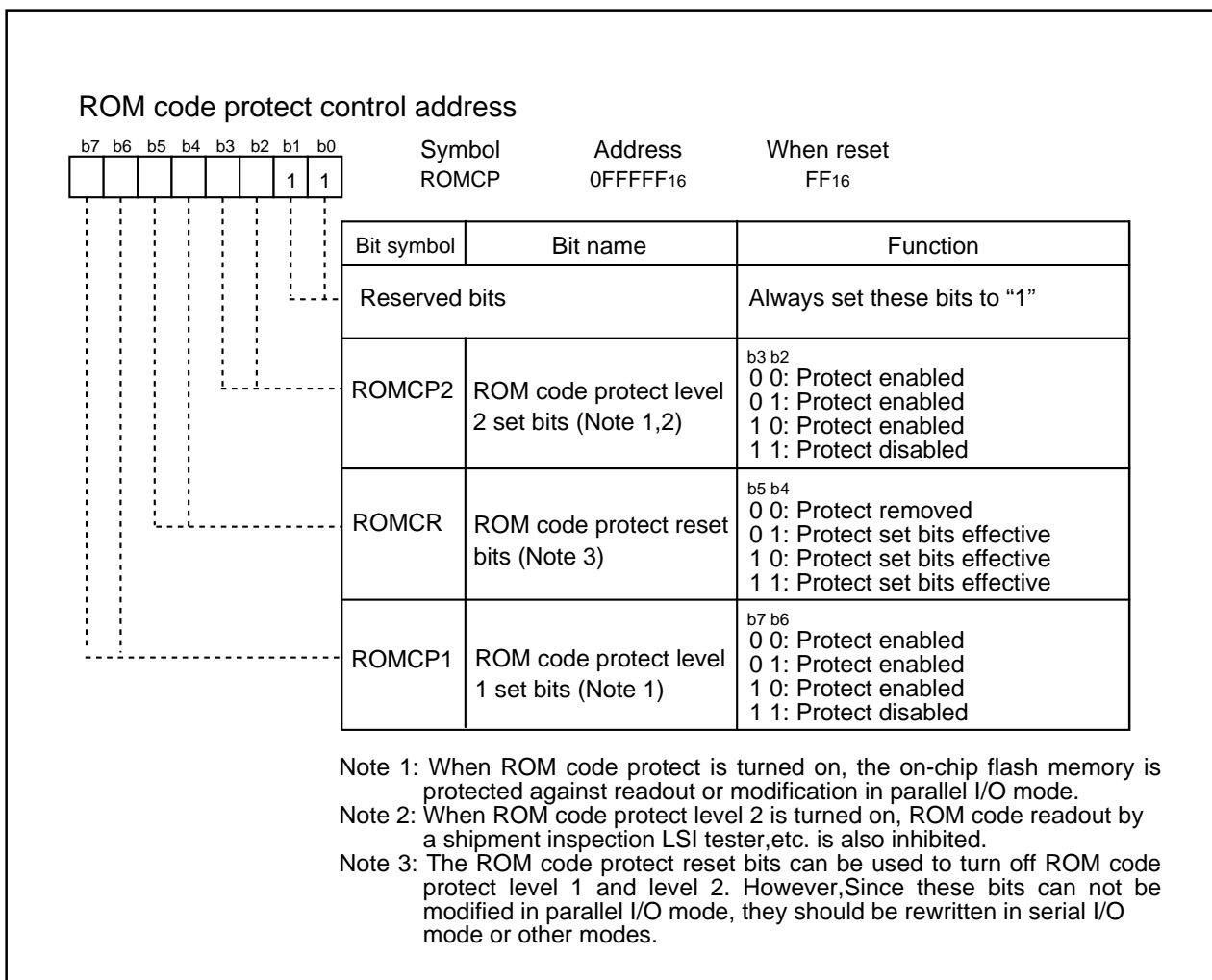


Fig.BB-7 ROM code protect control address

ID code check function

The function is used in standard serial I/O mode. If the flash memory is not blank, the ID code sent from serial burner is compared with that inside flash memory to check the agreement. If the ID codes do not match, the commands from serial burner are not accepted. Each ID code consists of 8-bit data, the areas of which, beginning from the 1st byte, are 0FFFDF₁₆, 0FFFE3₁₆, 0FFFEB₁₆, 0FFFEF₁₆, 0FFFF3₁₆, 0FFFF7₁₆, 0FFFFB₁₆. Write a program with the ID code at these addresses to the flash memory.

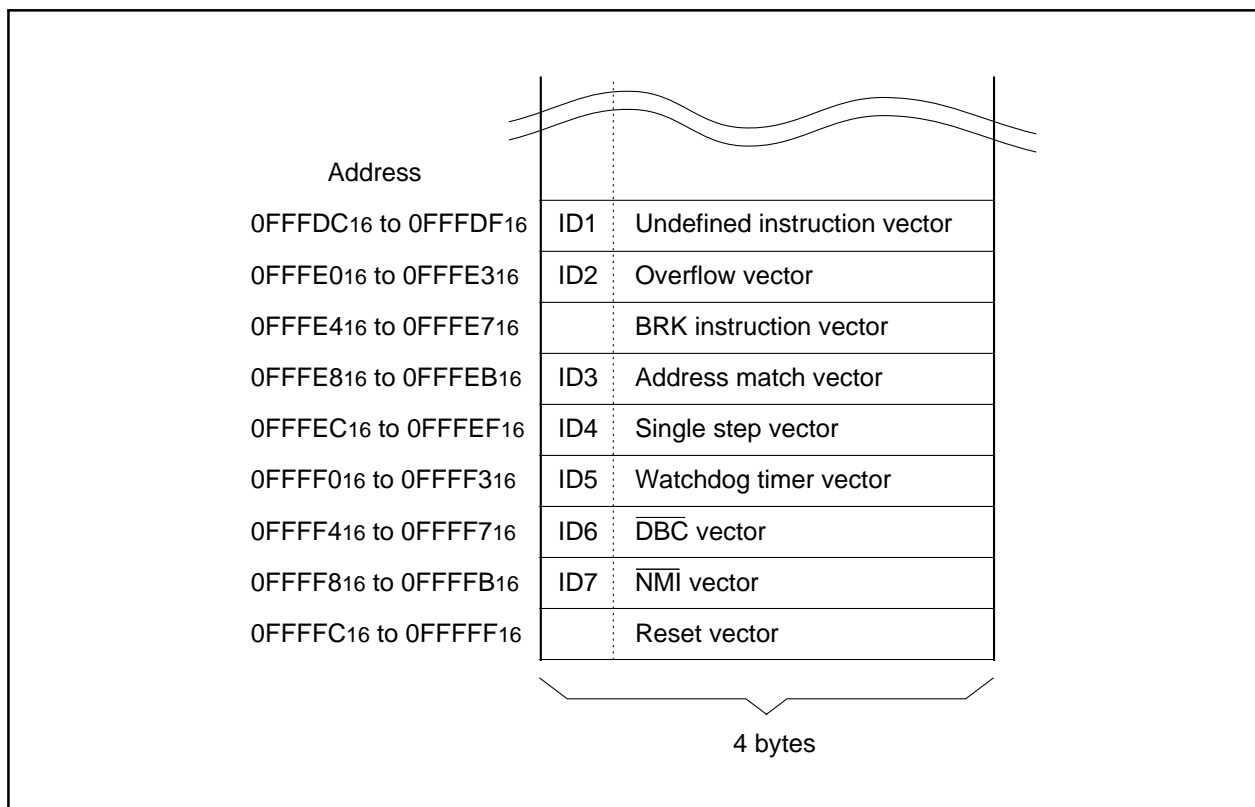


Fig.BB-8 ROM ID code addresses

Parallel I/O mode

Parallel I/O mode is to input and output the software command, address and data in parallel to access the on-chip flash memory (read, program, erase etc.).

