

**DESCRIPTION**

The 38K0 group is the 8-bit microcomputer based on the 740 family core technology. The 38K0 group has the USB function, an 8-bit bus interface, a Serial I/O, three 8-bit timers, and an 8-channel 10-bit A-D converter, which are available for the PC peripheral I/O device. The various microcomputers in the 38K0 group include variations of internal memory size and packaging. For details, refer to the section on part numbering.

**FEATURES**

- Basic machine-language instructions ..... 71
- The minimum instruction execution time ..... 0.25 μs  
(at 8 MHz system clock\*)  
System clock\*: Reference frequency to internal circuit except USB function
- Memory size  
ROM ..... 16 K to 32 K bytes  
RAM ..... 1024 to 2048 bytes
- Programmable input/output ports ..... 48
- Software pull-up resistors
- Interrupts ..... 15 sources, 15 vectors
- USB function (USB version 1.1 specification) ..... 4 endpoints
- External bus interface ..... 8-bit X 1 channel
- Timers ..... 8-bit X 3
- Watchdog timer ..... 16-bit X 1
- Serial I/O ..... 8-bit X 1 (UART or Clock-synchronized)
- A-D converter ..... 10-bit X 8 channels  
(8-bit reading available)

- LED direct drive port ..... 4
- Clock generating circuit  
(connect to external ceramic resonator or quartz-crystal oscillator)
- Power source voltage  
System clock/Internal clock division mode  
At 12 MHz/Through mode (φ = 12 MHz) .....  
..... 4.50 to 5.25 V (under planning)  
At 12 MHz/2-divide mode(φ = 6 MHz) .....  
..... 4.00 to 5.25 V (under development)  
At 8 MHz/Through mode (φ = 8 MHz) ..... 4.00 to 5.25 V  
At 6 MHz/Through mode (φ = 6 MHz) ..... 4.00 to 5.25 V  
At 6 MHz/Through mode (φ = 6 MHz) .....  
..... 3.00 to 4.00 V (under development)

**Remarks:** The mode under development will be available from Aug./2002.

- Power dissipation  
At 5 V power source voltage ..... 125 mW (typ.)  
(at 8 MHz system clock, in through mode)  
At 3.3 V power source voltage ..... 45 mW (typ.)  
(at 6 MHz system clock, in through mode)
- Operating temperature range ..... -20 to 85°C
- Packages  
FP ..... 64P6U-A (64-pin 14 X 14 mm LQFP)  
HP ..... 64P6Q-A (64-pin 10 X 10 mm LQFP)

**Notes**

1. The specifications of this product are subject to change because it is under development. Inquire the use of Mitsubishi Electric Corporation.
2. The flash memory version cannot be used for application embedded in the MCU card.

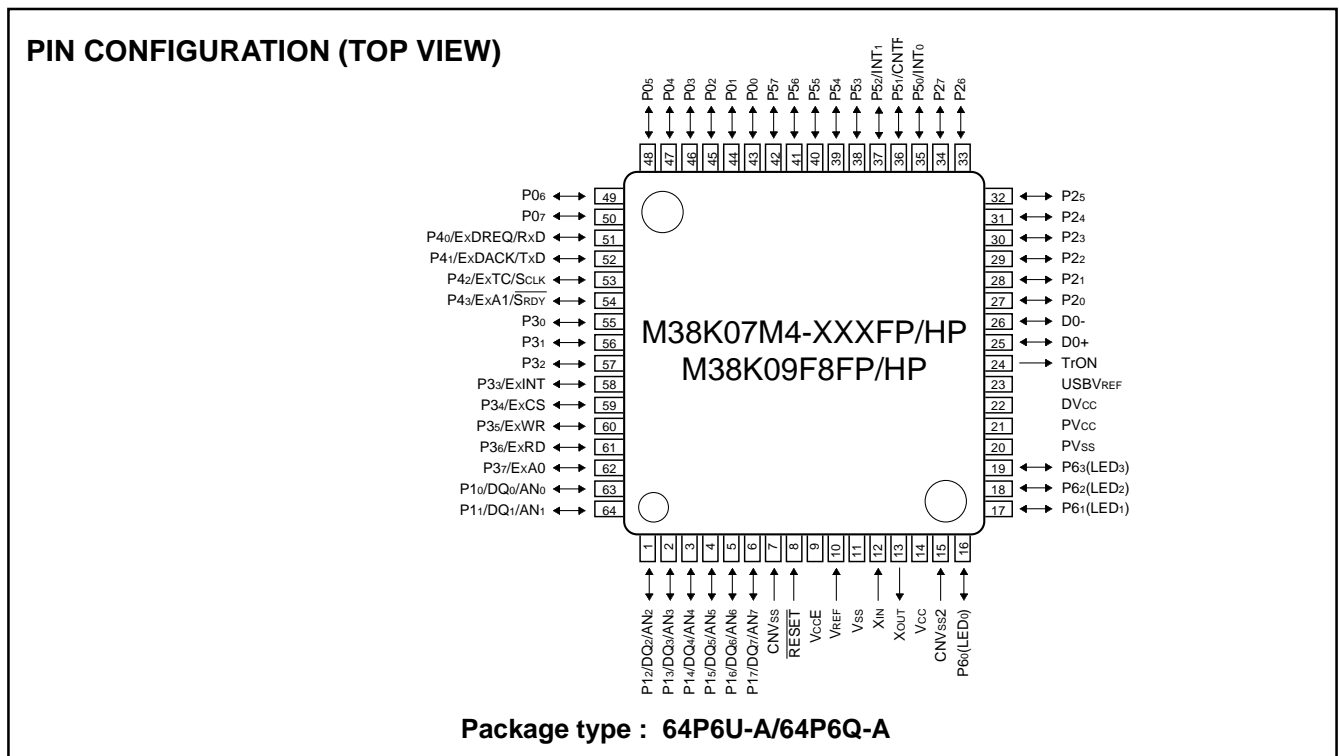


Fig. 1 Pin configuration of 38K0 group

FUNCTIONAL BLOCK DIAGRAM (Package : 64P6U-A/64P6Q-A)

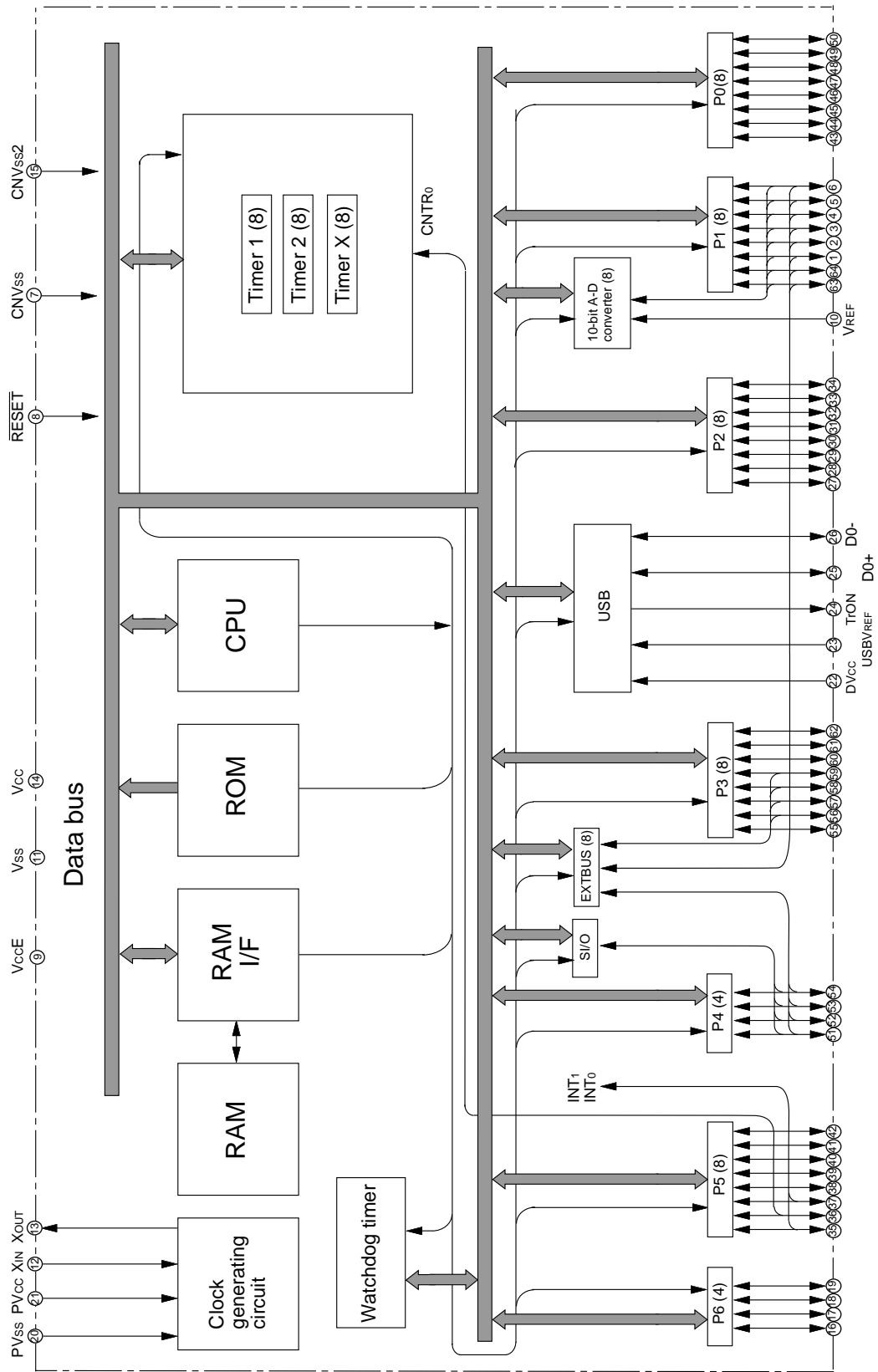


Fig. 2 Functional block diagram

## PIN DESCRIPTION

Table 1. Pin description

Pin	Name	Function	
			Function except a port function
Vcc, Vss	Power source	• Apply voltage of 3.0 V – 5.25 V to Vcc, and 0 V to Vss.	
VccE	Analog power source	• Power source pin for ports P1, P3, P4 and analog circuit. Connect this pin to Vcc.	
CNVss	CNVss	• This pin controls the operation mode of the chip. Connect this pin to Vss. In the flash memory mode, this pin becoems VPP power source input pin.	
CNVss2	CNVss2	• This pin controls the operation mode of the chip. Connect this pin to Vss.	
VREF	Analog reference voltage input	• Reference voltage input pin for A-D converter.	
DVcc PVcc, PVss	Analog power source	• Power source pin for analog circuit. • Connect the DVcc and PVcc pins to Vcc, and the PVss pin to Vss.	
RESET	Reset input	• Reset input pin for active “L”	
XIN	Clock input	• Input and output pins for the main clock generating circuit. • Connect a ceramic resonator or a quartz-crystal oscillator between the XIN and XOUT pins to set the oscillation frequency. • If an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.	
XOUT	Clock output		
USBVREF	USB reference power source	• Power source pin for USB port circuit. In Vcc = 4.00 to 5.25 V use the built-in USB reference voltage circuit. In Vcc = 3.00 to 4.00 V apply 3.3 V power supply from the external because use of the built-in USB reference voltage circuit is prohibited in this voltage range. In Vcc = 3.00 to 3.60 V connect this pin to Vcc.	
TrON	USB reference voltage output	• Output pin to pull-up D0+ by 1.5 kΩ external resistor.	
D0+, D0-	USB upstream I/O	• USB upstream I/O port • USB input level • USB output level output structure	
P00–P07	I/O port P0	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure • Pull-up control is enabled.	• Key input pins (key-on wake up interrupt)
P10/DQ0/AN0– P17/DQ7/AN7	I/O port P1	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	• A-D converter input pins • External bus interface function pins
P20–P27	I/O port P2	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	
P30–P32	I/O port P3	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	
P33/ExINT P34/ExCS P35/ExWR P36/ExRD P37/ExA0			• External bus interface function pins
P40/ExDREQ/RxD P41/ExDACK/TxD P42/ExTC/SCLK P43/ExA1/SRDY	I/O port P4	• 4-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	• Serial I/O function pins • External bus interface function pins
P50/INT0 P51/CNTR0 P52/INT1 P53–P57	I/O port P5	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	• Interrupt input pin
			• Timer X fuction pin
			• Interrupt input pin
P60–P63	I/O port P6	• 4-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure • Output large current for LED drive is enabled.	

**PART NUMBERING**

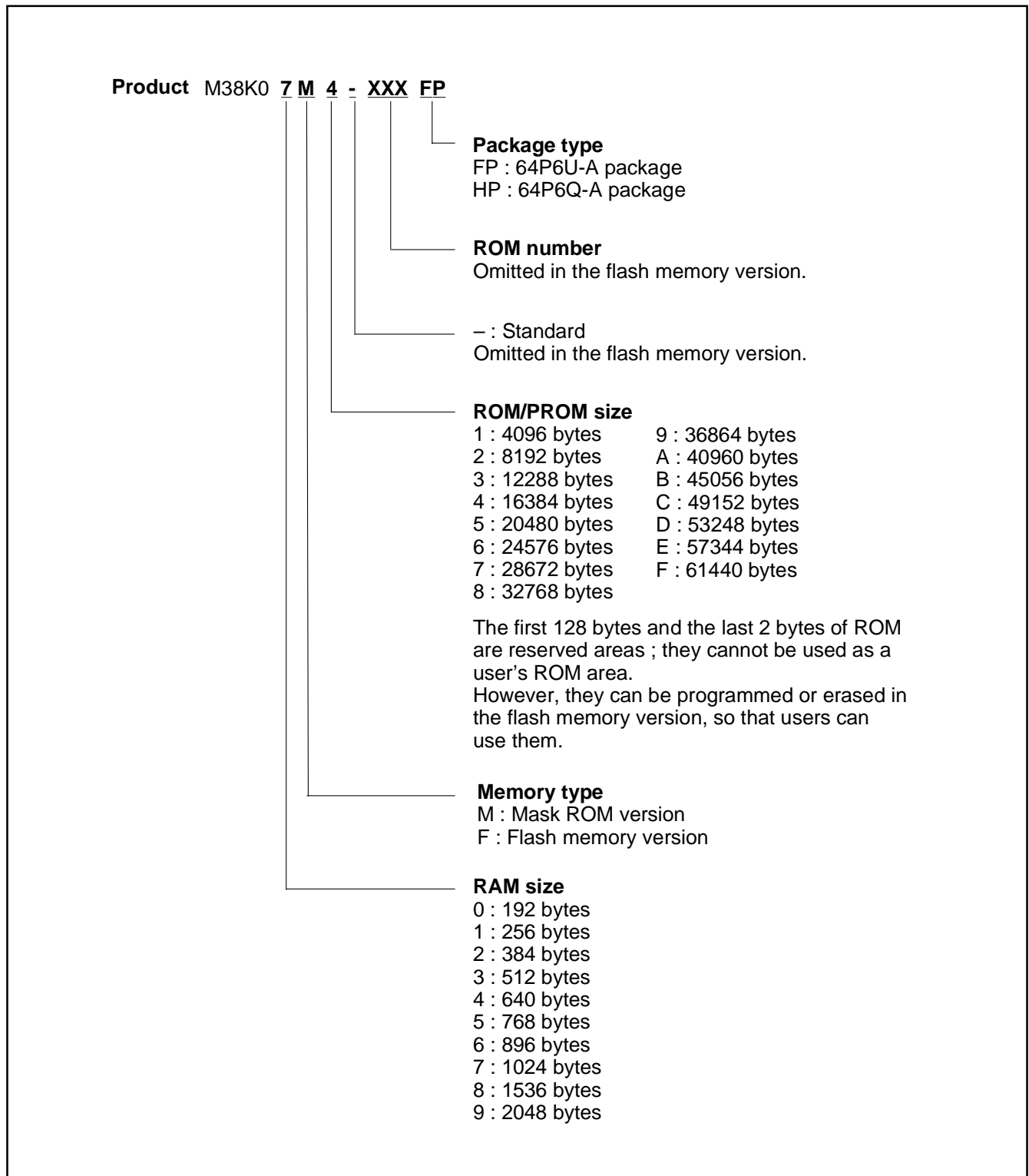


Fig. 3 Part numbering

**GROUP EXPANSION**

Mitsubishi plans to expand the 38K0 group as follows.

**Memory Type**

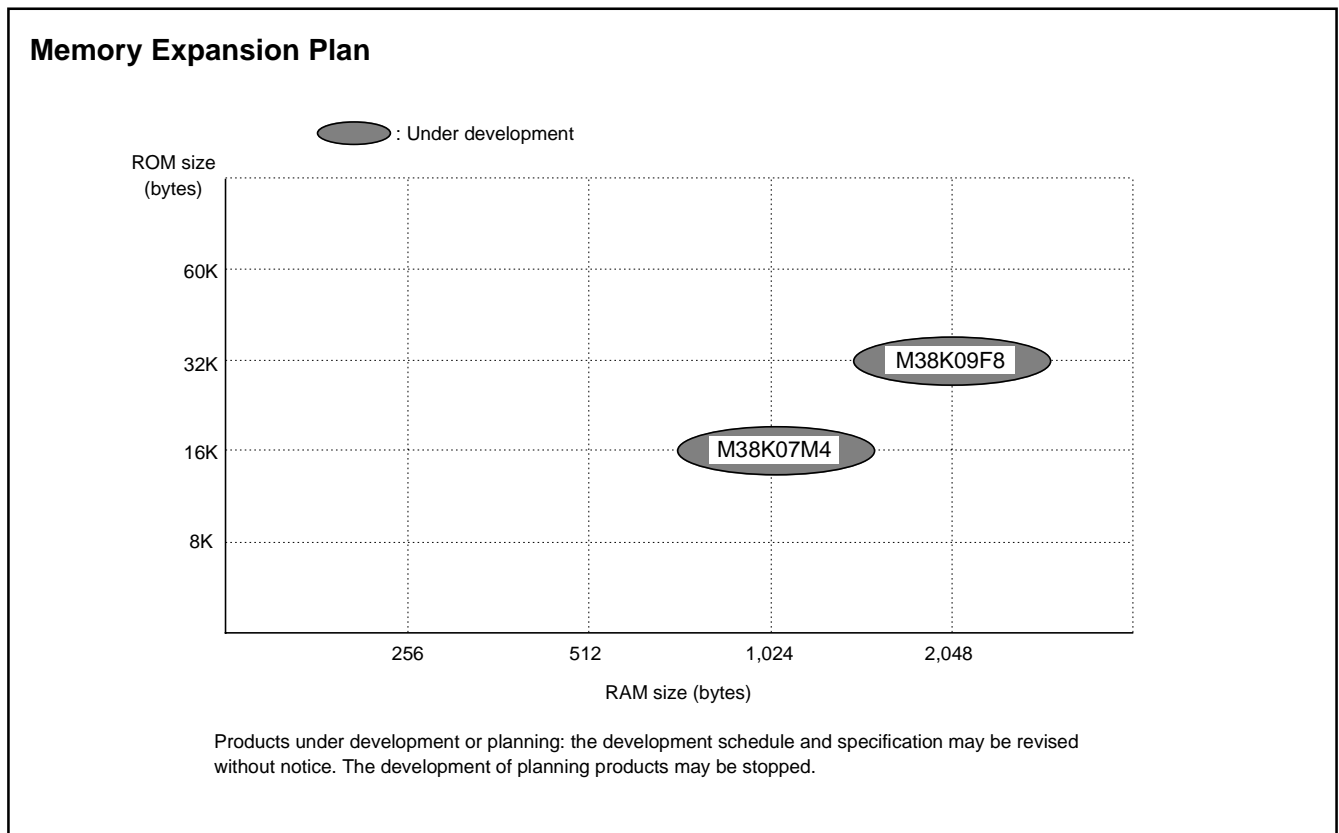
Support for mask ROM and flash memory versions.

**Memory Size**

Flash memory size ..... 32 Kbytes  
 Mask ROM size ..... 16 Kbytes  
 RAM size ..... 1024 to 2048 bytes

**Packages**

64P6U-A ..... 0.8 mm-pitch plastic molded LQFP  
 64P6Q-A ..... 0.5 mm-pitch plastic molded LQFP  
 100D0M ..... 0.65 mm-pitch metal seal PIGGY BACK



**Fig. 4 Memory expansion plan**

Currently products are listed below.

**Table 2. List of products**

**As of February 2002**

Product	ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38K07M4-XXXXFP	16384 (16254)	1024	64P6U-A	Mask ROM version
M38K07M4-XXXHP			64P6Q-A	
M38K09F8FP	32768 (32638)	2048	64P6U-A	Flash memory version
M38K09F8HP			64P6Q-A	
M38K09RFS	—	2048	100D0M	Emulator MCU (for program evaluation)

**FUNCTIONAL DESCRIPTION  
 CENTRAL PROCESSING UNIT (CPU)**

The 38K0 group uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine instructions or the 740 Family Software Manual for details on the instruction set.

Machine-resident 740 family instructions are as follows:

The FST and SLW instruction cannot be used.

The STP, WIT, MUL, and DIV instruction can be used.

The CPU has the 6 registers. The register structure is shown in Figure 5.

**[Accumulator (A)]**

The accumulator is an 8-bit register. Data operations such as data transfer, etc., are executed mainly through the accumulator.

**[Index Register X (X)]**

The index register X is an 8-bit register. In the index addressing modes, the value of the OPERAND is added to the contents of register X and specifies the real address.

**[Index Register Y (Y)]**

The index register Y is an 8-bit register. In partial instruction, the value of the OPERAND is added to the contents of register Y and specifies the real address.

**[Stack Pointer (S)]**

The stack pointer is an 8-bit register used during subroutine calls and interrupts. This register indicates start address of stored area (stack) for storing registers during subroutine calls and interrupts. The low-order 8 bits of the stack address are determined by the contents of the stack pointer. The high-order 8 bits of the stack address are determined by the stack page selection bit. If the stack page selection bit is "0", the high-order 8 bits becomes "0016". If the stack page selection bit is "1", the high-order 8 bits becomes "0116".

Figure 6 shows the store and the return movement into the stack. If there are registers other than those described in Figure 5, the users need to store them with the program.

**[Program Counter (PC)]**

The program counter is a 16-bit counter consisting of two 8-bit registers PCH and PCL. It is used to indicate the address of the next instruction to be executed.

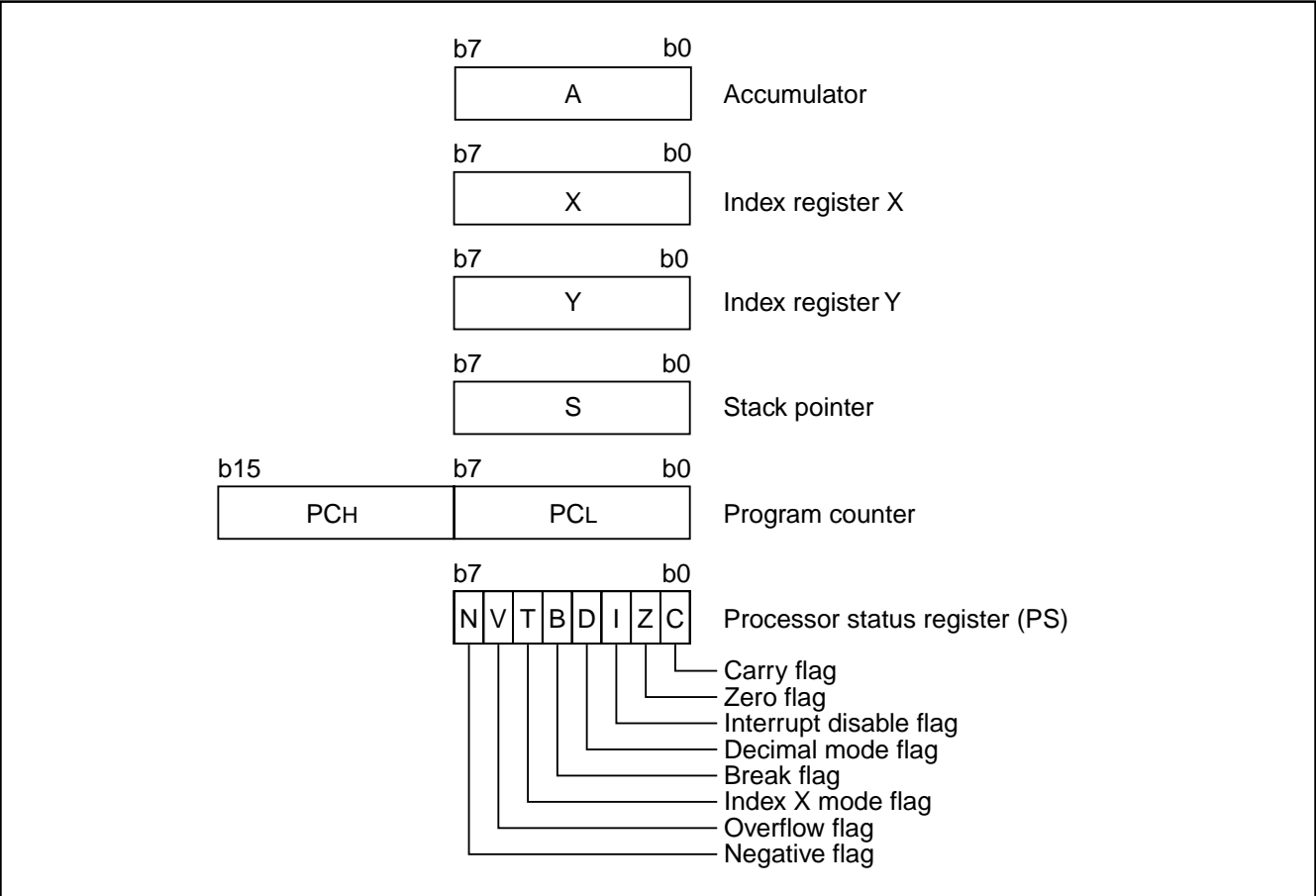


Fig. 5 740 Family CPU register structure

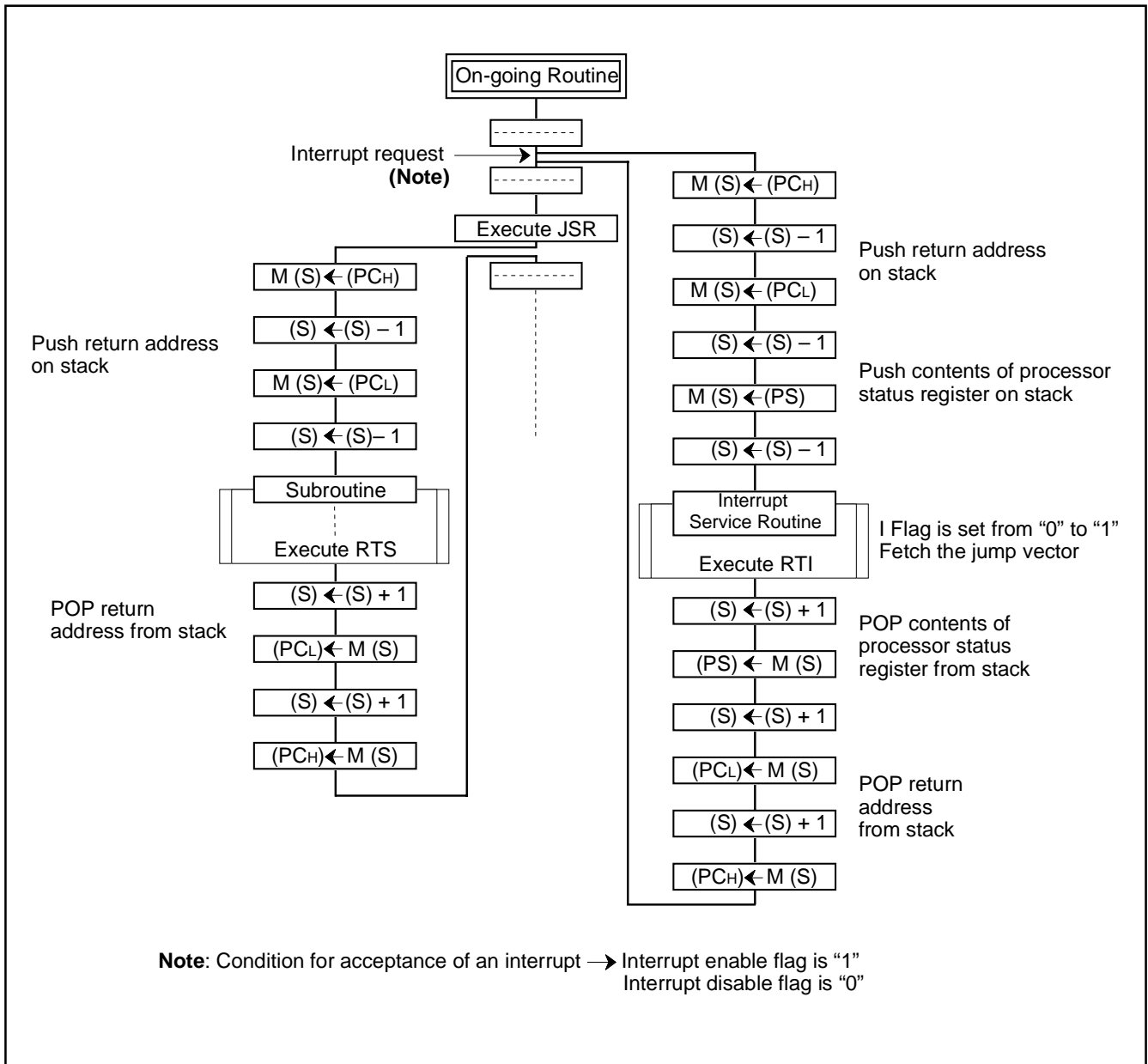


Fig. 6 Register push and pop at interrupt generation and subroutine call

Table 3 Push and pop instructions of accumulator or processor status register

	Push instruction to stack	Pop instruction from stack
Accumulator	PHA	PLA
Processor status register	PHP	PLP

### [Processor status register (PS)]

The processor status register is an 8-bit register consisting of 5 flags which indicate the status of the processor after an arithmetic operation and 3 flags which decide MCU operation. Branch operations can be performed by testing the Carry (C) flag, Zero (Z) flag, Overflow (V) flag, or the Negative (N) flag. In decimal mode, the Z, V, N flags are not valid.

•Bit 0: Carry flag (C)

The C flag contains a carry or borrow generated by the arithmetic logic unit (ALU) immediately after an arithmetic operation. It can also be changed by a shift or rotate instruction.

•Bit 1: Zero flag (Z)

The Z flag is set if the result of an immediate arithmetic operation or a data transfer is "0", and cleared if the result is anything other than "0".

•Bit 2: Interrupt disable flag (I)

The I flag disables all interrupts except for the interrupt generated by the BRK instruction.

Interrupts are disabled when the I flag is "1".

•Bit 3: Decimal mode flag (D)

The D flag determines whether additions and subtractions are executed in binary or decimal. Binary arithmetic is executed when this flag is "0"; decimal arithmetic is executed when it is "1". Decimal correction is automatic in decimal mode. Only the ADC

•Bit 4: Break flag (B)

The B flag is used to indicate that the current interrupt was generated by the BRK instruction. The BRK flag in the processor status register is always "0". When the BRK instruction is used to generate an interrupt, the processor status register is pushed onto the stack with the break flag set to "1".

•Bit 5: Index X mode flag (T)

When the T flag is "0", arithmetic operations are performed between accumulator and memory. When the T flag is "1", direct arithmetic operations and direct data transfers are enabled between memory locations.

•Bit 6: Overflow flag (V)

The V flag is used during the addition or subtraction of one byte of signed data. It is set if the result exceeds +127 to -128. When the BIT instruction is executed, bit 6 of the memory location operated on by the BIT instruction is stored in the overflow flag.

•Bit 7: Negative flag (N)

The N flag is set if the result of an arithmetic operation or data transfer is negative. When the BIT instruction is executed, bit 7 of the memory location operated on by the BIT instruction is stored in the negative flag.

Table 4 Set and clear instructions of each bit of processor status register

	C flag	Z flag	I flag	D flag	B flag	T flag	V flag	N flag
Set instruction	SEC	–	SEI	SED	–	SET	–	–
Clear instruction	CLC	–	CLI	CLD	–	CLT	CLV	–



[CPU Mode Register (CPUM)] 003B16

The CPU mode register contains the stack page selection bit and the internal system clock selection bit.

The CPU mode register is allocated at address 003B16.

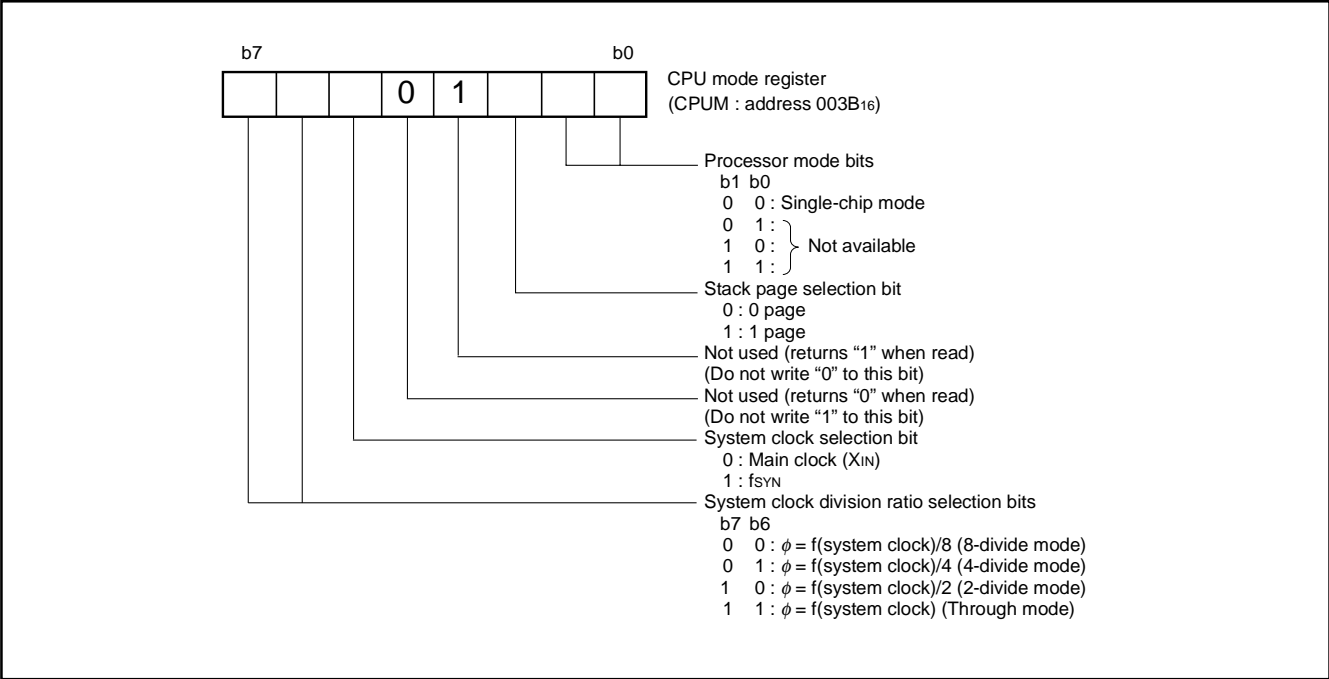


Fig. 7 Structure of CPU mode register

## MEMORY

### Special Function Register (SFR) Area

The Special Function Register area in the zero page contains control registers such as I/O ports and timers.

### RAM

RAM is used for data storage and for stack area of subroutine calls and interrupts.

### ROM

The first 128 bytes and the last 2 bytes of ROM are reserved for device testing and the rest is user area for storing programs. In the flash memory version, program and erase can be performed in the reserved area.

### Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

### Zero Page

The 256 bytes from addresses 0000<sub>16</sub> to 00FF<sub>16</sub> are called the zero page area. The internal RAM and the special function registers (SFR) are allocated to this area.

The zero page addressing mode can be used to specify memory and register addresses in the zero page area. Access to this area with only 2 bytes is possible in the zero page addressing mode.

### Special Page

The 256 bytes from addresses FF00<sub>16</sub> to FFFF<sub>16</sub> are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

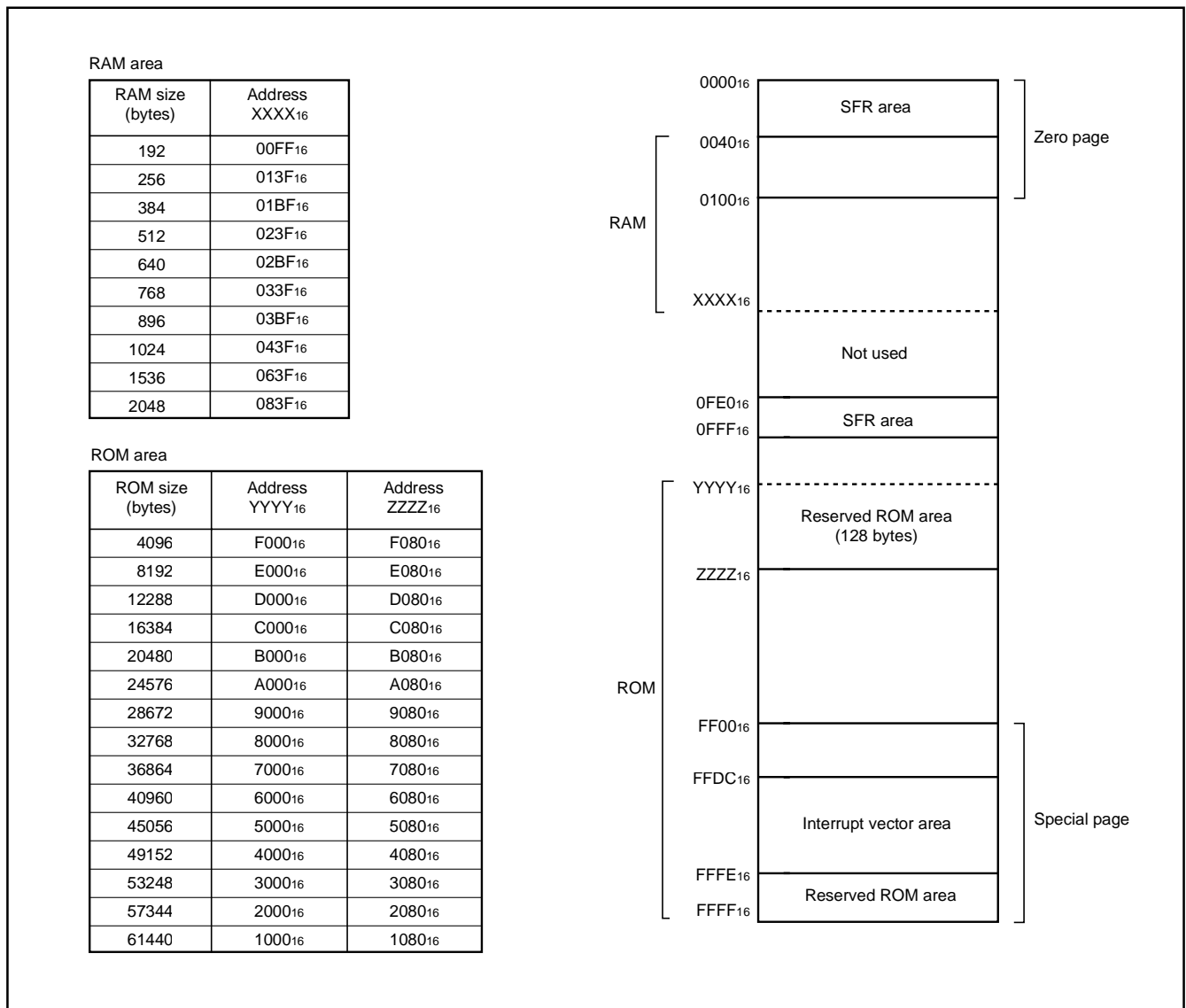


Fig. 8 Memory map diagram

0000 <sup>16</sup>	Port P0 (P0)	0020 <sup>16</sup>	Prescaler 12 (PRE12)
0001 <sup>16</sup>	Port P0 direction register (P0D)	0021 <sup>16</sup>	Timer 1 (T1)
0002 <sup>16</sup>	Port P1 (P1)	0022 <sup>16</sup>	Timer 2 (T2)
0003 <sup>16</sup>	Port P1 direction register (P1D)	0023 <sup>16</sup>	Timer X mode register (TM)
0004 <sup>16</sup>	Port P2 (P2)	0024 <sup>16</sup>	Prescaler X (PREX)
0005 <sup>16</sup>	Port P2 direction register (P2D)	0025 <sup>16</sup>	Timer X (TX)
0006 <sup>16</sup>	Port P3 (P3)	0026 <sup>16</sup>	Transmit/Receive buffer register (TB/RB)
0007 <sup>16</sup>	Port P3 direction register (P3D)	0027 <sup>16</sup>	Serial I/O status register (SIOSTS)
0008 <sup>16</sup>	Port P4 (P4)	0028 <sup>16</sup>	Reserved ( <b>Note</b> )
0009 <sup>16</sup>	Port P4 direction register (P4D)	0029 <sup>16</sup>	Reserved ( <b>Note</b> )
000A <sup>16</sup>	Port P5 (P5)	002A <sup>16</sup>	Reserved ( <b>Note</b> )
000B <sup>16</sup>	Port P5 direction register (P5D)	002B <sup>16</sup>	Reserved ( <b>Note</b> )
000C <sup>16</sup>	Port P6 (P6)	002C <sup>16</sup>	Reserved ( <b>Note</b> )
000D <sup>16</sup>	Port P6 direction register (P6D)	002D <sup>16</sup>	Reserved ( <b>Note</b> )
000E <sup>16</sup>	Reserved ( <b>Note</b> )	002E <sup>16</sup>	Reserved ( <b>Note</b> )
000F <sup>16</sup>	Reserved ( <b>Note</b> )	002F <sup>16</sup>	Reserved ( <b>Note</b> )
0010 <sup>16</sup>	USB control register (USBCON)	0030 <sup>16</sup>	EXB interrupt source enable register (EXBICON)
0011 <sup>16</sup>	USB address enable register (USBAE)	0031 <sup>16</sup>	EXB interrupt source register (EXBIREQ)
0012 <sup>16</sup>	USB address 0 register (USBA0)	0032 <sup>16</sup>	Reserved ( <b>Note</b> )
0013 <sup>16</sup>	USB address 1 register (USBA1)	0033 <sup>16</sup>	EXB index register (EXBINDEX)
0014 <sup>16</sup>	Frame number register Low (FNUML)	0034 <sup>16</sup>	EXB field register 1 (EXBREG1)
0015 <sup>16</sup>	Frame number register High (FNUMH)	0035 <sup>16</sup>	EXB field register 2 (EXBREG2)
0016 <sup>16</sup>	USB interrupt source enable register (USBICON)	0036 <sup>16</sup>	A-D control register (ADCON)
0017 <sup>16</sup>	USB interrupt source register (USBIREQ)	0037 <sup>16</sup>	A-D conversion register Low (ADL)
0018 <sup>16</sup>	Endpoint index register (USBINDEX)	0038 <sup>16</sup>	A-D conversion register High (ADH)
0019 <sup>16</sup>	Endpoint field register 1 (EPXXREG1)	0039 <sup>16</sup>	Watchdog timer control register (WDTCN)
001A <sup>16</sup>	Endpoint field register 2 (EPXXREG2)	003A <sup>16</sup>	Reserved ( <b>Note</b> )
001B <sup>16</sup>	Endpoint field register 3 (EPXXREG3)	003B <sup>16</sup>	CPU mode register (CPUM)
001C <sup>16</sup>	Endpoint field register 4 (EPXXREG4)	003C <sup>16</sup>	Interrupt request register 1(IREQ1)
001D <sup>16</sup>	Endpoint field register 5 (EPXXREG5)	003D <sup>16</sup>	Interrupt request register 2(IREQ2)
001E <sup>16</sup>	Endpoint field register 6 (EPXXREG6)	003E <sup>16</sup>	Interrupt control register 1(ICON1)
001F <sup>16</sup>	Endpoint field register 7 (EPXXREG7)	003F <sup>16</sup>	Interrupt control register 2(ICON2)
0FE0 <sup>16</sup>	Serial I/O control register (SIOCON)	0FF0 <sup>16</sup>	Port P0 pull-up control register (PULL0)
0FE1 <sup>16</sup>	UART control register (UARTCON)	0FF1 <sup>16</sup>	Reserved ( <b>Note</b> )
0FE2 <sup>16</sup>	Baud rate generator (BRG)	0FF2 <sup>16</sup>	Port P5 pull-up control register (PULL5)
0FE3 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF3 <sup>16</sup>	Interrupt edge selection register (INTEDGE)
0FE4 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF4 <sup>16</sup>	Reserved ( <b>Note</b> )
0FE5 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF5 <sup>16</sup>	Reserved ( <b>Note</b> )
0FE6 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF6 <sup>16</sup>	Reserved ( <b>Note</b> )
0FE7 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF7 <sup>16</sup>	Reserved ( <b>Note</b> )
0FE8 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF8 <sup>16</sup>	PLL control register (PLLCON)
0FE9 <sup>16</sup>	Reserved ( <b>Note</b> )	0FF9 <sup>16</sup>	Reserved ( <b>Note</b> )
0FEA <sup>16</sup>	Reserved ( <b>Note</b> )	0FFA <sup>16</sup>	Reserved ( <b>Note</b> )
0FEB <sup>16</sup>	Reserved ( <b>Note</b> )	0FFB <sup>16</sup>	MISRG
0FEC <sup>16</sup>	Endpoint field register 8 (EPXXREG8)	0FFC <sup>16</sup>	Reserved ( <b>Note</b> )
0FED <sup>16</sup>	Endpoint field register 9 (EPXXREG9)	0FFD <sup>16</sup>	Reserved ( <b>Note</b> )
0FEE <sup>16</sup>	Reserved ( <b>Note</b> )	0FFE <sup>16</sup>	Flash memory control register (FMCR)
0FEF <sup>16</sup>	Reserved ( <b>Note</b> )	0FFF <sup>16</sup>	Reserved ( <b>Note</b> )

**Note:** Do not write any data to these addresses, because these areas are reserved.

Fig. 9 Memory map of special function register (SFR)

**I/O PORTS**

The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, and each pin can be set to be input port or output port.

When "0" is written to the bit corresponding to a pin, that pin becomes an input pin. When "1" is written to that bit, that pin becomes an output pin.

If data is read from a pin set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.

**Table 5 I/O ports functions**

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Diagram No.	
P00-P07	Port P0	Input/output, individual bits	CMOS compatible input level CMOS 3-state output	Key-on wake up	Port P0 pull-up control register	(1)	
P10-P17	Port P1			A-D conversion input External bus interface function I/O	A-D control register EXB control register	(2)	
P20-P27	Port P2			—————	—————	(3)	
P30-P32	Port P3			—————	—————	(4)	
P33/ExINT				External bus interface function output	EXB control register	(5)	
P34/ExCS P35/ExWR P36/ExRD P37/ExA0				External bus interface function input	EXB control register	(6)	
P40/RxD/ ExDREQ				Serial I/O input External bus interface function output	Serial I/O control register EXB control register	(7)	
P41/TxD/ ExDACK	Serial I/O output External bus interface function input			Serial I/O control register EXB control register	(8)		
P42/SCLK/ ExTC	Serial I/O I/O External bus interface function input			Serial I/O control register EXB control register	(9)		
P43/SRDY/ ExA1	Serial I/O output External bus interface function input			Serial I/O control register EXB control register	(10)		
P50/INT0 P52/INT1	Port P5			CMOS compatible input level CMOS 3-state output	External interrupt input	Port P5 pull-up control register Interrupt edge selection register	(11)
P51/CNTR0					Timer X function I/O	Timer X mode register	(12)
P53-P57					—————	—————	(13)
P60-P63	Port P6			—————	—————	(14)	

**Note:** Make sure that the input level at each pin is either 0 V or Vcc during execution of the STP instruction. When an input level is at an intermediate potential, a current will flow from Vcc to Vss through the input-stage gate.

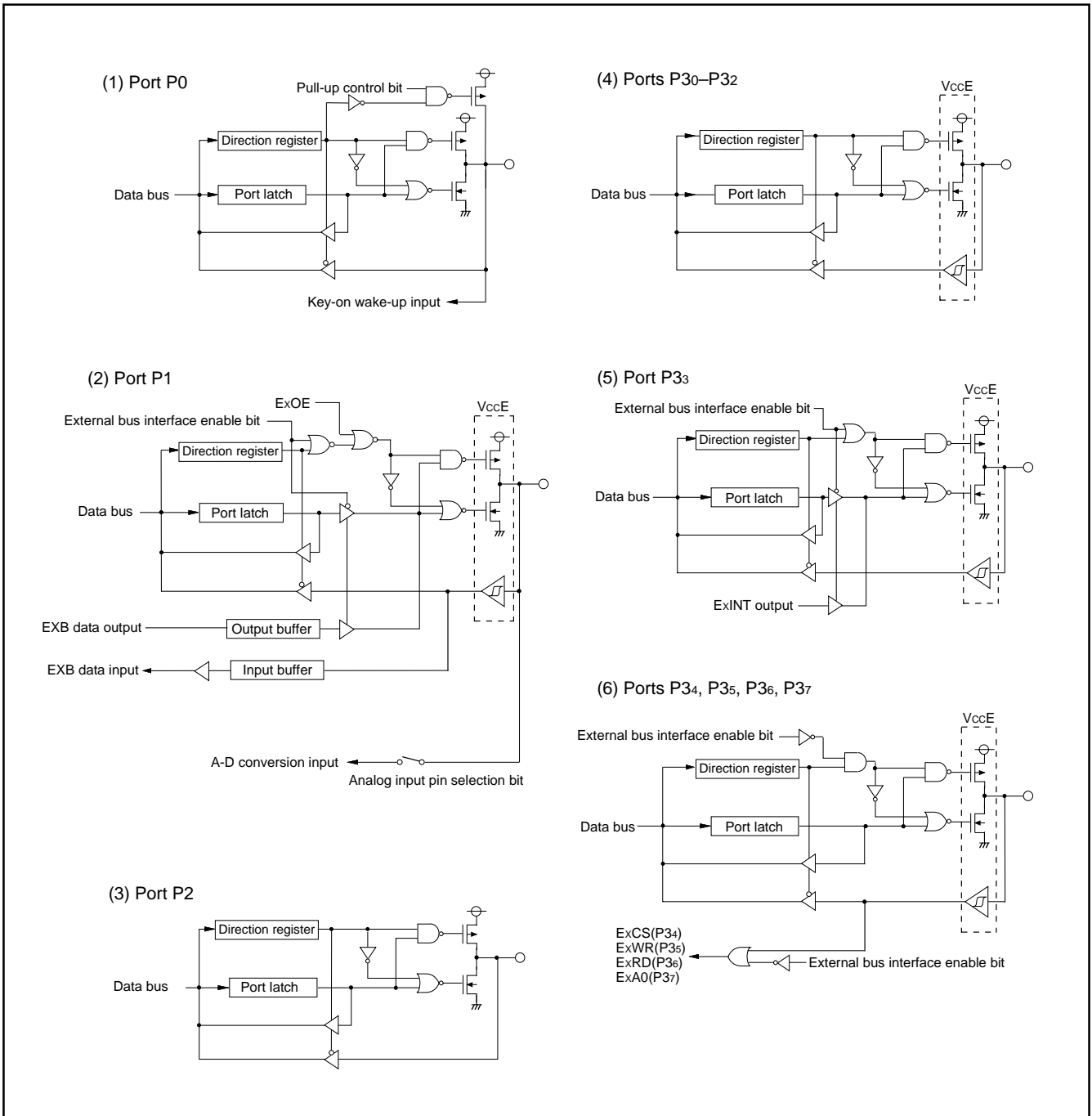


Fig. 10 Port block diagram (1)

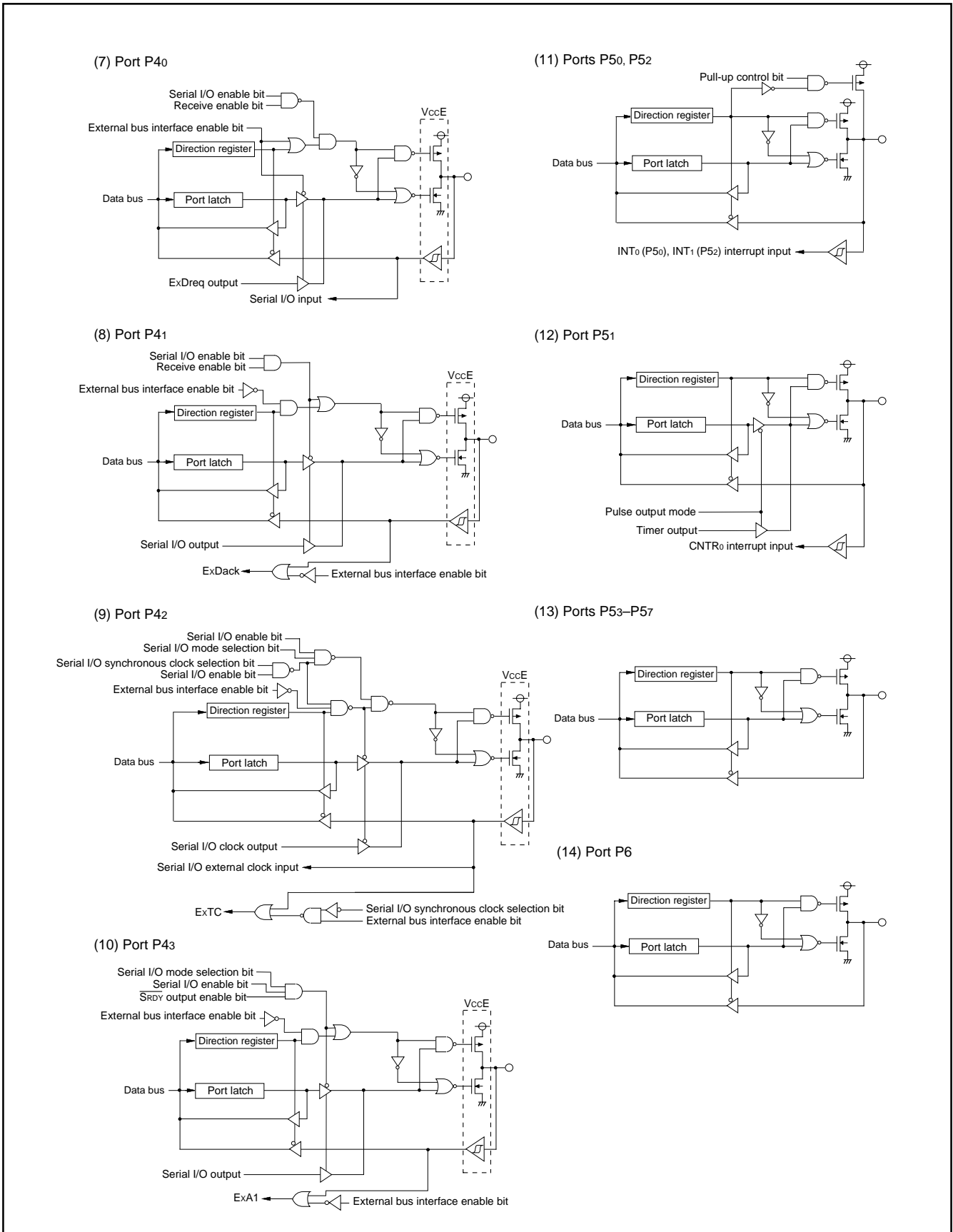


Fig. 11 Port block diagram (2)

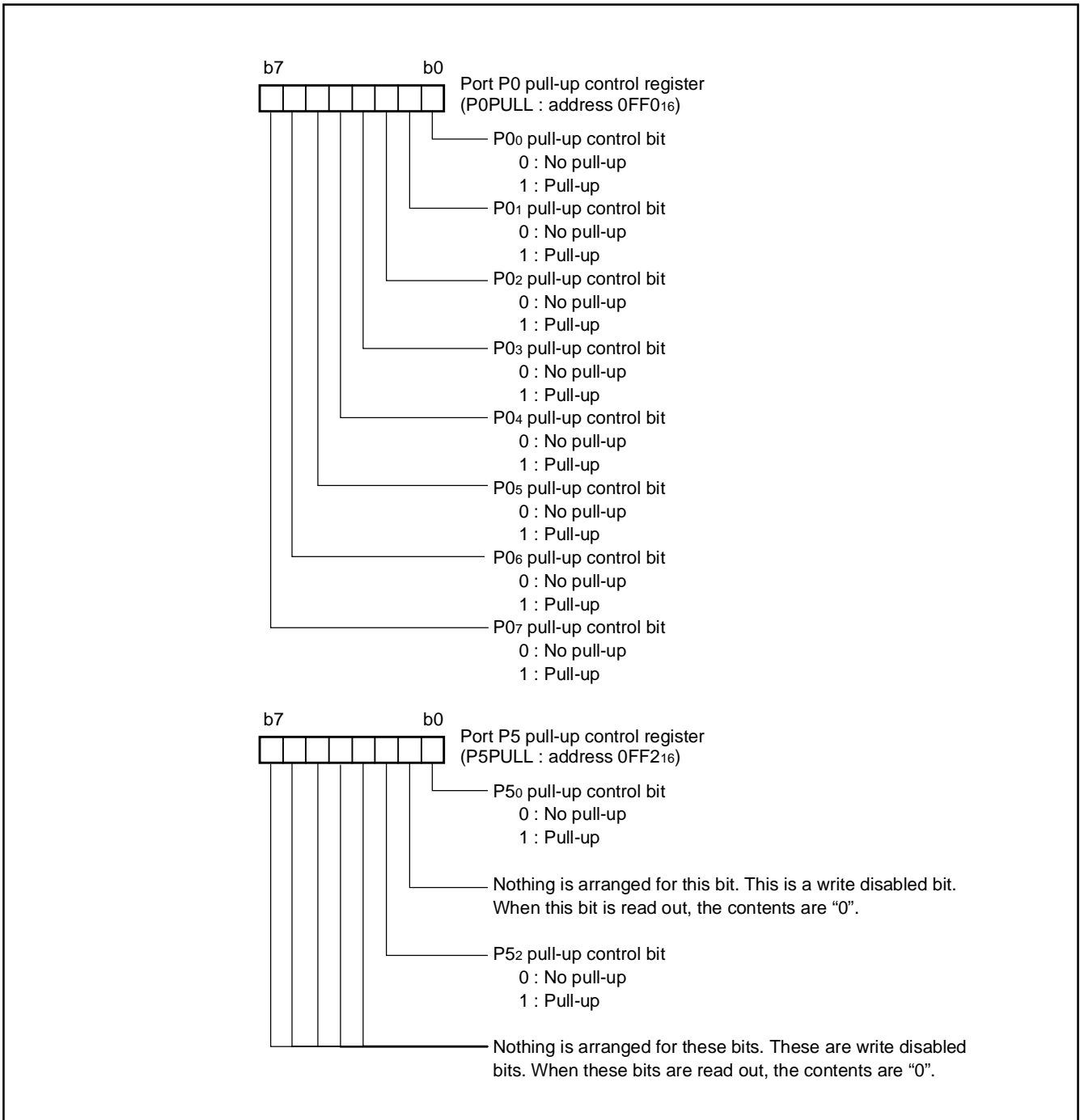


Fig. 12 Structure of port I/O-related registers

## INTERRUPTS

Interrupts occur by fifteen sources: four external, ten internal, and one software.

### Interrupt Control

Each interrupt is controlled by an interrupt request bit, an interrupt enable bit, and the interrupt disable flag except for the software interrupt set by the BRK instruction. An interrupt occurs if the corresponding interrupt request and enable bits are "1" and the interrupt disable flag is "0".

Interrupt enable bits can be set or cleared by software.

Interrupt request bits can be cleared by software, but cannot be set by software.

The BRK instruction cannot be disabled with any flag or bit. The I flag disables all interrupts except the BRK instruction interrupt.

When several interrupts occur at the same time, the interrupts are received according to priority.

### Interrupt Operation

By acceptance of an interrupt, the following operations are automatically performed:

1. The contents of the program counter and the processor status register are automatically pushed onto the stack.
2. The interrupt disable flag is set and the corresponding interrupt request bit is cleared.
3. The interrupt jump destination address is read from the vector table into the program counter.

### ■Notes on interrupts

When setting the followings, the interrupt request bit may be set to "1".

- When setting external interrupt active edge

Related register: Interrupt edge selection register (address 0FF316), Timer X mode register (address 002316)

When not requiring for the interrupt occurrence synchronized with these setting, take the following sequence.

- ①Set the corresponding interrupt enable bit to "0" (disabled).
- ②Set the interrupt edge select bit (active edge switch bit).
- ③Set the corresponding interrupt request bit to "0" after 1 or more instructions have been executed.
- ④Set the corresponding interrupt enable bit to "1" (enabled).

**Table 6 Interrupt vector addresses and priority**

Interrupt Source	Priority	Vector Addresses (Note 1)		Interrupt Request Generating Conditions
		High	Low	
Reset (Note 2)	1	FFFD16	FFFC16	At reset
USB bus reset	2	FFFB16	FFFA16	At detection of USB bus reset signal (2.5 μs interval SE0)
USB SOF	3	FFF916	FFF816	At detection of USB SOF signal
USB device	4	FFF716	FFF616	At detection of resume signal (K state or SE0) or suspend signal (3 ms interval bus idle), or at completion of transaction
External bus	5	FFF516	FFF416	At completion of reception or transmission or at completion of DMA transmission
INT0	6	FFF316	FFF216	At detection of either rising or falling edge of INT0 input
Timer X	7	FFF116	FFF016	At timer X underflow
Timer 1	8	FFEF16	FFEE16	At timer 1 underflow
Timer 2	9	FFED16	FFEC16	At timer 2 underflow
INT1	10	FFE916	FFE816	At detection of either rising or falling edge of INT1 input
<b>(Note 3)</b>	—	FFE916	FFE816	<b>(Note 4)</b>
Serial I/O reception	11	FFE716	FFE616	At completion of serial I/O data reception
Serial I/O transmission	12	FFE516	FFE416	At completion of serial I/O data transmission
CNTR0	13	FFE316	FFE216	At detection of either rising or falling edge of CNTR0 input
Key-on wake up	14	FFE116	FFE016	At falling of conjunction of input level for port P2 (at input mode)
A-D conversion	15	FFDF16	FFDE16	At completion of A-D conversion
BRK instruction	16	FFDD16	FFDC16	At BRK instruction execution

**Notes 1:** Vector addresses contain interrupt jump destination addresses.

**2:** Reset function in the same way as an interrupt with the highest priority.

**3:** Nothing is arranged in these vector addresses.

**4:** Fix bit 1 of interrupt control register 2 (address 003F16) to "0".



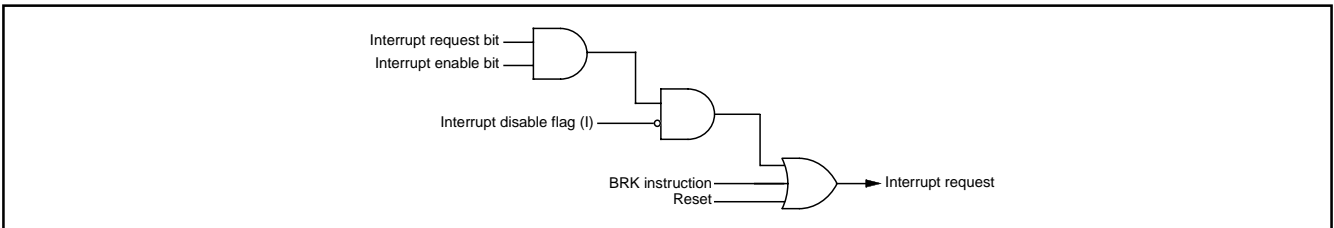


Fig. 13 Interrupt control

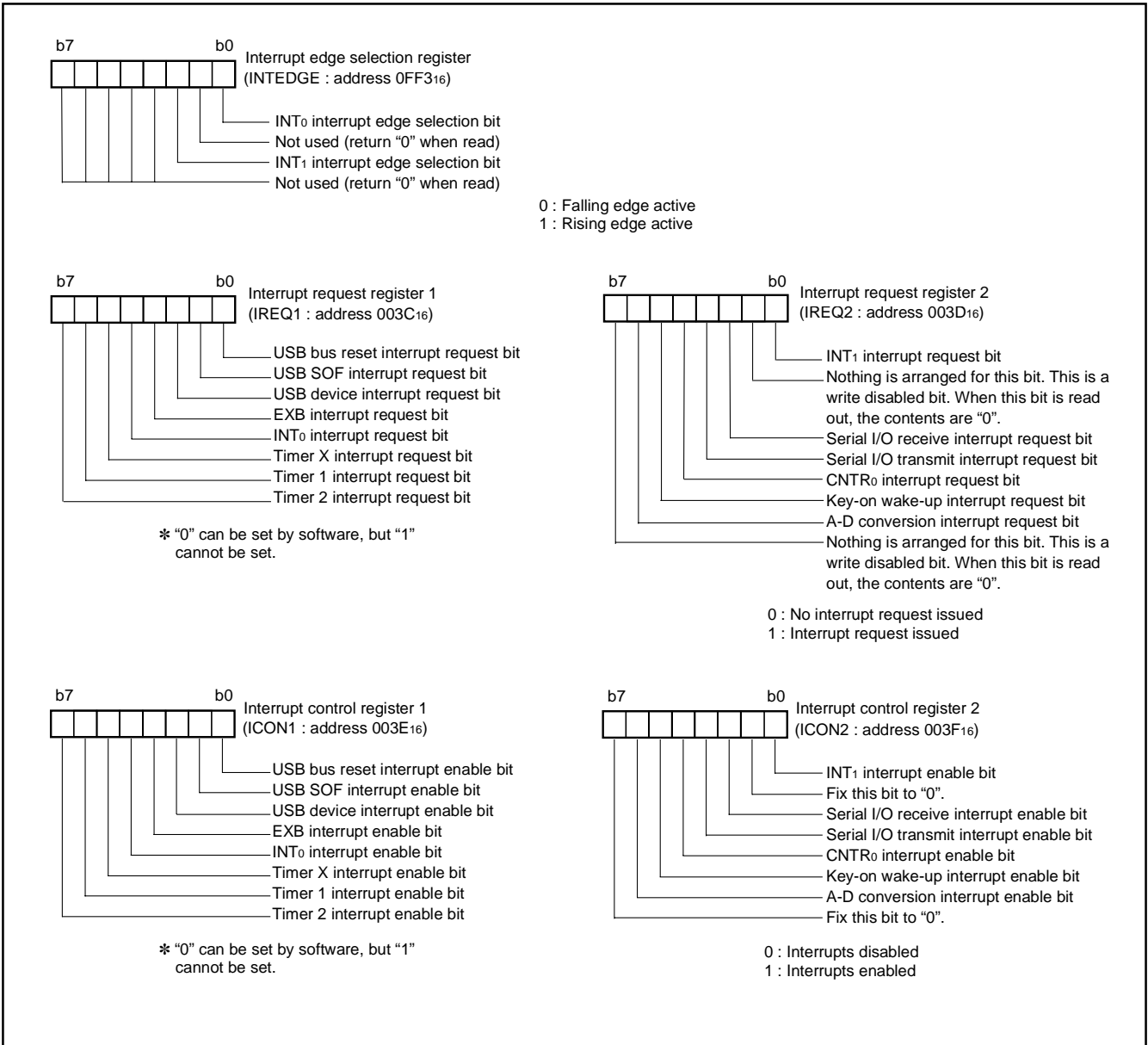


Fig. 14 Structure of interrupt-related registers

**Key Input Interrupt (Key-on Wake Up)**

A Key-on wake up interrupt request is generated by applying a falling edge to any pin of port P0 that have been set to input mode. In other words, it is generated when AND of input level goes from

"1" to "0". An example of using a key input interrupt is shown in Figure 15, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports P00-P03.

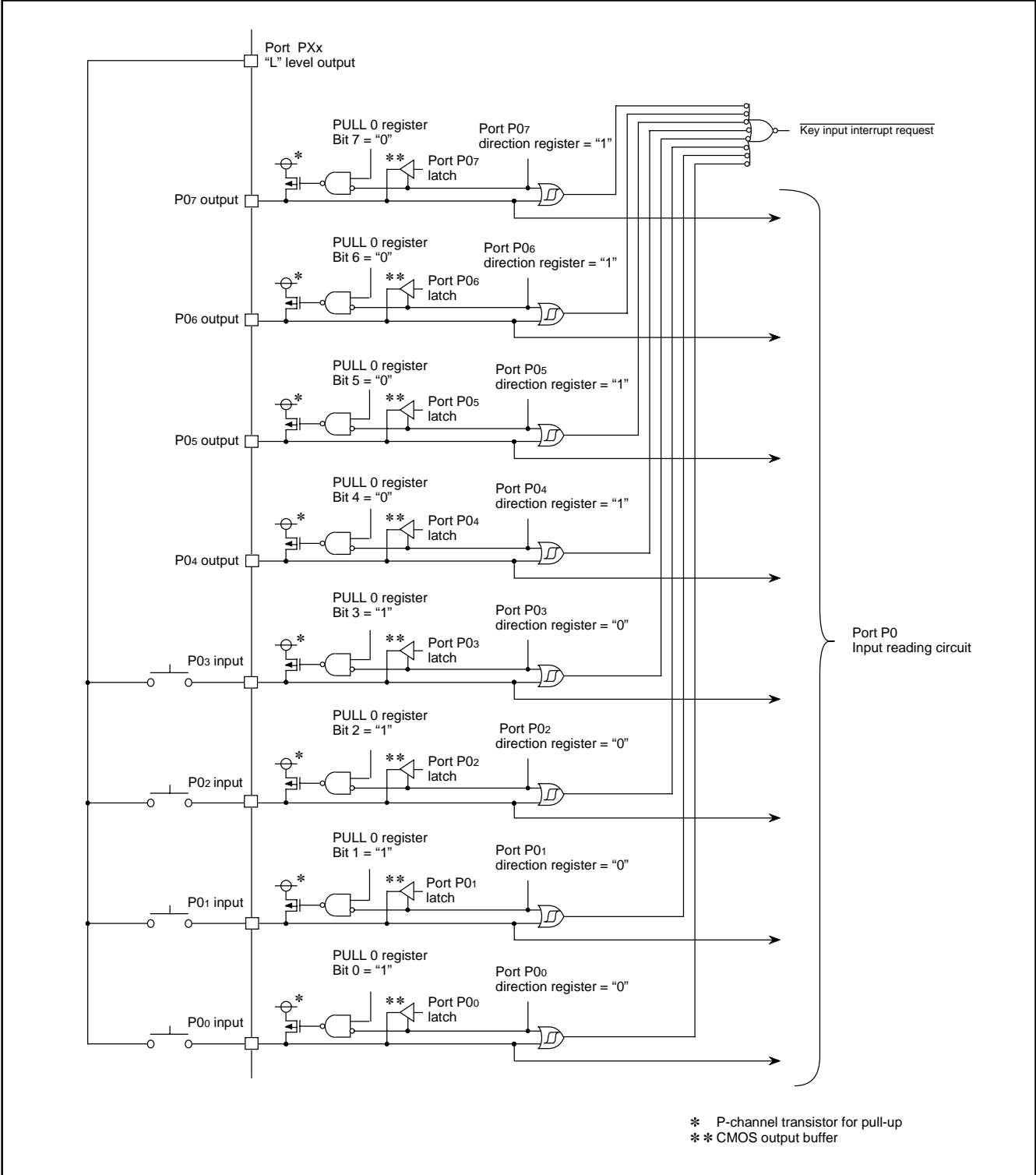


Fig. 15 Connection example when using key input interrupt and port P0 block diagram

## TIMERS

The 38K0 group has three timers: timer X, timer 1, and timer 2. The division ratio of each timer or prescaler is given by  $1/(n + 1)$ , where  $n$  is the value in the corresponding timer or prescaler latch. All timers are down count timers. When the timer reaches "0016", an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When a timer underflows, the interrupt request bit corresponding to that timer is set to "1".