Please use the specific device (programmer) supported for M16C/6KA Group. Referring to the guideline etc. of each device manufacture for the usage.

User ROM area and boot ROM area

In parallel I/O mode, both user ROM area and boot ROM area showed in Fig.AB-1 can be reprogrammed. The access method to both areas is the same.

The size of boot ROM area is 4K bytes. The addresses are allocated in $0FF000_{16}$ – $0FFFFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)

In the boot ROM area, erase block operation is applied to only one 4K bytes block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from Renesas factory. Therefore, if the standard serial I/O mode is used, the rewriting to the boot ROM area is not necessary.

Table EE-1 Pin function (Flash memory standard serial I/O mode)

Pin name	Name	I/O	
Vcc, Vss	Power supply		Apply $3.3 \pm 0.3V$ to Vcc, apply 0V to Vss
M0	M0	I	Connect to Vcc
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While reset is "L", 20 cycles or more clocks input to XIN pin are needed.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT. If external clock is used, input it to XIN pin and open the XOUT pin.
XOUT	Clock output	O	
M1	M1	I	Connect to Vss
AVcc, AVss	Analog power supply		Connect AVss to Vss, AVcc to Vcc
VREF	Reference voltage	I	The input pin of reference voltage of AD converter
P00–P07	Input port P0	I	Input "H", "L" or open
P10–P17	Input port P1	I	Input "H", "L" or open
P20–P27	Input port P2	I	Input "H", "L" or open
P30–P37	Input port P3	I	Input "H", "L" or open
P40–P47	Input port P4	I	Input "H", "L" or open
P50–P57	Input port P5	I	Input "H", "L" or open
P60–P63	Input port P6	I	Input "H", "L" or open
P64	BUSY output	O	The output pin of BUSY signal
P65	SCLK input	I	The input pin of serial clock
P66	RxD input	I	The input pin if serial data
P67	TxD input	O	The output pin of serial data
P70–P77	Input port P7	I	Input "H", "L" or open
P80–P84 P86, P87	Input port P8	I	Input "H", "L" or open
P85	NMI input	I	Connect to Vcc
P90–P97	Input port P9	I	Input "H", "L" or open
P100–P107	Input port P10	I	Input "H", "L" or open
P110–P117	Input port P11	I	Input "H", "L" or open
P120–P127	Input port P12	I	Input "H", "L" or open
P130–P137	Input port P13	I	Input "H", "L" or open
P140–P147	Input port P14	I	Input "H", "L" or open
P150–P157	Input port P15	I	Input "H", "L" or open
P160, P161	Input port P16	I	Input "H", "L" or open

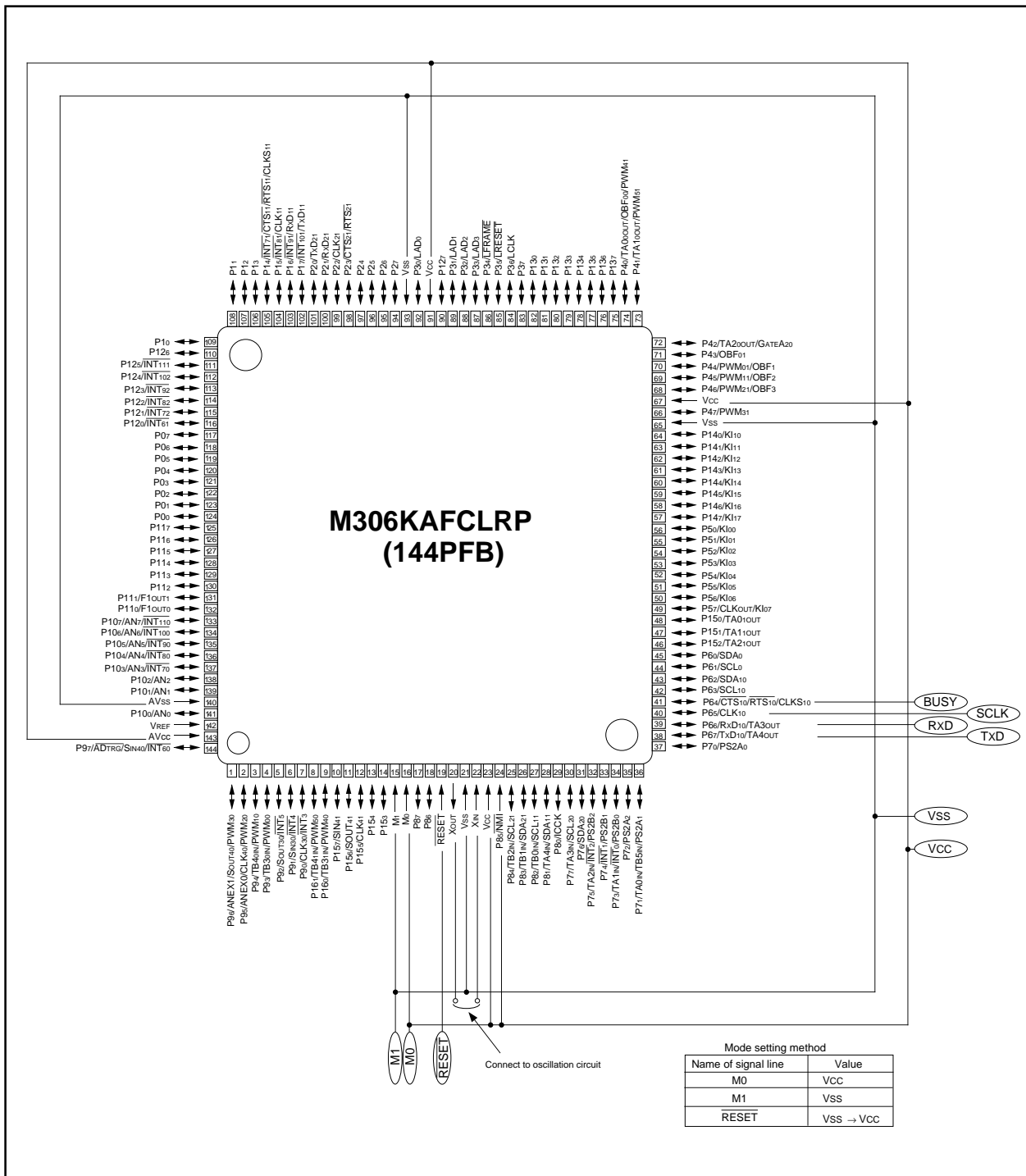


Fig.EE-1 Pin connections for serial I/O mode

Standard serial I/O mode

The standard serial I/O mode inputs and outputs the S/W commands, addresses and data needed to operate (read, program, erase etc.) the on-chip flash memory with a dedicated serial programmer.

Different from parallel I/O mode, in standard serial mode, CPU controls the flash memory reprogramming (uses the CPU reprogram mode) and the input of serial reprogram data etc. The standard serial I/O mode is started by connecting M0 to "H", M1 to "L" with the release of reset. (To connect M0 to "L" in normal microcomputer mode.)