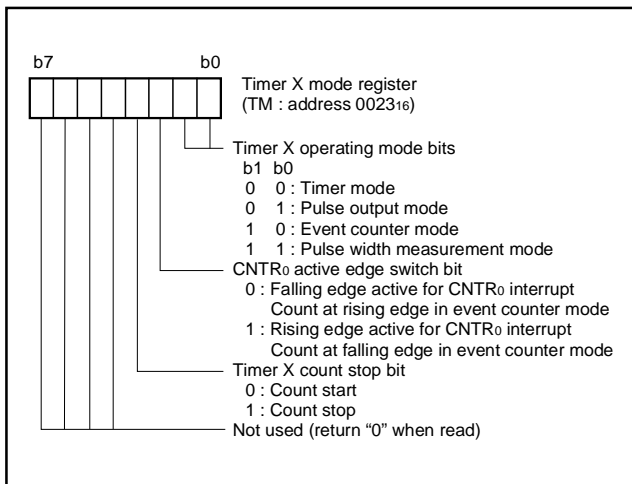


Fig. 16 Structure of timer X mode register

### Timer 1 and Timer 2

The count source of prescaler 12 is the system clock divided by 16. The output of prescaler 12 is counted by timer 1 and timer 2, and a timer underflow periodically sets the interrupt request bit.

### Timer X

Timer X can each select in one of four operating modes by setting the timer X mode register.

#### (1) Timer Mode

The timer counts the count source selected by timer count source selection bit.

#### (2) Pulse Output Mode

The timer counts the system clock divided by 16. Whenever the contents of the timer reach "0016", the signal output from the CNTR<sub>0</sub> pin is inverted. If the CNTR<sub>0</sub> active edge selection bit is "0", output begins at "H".

If it is "1", output starts at "L". When using a timer in this mode, set the corresponding port P51 direction register to output mode.

#### (3) Event Counter Mode

Operation in event counter mode is the same as in timer mode, except that the timer counts signals input through the CNTR<sub>0</sub> pin. When the CNTR<sub>0</sub> active edge selection bit is "0", the rising edge of the CNTR<sub>0</sub> pin is counted.

When the CNTR<sub>0</sub> active edge selection bit is "1", the falling edge of the CNTR<sub>0</sub> pin is counted.

#### (4) Pulse Width Measurement Mode

If the CNTR<sub>0</sub> active edge selection bit is "0", the timer counts the system clock divided by 16 while the CNTR<sub>0</sub> pin is at "H". If the CNTR<sub>0</sub> active edge selection bit is "1", the timer counts it while the CNTR<sub>0</sub> pin is at "L".

The count can be stopped by setting "1" to the timer X count stop bit in any mode. The corresponding interrupt request bit is set each time a timer underflows.

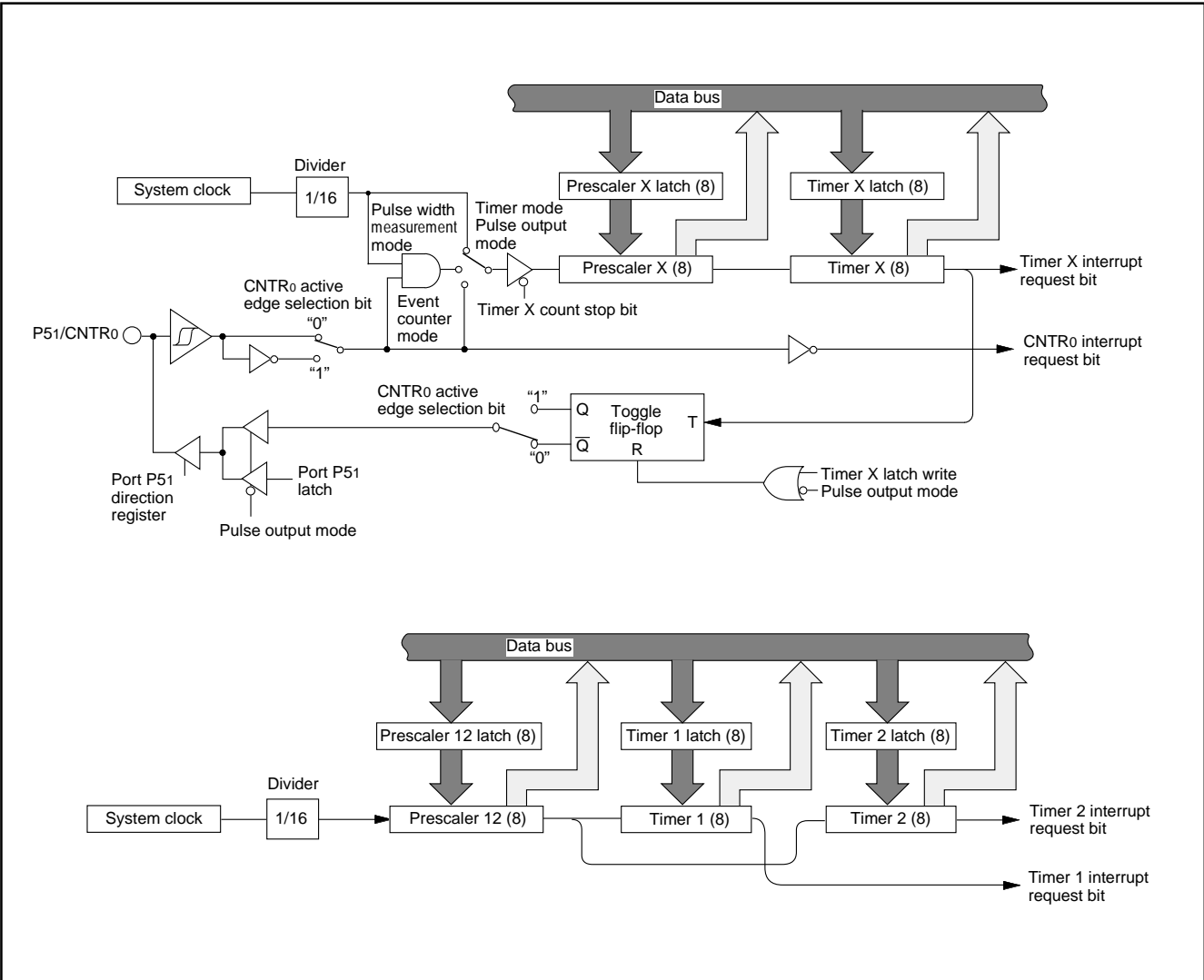


Fig. 17 Timer block diagram

**SERIAL I/O**

Serial I/O can be used as either clock synchronous or asynchronous (UART) serial I/O. A dedicated timer (baud rate generator) is also provided for baud rate generation.

**(1) Clock Synchronous Serial I/O Mode**

Clock synchronous serial I/O mode can be selected by setting the mode selection bit of the serial I/O control register (bit 6 of address 0FE0<sub>16</sub>) to "1".

For clock synchronous serial I/O, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the Transceive/Receive buffer register.

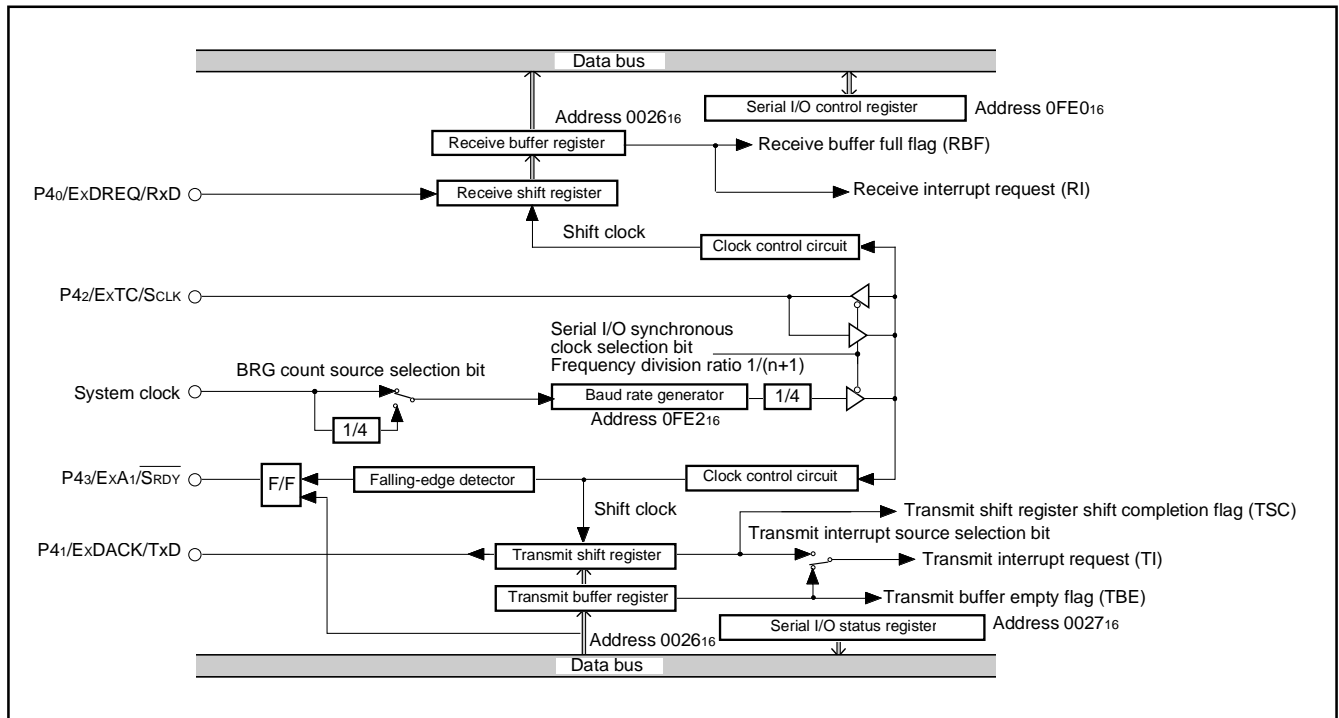


Fig. 18 Block diagram of clock synchronous serial I/O

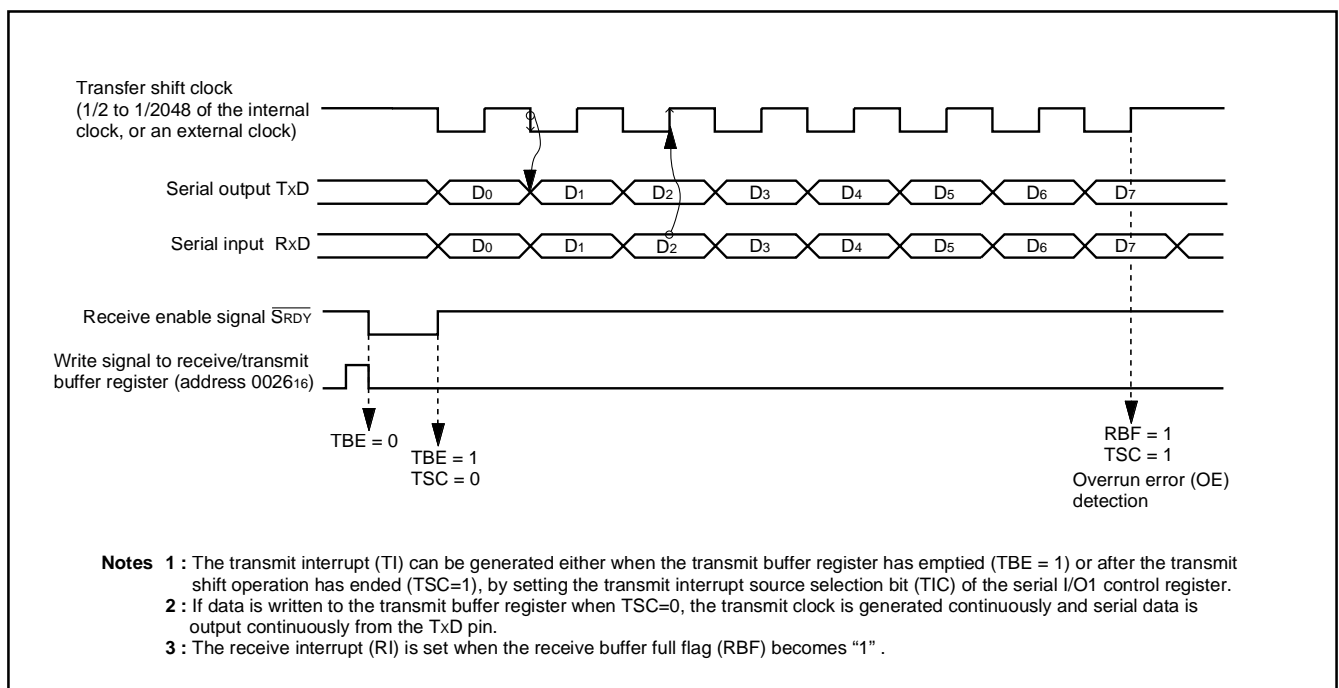


Fig. 19 Operation of clock synchronous serial I/O function

## (2) Asynchronous Serial I/O (UART) Mode

Clock asynchronous serial I/O mode (UART) can be selected by setting the serial I/O mode selection bit of the serial I/O control register to "0".

Eight serial data transfer formats can be selected, and the transfer formats used by a transmitter and receiver must be identical.

The transmit and receive shift registers each have a buffer regis-

ter, but the two buffers have the same address in memory. Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer, and receive data is read from the receive buffer.

The transmit buffer can also hold the next data to be transmitted, and the receive buffer register can hold a character while the next character is being received.

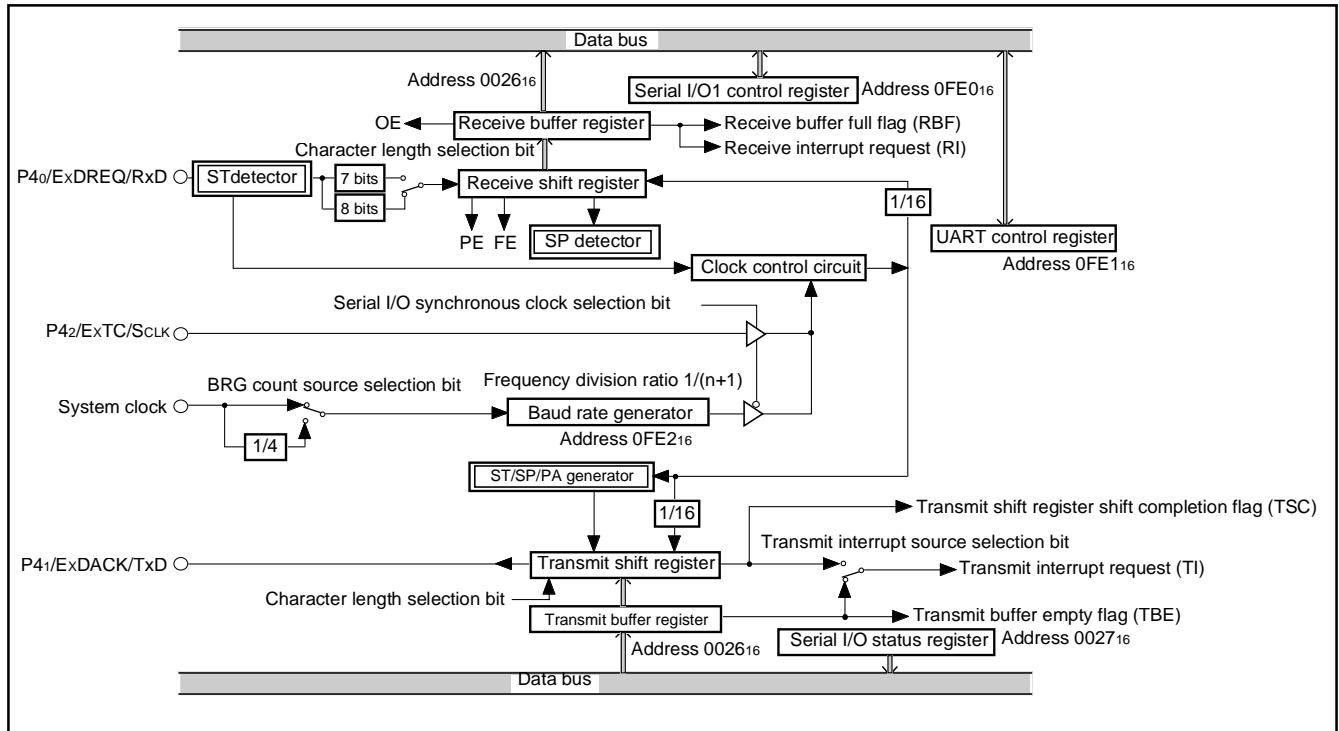


Fig. 20 Block diagram of UART serial I/O

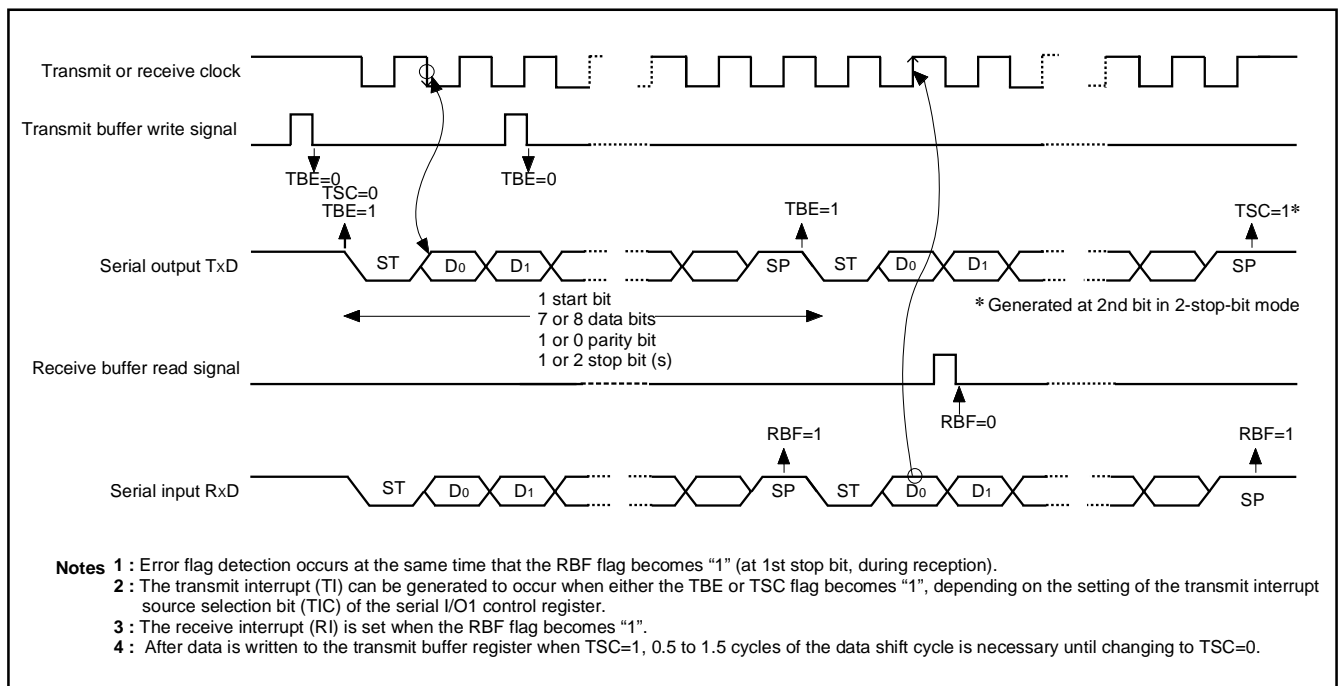


Fig. 21 Operation of UART serial I/O function

### **[Serial I/O Control Register (SIOCON)] 0FE0<sub>16</sub>**

The serial I/O control register contains eight control bits for the serial I/O function.

### **[UART Control Register (UARTCON)] 0FE1<sub>16</sub>**

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of an data transfer.

### **[Serial I/O Status Register (SIOSTS)] 0027<sub>16</sub>**

The read-only serial I/O status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O function and various errors.

Three of the flags (bits 4 to 6) are valid only in UART mode.

The receive buffer full flag (bit 1) is cleared to "0" when the receive buffer is read.

If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing "0" to the serial I/O enable bit SIOE (bit 7 of the serial I/O control register) also clears all the status flags, including the error flags.

All bits of the serial I/O status register are initialized to "0" at reset, but if the transmit enable bit (bit 4) of the serial I/O control register has been set to "1", the transmit shift register shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become "1".

### **[Transmit Buffer/Receive Buffer Register (TB/RB)] 0026<sub>16</sub>**

The transmit buffer register and the receive buffer register are located at the same address. The transmit buffer register is write-only and the receive buffer register is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer register is "0".

### **[Baud Rate Generator (BRG)] 0FE2<sub>16</sub>**

The baud rate generator determines the baud rate for serial transfer.

The baud rate generator divides the frequency of the count source by  $1/(n + 1)$ , where  $n$  is the value written to the baud rate generator.

### **■Notes on serial I/O**

When setting the transmit enable bit to "1", the serial I/O transmit interrupt request bit is automatically set to "1". When not requiring the interrupt occurrence synchronized with the transmission enabled, take the following sequence.

- ① Set the serial I/O transmit interrupt enable bit to "0" (disabled).
- ② Set the transmit enable bit to "1".
- ③ Set the serial I/O transmit interrupt request bit to "0" after 1 or more instructions have been executed.
- ④ Set the serial I/O transmit interrupt enable bit to "1" (enabled).

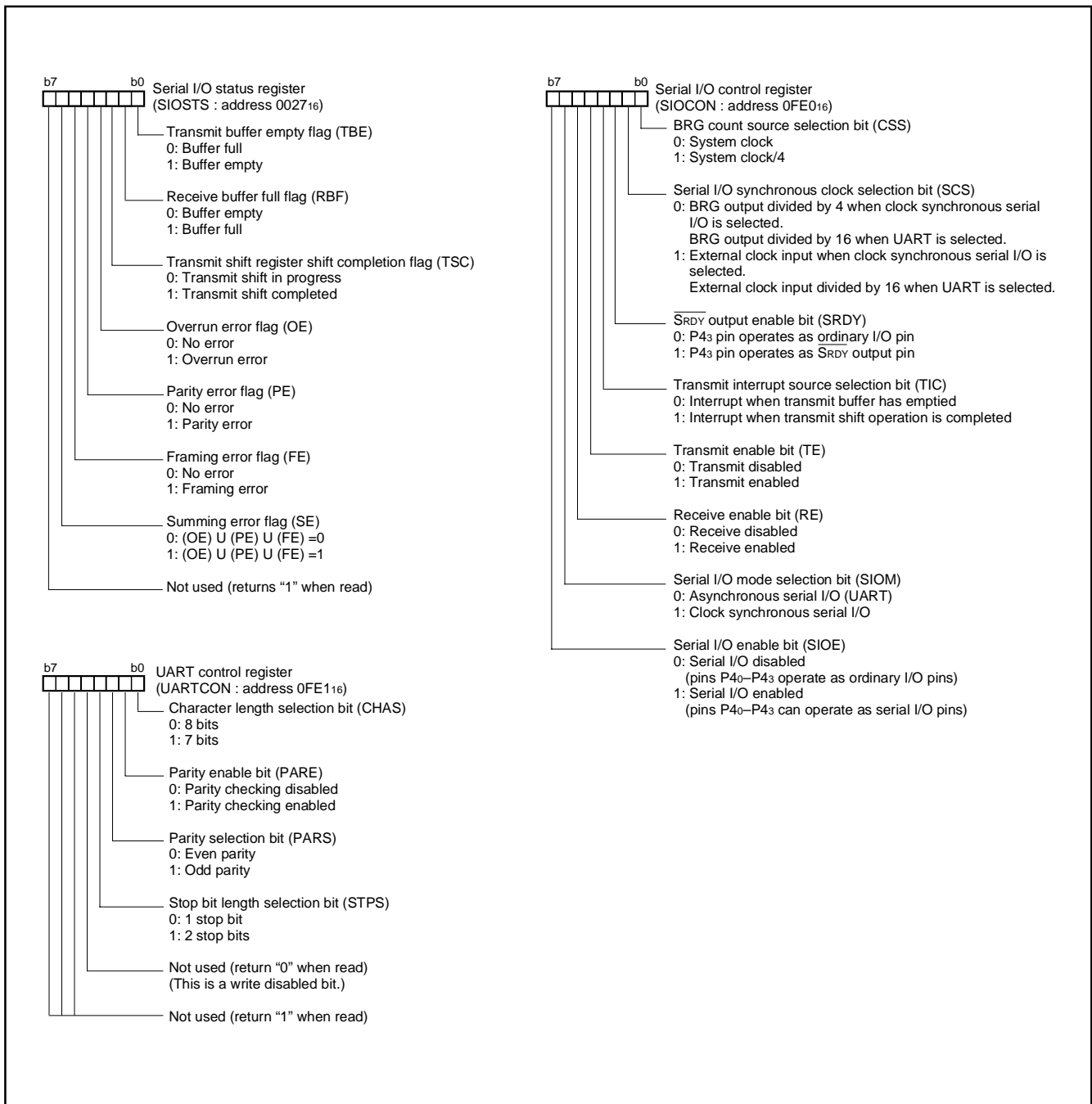


Fig. 22 Structure of serial I/O control registers



## USB FUNCTION

38K0 Group is equipped with a USB function control circuit (USBFCC) that enables effective interfacing with the host-PC. This circuit is in compliance with USB Specification Version 2.0 Full-Speed Transfer Mode (12 Mbps, equivalent to Version 1.1). This circuit also supports all four transfer-types specified in the standard USB specification.

The USBFCC has four endpoints that can select its transfer type. Although Endpoint 0 is fixed to Control Transfer, the Endpoints 1 to 3 can be set to Interrupt Transfer, Bulk Transfer, or Isochronous Transfer.

A dedicated circuit automatically performs stage management for Control Transfer and packet management for transactions, which are necessary for matching of data transmit/receive timing, error detection, and retry after error. This dedicated control circuit enables the user to develop a program or timing design very easily. Each endpoint can be programmed for data transfer conditions so that the endpoints are adaptive for all USB device class transfer systems.

The data buffer of each endpoint can be assigned to any area in the multi-channel RAM. This feature offers highly efficient memory usage by avoiding re-buffering and enabling simple data modification.

The transmit/receive data is directly transferred to the data buffer via the control circuit (direct RAM access type) without disturbing the CPU operation. This mechanism enables the CPU to transfer data smoothly with no drop in performance. In addition to this buffer function, a double-buffer setting will keep a re-buffering stall at a minimum and increase the overall data throughput (max. 64 bytes X 2 channels).

As other special signals control, the endpoints have detection functions for the USB bus reset signal, resume signal, suspend signal, and SOF signal, and also have a remote wake-up signal transmit function.

When completing data transfer or receiving a special signal, the endpoint generates the corresponding interrupt to the CPU (3 vectors/18 factors).

With all this essential yet comprehensive built-in hardware, your system using the 38K2 group will be ready for any USB application that comes its way.

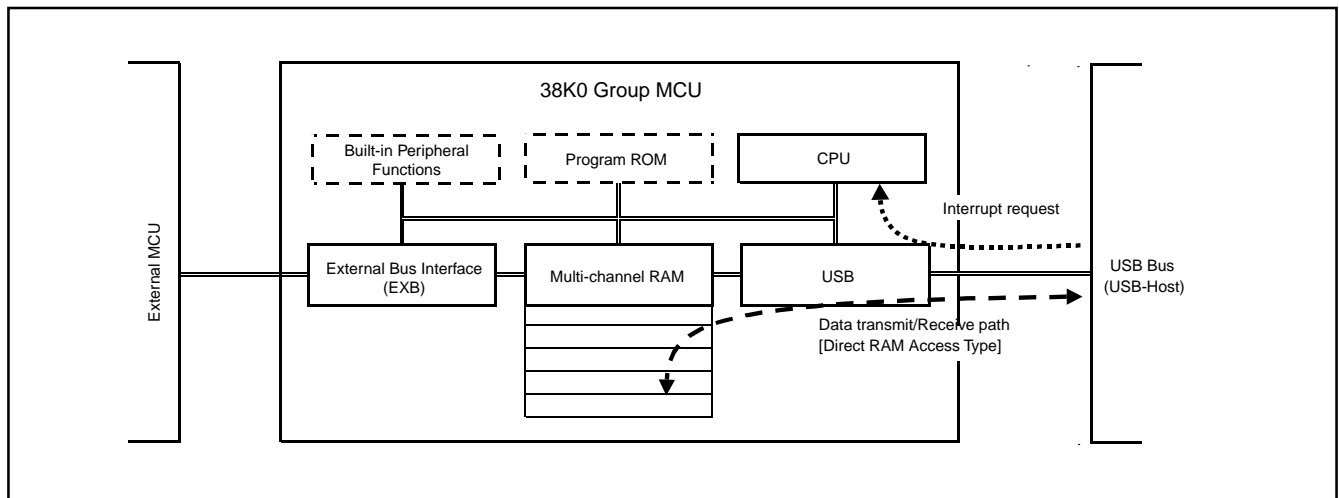


Fig. 23 USB function overview

## USB Data Transfer

The USB specification promises 12 Mbps data transfer in the full-speed mode, that is equivalent to 1.5 M bytes per second of data transactions.

However, in USB data transfer, bit-stuffing may be executed depending on the bit patterns of the transfer data, possibly resulting in 1-byte data (normally 8 bits) handled as up to 10 bits.

Because USB uses asynchronous transfers, the clock cycle of the USB internal reference clock may change to adjust to the clock phase. Therefore, the access timing of the USBFCC for the multi-channel RAM will change owing to the frequency of internal clock  $\phi$ :

When the USBFCC is operating at  $\phi = 8$  MHz, access for a normal

transfer is performed every 5 to 6 cycles and access for a bit-stuffing transfer is performed in up to 7 cycles.

If the EXB function is enabled in the above conditions, this function generates a maximum wait of 1 clock cycle, so that the access is performed every 4 to 8 cycles.

When operating at  $\phi = 6$  MHz, a normal access is performed every 4 cycles. If the clock-phase correction of the reference clock occurs, access is performed every 3 to 5 cycles.

If bit stuffing occurs at this clock rate, the access cycle will be extended to up to 6 cycles. When the EXB function that generates a maximum 1-wait cycle is used in this condition, the access cycle will be 2 (min.) to 7 (max.) cycles.

## USB Function Control Circuit (USBFCC) Block Diagram

The following diagram shows the USBFCC block diagram. The circuit comprises:

- (1) Serial Interface Engine (SIE)
- (2) Device Control Unit (DCU)
- (3) Internal Memory Interface (MIF)
- (4) CPU Interface (CIF)

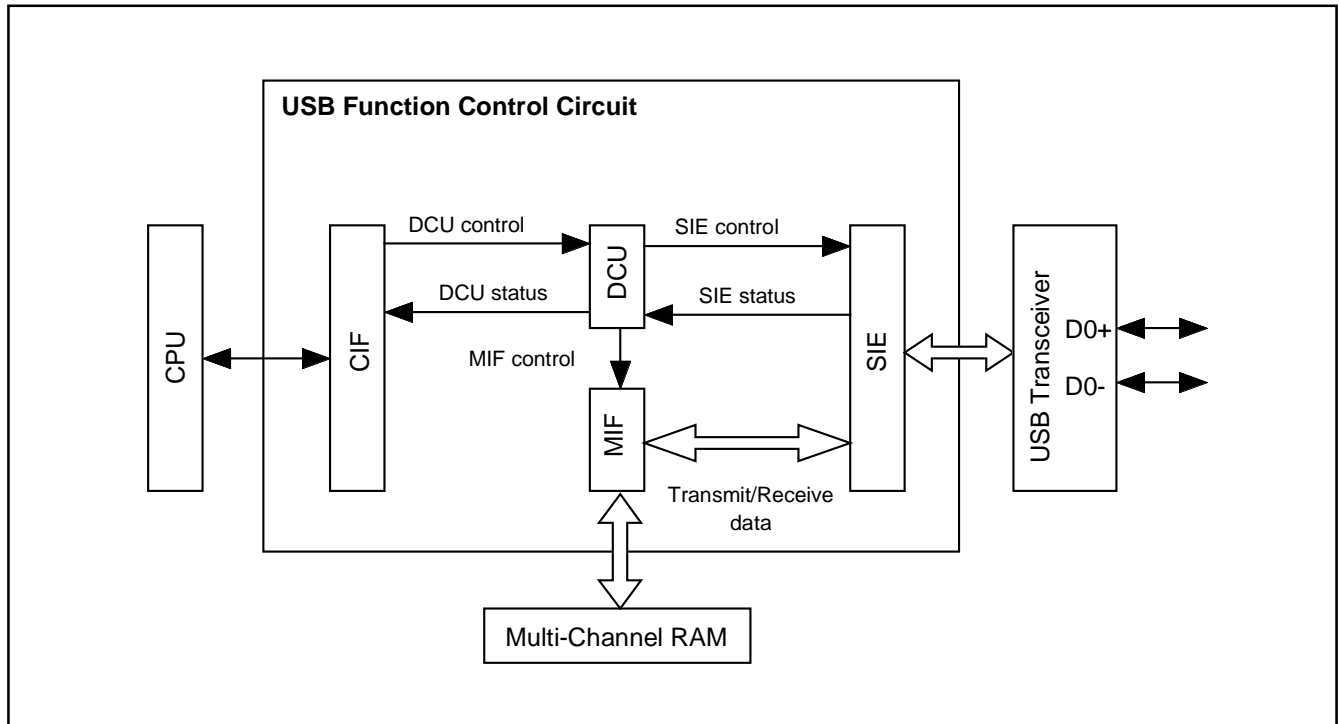


Fig. 24 USB Function Control Circuit (USBFCC) block diagram

### (1) Serial Interface Engine (SIE)

The SIE performs the following USB lower-layer protocols (packets, transactions):

- Sampling of receive data and clock, generation of transmit clock
- Serial-to-parallel conversion of transmit/receive data
- NRZI (Non Return Zero Invert) encode/decode
- Bit stuffing/unstuffing
- SYNC (Synchronization Pattern) detection, EOP (End of Packet) detection
- USB address detection, endpoint detection
- CRC (Cyclic Redundancy Check) generation and checking

### (2) Device Control Unit (DCU)

The DCU manages the following USB upper-layer protocols (address/endpoint and control-transfer sequence):

- Status control for each endpoint
- Control-transfer sequence control
- Memory interface status control

### (3) Memory Interface (MIF)

The MIF controls the flow of data transfer between the SIE and the multi-channel RAM under the management of the DCU.

### (4) CPU Interface (CIF)

The CIF performs the following functions:

- Mode setting via registers, DCU control signal generation, DCU status signal reading
- Interrupt signal generation
- Internal bus interface control.



### Endpoint Buffer Area Setting

The buffer area used in data transfer can be assigned to any area of the multi-channel RAM for each endpoint.