This control program is written in boot ROM area when the product is shipped from Renesas factory. Make sure that the standard serial I/O mode cannot be used if boot ROM area is written in parallel I/O mode. Fig EE-1 shows the pin connections for standard serial I/O mode. The input and output of serial data are processed in CLK10, RxD10, TxD10, RTS10 (BUSY) 4 pins of the UART1.

The CLK10 is clock input pin, which clock is input externally. The TxD10 is CMOS output pin. The RTS10 (BUSY) pin outputs "L" when ready for reception and outputs "H" when reception starts. The serial data are transferred in 8-bit unit.

In standard serial I/O mode, only the user ROM area shown in Fig.AB-1 can be reprogrammed. Boot ROM area cannot be reprogrammed.

In the standard serial I/O mode, a 7-byte ID code is used. If the flash memory is not blank, commands sent from programmer are not accepted unless the ID code matches.

Outline (standard serial I/O mode)

In standard serial I/O mode, S/W commands, addresses, and data etc. are input and output with the peripheral device (serial programmer) using 4-wire clock-synchronized serial I/O (UART1). In reception, S/W commands, addresses and program data are read from RxD₁₀ pin synchronized with the rising edge of the transfer clock that is input to the CLK₁₀ pin. In transmission, the read data and status are output to TxD₁₀ pin synchronized with the falling edge of the transfer clock.

The TxD₁₀ is CMOS output pin. Transfer is in 8-bit unit with LSB first.

During transmission, reception, erasing and programming, the RTS₁₀ (BUSY) pin is "H". Accordingly, always start the next transfer after the RTS₁₀ (BUSY) pin becomes "L".

The read after the input of S/W commands can get memory data and status register. Reading the status register can check the flash memory operation status, the normal/error end of erasing or programming operation. The following are the explanation of S/W commands, status register etc.

S/W commands

Table EE-2 lists the S/W commands. In standard serial I/O mode, the S/W commands, which transferred from RxD pin, control of erase, program and read etc. The S/W commands in standard serial I/O mode are similar with that in parallel I/O mode. ID check function, download function, version information output function, boot ROM area output function and read check data, 5 commands are added.

Table EE-2 The list of S/W commands (standard serial I/O mode)

	Control command	1 st byte transfer	2 nd byte	3 rd byte	4 th byte	5 th byte	6 th byte	–	If ID unmatched
1	Page read	FF ₁₆	Address (middle)	Address (high)	Data output	Data output	Data output	259 th byte data output	Not acceptable
2	Page program	41 ₁₆	Address (middle)	Address (high)	Data input	Data input	Data input	259 th byte data input	Not acceptable
3	Block erase	20 ₁₆	Address (middle)	Address (high)	D0 ₁₆				Not acceptable
4	Read status register	70 ₁₆	SRD output	SRD1 output					Acceptable
5	Clear status register	50 ₁₆							Not acceptable
6	ID check function	F5 ₁₆	Address (low)	Address (middle)	Address (high)	ID size	ID1	–ID7	Acceptable
7	Download function	FA ₁₆	Address (low)	Address (high)	Check sum	Data input	No. of times required		Not acceptable
8	Version information output function	FB ₁₆	Version data output	Version data output	Version data output	Version data output	Version data output	–9 th byte Version data output	Acceptable
9	Boot ROM area output function	FC ₁₆	Address (middle)	Address (high)	Data output	Data output	Data output	–259 th byte data output	Not acceptable
10	Read check data	FD ₁₆	Check data (low)	Check data (high)					Not acceptable

Note 1: Shading indicates transfer from flash memory on chip microcomputer to serial programmer.

The else indicates transfer from serial programmer to flash memory on chip microcomputer.

Note 2: SRD means status register data. SRD1 means status register 1 data.

Note 3: All commands are acceptable if the flash memory is blank.

The following are the descriptions of S/W commands

Page read command

The command reads the specified page (256 bytes) of the flash memory sequentially one byte at a time. Execute the page read command as following:

- (1) Transfer the "FF16" command code in the 1st byte.
- (2) Transfer addresses A8– A15 and A16– A23 in the 2nd and 3rd byte respectively.
- (3) From the 4th byte onward, data (D7–D0) of the page specified by the address (A23–A8) will be output sequentially from the smallest address sync with the falling edge of the clock.

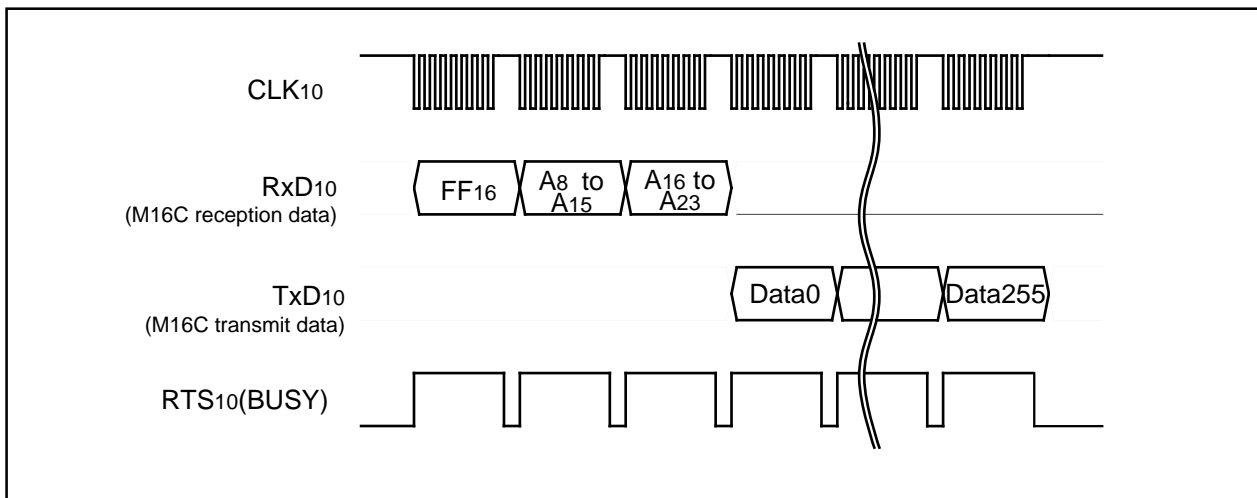


Fig.EE-2 Timing of page read

Read status register command

The command is for reading status information. When command code "7016" is sent in the 1st byte, the contents of status register (SRD) and status register 1 (SRD1) will be output in the 2nd and 3rd byte respectively sync with the falling edge of the clock.