#### ●Buffer area beginning address

The buffer area configuration register (address 0FED<sub>16</sub>) defines the beginning address of the buffer area (every 32 bytes) for each Endpoint. However, the only RAM area is configurable.

- 00h [Address 0000<sub>16</sub>], 01h [Address 0020<sub>16</sub>]: Not configurable
- 02h [Address 0040<sub>16</sub>] to 1Fh [Address 03E0<sub>16</sub>]: Configurable

#### ●Interrupt-source dependant buffer area offset address

An offset value is added to the beginning address of each source, which is specified by the interrupt source register (address 001D<sub>16</sub>), for each endpoint.

This section describes in detail the beginning address specified by the buffer area set register as offset address 00h, according to each endpoint.

#### (1) Endpoint 00

Endpoint 00 has two kinds of interrupt sources for accessing the buffer. The respective address offsets are:

- BSRDY00 (SETUP Buffer Ready Interrupt): Offset address = 00h
- BRDY00 (OUT or IN Buffer Ready Interrupt):  
Offset address = 08h

#### (2) Endpoint 01

The buffer area offset address for each interrupt source for of Endpoint 01 varies according to the contents of the EP01 set register (address 0019<sub>16</sub>).

- In single buffer mode (DBLB01 = "0"): Endpoint 01 has only one interrupt source for accessing the buffer.  
B0RDY01 (Buffer 0 Ready Interrupt): Offset address = 00h

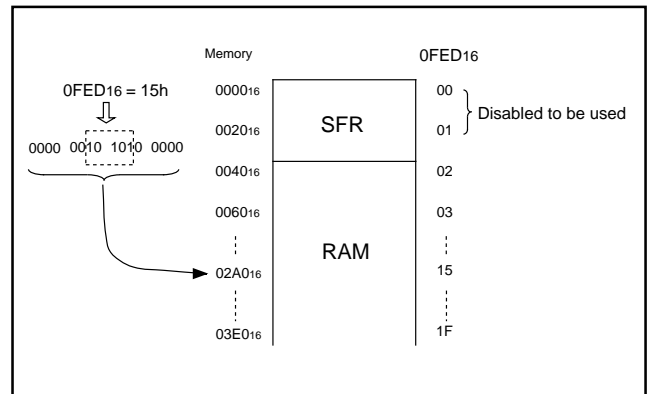


Fig. 26 Example setting of buffer area beginning address

- In double buffer mode (DBLB01 = "1"):

Endpoint 01 has two kinds of interrupt sources for accessing the buffer.

B0RDY01 (Buffer 0 Ready Interrupt): Offset address = 00h

B1RDY01 (Buffer 1 Ready Interrupt):

The offset address varies according to the double buffer beginning address set bit (BSIZ01).

- Offset address = 08h when BSIZ01 = 00
- Offset address = 10h when BSIZ01 = 01
- Offset address = 40h when BSIZ01 = 10
- Offset address = 80h when BSIZ01 = 11

#### (3) Endpoints 02 and 03

Same as Endpoint 01.

### Notes

The selected RAM area must be within addresses 0040<sub>16</sub> to 03FF<sub>16</sub>.

Make sure the buffer area beginning address is set in agreement with the offset address and the number of transmit/receive data bytes.

This is particularly important when in the double buffer mode or when handling 64-byte data.

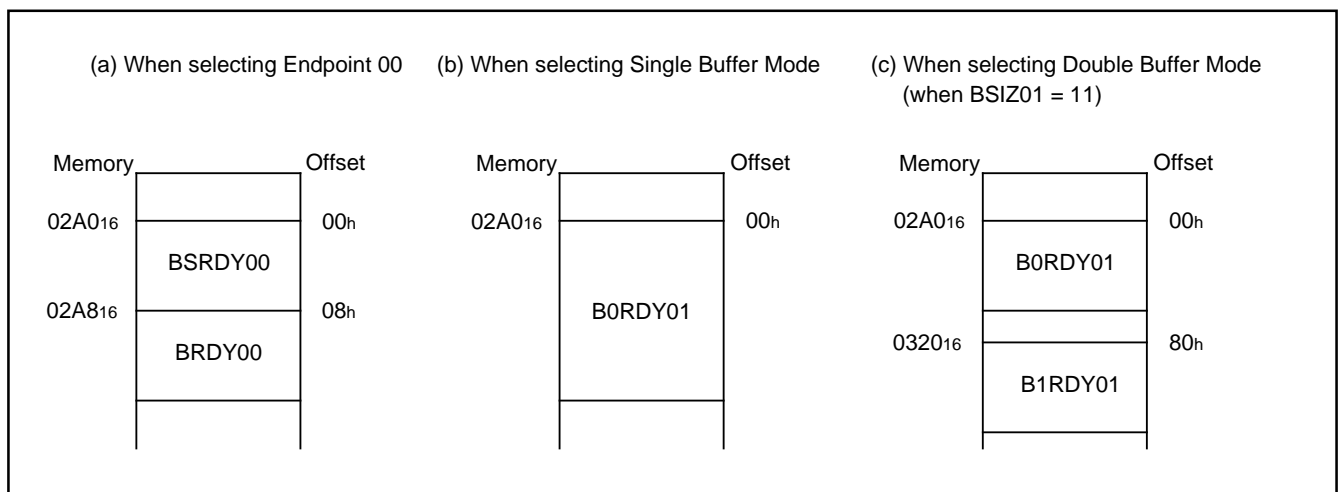


Fig. 27 Examples of interrupt source dependant buffer area offset address

## USB Interrupt Function

USB Interrupt Control Circuit (USBINTCON) has 3 requests and 16 USB-device interrupt request sources. Each interrupt source register enables the user to easily determine which interrupt has occurred.

Table 7 shows the list of USB interrupt sources.

**Table 7 USB interrupt sources**

Interrupt request bit (IREQ1: Address 003C16)	USB interrupt bit (USBIREQ: Address 001716)	Interrupt source
USB bus reset	—	At USB bus reset signal detection: After enabling the USB module (USBE = "1"), an interrupt request occurs when 2.5 $\mu$ s SE0 state is detected in D0+/D0- port. (Equivalent to 120-clock length when fUSB = 48 MHz)
USB SOF	—	At SOF packet receive: After enabling the USB module (USBE = "1"), an interrupt request occurs when SOF packet is detected in D0+/D0- port. Its occurrence does not depend on frame-time or CRC value after SOF packet is transferred. (Normally, SOF packet detection occurs only when fUSB = 48 MHz)
USB device	EP00	At Endpoint 00 data transfer complete: •Buffer ready (read/write enabled state) •Control transfer completed •Status stage transition •SETUP buffer ready (read enabled state) •Control transfer error
	EP01	At Endpoint 01 data transfer complete: •Buffer 0 ready (read/write enabled state) •Buffer 1 ready (read/write enabled state) •Transfer error
	EP02	At Endpoint 02 data transfer complete: •Buffer 0 ready (read/write enabled state) •Buffer 1 ready (read/write enabled state) •Transfer error
	EP03	At Endpoint 03 data transfer complete: •Buffer 0 ready (read/write enabled state) •Buffer 1 ready (read/write enabled state) •Transfer error
	SUS	At suspend signal detection: After enabling the USB module (USBE = "1"), an interrupt request occurs when 3 ms J state is detected in D0+/D0- port. (Equivalent to 144,000 clock-length when fUSB = 48MHz)
	RSM	At resume signal detection: After enabling the USB module (USBE = "1") and resume interrupt (RSME = "1"), an interrupt request occurs when a bus state change (J state to SE0 or K state) is detected in D0- port.

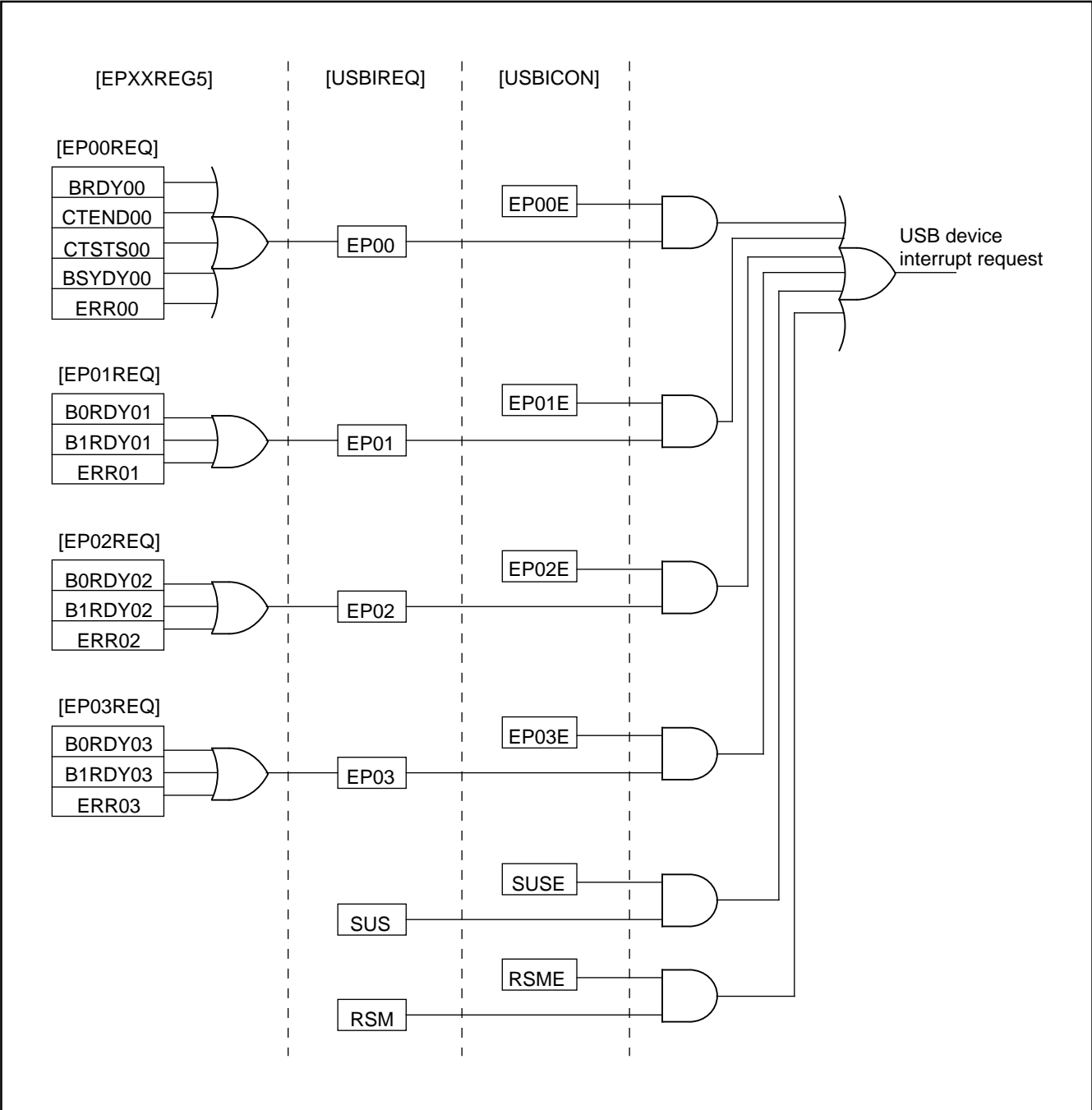


Fig. 28 USB device interrupt control

### USB Register List

The USB register list is shown below.

Address	Register Name	SYMBOL	USB SFR							
			bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0010 <sub>16</sub>	USB control register	USBCON	USBE	UCLKCON	USBDIFE	VREFE	VREFCON	TRONE	TRONCON	WKUP
0011 <sub>16</sub>	USB Function enable register	USBAE								AD0E
0012 <sub>16</sub>	USB function address register	USBA0	USBADD0[6:0]							
0013 <sub>16</sub>			FNUM[7:0]							
0014 <sub>16</sub>	Frame number register Low	FNUML	FNUM[7:0]							
0015 <sub>16</sub>	Frame number register High	FNUMH	FNUM[10:8]							
0016 <sub>16</sub>	USB interrupt source enable register	USBIICON	RSME	SUSE			EP03E	EP02E	EP01E	EP00E
0017 <sub>16</sub>	USB interrupt source register	USBIREQ	RSM	SUS			EP03	EP02	EP01	EP00
0018 <sub>16</sub>	Endpoint index register	USBINDEX	EPIDX[1:0]							
0019 <sub>16</sub>	Endpoint field register 1	EPXXREG1								
001A <sub>16</sub>	Endpoint field register 2	EPXXREG2								
001B <sub>16</sub>	Endpoint field register 3	EPXXREG3								
001C <sub>16</sub>	Endpoint field register 4	EPXXREG4								
001D <sub>16</sub>	Endpoint field register 5	EPXXREG5								
001E <sub>16</sub>	Endpoint field register 6	EPXXREG6								
001F <sub>16</sub>	Endpoint field register 7	EPXXREG7								
0FEC <sub>16</sub>	Endpoint field register 8	EPXXREG8								
0FED <sub>16</sub>	Endpoint field register 9	EPXXREG9								
(1) Endpoint 00										
0019 <sub>16</sub>	EP00 stage register	EP00STG								SETUP00
001A <sub>16</sub>	EP00 control register 1	EP00CON1							PID00[1:0]	
001B <sub>16</sub>	EP00 control register 2	EP00CON2								BVAL00
001C <sub>16</sub>	EP00 control register 3	EP00CON3								CTENDE00
001D <sub>16</sub>	EP00 interrupt source register	EP00REQ				ERR00	BSRDY00	CTSTS00	CTEND00	BRDY00
001E <sub>16</sub>	EP00 transmit/receive byte number register	EP00BYT						BBYT00[3:0]		
001F <sub>16</sub>										
0FEC <sub>16</sub>										
0FED <sub>16</sub>	EP00 buffer area set register	EP00BUF						BADD00[4:0]		
(2) Endpoint 01										
0019 <sub>16</sub>	EP01 set register	EP01CFG	TYP01[1:0]		DIR01	ITMD01	SQCL01	DBLB01	BSIZ01[1:0]	
001A <sub>16</sub>	EP01 control register 1	EP01CON1							PID01[1:0]	
001B <sub>16</sub>	EP01 control register 2	EP01CON2								BOVAL01
001C <sub>16</sub>	EP01 control register 3	EP01CON3								B1VAL01
001D <sub>16</sub>	EP01 interrupt source register	EP01REQ						ERR01	B1RDY01	B0RDY01
001E <sub>16</sub>	EP01 byte number register 0	EP01BYT0					B0BYT01[6:0]			
001F <sub>16</sub>	EP01 byte number register 1	EP01BYT1					B1BYT01[6:0]			
0FEC <sub>16</sub>	EP01 MAX. packet size register	EP01MAX					MXPS01[6:0]			
0FED <sub>16</sub>	EP01 buffer area set register	EP01BUF						BADD01[4:0]		
(3) Endpoint 02										
0019 <sub>16</sub>	EP02 set register	EP02CFG	TYP02[1:0]		DIR02	ITMD02	SQCL02	DBLB02	BSIZ02[1:0]	
001A <sub>16</sub>	EP02 control register 1	EP02CON1							PID02[1:0]	
001B <sub>16</sub>	EP02 control register 2	EP02CON2								BOVAL02
001C <sub>16</sub>	EP02 control register 3	EP02CON3								B1VAL02
001D <sub>16</sub>	EP02 interrupt source register	EP02REQ						ERR02	B1RDY02	B0RDY02
001E <sub>16</sub>	EP02 byte number register 0	EP02BYT0					B0BYT02[6:0]			
001F <sub>16</sub>	EP02 byte number register 1	EP02BYT1					B1BYT02[6:0]			
0FEC <sub>16</sub>	EP02 MAX. packet size register	EP02MAX					MXPS02[6:0]			
0FED <sub>16</sub>	EP02 buffer area set register	EP02BUF						BADD02[4:0]		
(4) Endpoint 03										
0019 <sub>16</sub>	EP03 set register	EP03CFG	TYP03[1:0]		DIR03	ITMD03	SQCL03	DBLB03	BSIZ03[1:0]	
001A <sub>16</sub>	EP03 control register 1	EP03CON1							PID03[1:0]	
001B <sub>16</sub>	EP03 control register 2	EP03CON2								BOVAL03
001C <sub>16</sub>	EP03 control register 3	EP03CON3								B1VAL03
001D <sub>16</sub>	EP03 interrupt source register	EP03REQ						ERR03	B1RDY03	B0RDY03
001E <sub>16</sub>	EP03 byte number register 0	EP03BYT0					B0BYT03[6:0]			
001F <sub>16</sub>	EP03 byte number register 1	EP03BYT1					B1BYT03[6:0]			
0FEC <sub>16</sub>	EP03 MAX. packet size register	EP03MAX					MXPS03[6:0]			
0FED <sub>16</sub>	EP03 buffer area set register	EP03BUF						BADD03[4:0]		

 : Not used

Fig. 29 USB related registers

## USB Related Registers

The USB related registers are shown below.