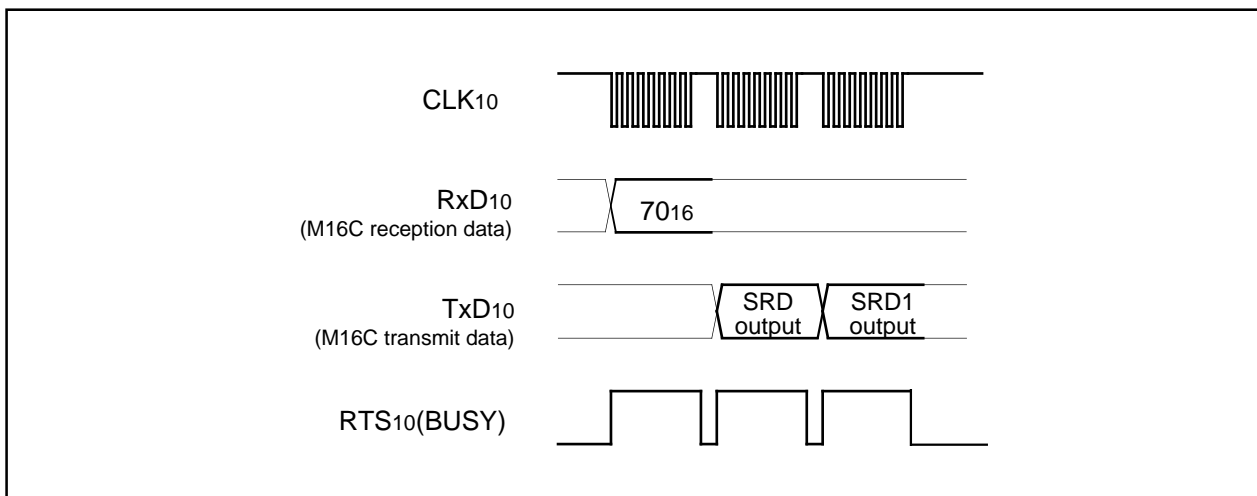


Fig.EE-3 Timing of read status register

Clear status register command

The command clears the bits (SR4–SR5), which are set when operation ended in error. When command code “50₁₆” is sent in the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS₁₀ (BUSY) signal changes from “H” to “L”.

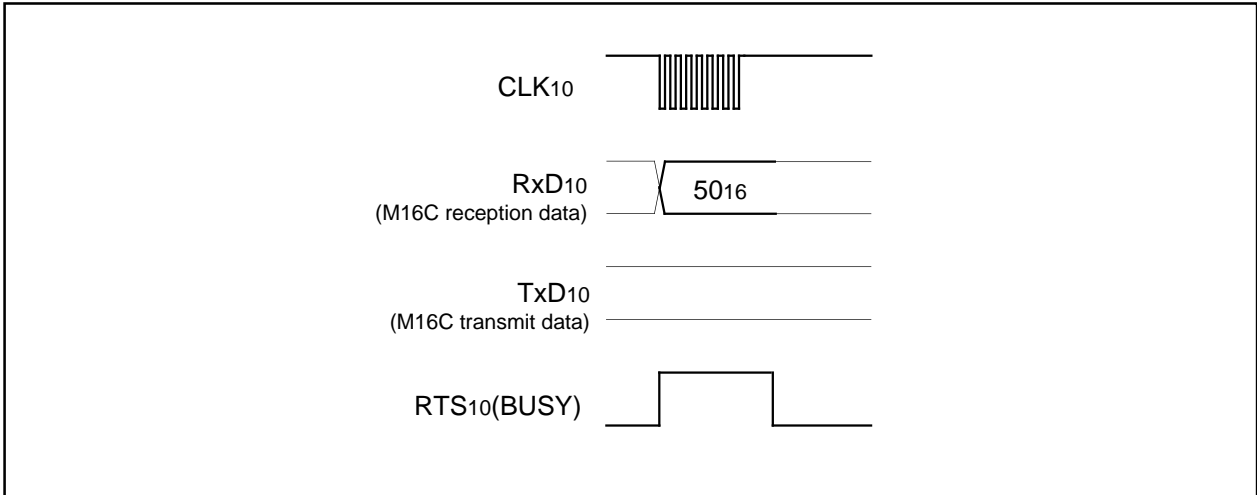


Fig.EE-4 Timing of clear status register

Page program command

The command programs the specified page (256 bytes) of flash memory sequentially one byte a time. Execute the command as follows:

- (1) Transfer the command code “41₁₆” in the 1st byte.
- (2) Transfer addresses A₁₅–A₈ and A₂₃–A₁₆ in the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, after inputting 256 bytes program data (A₇–A₀) from the smallest address of the specified page, the page program operation will be executed automatically.

When the reception for the next 256 bytes is setup, the RTS₁₀ (BUSY) signal changes from “H” to “L”. The result of the page program can be known by reading the status register. For more detail, see the section on the status register.

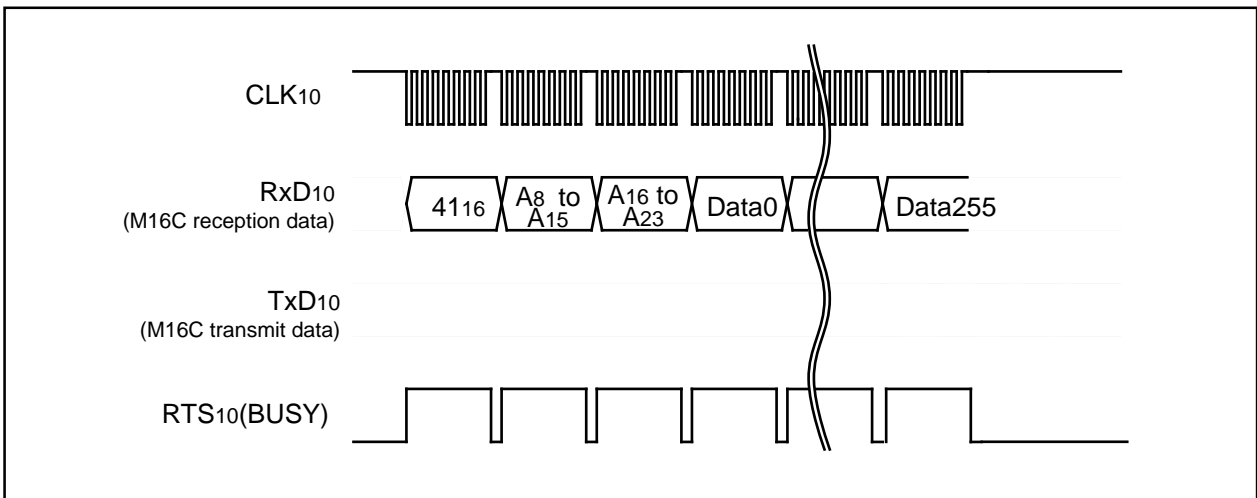


Fig.EE-5 Timing of page program

Block erase command

The command erases the data in the specified block. Execute the command as follows:

- (1) Transfer the command code "2016" in the 1st byte.
- (2) Transfer addresses A15–A8 and A23–A16 in the 2nd and 3rd bytes respectively.
- (3) After transferring the verify command code "D016" in the 4th byte, the erase operation starts for the specified block of the flash memory. Issue the biggest address of the specified block to A23–A8.

After the completion of block erase, the RTS10 (BUSY) signal changes from "H" to "L". The result of the block erase can be know by reading the status register. For more detail, see the section on the status register.

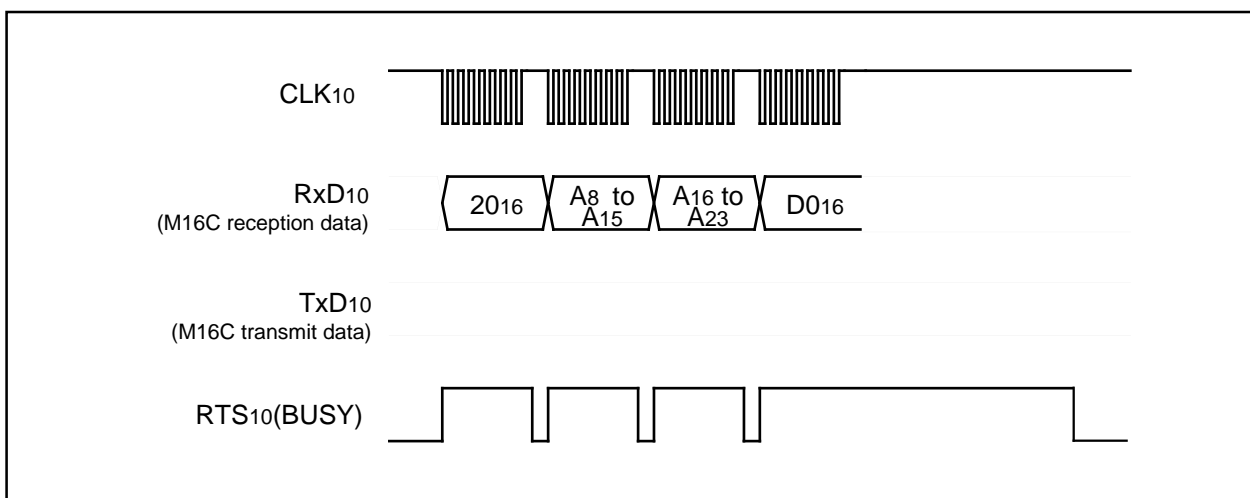


Fig.EE-6 Timing of block erase

Download function

The command downloads an execution program to RAM. Execute the command as follows:

- (1) Transfer the command code "FA16" in the 1st byte.
- (2) Transfer the program size in the 2nd and 3rd bytes.
- (3) Transfer the checksum in the 4th byte. Check sum is calculated from all transferred data from the 5th byte onward.
- (4) The execution program is transferred from 5th byte onward.

After the entire program data have been transferred, the downloaded execution program will be executed if the checksum matches.

The program size allowed to transfer varies according to the size of on-chip RAM.

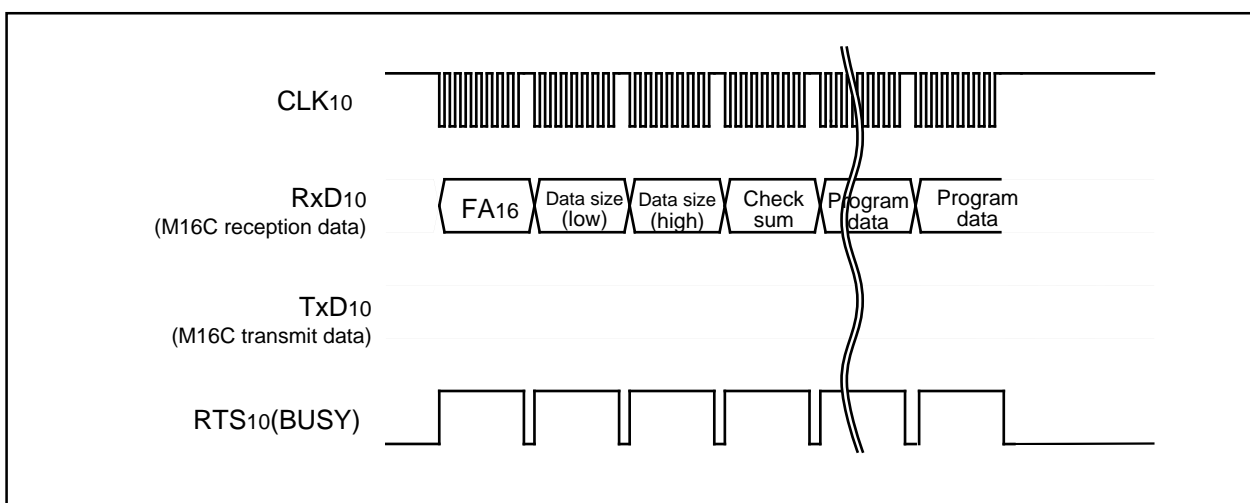


Fig.EE-7 Timing of download function

Version information output function

The version information of the control program stored in boot ROM area can be output by the function. Execute the command as follows:

- (1) Transfer the command code "FB16" in the 1st byte.
- (2) From the 2nd byte onward, the version information will be output. The information is composed of 8 ASCII character code.

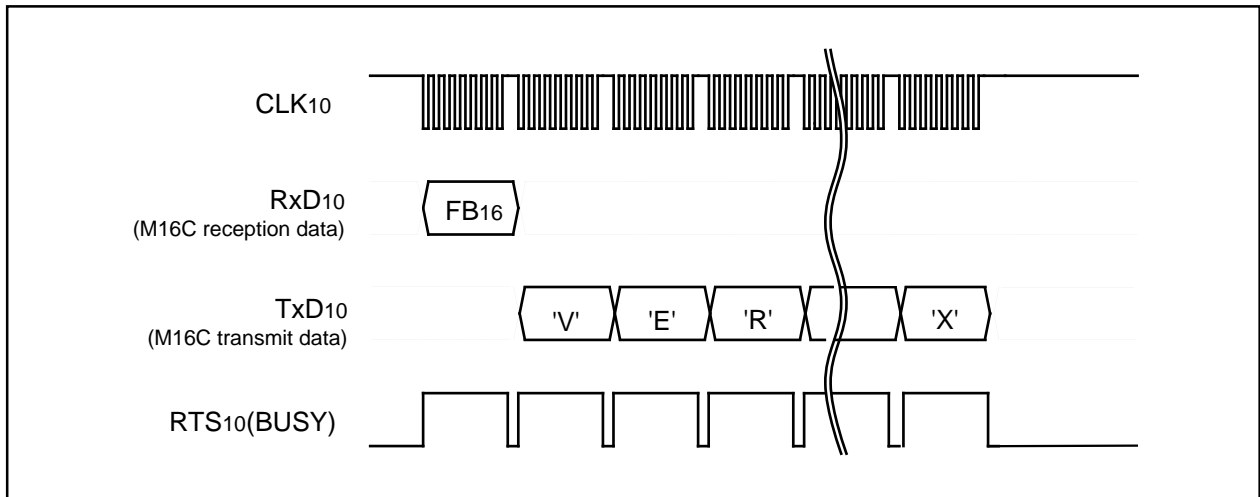


Fig.EE-8 Timing of version information output function

Boot ROM area output function

The control program stored in boot ROM area can be read out in page (256 bytes) unit by the function. Execute the command as follows:

- (1) Transfer the command code "FC16" in the 1st byte.
- (2) Transfer addresses A15–A8 and A23–A16 in the 2nd and 3rd bytes respectively.

From the 4th byte onward, the data (D7–D0) specified in page (256 bytes) address A23–A8 will be output sequentially from the smallest address in sync with the rising edge of the clock.