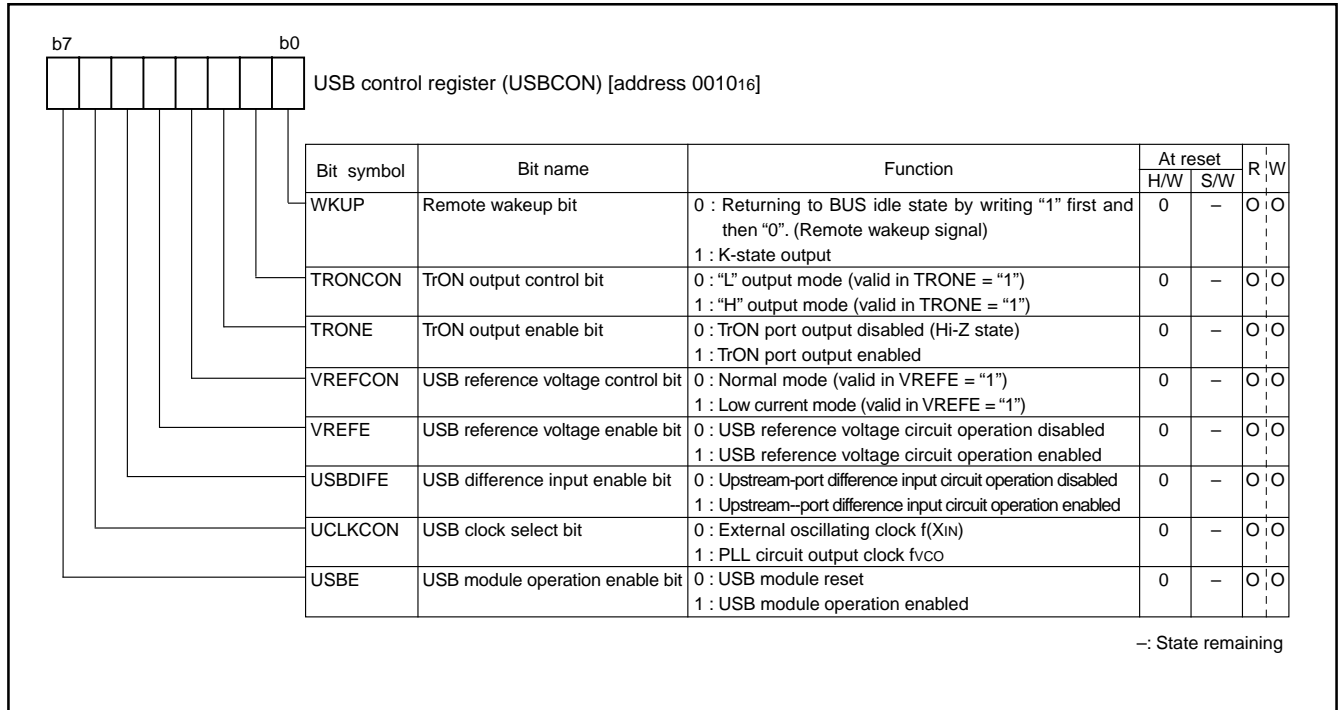


Fig. 30 Structure of USB control register

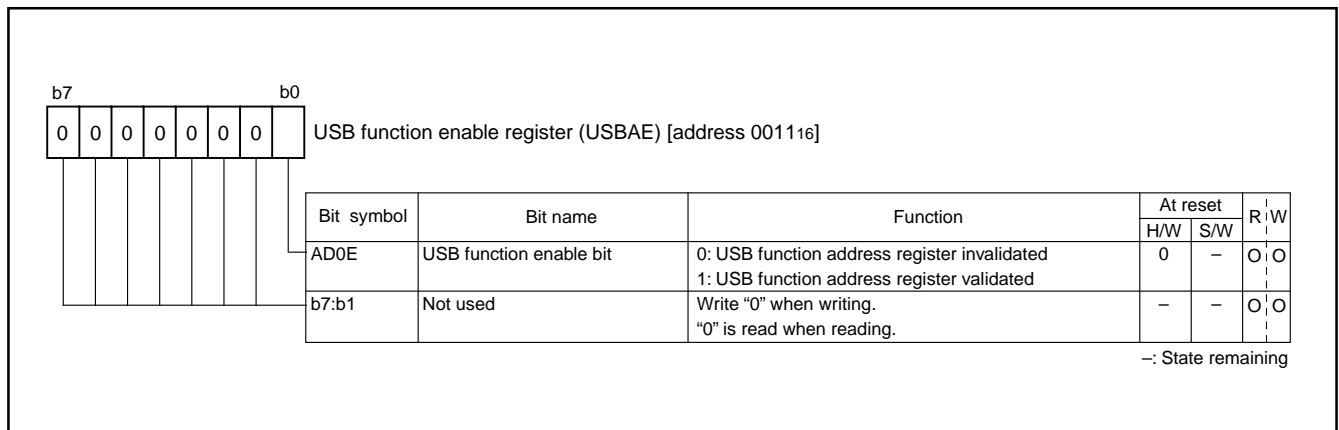


Fig. 31 Structure of USB function enable register



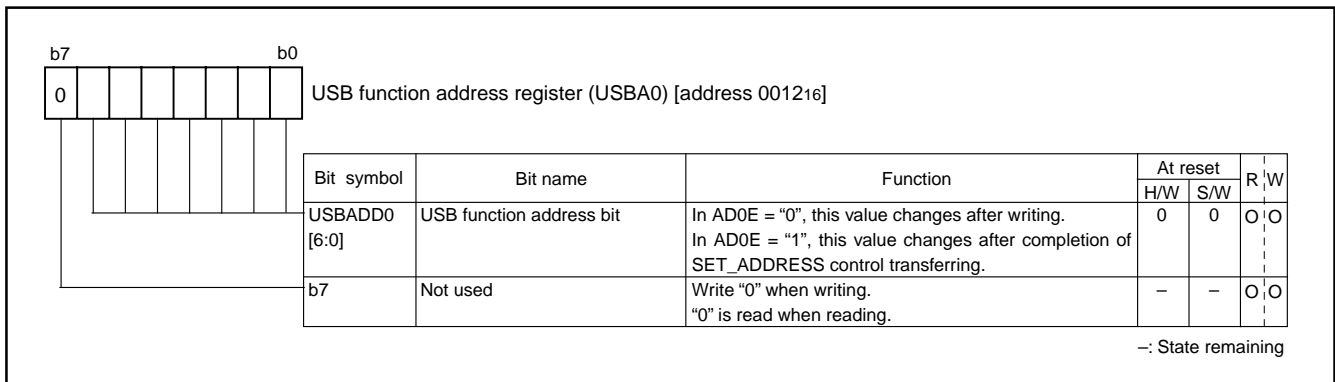


Fig. 32 Structure of USB function address register

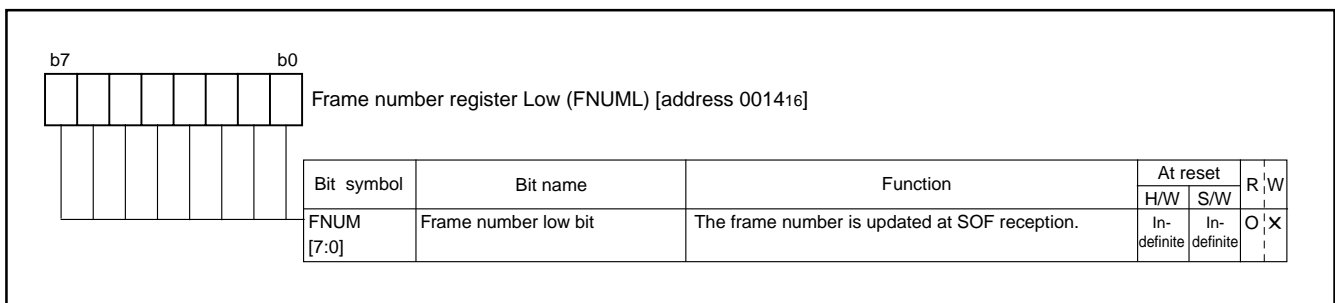


Fig. 33 Structure of Frame number register Low

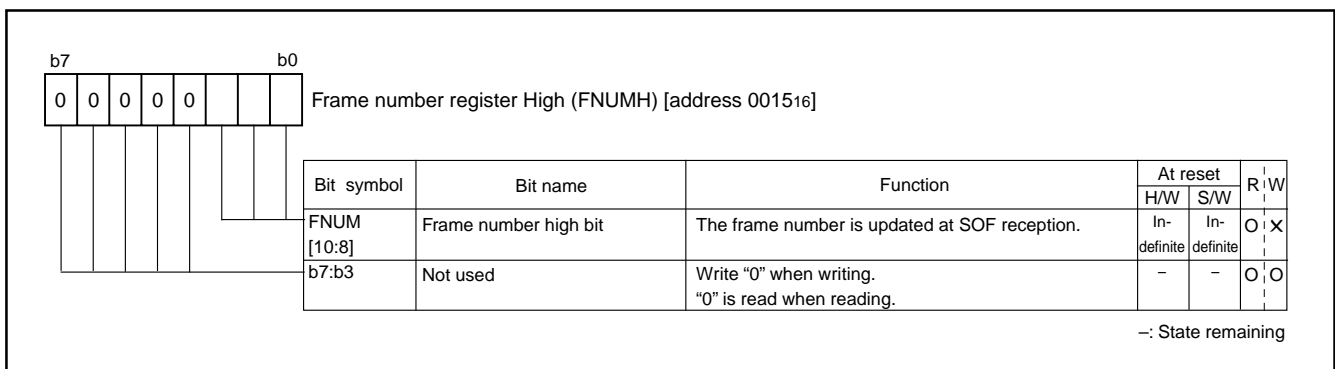


Fig. 34 Structure of Frame number register High

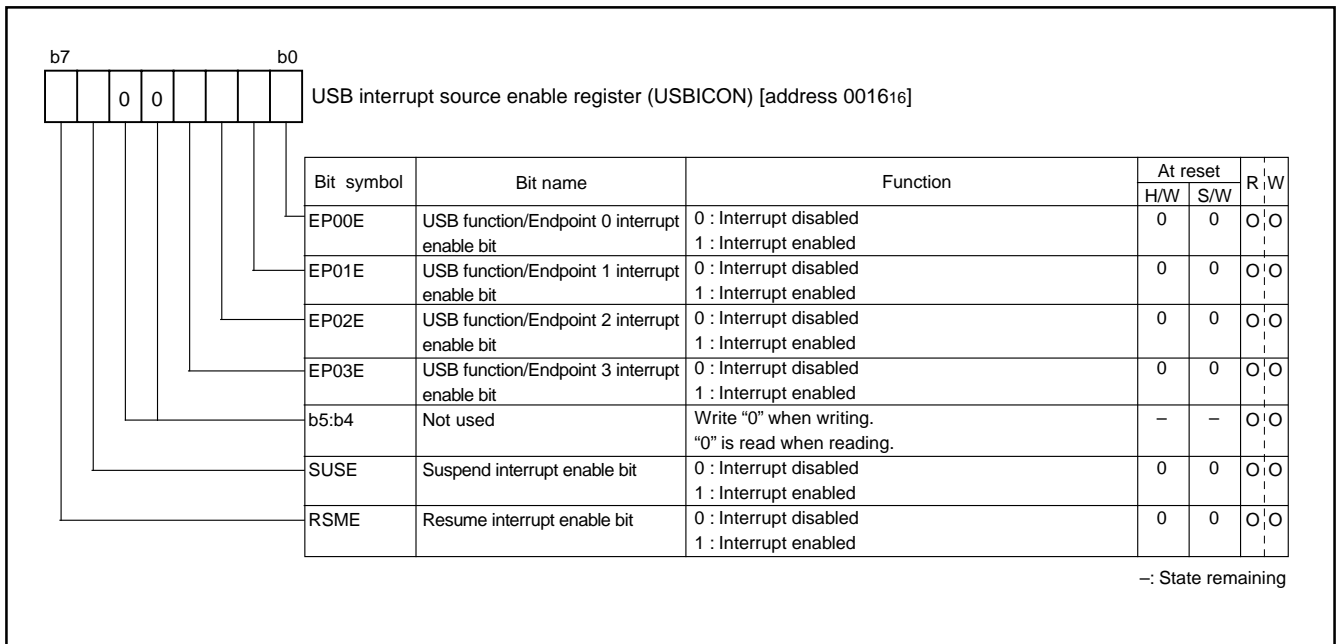


Fig. 35 Structure of USB interrupt source enable register

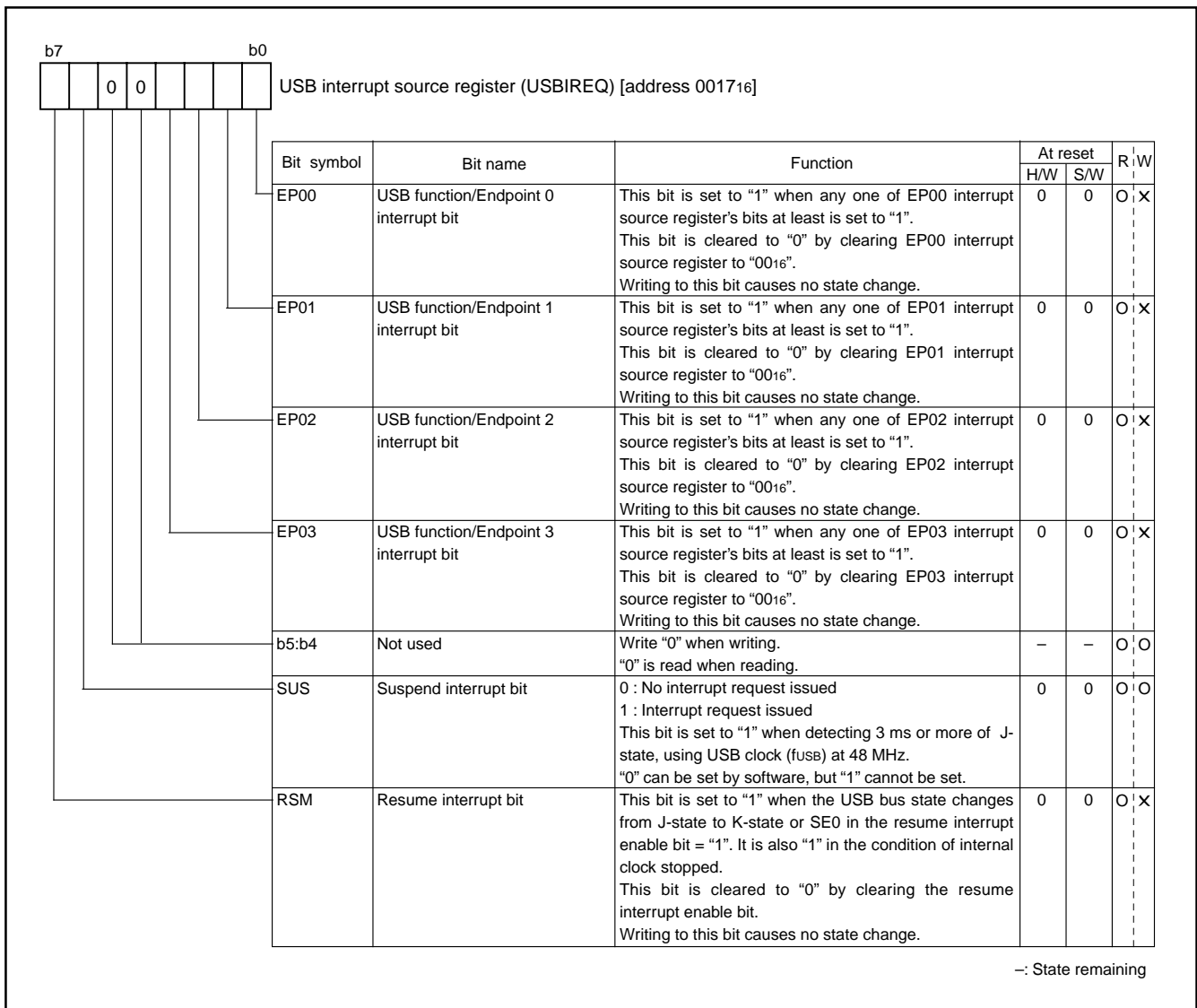


Fig.36 Structure of USB interrupt source register

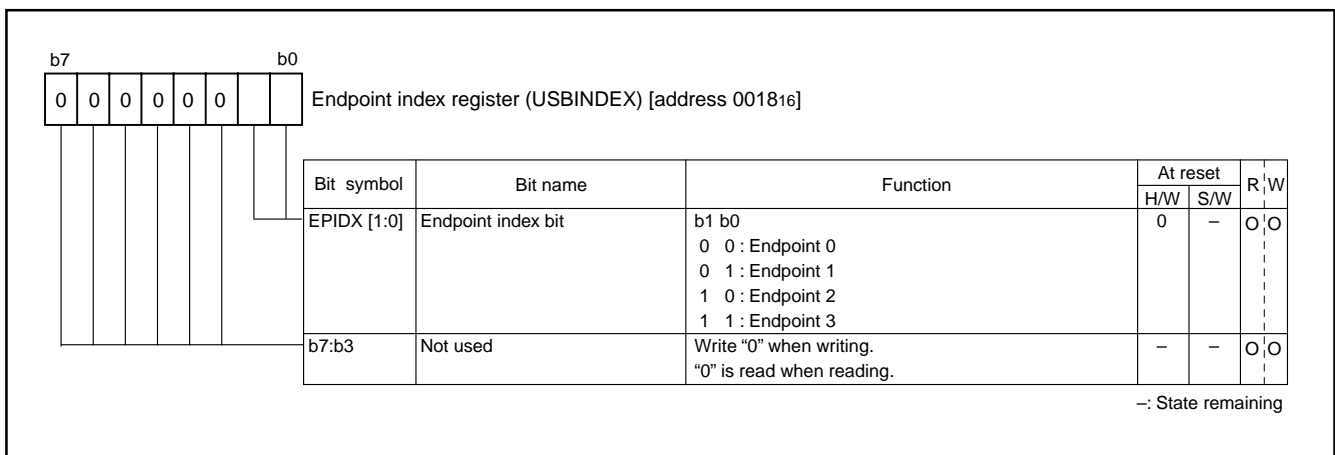


Fig. 37 Structure of Endpoint index register

(1) Endpoint 00

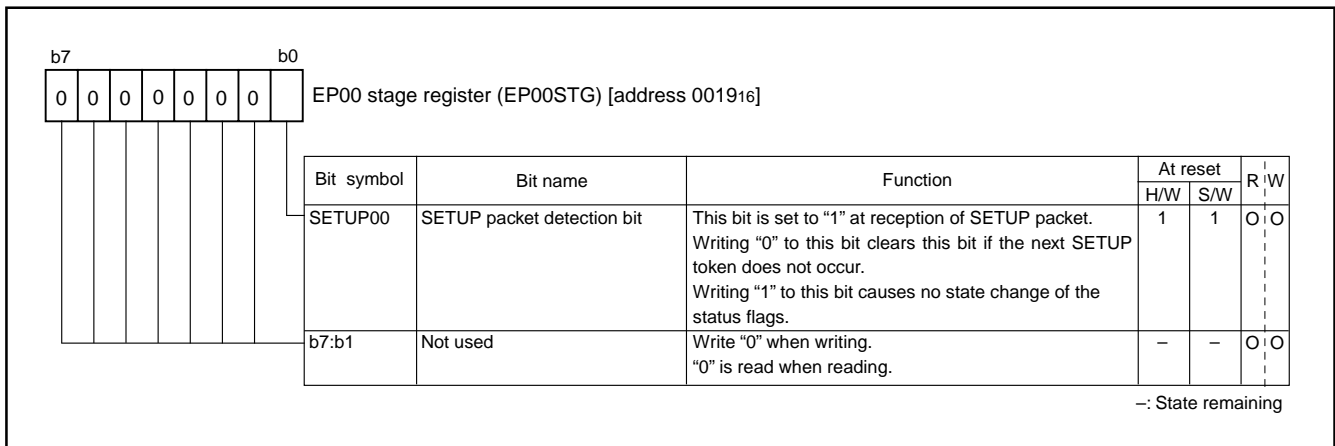


Fig. 38 Structure of EP00 stage register

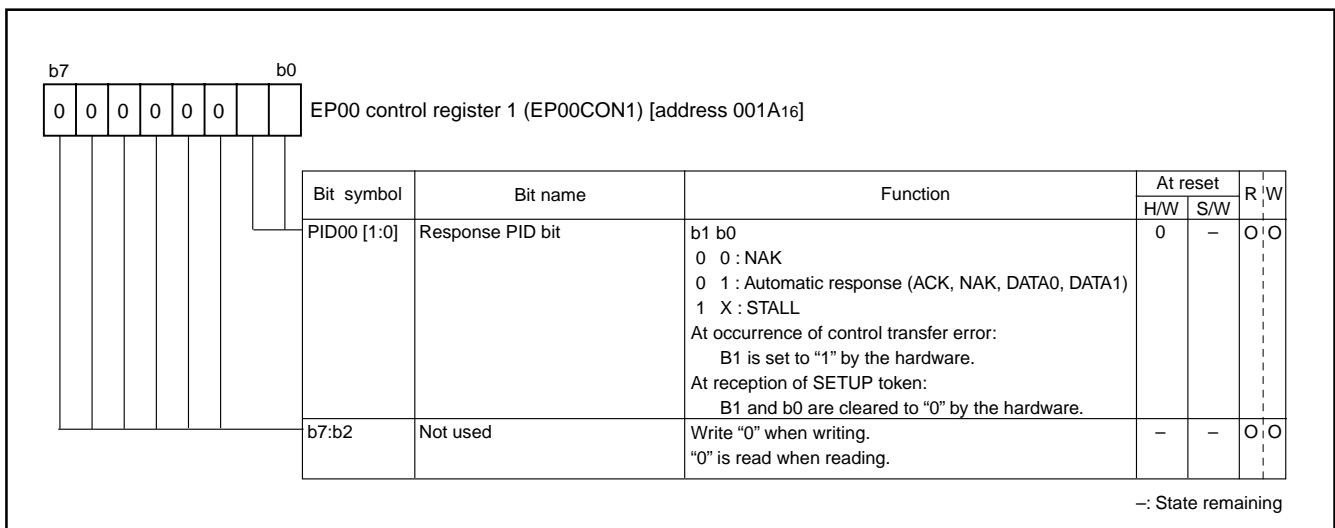


Fig. 39 Structure of EP00 control register 1

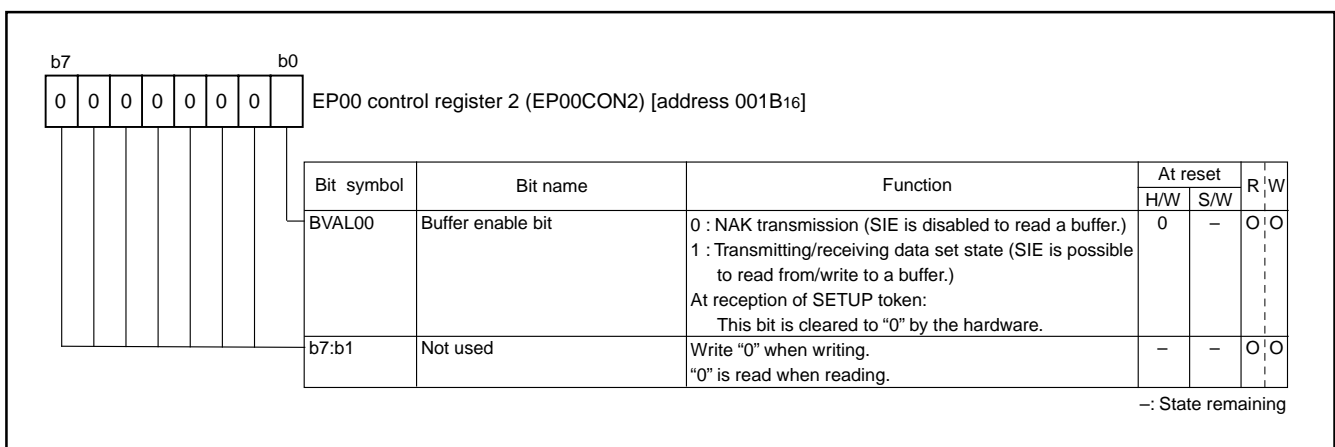


Fig. 40 Structure of EP00 control register 2

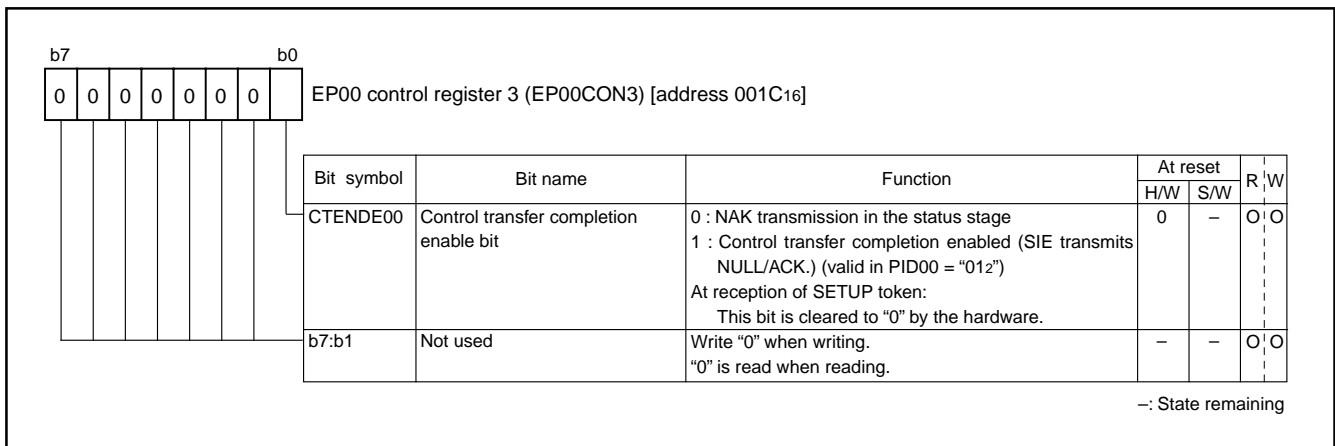


Fig. 41 Structure of EP00 control register 3

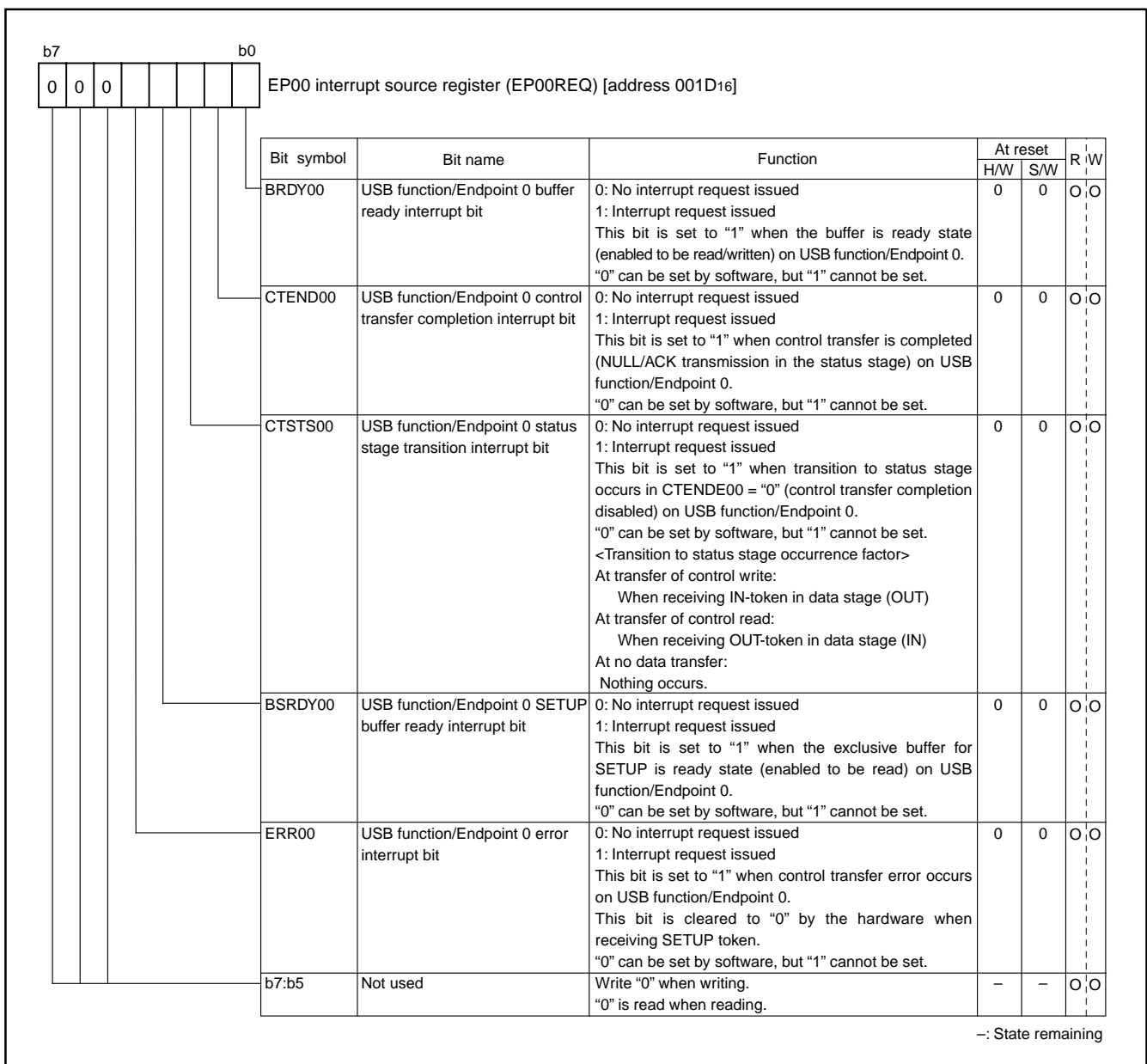


Fig. 42 Structure of EP00 interrupt source register

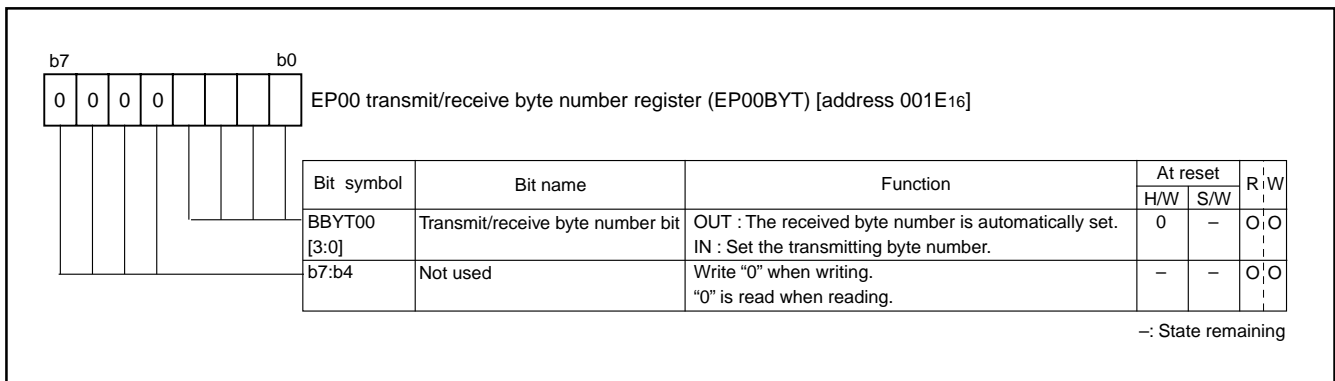


Fig. 43 Structure of EP00 transmit/receive byte number register

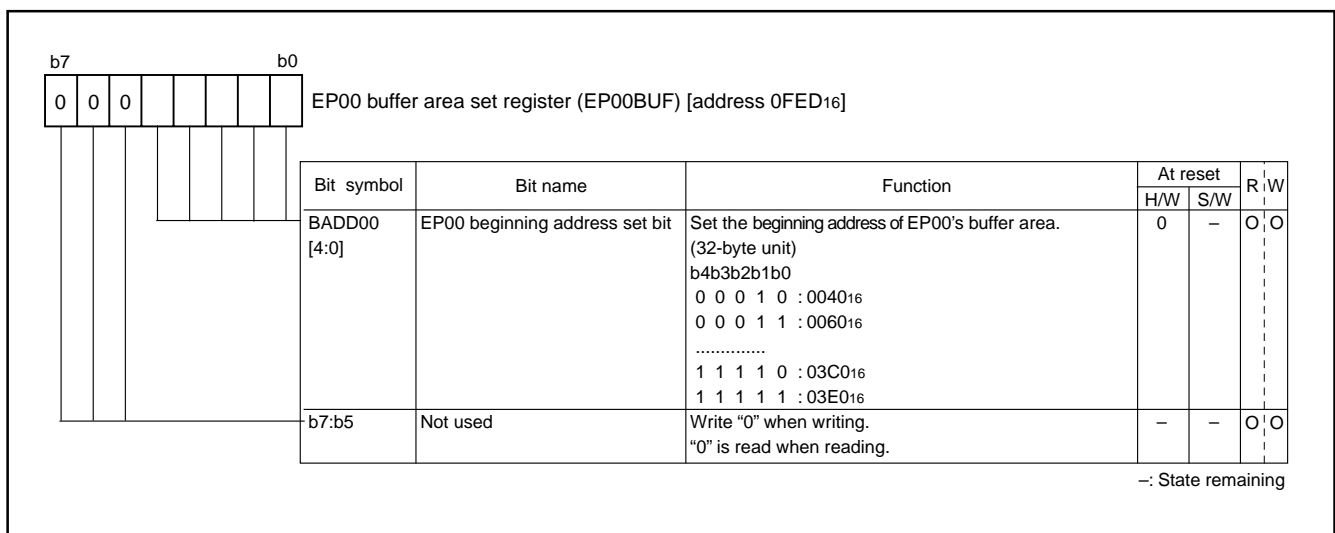


Fig. 44 Structure of EP00 buffer area set register

(2) Endpoint 01

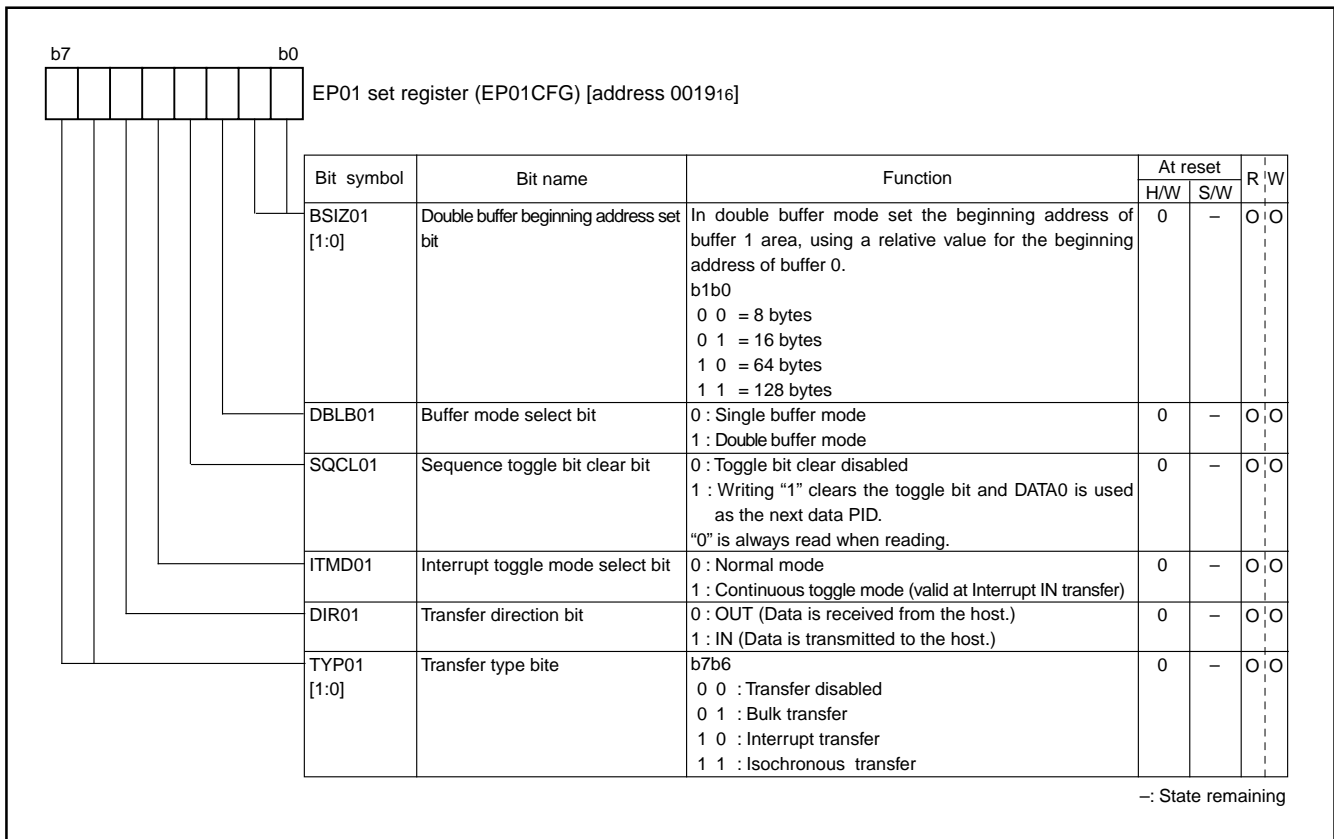


Fig. 45 Structure of EP01 set register

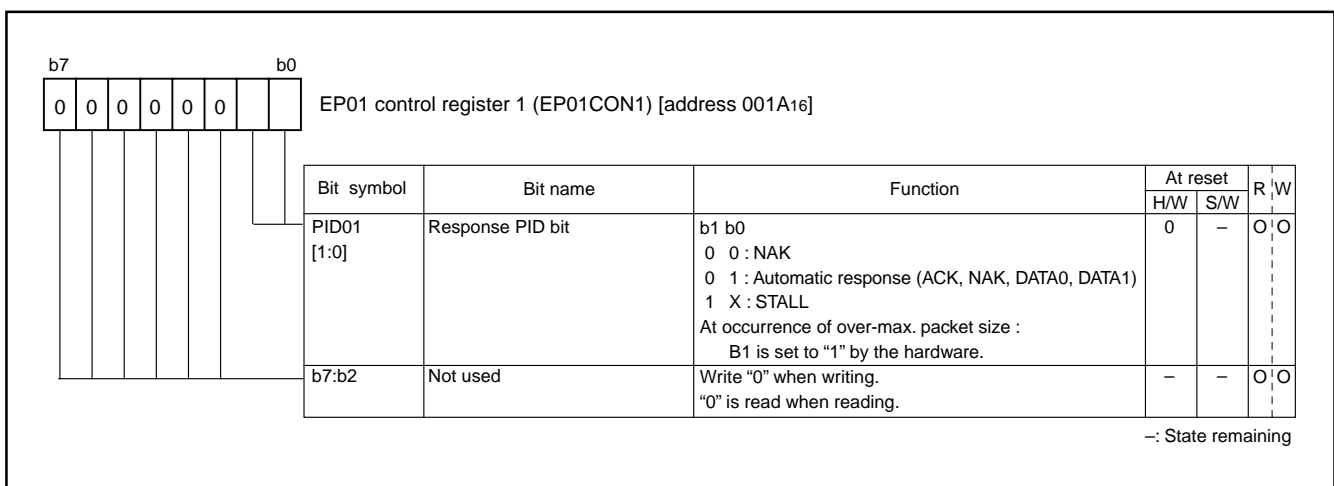


Fig. 46 Structure of EP01 control register 1

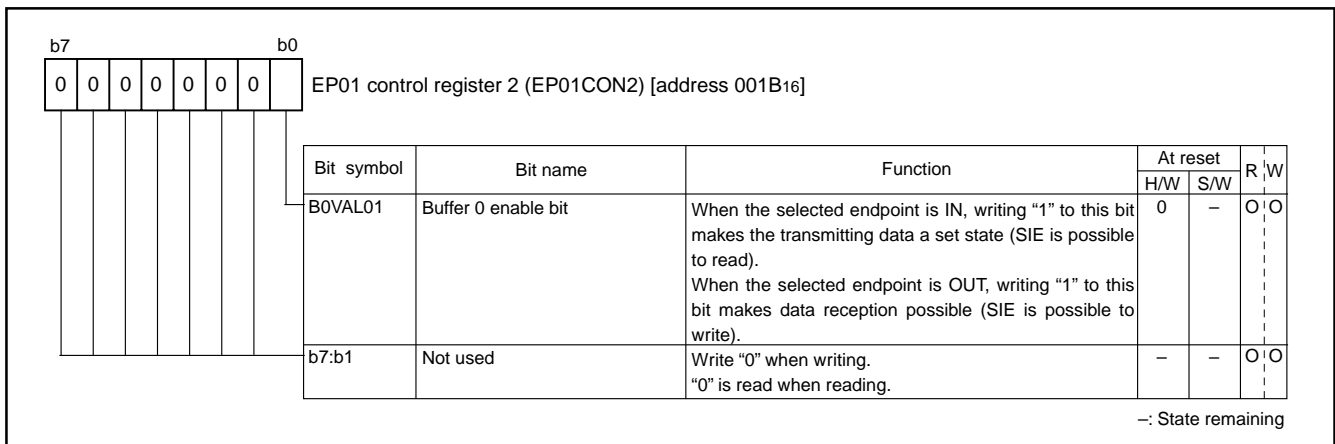


Fig. 47 Structure of EP01 control register 2

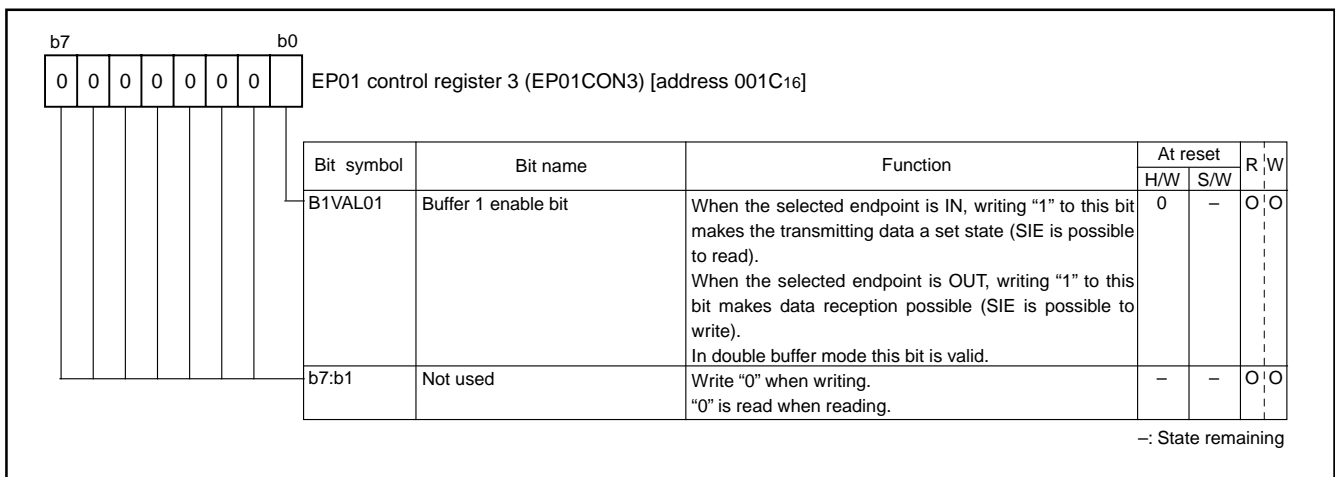


Fig. 48 Structure of EP01 control register 3

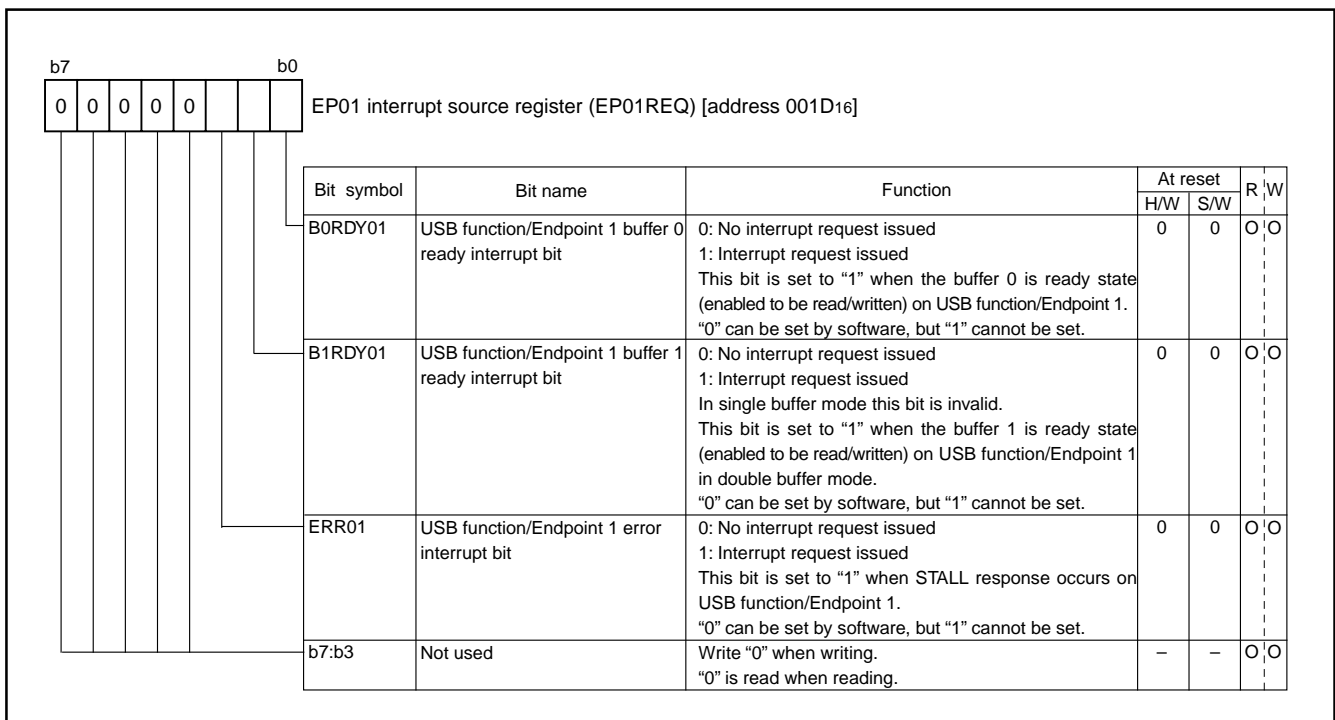


Fig. 49 Structure of EP01 interrupt source register



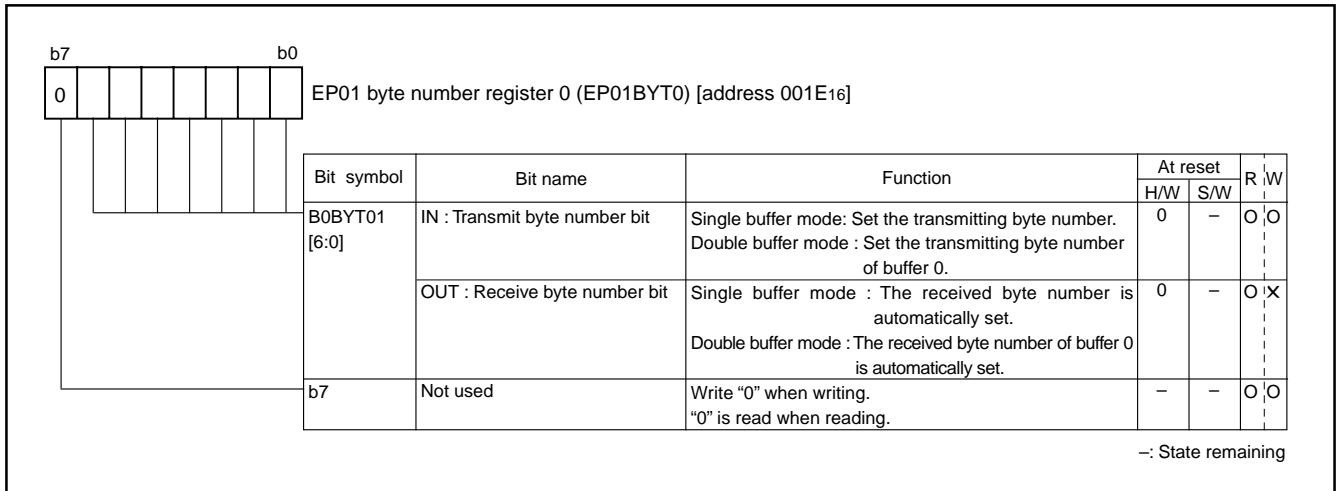


Fig. 50 Structure of EP01 byte number register 0

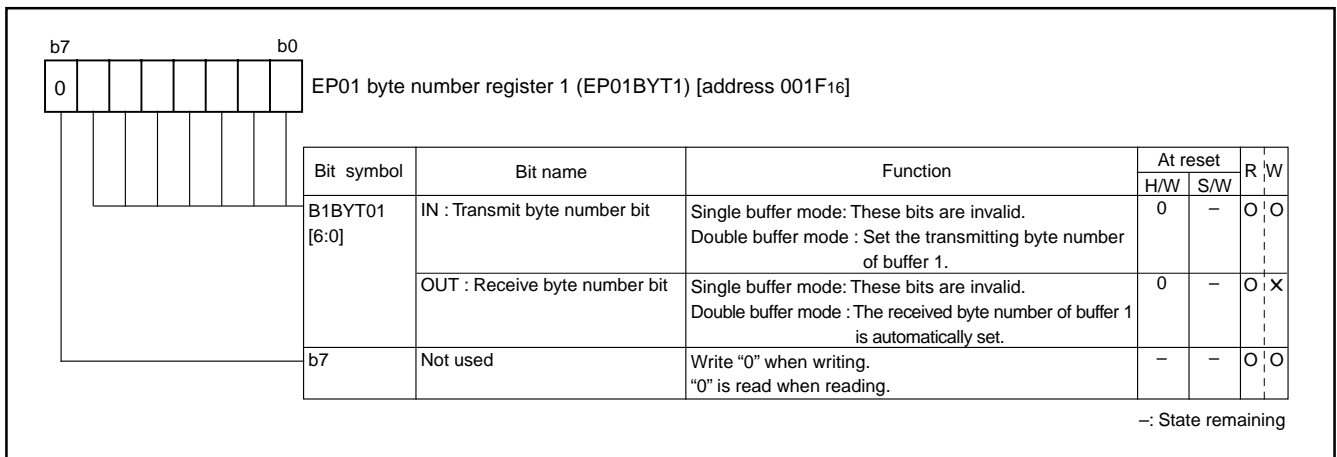


Fig. 51 Structure of EP01 byte number register 1

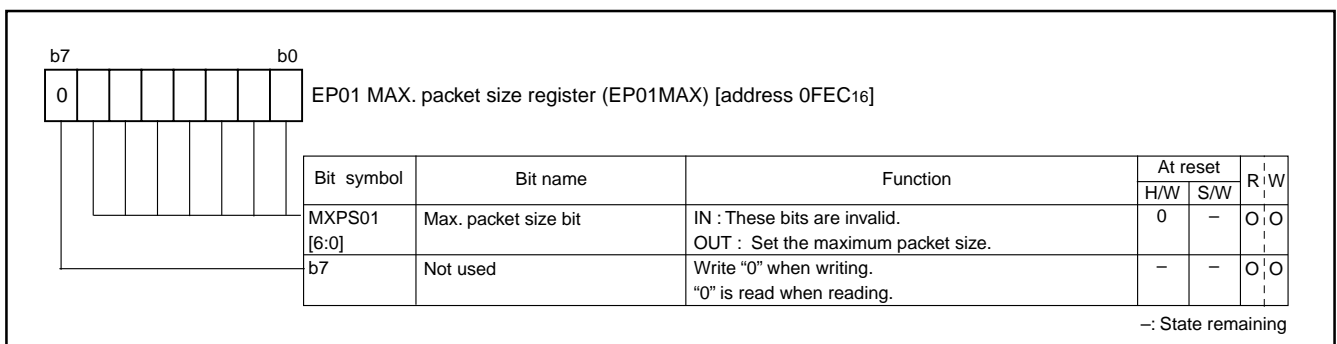


Fig. 52 Structure of EP01 MAX. packet size register

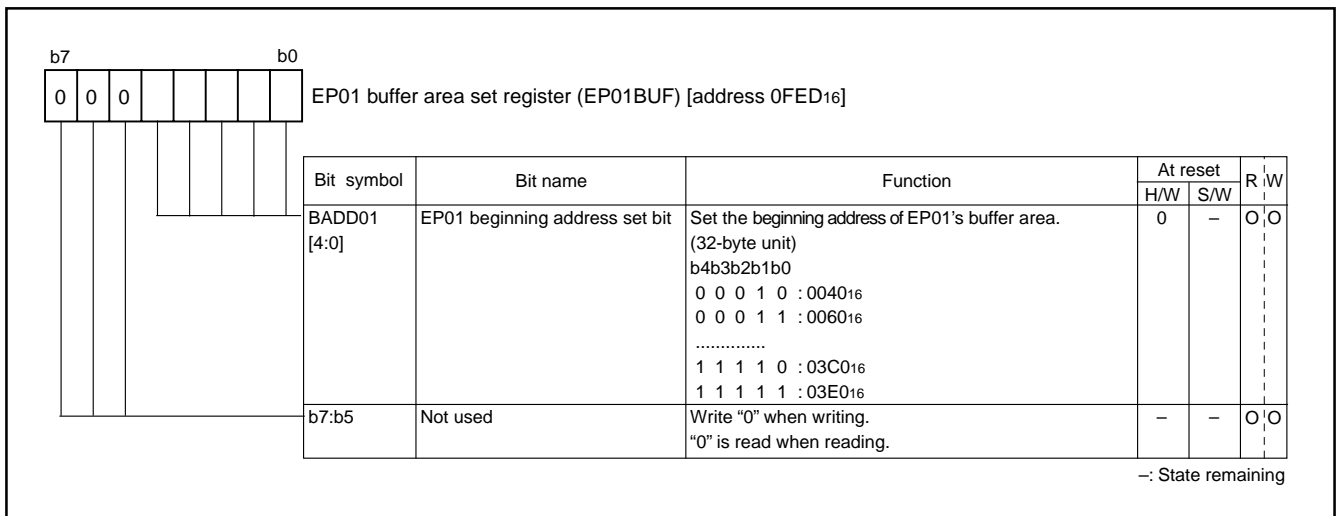


Fig. 53 Structure of EP01 buffer area set register

(3) Endpoint 02

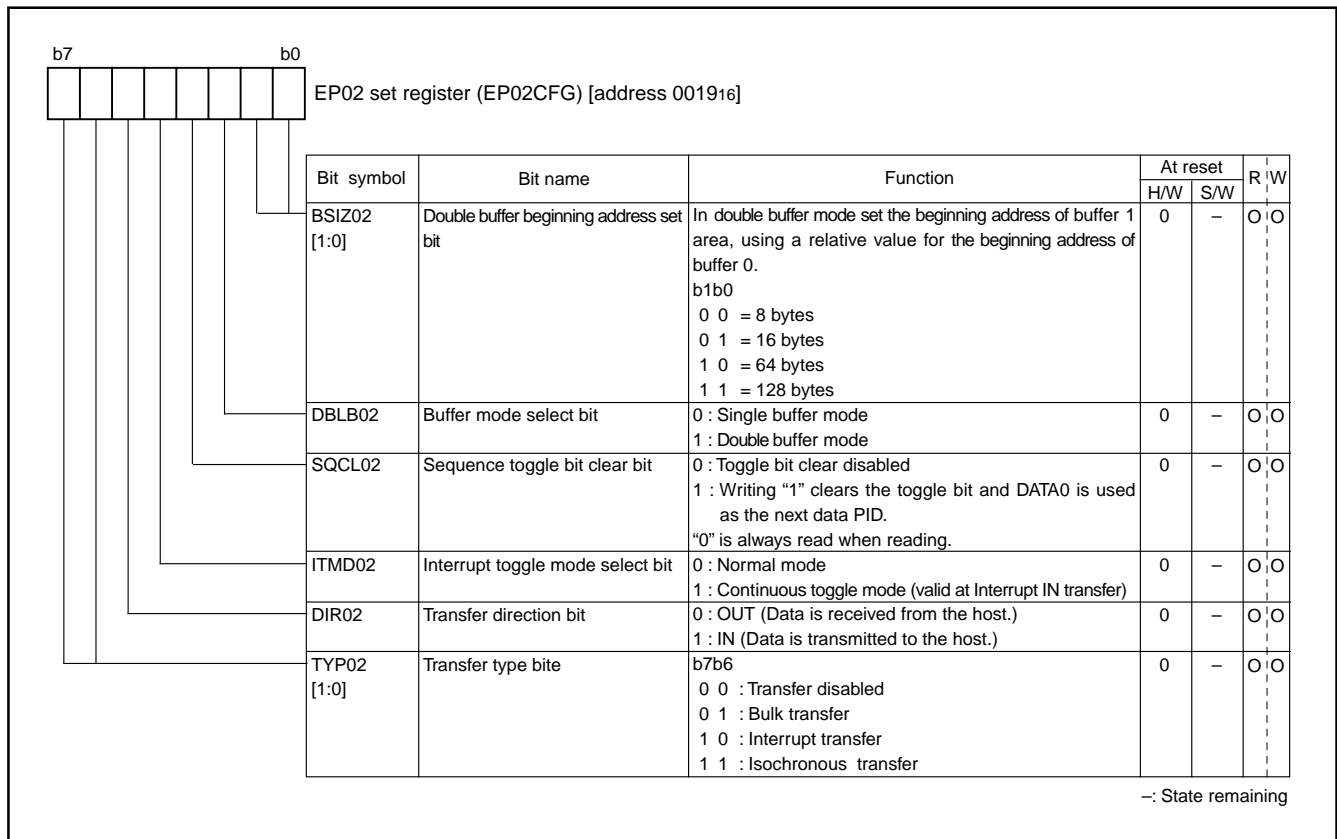


Fig. 54 Structure of EP02 set register

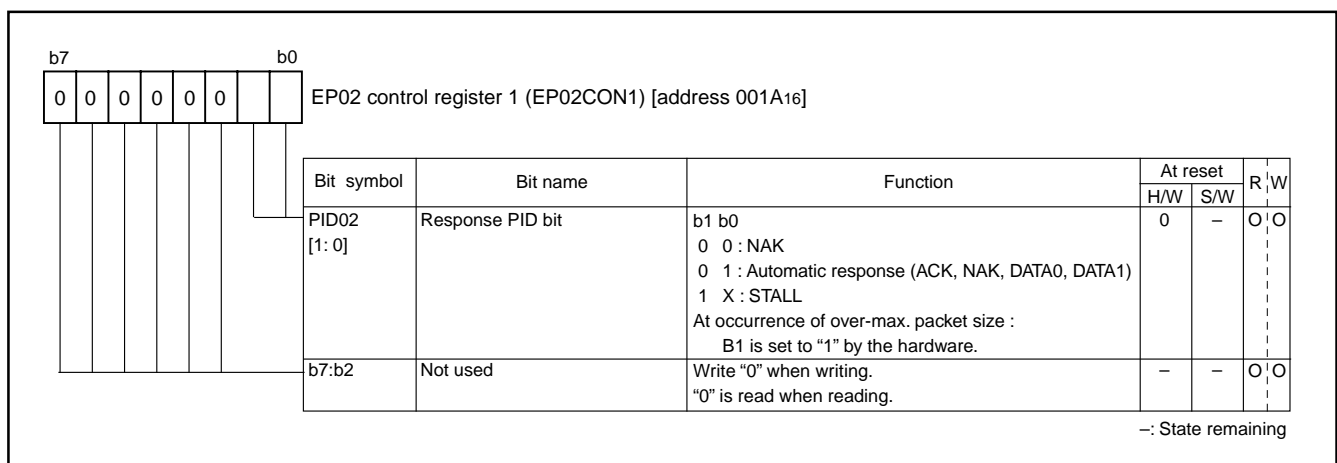


Fig. 55 Structure of EP02 control register 1

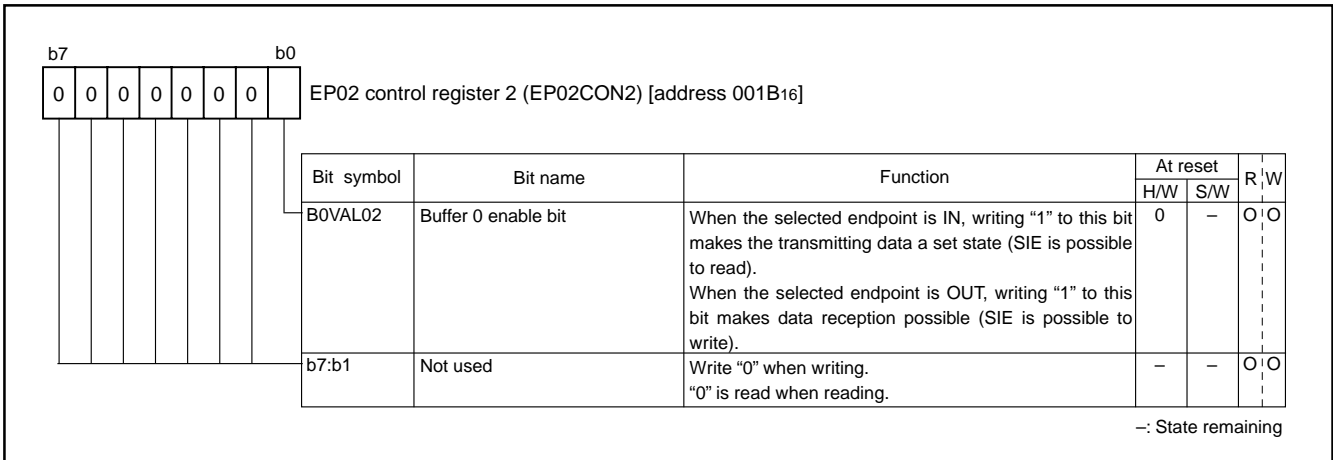


Fig. 56 Structure of EP02 control register 2

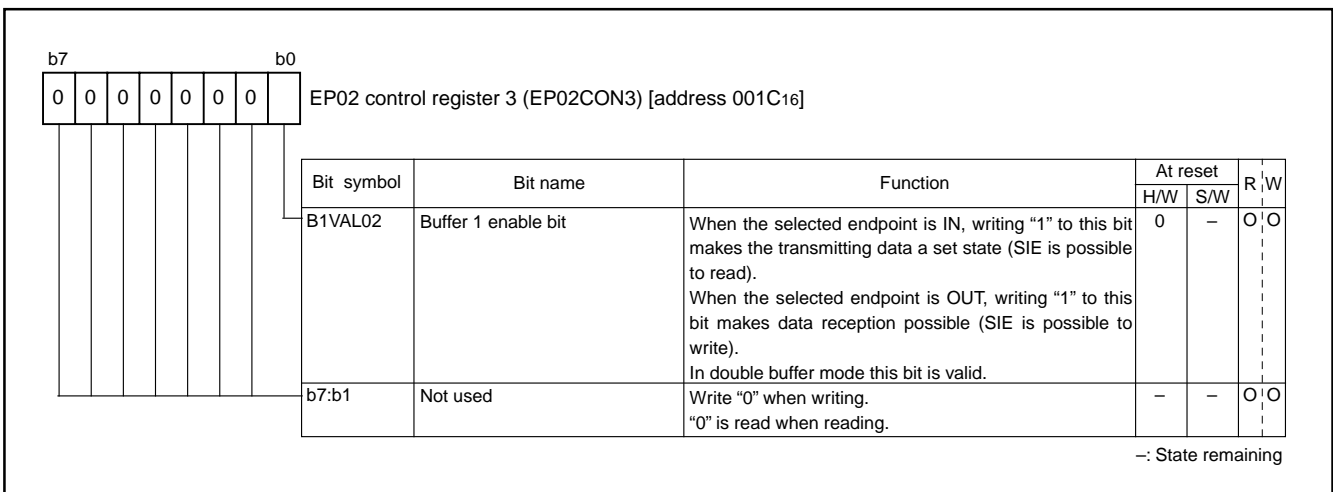


Fig. 57 Structure of EP02 control register 3

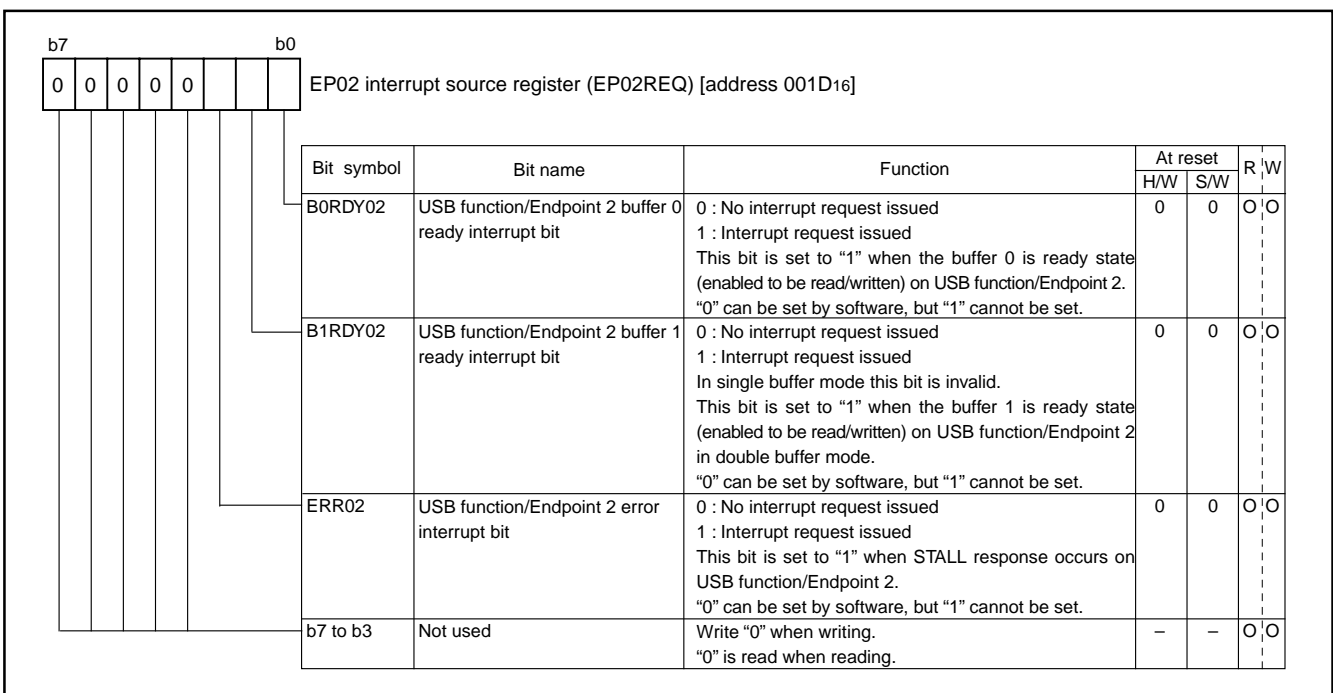


Fig. 58 Structure of EP02 interrupt source register

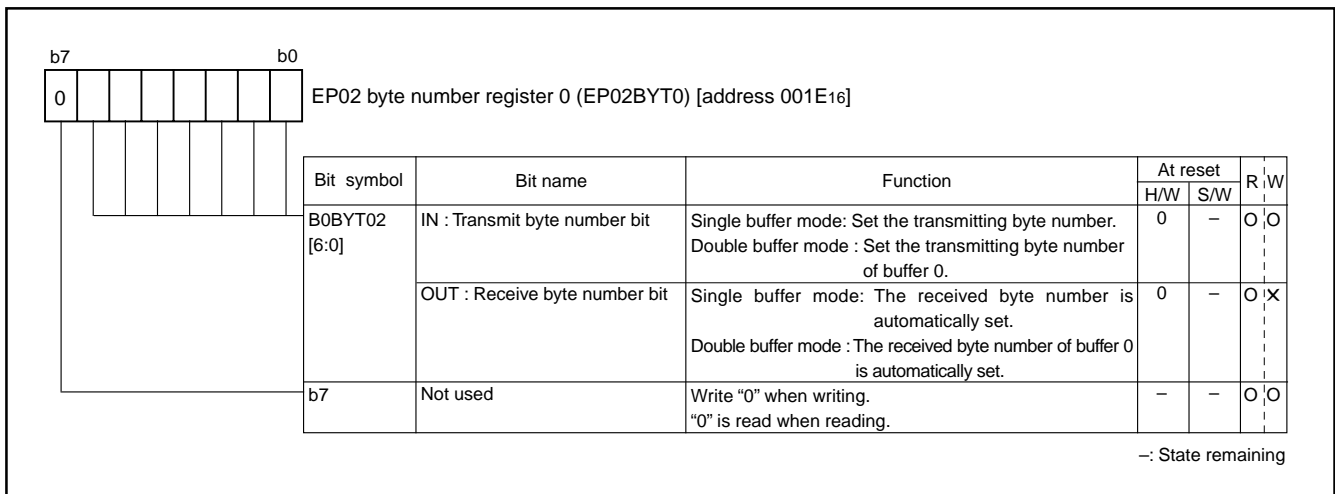


Fig. 59 Structure of EP02 byte number register 0

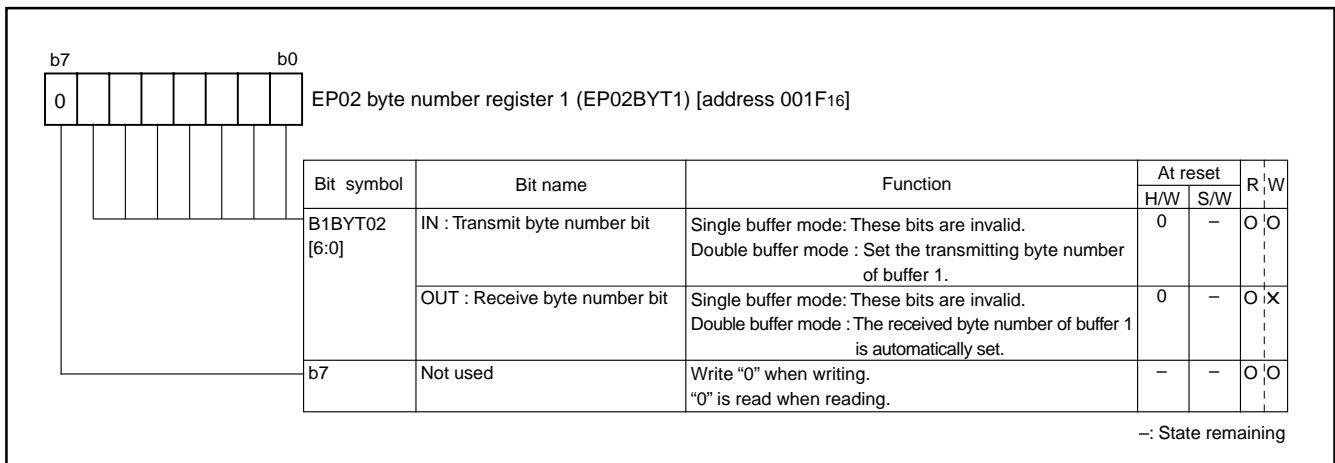


Fig. 60 Structure of EP02 byte number register 1

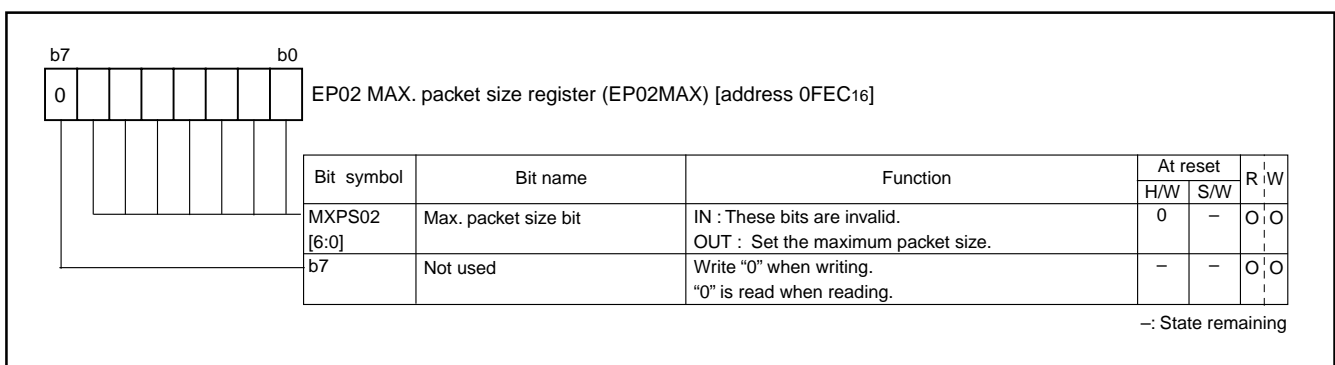


Fig. 61 Structure of EP02 MAX. packet size register

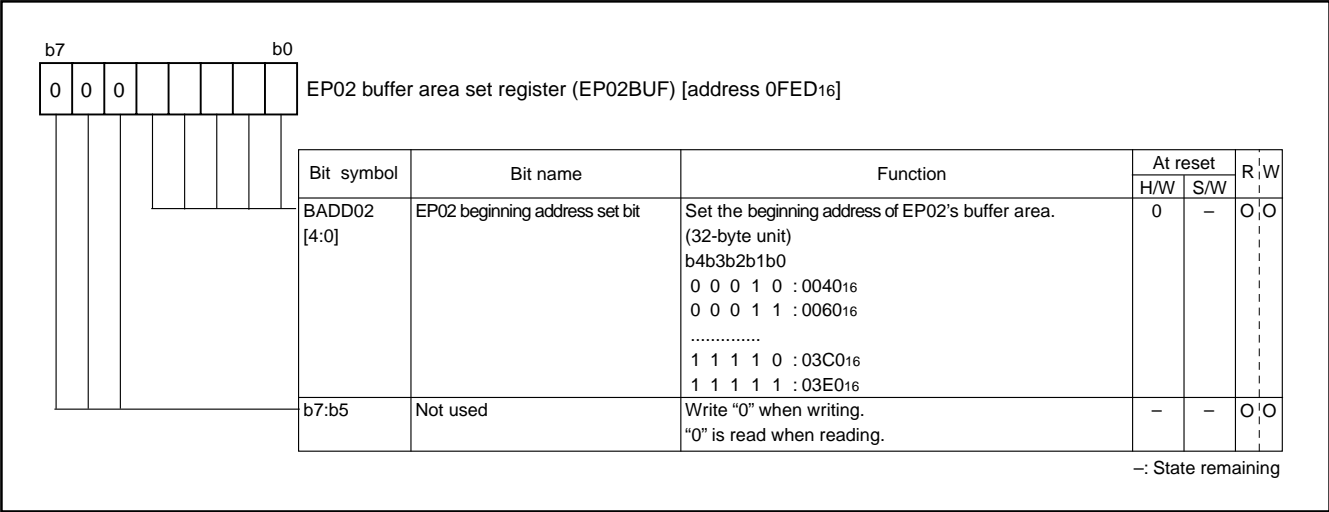


Fig. 62 Structure of EP02 buffer area set register

(4) Endpoint 03

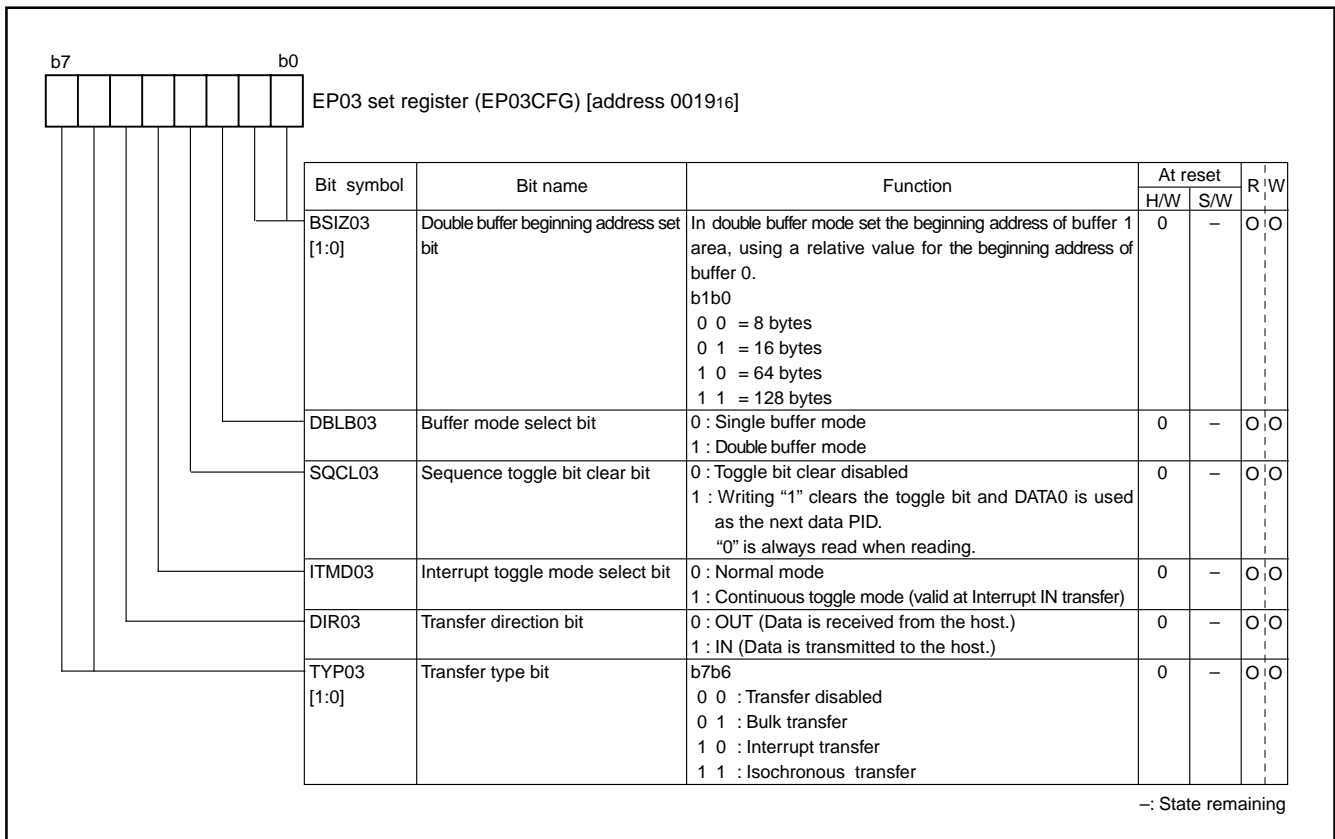


Fig. 63 Structure of EP03 set register

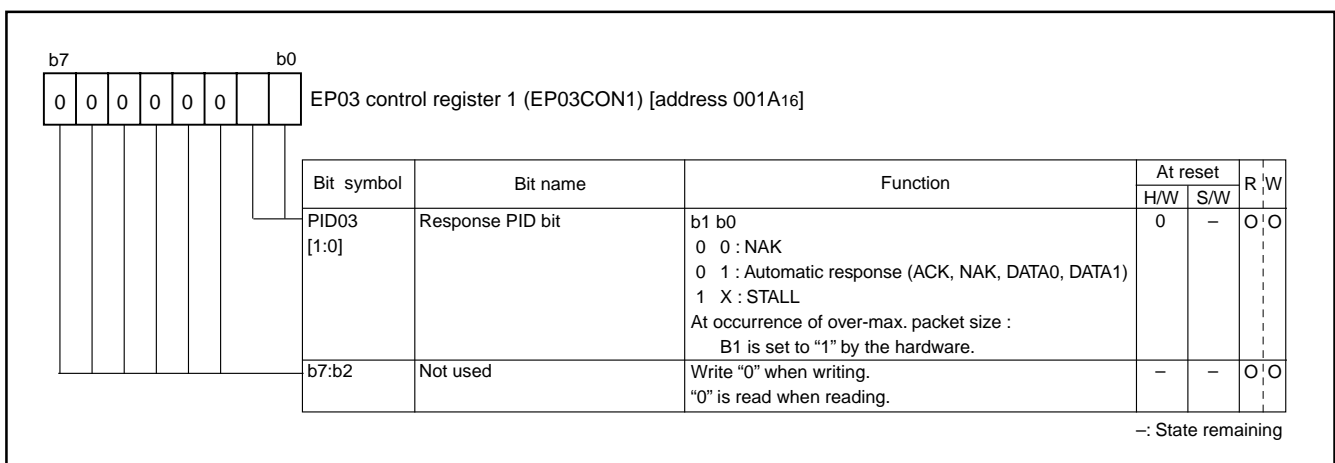


Fig. 64 Structure of EP03 control register 1

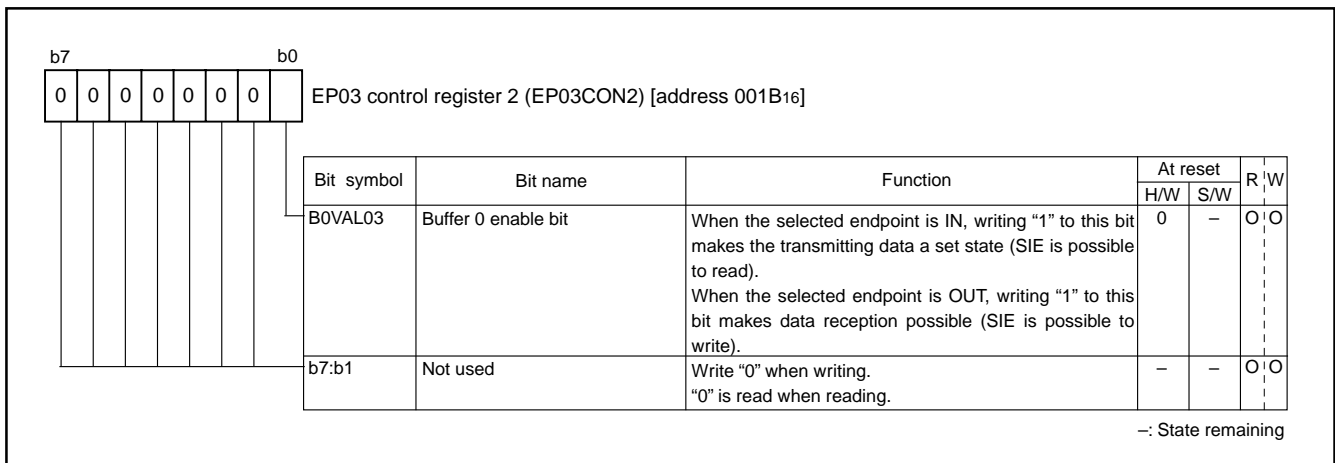


Fig. 65 Structure of EP03 control register 2

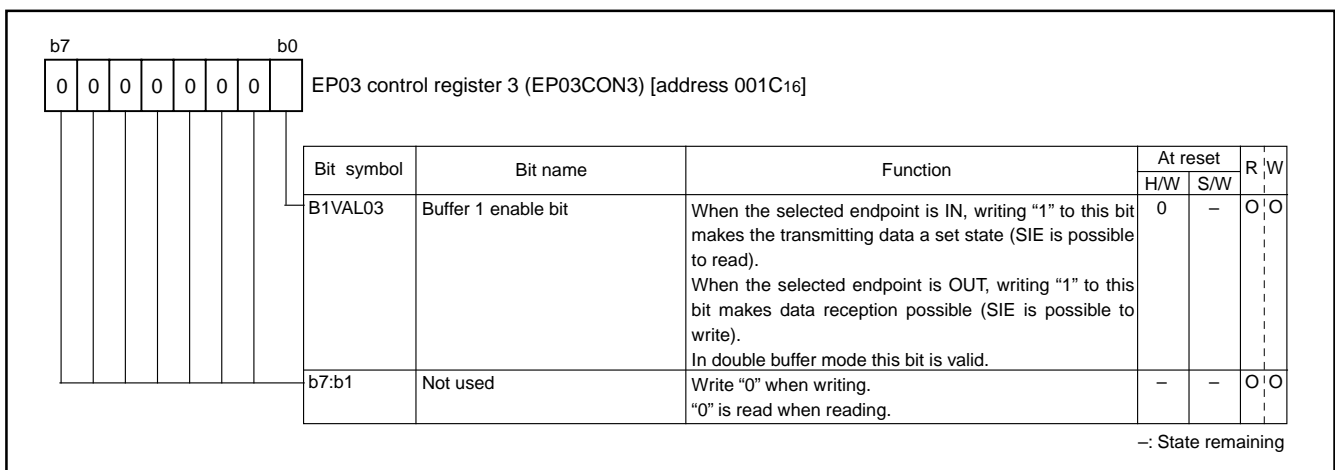


Fig. 66 Structure of EP03 control register 3

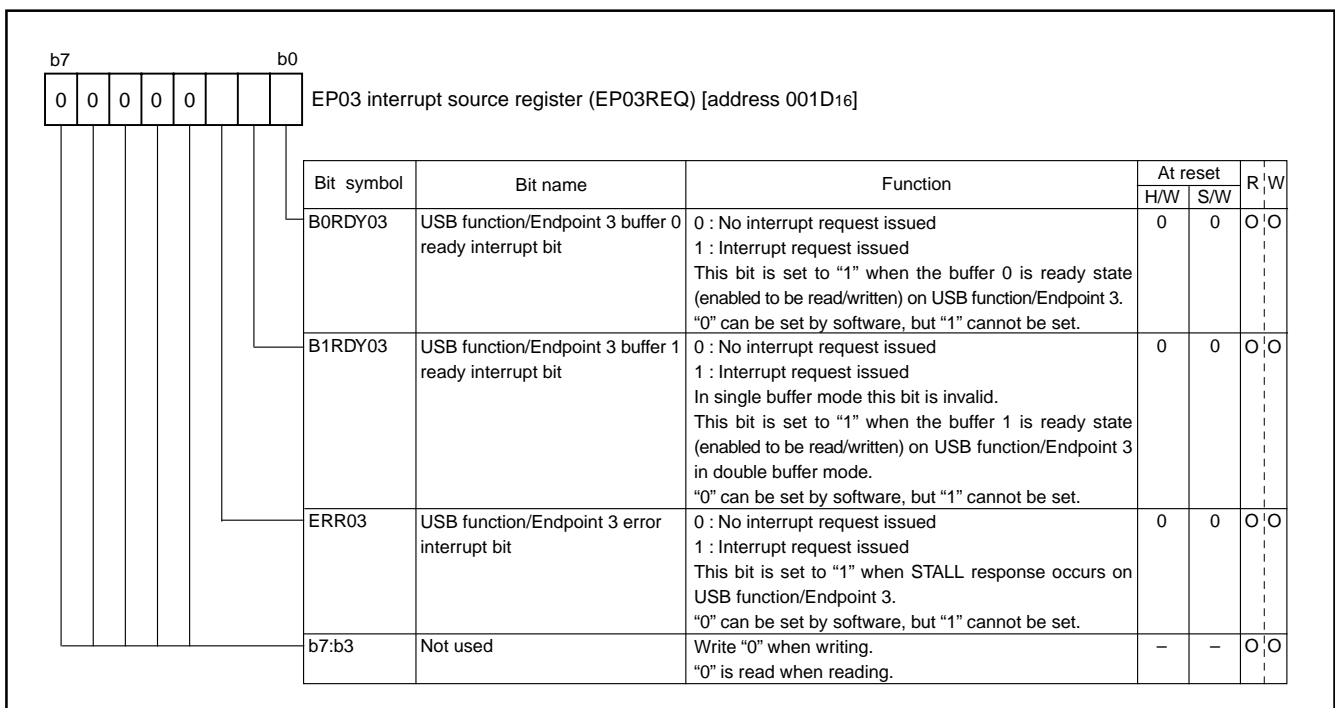


Fig. 67 Structure of EP03 interrupt source register



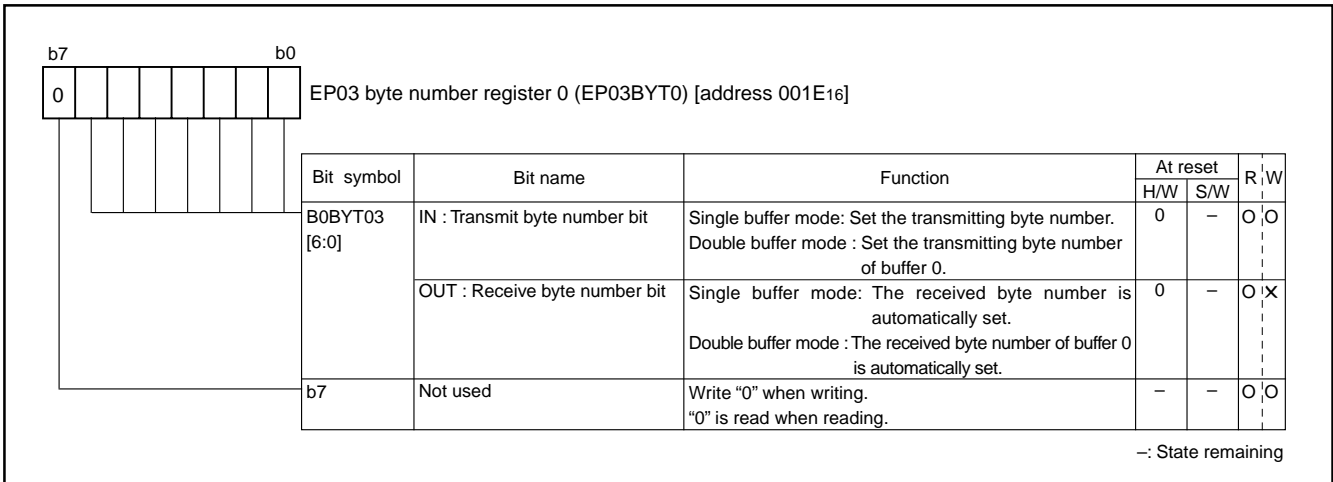


Fig. 68 Structure of EP03 byte number register 0

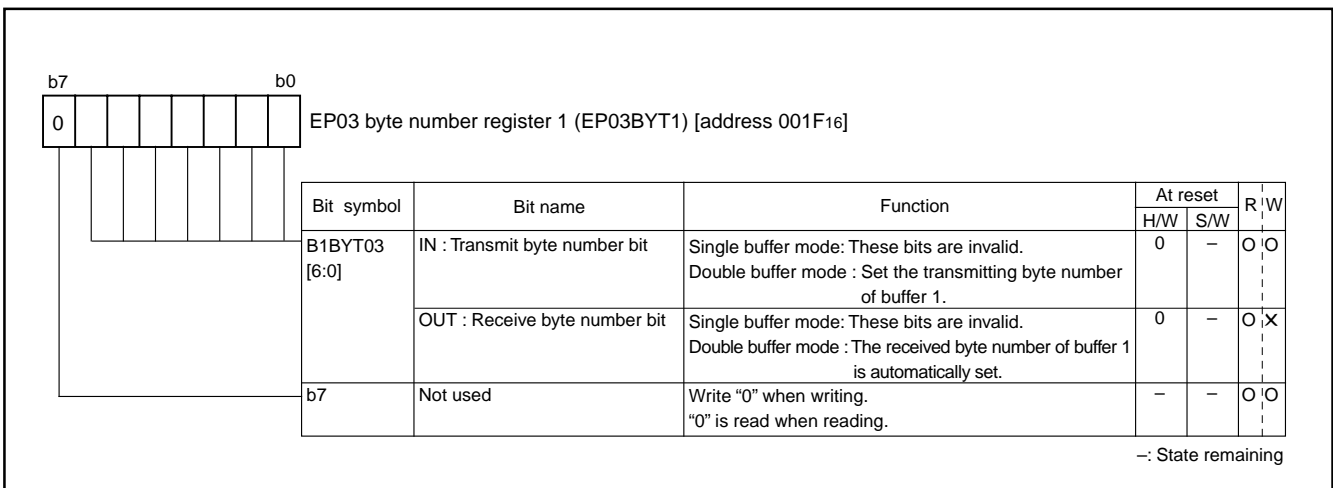


Fig. 69 Structure of EP03 byte number register 1

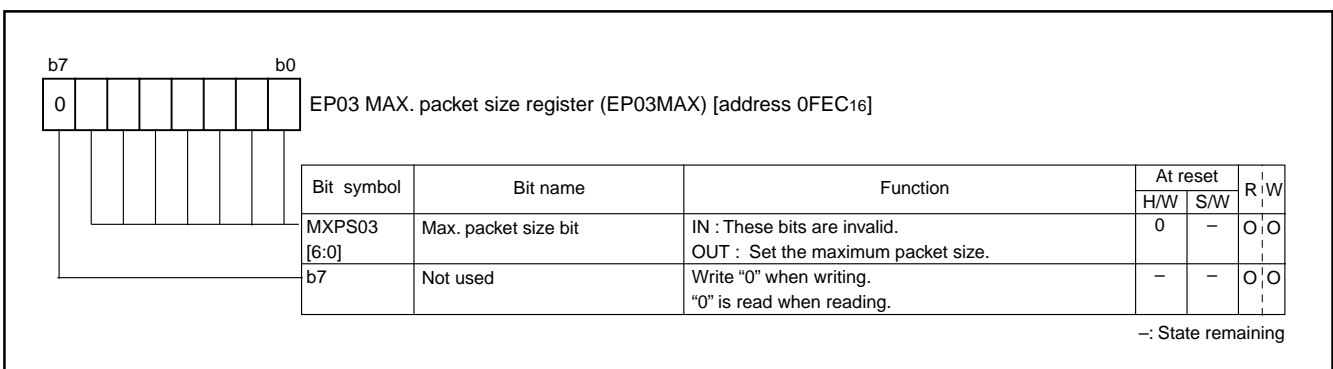


Fig. 70 Structure of EP03 MAX. packet size register

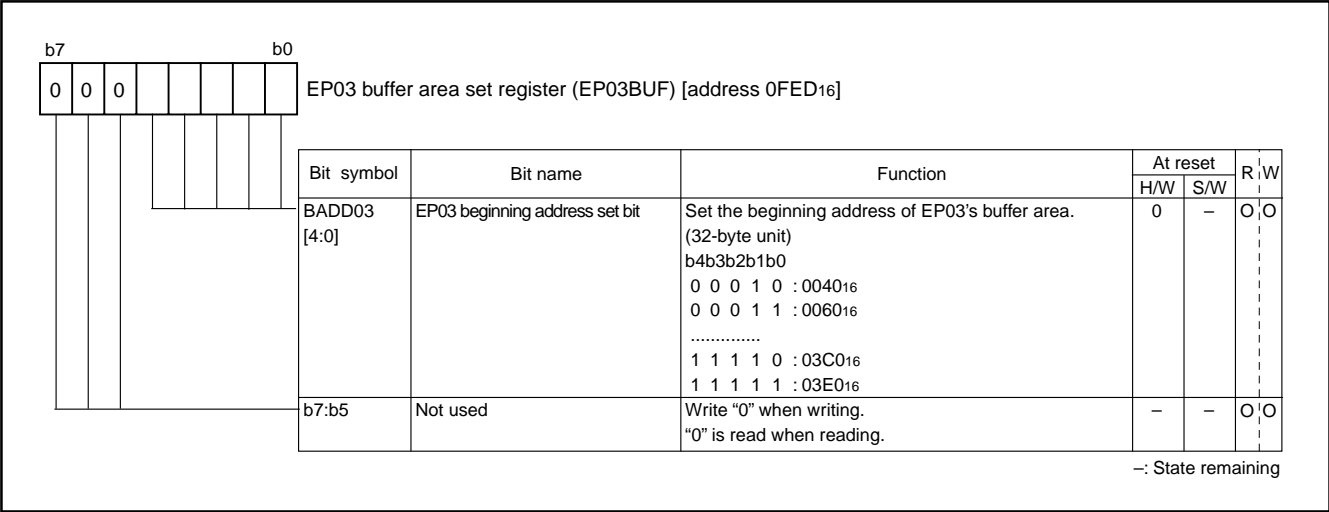


Fig. 71 Structure of EP03 buffer area set register

### EXTERNAL BUS INTERFACE (EXB)

The external bus interface (EXB) controls the data transfer between the external MCU and the 38K0 group's CPU or its

memory (multichannel RAM). The external bus interface is shown below.

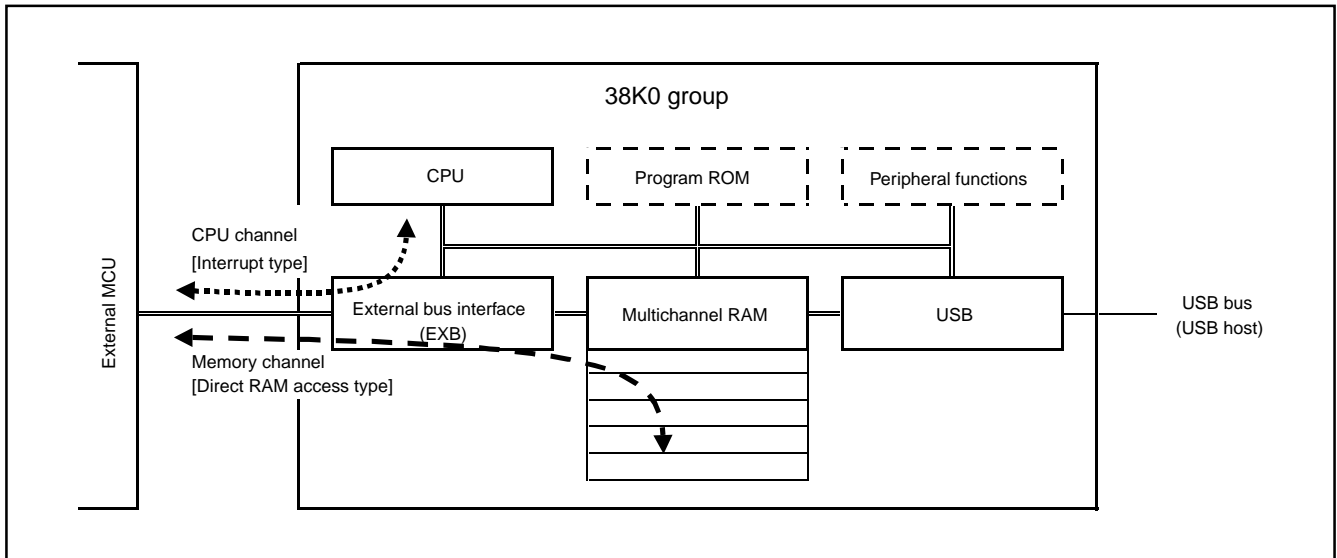


Fig. 72 External bus interface

#### ●CPU channel

It is a data transfer course by the interrupt processing between the external MCU and the 38K0 group's CPU.

#### ●Memory channel

It is a data transfer course by direct RAM access of the memory channel controller between the external MCU and the 38K0 group's memory (multichannel RAM)

#### ●Data transfer of memory channel

When the burst mode is selected with the burst bit of the memory channel operation mode register, data transfer can be carried out at the highest speed. After the external bus interface detects a rise of external read signal/write signal and synchronizes it with the internal clock  $\phi$ , it completes the data transfer between the transmit/receive buffer and the multichannel RAM in two clocks.

However, the waiting time of two clocks at a maximum is generated to access the multichannel RAM in USB being operating because the USB has priority to access.

Therefore, it is necessary to set up the access interval which fills the following timing with the external MCU bus side.

In  $\phi = 8$  MHz, data transfer at about 2 Mbytes/second is possible at a maximum. When there is access simultaneously from the USB, it is about 1.3 Mbytes/second.

In  $\phi = 6$  MHz, data transfer at about 1.5 Mbytes/second is possible at a maximum. When there is access simultaneously from the USB, it is about 1 Mbytes/second.

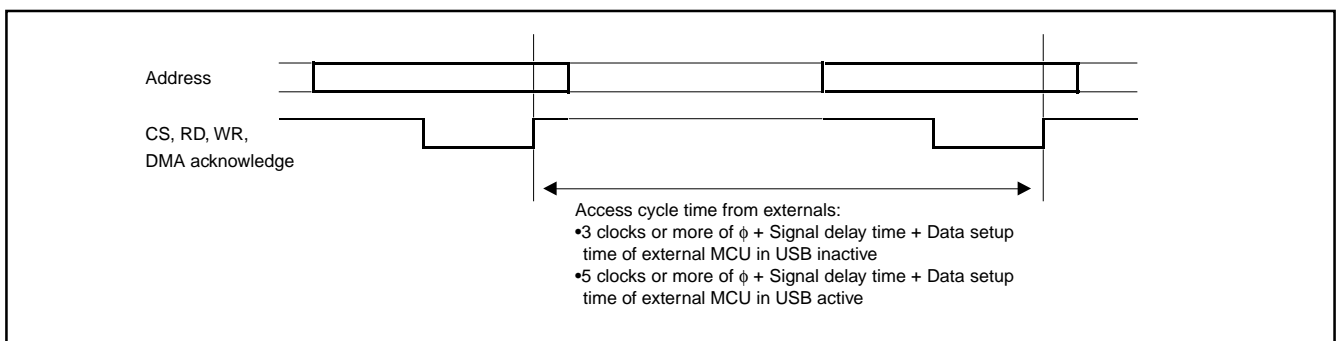


Fig. 73 Data transfer timing of memory channel

**EXB Pin Assignment**

The external bus interface (EXB) pins are shown below.

The 38K0 group can transmit/receive a data to/from an external MCU, using the following signals:

- Control input signal ..... 4 (ExCS, ExA0, ExRD, ExWR)
- Data input/output pin ..... 8 (DQ0 to DQ7)
- Interrupt output signal ..... 1 (ExINT)

Additionally, the DMA interface signal and the buffer status read select signal of 38K0 group can be set up per one by the program.

- Control input signal ..... 3 (ExTC, ExDACK, ExRD, ExA1)
- Interrupt output signal ..... 1 (ExDREQ)

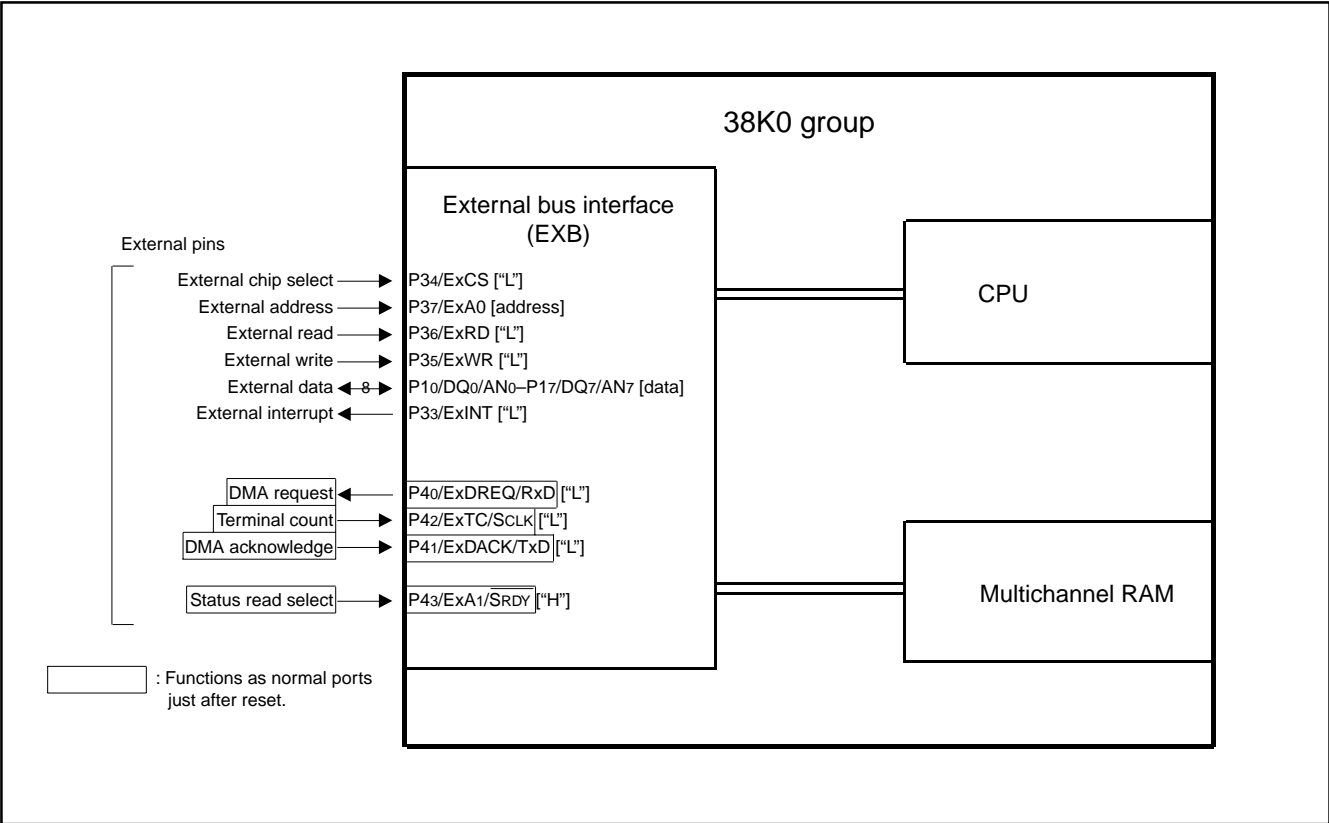


Fig. 74 External bus interface (EXB) pin assignment

### EXB Block Diagram

The block diagram of external bus interface (EXB) is shown below.

The external bus interface (EXB) consists of:

- (1) External I/O interface part
- (2) CPU interface part
- (3) Internal memory interface part
- (4) Transmit/Receive data buffer part

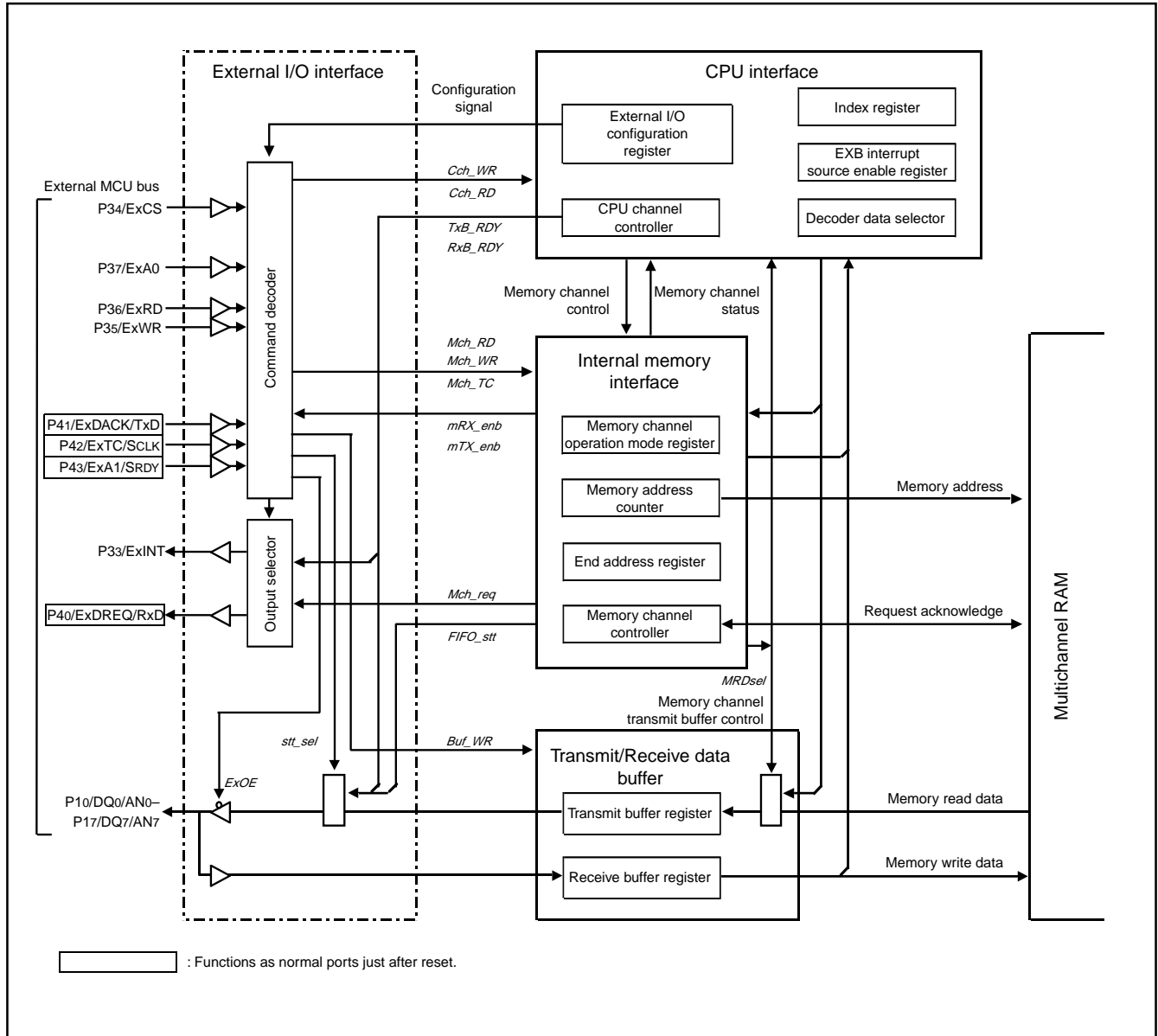


Fig. 75 Block diagram of external bus interface (EXB)

## (1) External I/O Interface Part

The external I/O interface part consists of a command decoder and an output selector. A command decoder generates the following signals to each unit.

### ●CPU interface part

- CPU channel read (Cch\_RD)
- CPU channel write (Cch\_WR)

### ●Internal memory interface part

- Memory channel read (Mch\_RD)
- Memory channel write (Mch\_WR)
- Memory channel terminal count (Mch\_TC)

### ●Transmit/receive data buffer part

- Buffer write (Buf\_WR)

### ●External I/O interface part

- Status selection (stt\_sel)
- Output enable (ExOE)

Access to the CPU channel can be controlled only by setup of external signals.

Access to the memory channel can be controlled by the value of the external I/O configuration register and the state (mRX\_enb, mTX\_enb signals) of the internal memory interface part.

The output selector has the function which selects from the state of CPU channel (TxB\_RDY and RxD\_RDY) and the state of memory channel (Mch\_req) as the signal assigned to P33/ExlNT pin and P40/ExDREQ/RxD pin.

## (2) CPU Interface Part

The CPU interface part consists of the decoder/data selector of the CPU channel, the CPU write register and CPU channel controller

### ●Decoder/data selector of CPU channel

A write operation to the CPU register is performed by generating a write signal for each register with an address decode signal and a write signal.

A read operation from the CPU register is performed by generating an output enable signal of the internal data bus with an module select signal and a read signal and generating a select signal for each register with an address decode signal.

### ●CPU write register

There are three CPU write registers as follows:

- EXB interrupt source enable register
- Index register
- External I/O configuration register

The EXB interrupt source register is a read-only register.

A status signal of the CPU channel controller and a status signal of the memory channel controller in the internal memory interface part are generated.

### ●CPU channel controller

The CPU channel controller generates the following signals, using bits 0 and 1 (RXB\_ENB, TXB\_ENB) of EXB interrupt source enable register.

- Memory channel transmitting buffer control signal (MRD\_sel), generated in the internal memory interface part
- CPU channel command signal (Cch\_RD, Cch\_WR), generated in the external I/O interface part
- Signals RxB\_RDY/RxB\_full and TxB\_RDY/TxB\_empty, generated with read/write signals from the CPU channel

### (3) Internal Memory Interface Part

The internal memory interface part consists of the CPU register and the memory channel controller.

#### ●CPU register

The CPU register consists of the follows:

- Memory channel operation mode register
- Memory address counter
- End address register

The CPU can set the beginning address into the memory address counter when the memory channel operation enable bit (MC\_ENB) of EXB interrupt source enable register is "0". When this bit is "1", the write operation from the CPU is invalid and each access from the external bus causes count-up operation.

#### ●Memory channel controller

The CPU register consists of the follows:

- Main sequencer
- Internal memory request signal generating circuit
- External memory channel request signal generating circuit
- Address end detection circuit
- Terminal end input processing circuit

### (4) Transmit/Receive Data Buffer Part

The transmit/receive data buffer part consists of the 8-bit transmit buffer register (TXBUF) and the 8-bit receive buffer register (RXBUF).

Both CPU channel and memory channel use the same transmit buffer register/receive buffer register to transfer a data to an external MCU bus.

### (5) External Pin

The external bus interface has the following pins to connect with an external MCU bus.

- Chip select ..... P34/ExCS
- Address ..... P37/ExA0
- Data ..... P10/DQ0/AN0 to P17/DQ7/AN7
- Read ..... P36/ExRD
- Write ..... P35/ExWR
- Interrupt request ..... P33/ExINT

It also has the following pins to connect with an external DMAC. Each pin can be programmed for an ordinary port function or a DMA interface pin function.

- DMA request ..... P40/ExDREQ/RxD
- DMA acknowledgment ..... P41/ExDACK/TxD
- Terminal count ..... P42/ExTC/SCLK

It also has the status read select pin (P43/ExA1/ $\overline{\text{SRDY}}$  pin) to confirm a ready status of the data buffer from an external MCU bus. This pin functions as a port just after reset. The status read select function can be set by a program.

- Status read select ..... P43/ExA1/ $\overline{\text{SRDY}}$

#### ●CPU channel: Communication with 38K0 group CPU

When a read/write operation is performed from an external MCU bus in address signal ExA0 = "H", the interrupt is generated and the 38K0 group CPU can confirm its access. The 38K0 group CPU judges the interrupt source and it starts a data transmission/reception with an external MCU bus.

#### ●Memory channel: Communication with 38K2 group memory multichannel RAM

When a read/write operation is performed from an external MCU bus in address signal ExA0 = "L", access to the multichannel RAM is performed. Then an address of the multichannel RAM is made by the external bus interface and it is increased at each access completion. Consequently, FIFO access is performed.

Even if a read/write operation is performed in DACK = "L" instead of ExCS = "L" and ExA0 = "L", FIFO access to the multichannel RAM is performed.

The beginning address and the end address must be set by the CPU in advance.

●P33/ExINT pin

Any one of the following signals for this pin can be selected:

- TxB\_RDY (transmit buffer ready) output
- RxB\_RDY (receive buffer ready) output
- Mch\_req (memory channel request) output

Either TxB\_RDY or RxB\_RDY is normally selected. The memory channel request is for an access request signal to the memory channel.

In a small system, a data transfer processing to the internal memory is performed in the interrupt routine. According to that situation, the 38K0 group has the function automatically to switch an interrupt factor attached on the interrupt pin by program.

●P40/ExDREQ/RxD pin

This pin is a port at the initial state. Which signal can be set by program.

- RxB\_RDY (receive buffer ready) output
- Mch\_req (memory channel request) output

Mch\_req of DMAC is normally selected. The output method of the memory channel request signal depends on the burst bit (BURST) of memory channel operation mode register. When the burst bit is "0", this signal is periodically output at each 1-byte transfer. (See Figures 87 and 90.)

When the burst bit is "1", this signal is continuously output while the memory address counter is counting from the beginning address to the end address (See Figures 88 and 91.)

●P41/ExDACK/TxD pin

This pin is a port at the initial state. The DMA acknowledge signal can be set by program.

The DMA acknowledge signal DACK = "L" is the same state as that of CS = "L" and A0 = "L". Access to multichannel RAM is started by a rise of read signal or write signal which is set during this term.

**Note:** If the DMA acknowledge signal and the chip select signal are simultaneously active (DACK = "L" and CS = "L"), also set the address signal A0 to "L". If A0 is "H", the memory channel and the CPU channel are activated simultaneously and it might cause some error.

●P42/ExTC/SCLK pin

This pin is a port at the initial state. The terminal count signal can be set by program.

If the terminal count signal is set at one bus cycle while a memory channel operation write is being performed, the 38K0 group confirms that its bus cycle is the write cycle of the last data and sets the memory channel status bits to "112", and the interrupt is generated and the memory channel operation ends even if the memory address counter has not reached the end address.

The CPU can obtain the last address where the data is written by reading out the value of memory address counter. (See Figures 87 and 88.)



## EXB Register List

The EXB register list is shown below.

Address	Register Name	SYMBOL	EXB SFR								
			bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
0030 <sub>16</sub>	EXB interrupt source enable register	EXBICON	X		X		X		MC_ENB	TXB_ENB	RXB_EMB
0031 <sub>16</sub>	EXB interrupt source register	EXBIREQ	X		X		X		MC_STS[1:0]	TXB_EMPTY	RXB_FULL
0033 <sub>16</sub>	EXB index register	EXBINDEX	0	0	0	0	0	INDEX[2:0]			
0034 <sub>16</sub>	Register window 1 (low)	EXBREG1	LOW_WIN[7:0]								
0035 <sub>16</sub>	Register window 2 (high)	EXBREG2	HIGH_WIN[7:0]								

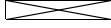
 : Not used  
 0 : "0" fixed

Fig. 76 EXB related registers (1)

### •EXB interrupt source enable register

This register enables/disables access from an external bus and an internal interrupt.

### •EXB interrupt source register

This register indicates the state of CPU channel's transmit/receive buffer register and the memory channel. The same value can be read out from the external MCU bus by using the buffer status read select signal (A1 pin = "H").

### •EXB index register/Register windows 1, 2

The accessible register is switched by treating addresses 0034<sub>16</sub> and 0035<sub>16</sub> as a register window depending on the value of EXB index register at address 0033<sub>16</sub>.

Index	low high	Register Name	SYMBOL	EXB SFR							
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00 <sub>16</sub>	low	External I/O configuration register	EXBCFGL	X		X		A1_CTR	INT_CTR[2:0]		EXB_CTR
	high		EXBCFGH	X		X		TC_CTR	DAK_CTR[1:0]	DRQ_CTR[1:0]	
01 <sub>16</sub>	low	Transmit/Receive buffer register	RXBUF/TXBUF	At CPU read : RXBUF[7:0] At CPU write : TXBUF[7:0]							
	high		—	X							
02 <sub>16</sub>	low	Memory channel operation mode register	MCHMOD	X		X		BURST		MC_DIR[1:0]	
	high		—	X							
03 <sub>16</sub>	low	Memory address counter	MEMADL	IM_A[7:0]							
	high		MEMADH	0	0	0	0	0	IM_A[10:8]		
04 <sub>16</sub>	low	End address register	ENDADL	END_A[7:0]							
	high		ENDADH	0	0	0	0	0	END_A[10:8]		

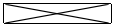
 : Not used  
 0 : "0" fixed

Fig. 77 EXB related registers (2)

### •External I/O configuration register

This register selects the function of each pin.

### •Transmit/Receive buffer register

This register consists of the receive buffer register (RXBUF) and the transmit buffer register (TXBUF)

### •Memory channel operation mode register

This register sets the operation mode of the memory channel.

### •Memory address counter

This is a counter to set the beginning address which FIFO accesses. This register is increased by access from the external MCU bus.

### •End address register

This register is to set the end address which FIFO accesses.

## EXB Related Registers

The EXB related registers are shown below.