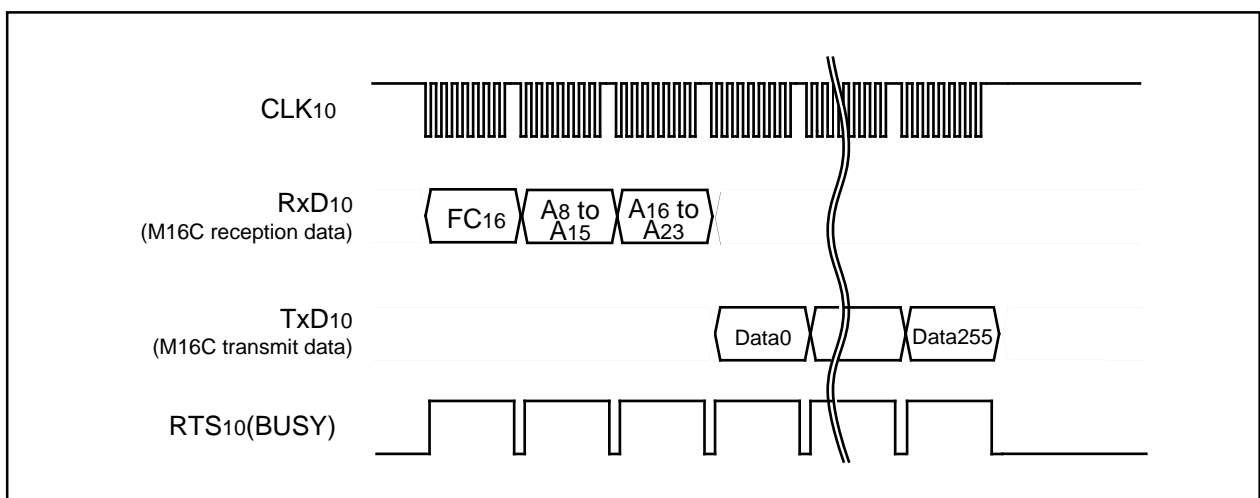


Fig.EE-9 Timing of boot ROM area output function

ID check function

The command checks the ID code. Execute the command as follows:

- (1) Transfer the command code "F5₁₆" in the 1st byte.
- (2) Transfer addresses A₇–A₀, A₁₅–A₈ and A₂₃–A₁₆ of 1st ID code (ID1) in the 2nd, 3rd and 4th bytes respectively.
- (3) Transfer the number of the ID code in the 5th byte.
- (4) From the 6th byte onward, transfer the IDs from the 1st ID code (ID1).

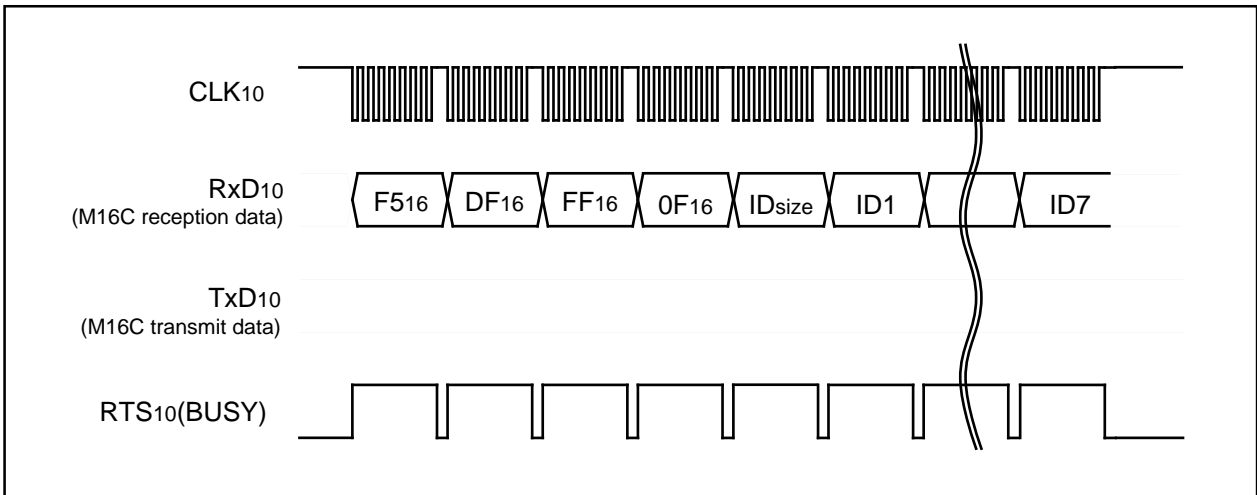


Fig.EE-10 Timing of ID check function

ID code

If the flash memory is not blank, the input ID codes are compared with that written in flash memory. If they do not match, the input commands will not be accepted. Each ID code contains 8 bits data. Beginning from the 1st ID byte, the address of each ID code is 0FFFDF₁₆, 0FFFE0₁₆, 0FFFE3₁₆, 0FFFE4₁₆, 0FFFE7₁₆, 0FFFE8₁₆, 0FFFEB₁₆, 0FFFEF₁₆, 0FFFF3₁₆, 0FFFF7₁₆ and 0FFFFB₁₆ respectively. Write the program with the ID codes in these addresses to the flash memory.

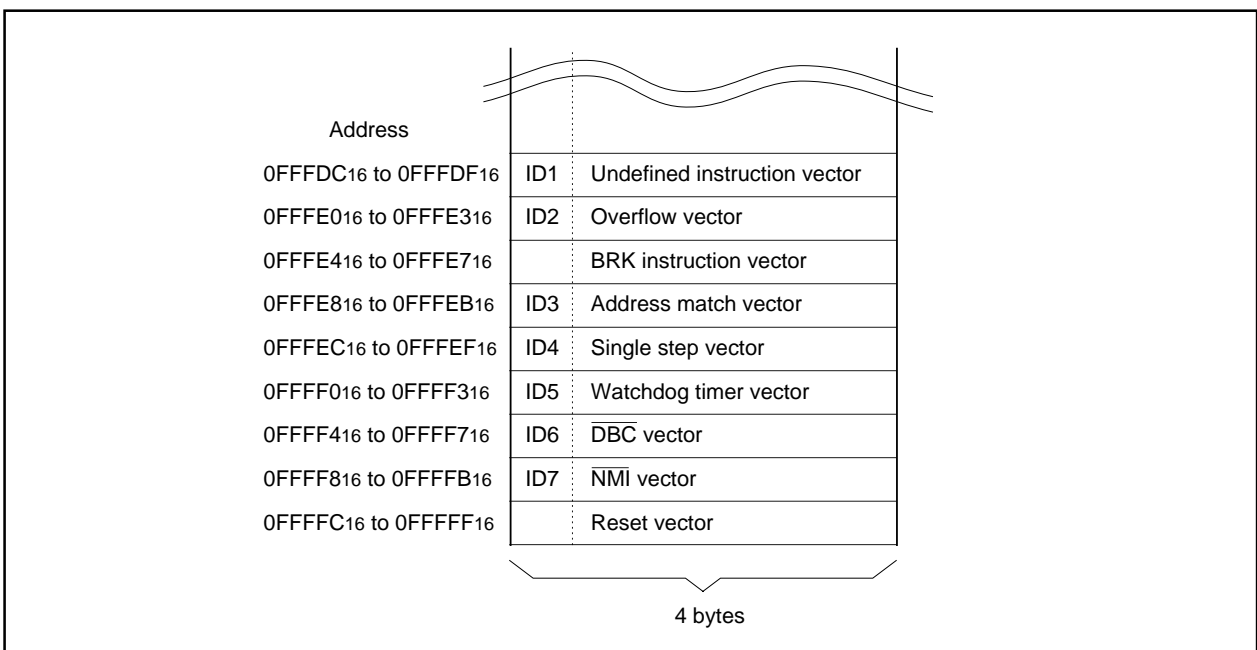


Fig.EE-11 ID code addressed

Read check data

Read check data command is for conforming if the reprogram data sent after page program command have been received correctly.

- (1) Transfer the command code "FD16" in the 1st byte.
- (2) Transfer check data (low) and check data (high) in the 2nd and 3rd bytes respectively.

When using read check data command, the command should be issued at first to initialize the check data. The next is to issue the page program command and related reprogram data. After that, by issuing the read check data command again, the check data for the reprogram data issued between the two read check data command can be read out.

Adding the reprogram data in byte unit and then calculating the lower 2 bytes of the added data in two's complement gives out the check data.

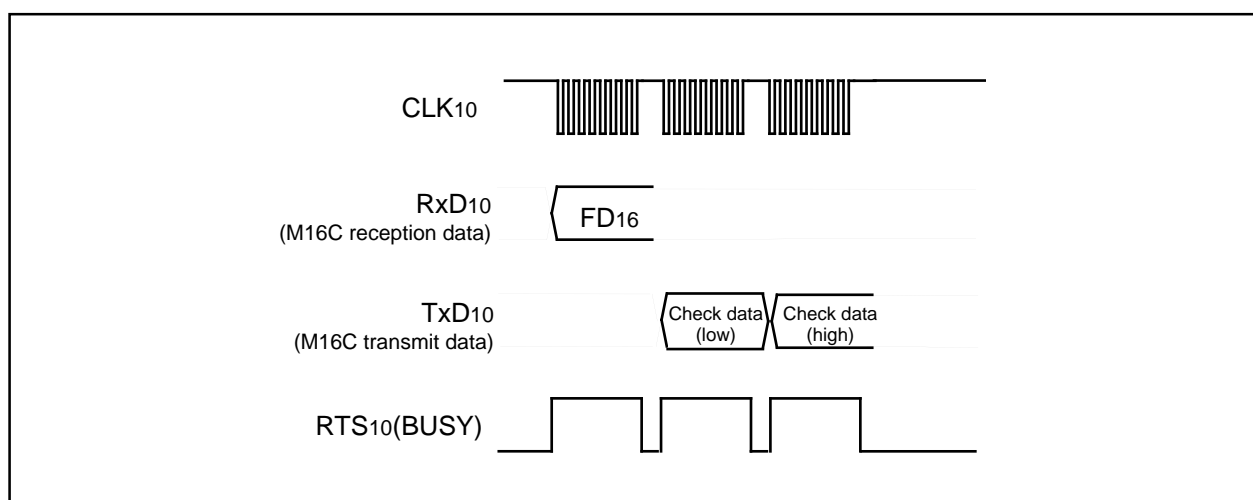


Fig.EE-12 Timing of read check dt command

Status register

Status register indicates if the operation to the flash memory ends successfully or in error. It can be read by issuing the read status register command (70₁₆). The status register can be cleared by issuing the clear status register command (50₁₆).

Table EE-3 shows the definition of each bit of the register.

After reset, status register outputs "80₁₆".

Table EE-3 Status register (SRD)

Symbol	Status	Definition	
		"1"	"0"
SR7 (D7)	Sequencer status	Ready	Busy
SR6 (D6)	Reserved	-	-
SR5 (D5)	Erase status	Terminated in error	Terminated normally
SR4 (D4)	Program status	Terminated in error	Terminated normally
SR3 (D3)	Reserved	-	-
SR2 (D2)	Reserved	-	-
SR1 (D1)	Reserved	-	-
SR0 (D0)	Reserved	-	-

Sequencer status (SR7)

After power-on, the sequencer status is set to "1" (ready).

The bit is set to "0" (busy) during program and erase operations and is set to "1" upon the completion of these operations.

Erase status (SR5)

Erase status indicates the status of erase operation. When erase error occurs, it is set to "1".

The bit becomes "0" when it is cleared.

Program status (SR4)

Program status indicates the status of program operation. When program error occurs, it is set to "1".

The bit becomes "0" when it is cleared.

Status register 1 (SRD1)

Status register 1 indicates the status of serial communication, the result of ID codes comparison, the result of checksum comparison etc. It can be read after SDR by issuing the read status register command (70₁₆). The register can be cleared by issuing the clear status register command (50₁₆).

Table EE-4 shows the definition of each bit of the register.

After power on, status register 1 outputs "00₁₆".

Table EE-4 Status register (SRD1)

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Check sum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 Not verified 01 Verified with mismatch 10 Reserved 11 Verified with match	
SR9 (bit1)	Timeout of data reception	Timeout	Normal operation
SR8 (bit0)	Reserved	-	-

Boot update completed bit (SR15)

The flag indicates that if the control program has been downloaded to RAM with download function.

Check sum match bit (SR12)

The flag indicates if the check sum is matched when downloading the control program with download function.

ID check completed bits (SR11, SR10)

These bits indicate the result of ID checks. Some commands cannot be accepted without the ID checks.

Timeout of data reception bit (SR9)

The flag indicates if timeout occurs during data reception. If the bit is set to "1" during data reception, microcomputer will discard the received data and return to wait state.

Full status check

By performing full status check, the execution result of erase and program operations can be known. Fig.EE-13 shows the full status check flowchart and the method to deal with the error.

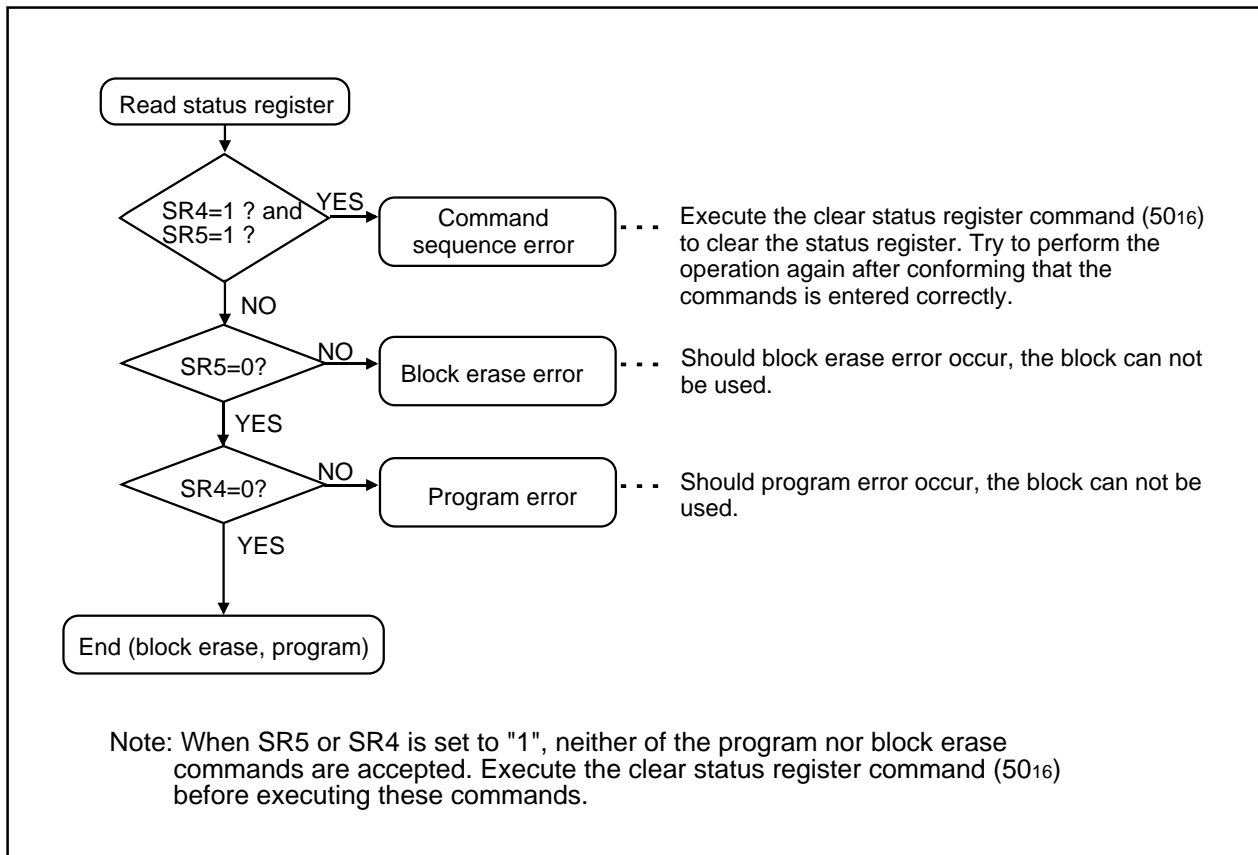


Fig.EE-13 Full status check flowchart and the method to deal with errors

Circuit applied for standard serial I/O mode (example)

The figure below shows a circuit applied for standard serial I/O mode. The control pins vary by different programmer. Refer to programmer manual for the detail.

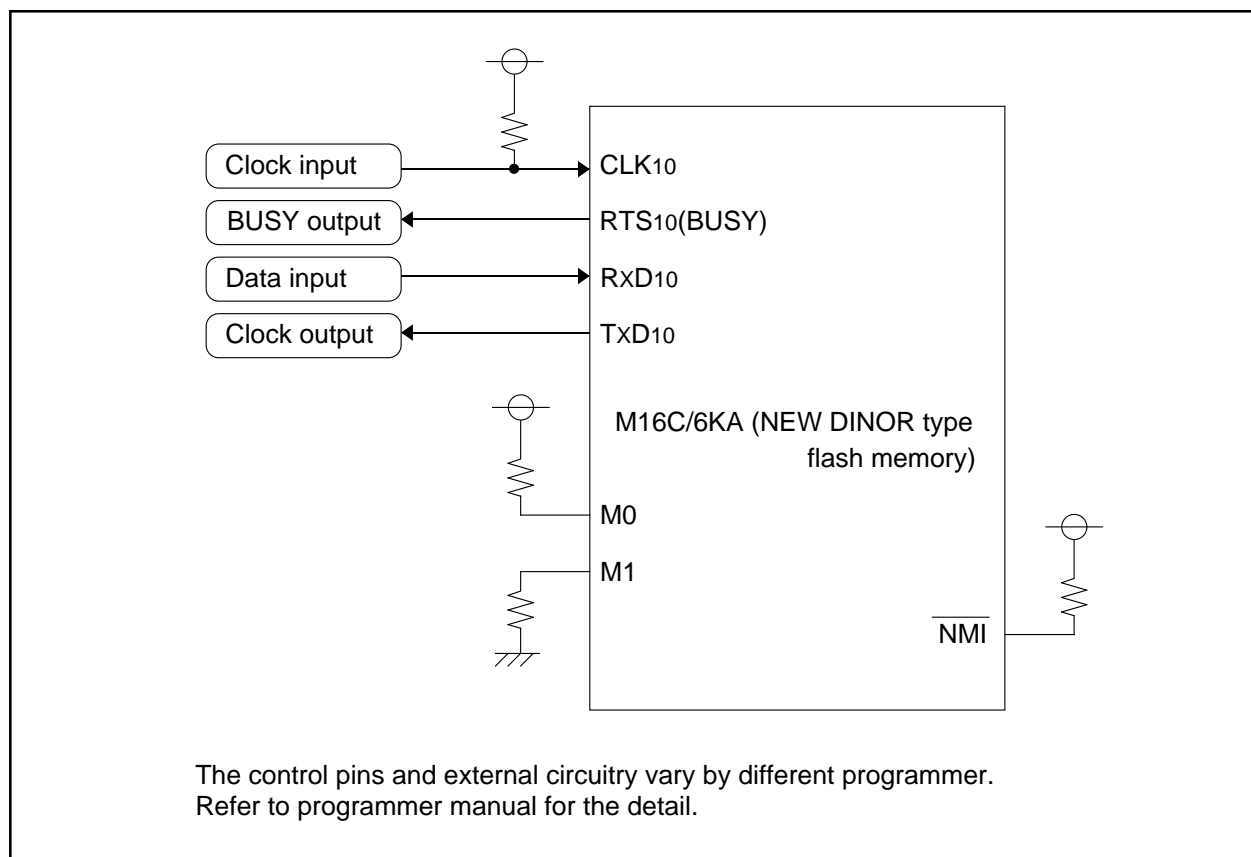


Fig.EE-14 Example circuit applied for the standard serial I/O mode

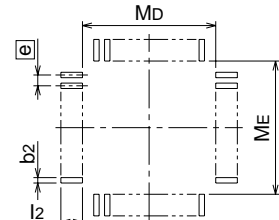
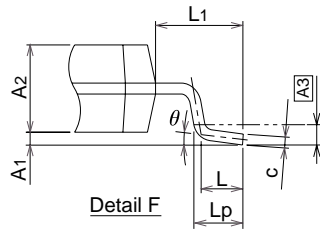
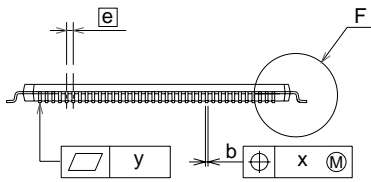
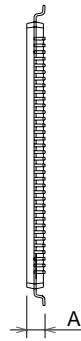
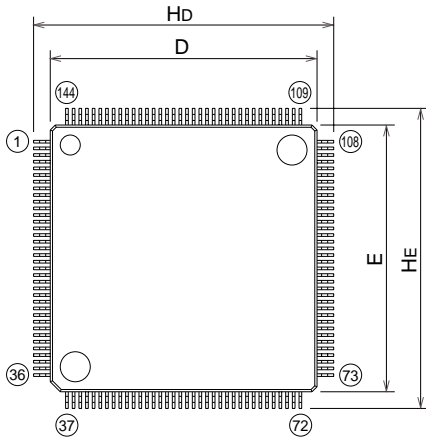
Package

144PFB-A

(MMP)

Plastic 144pin 16 16mm body TQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
TQFP144-P-1616-0.40	-	0.62	Cu Alloy



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	1.2
A1	0.05	0.1	0.15
A2	-	1.0	-
b	0.13	0.18	0.23
c	0.105	0.125	0.175
D	15.9	16.0	16.1
E	15.9	16.0	16.1
e	-	0.4	-
Hd	17.8	18.0	18.2
HE	17.8	18.0	18.2
L	0.4	0.5	0.6
L1	-	1.0	-
Lp	0.45	0.6	0.75
A3	-	0.25	-
x	-	-	0.07
y	-	-	0.08
θ	0	-	8
b2	-	0.225	-
l2	1.0	-	-
MD	-	16.4	-
ME	-	16.4	-

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
 2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
 3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors.
Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
 4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
 5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
 6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
 7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
 8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.
-



RENESAS SALES OFFICES

<http://www.renesas.com>

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500 Fax: <1> (408) 382-7501

Renesas Technology Europe Limited.

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH

Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.

7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.

FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.

26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.

1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