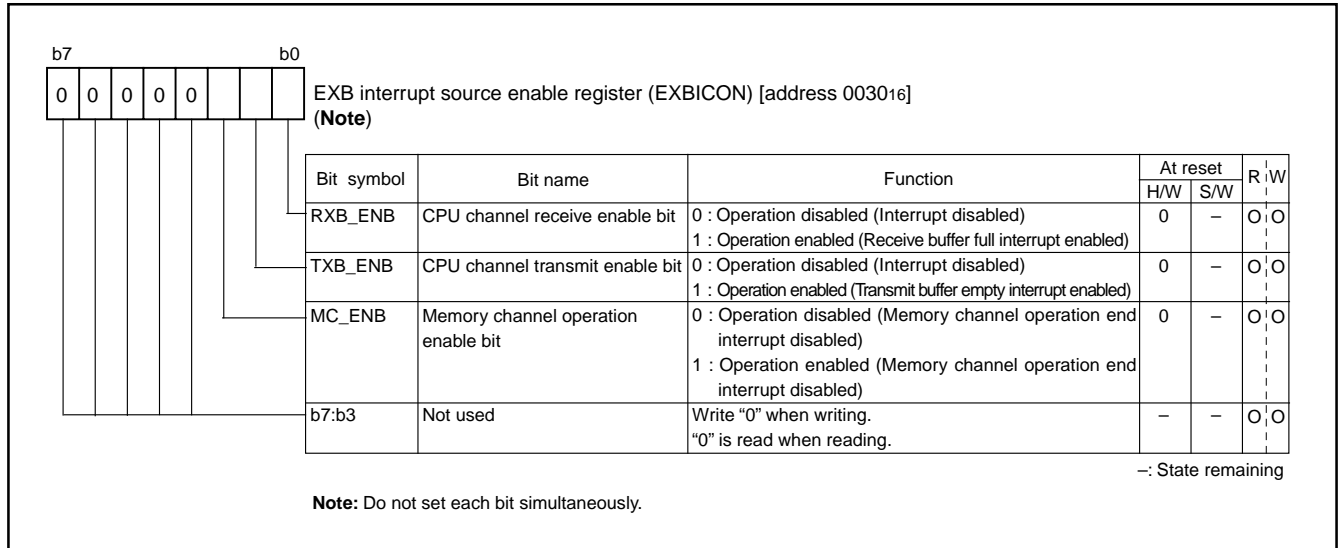


Fig. 78 Structure of EXB interrupt source enable register

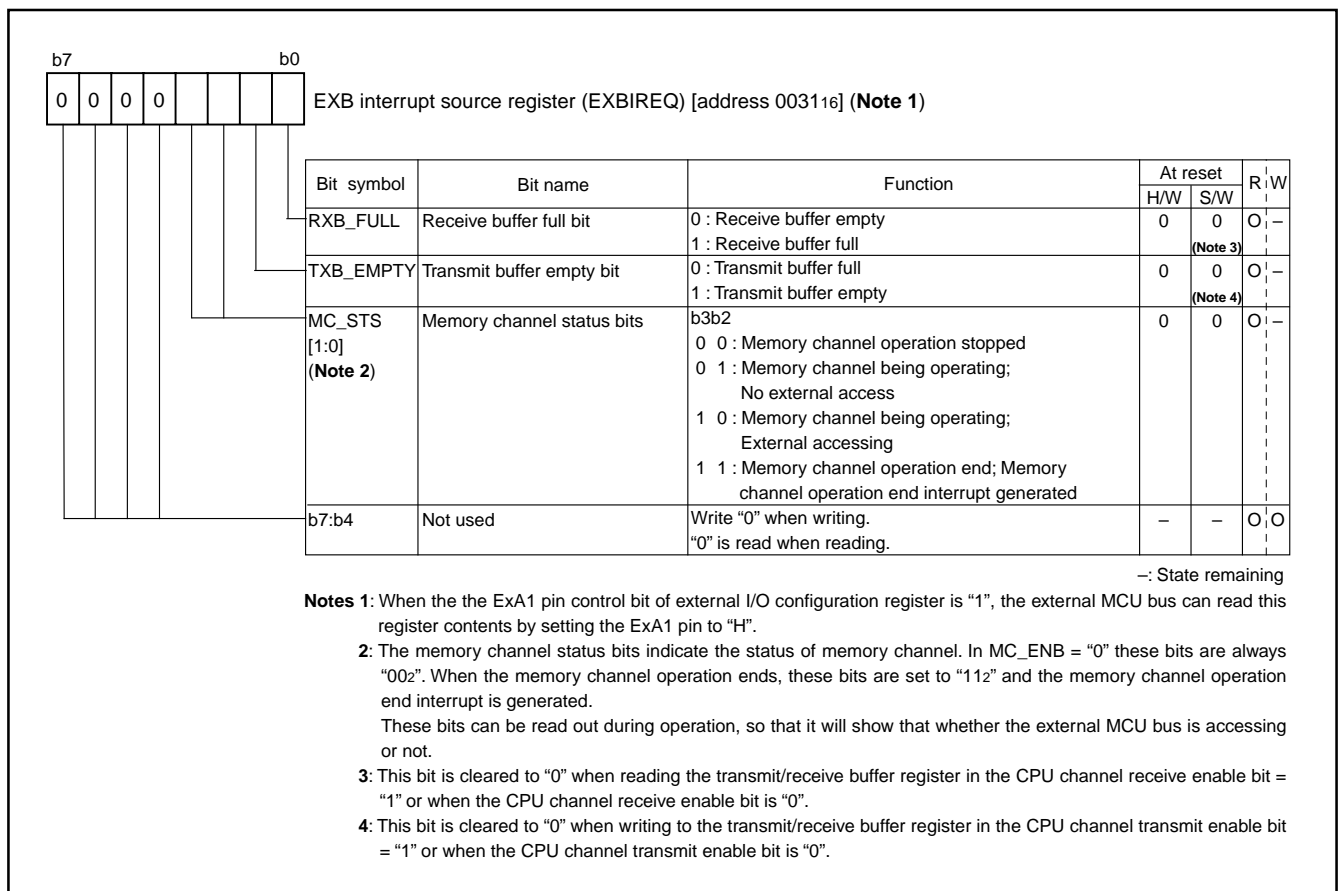


Fig. 79 Structure of EXB interrupt source register

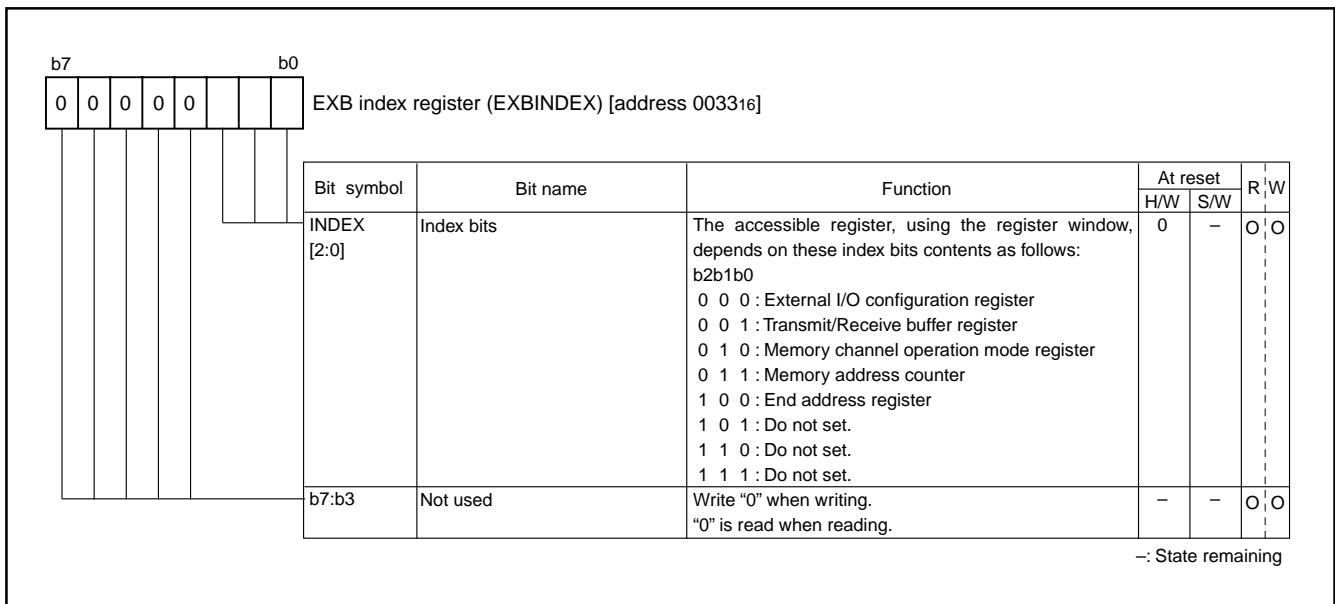


Fig. 80 Structure of EXB index register

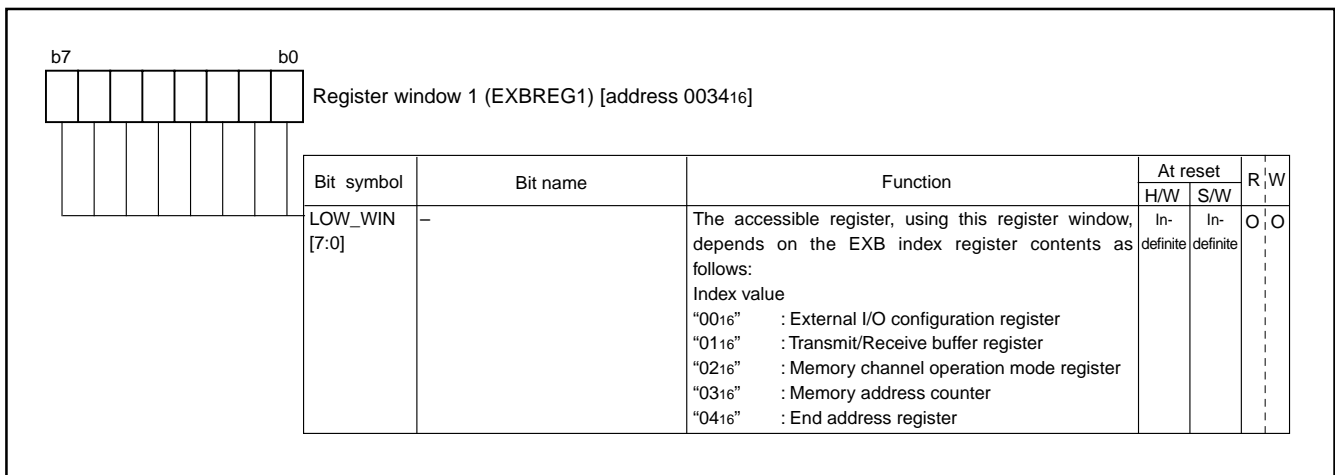


Fig. 81 Structure of Register window 1

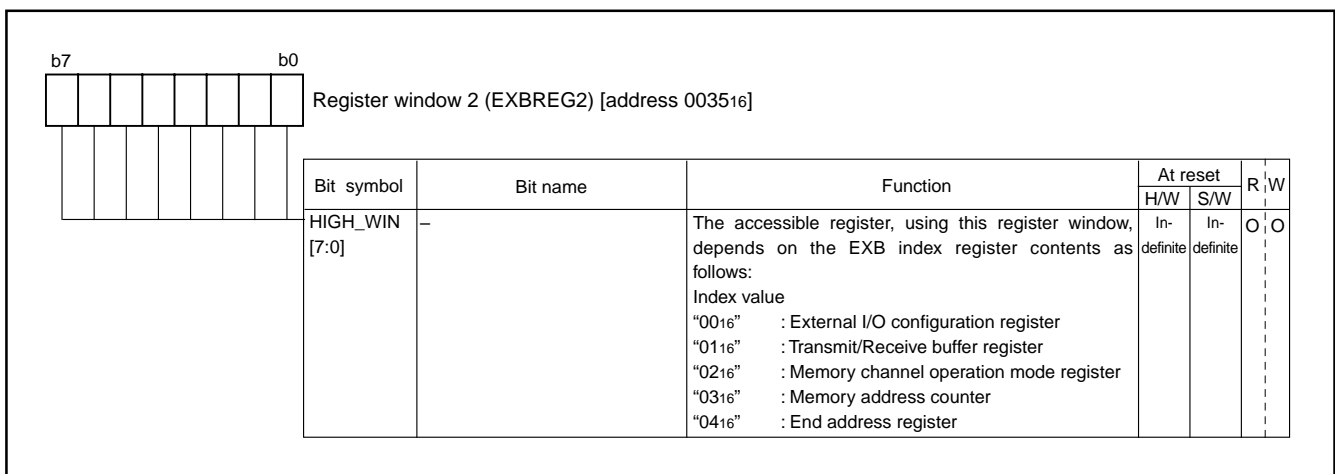


Fig. 82 Structure of Register window 2

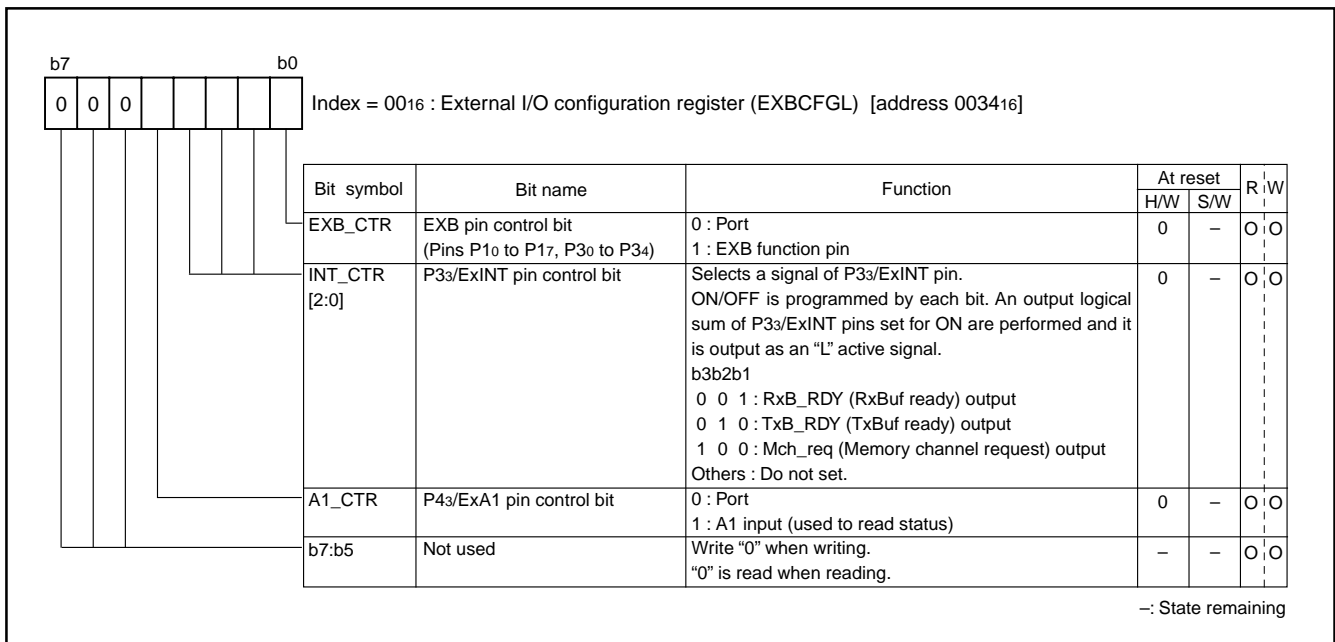


Fig. 83 Index00[low]; Structure of External I/O configuration register

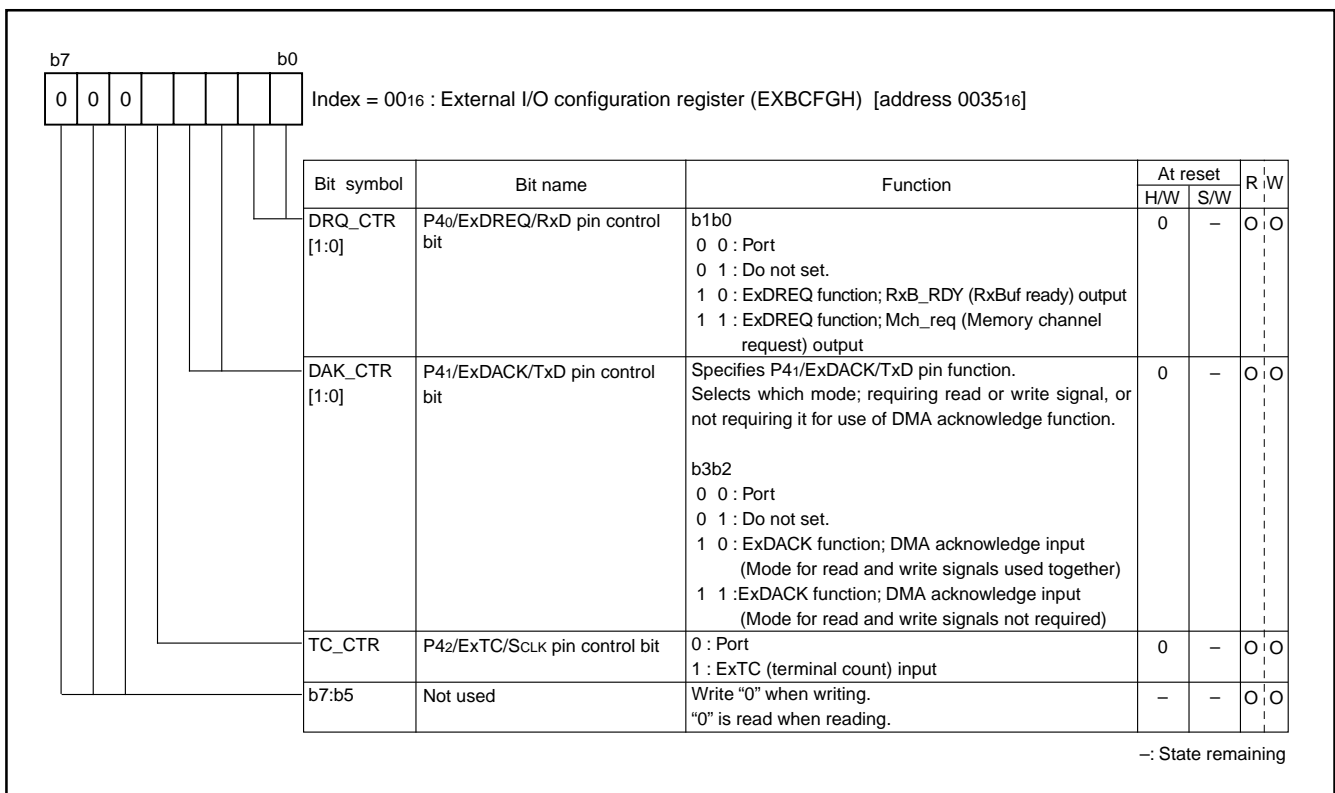


Fig. 84 Index00[high]; Structure of External I/O configuration register

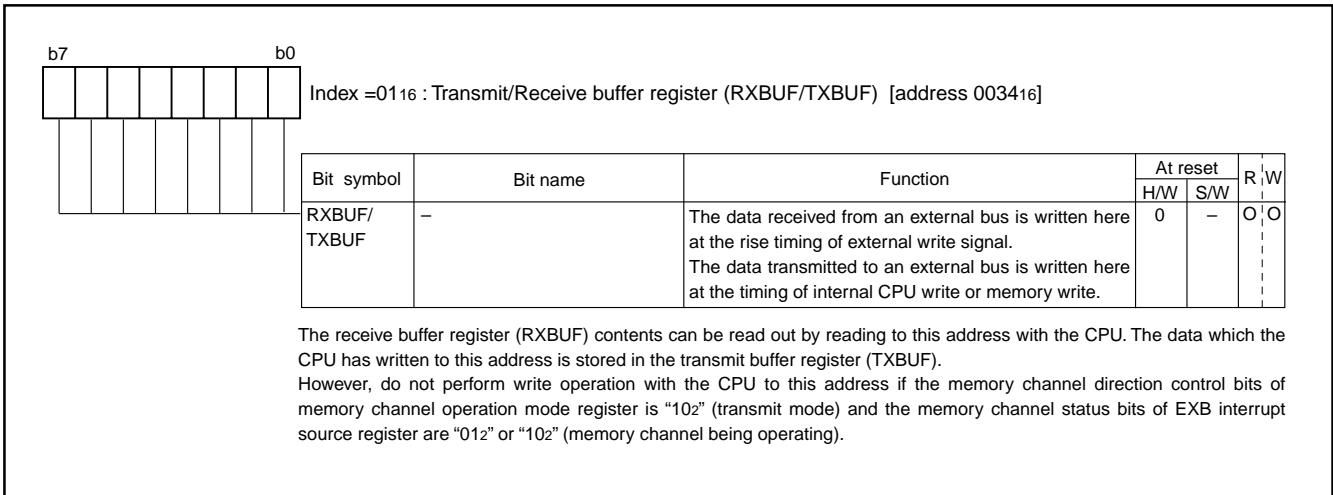


Fig. 85 Index01[low]; Structure of Transmit/Receive buffer register

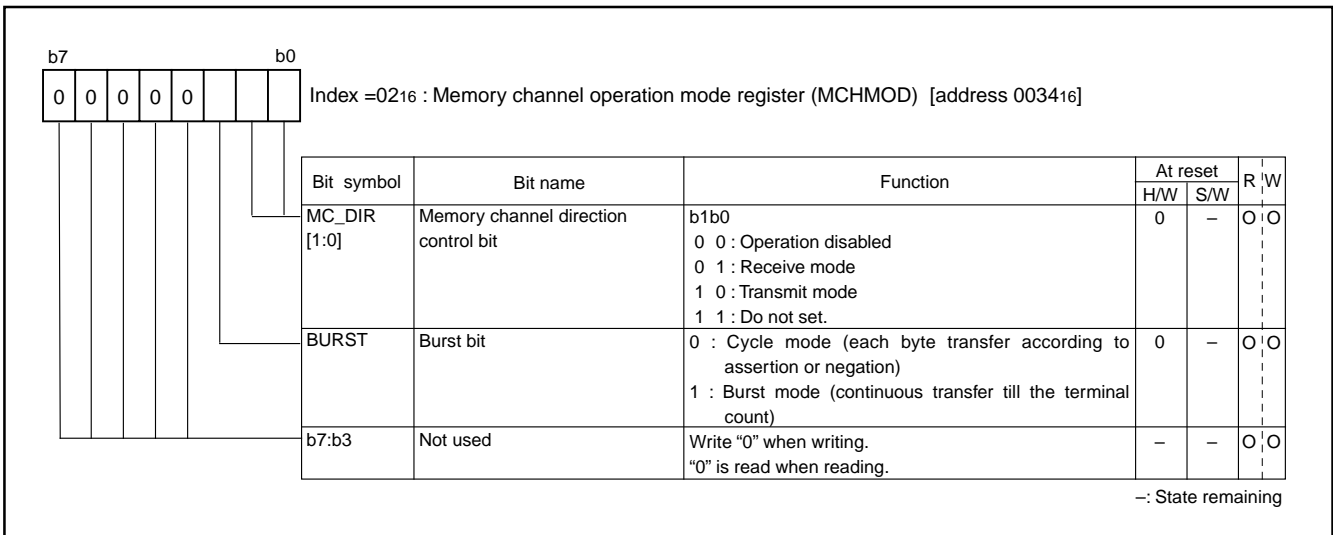


Fig. 86 Index02[low]; Structure of Memory channel operation mode register

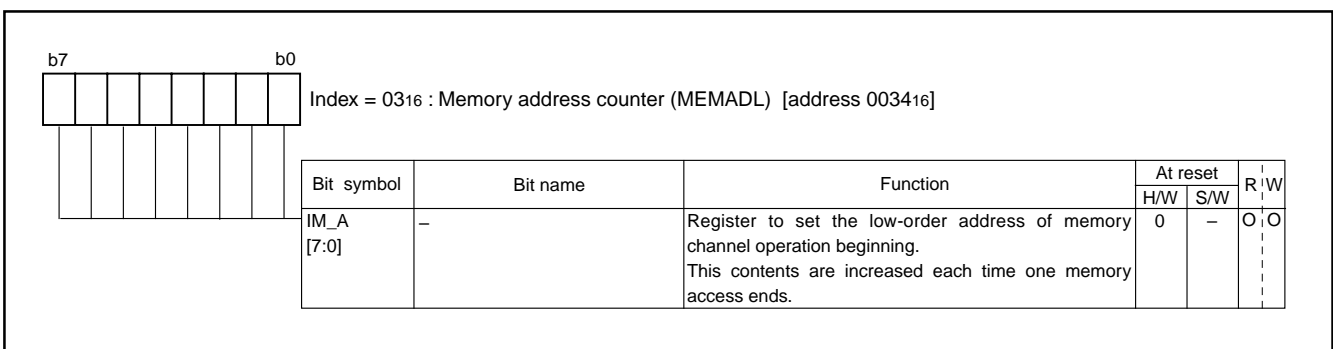


Fig. 87 Index03[low]; Structure of Memory address counter

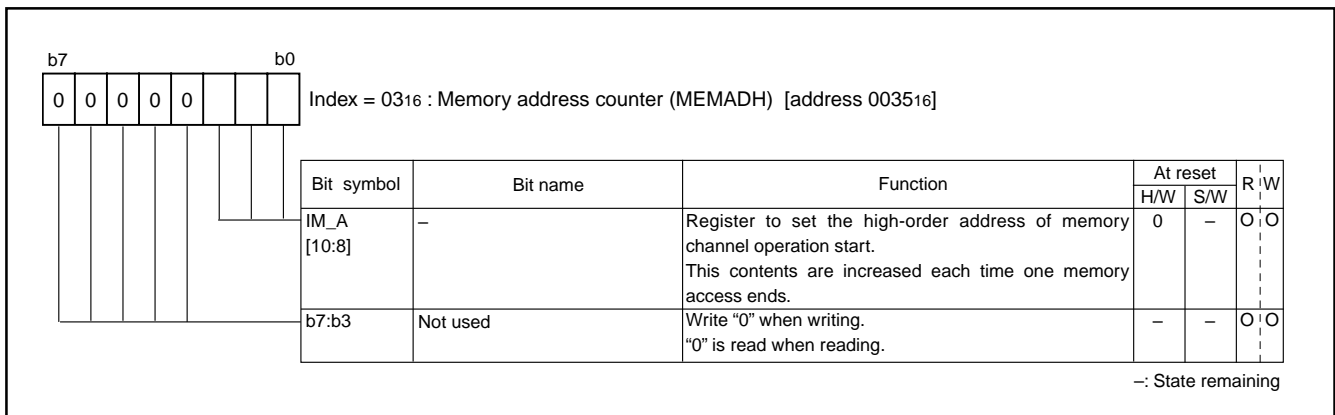


Fig. 88 Index03[high]; Structure of Memory address counter

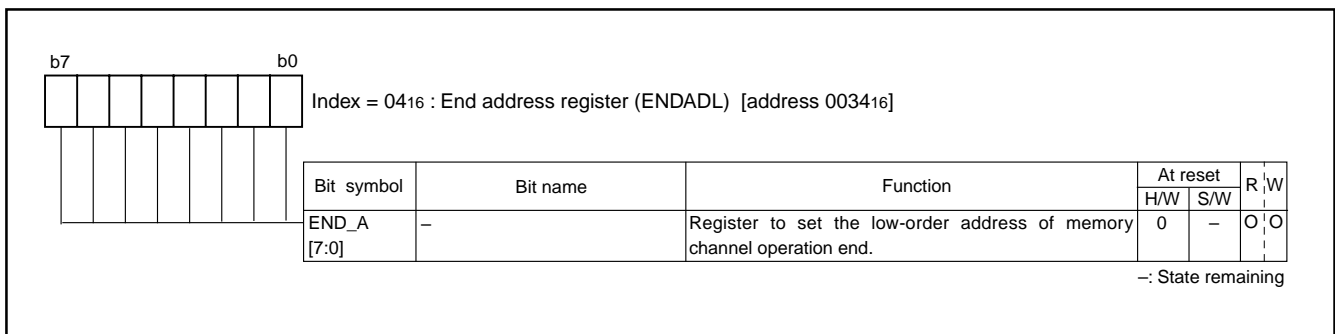


Fig. 89 Index04[low]; Structure of End address register

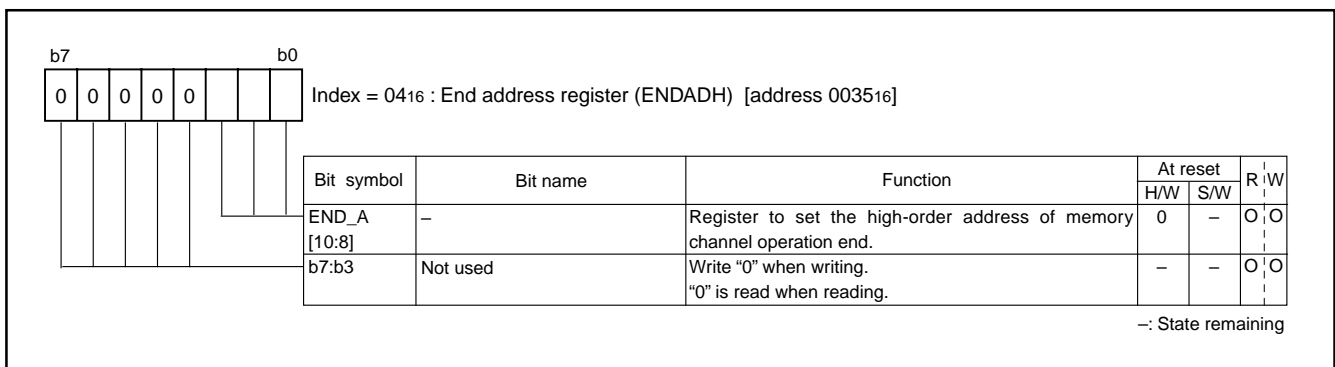


Fig. 90 Index04[high]; Structure of End address register

### EXB Operation Timing Diagram (1) CPU Channel Receiving Operation

CPU channel receiving operation is shown below.

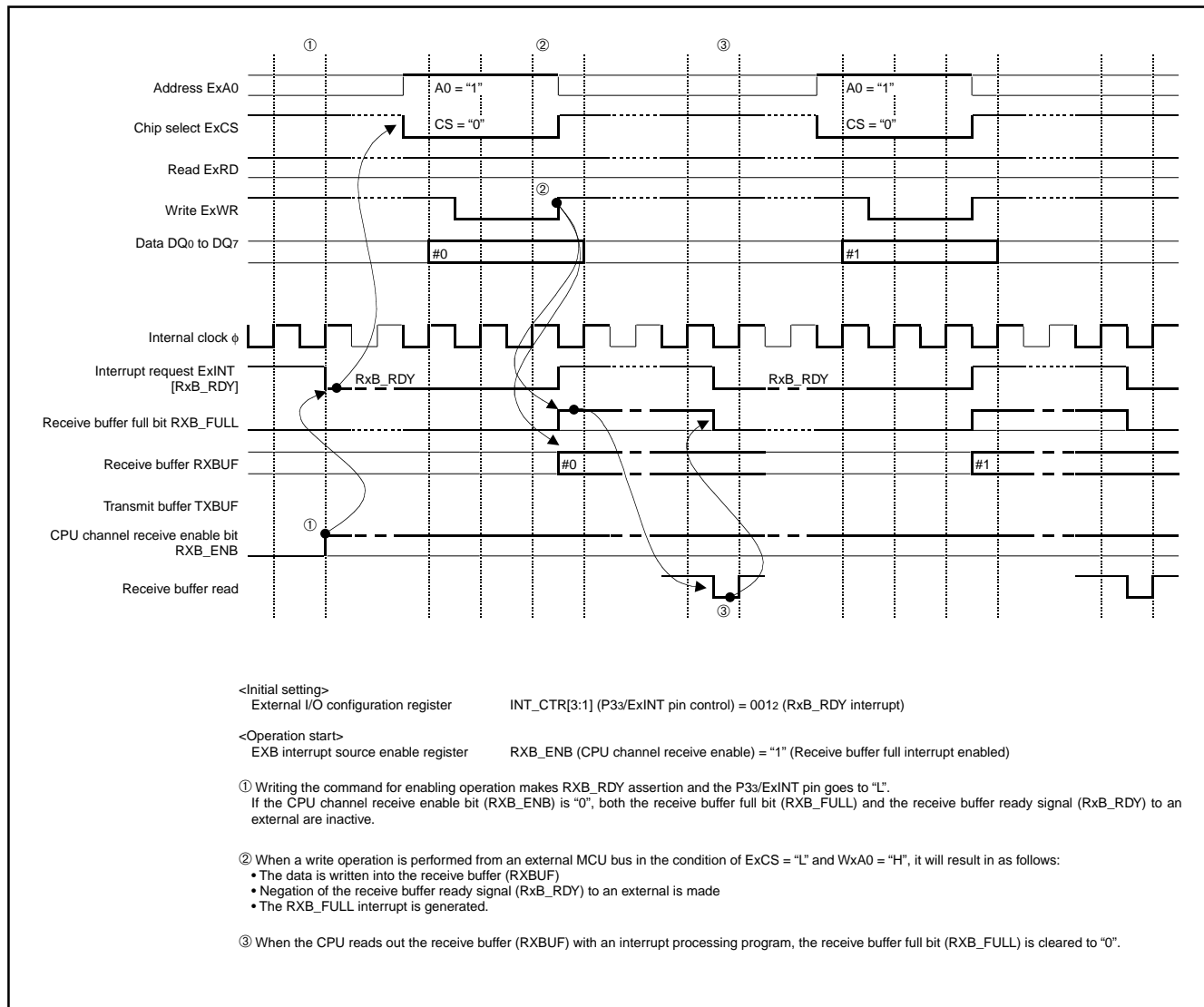


Fig. 91 CPU channel receiving operation

## (2) CPU Channel Transmitting Operation

CPU channel transmitting operation is shown below.

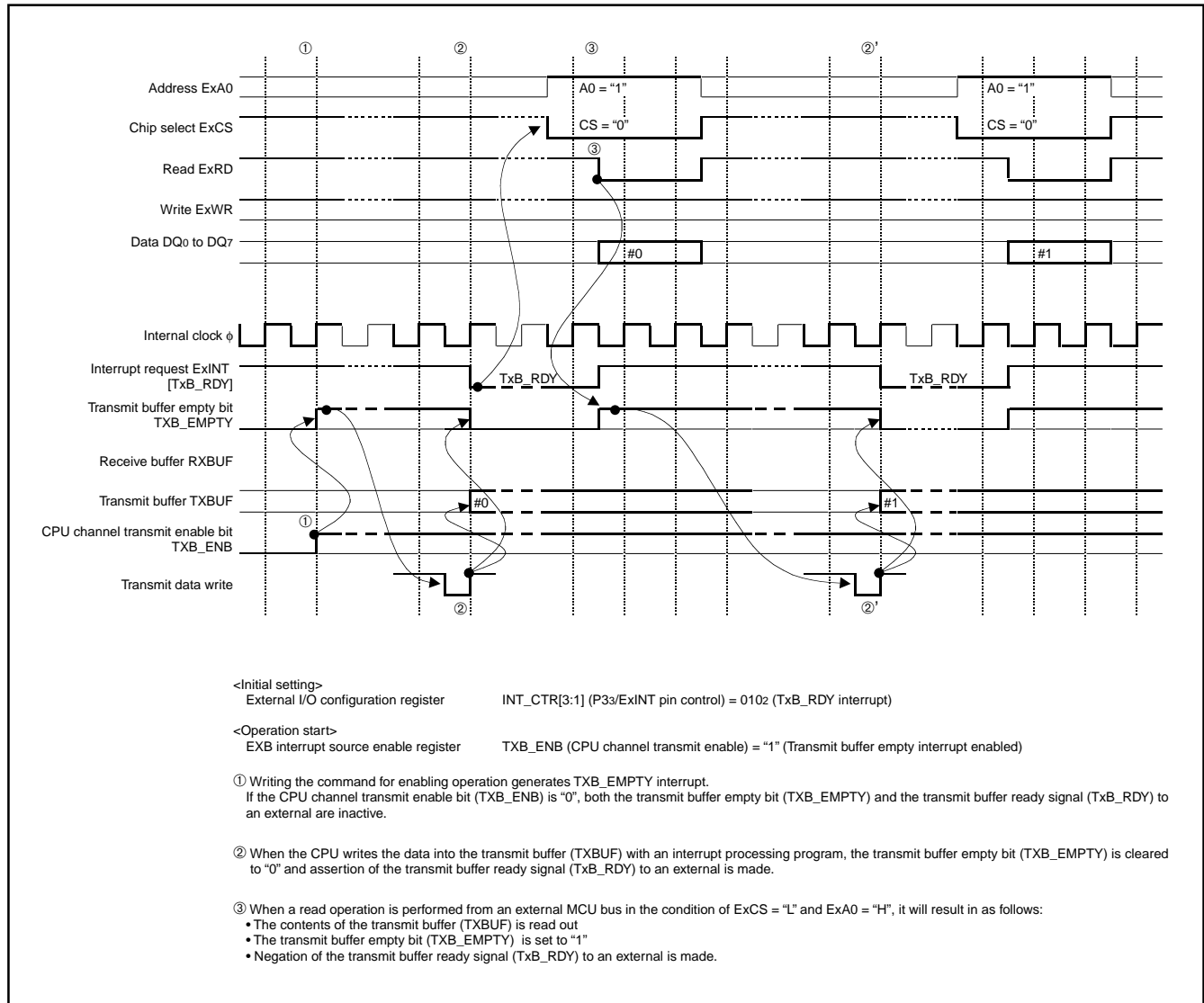


Fig. 92 CPU channel transmitting operation













### MULTICHANNEL RAM

The 38K0 group has the built-in multichannel RAM including the small logic circuit (RAM I/F) instead of ordinary RAM.

The multichannel RAM has the USB channel and the EXB channel in addition to the CPU channel.

The multichannel RAM controls access from CPU, USB and EXB, synchronizing control with  $\phi$ . The USB transfer rate is about 1.5 Mbytes/second. Access to the multichannel RAM is performed at every about 5.3 clocks in  $\phi = 8$  MHz, or at every about 4 clocks in  $\phi = 6$  MHz. The USB's access has priority to the EXB's.

The one wait function ( $\overline{\text{ONW}}$  function) of 38000 series CPU is used internally to control access with the CPU. When receiving an access request from the USB or the EXB, the multichannel RAM outputs  $\overline{\text{ONW}}$  signal to wait the CPU for one clock, and access of the USB or the EXB is performed.

If the multichannel RAM is outputting  $\overline{\text{ONW}}$  signal while the CPU is in the state of reading/writing for the RAM area, the CPU read cycle or write cycle is extended by 1 period of  $\phi$ .

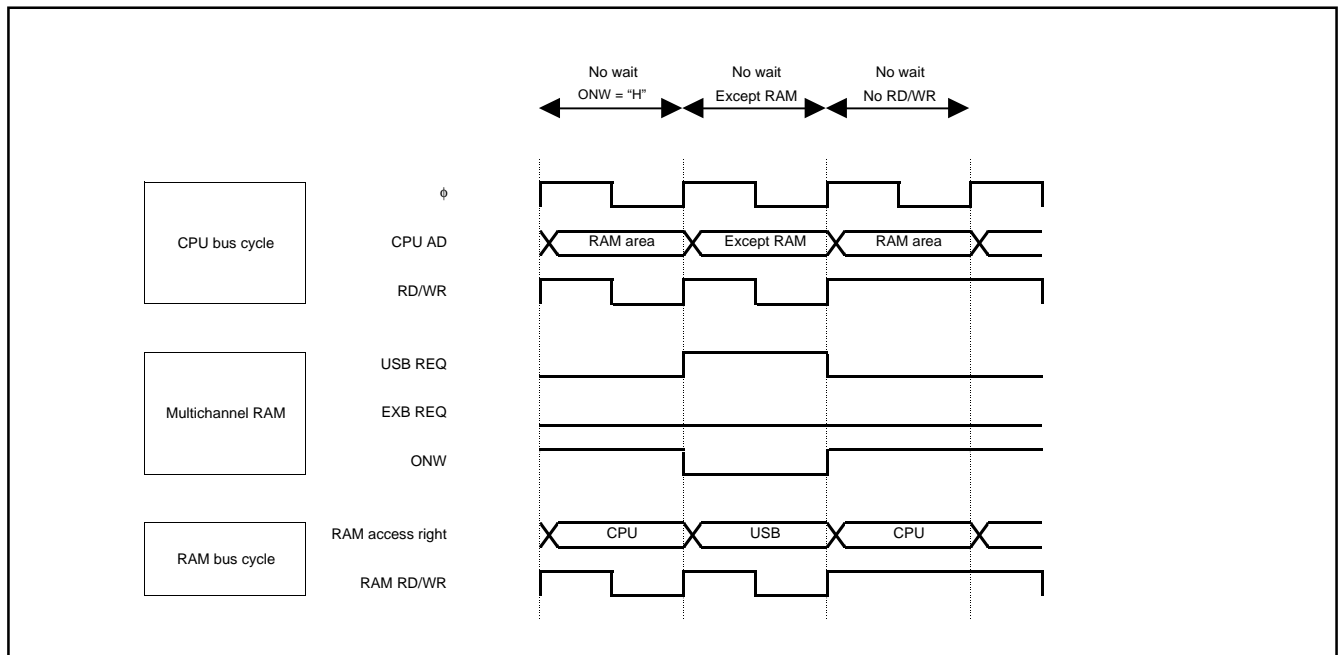


Fig. 98 Multichannel RAM timing diagram (no wait)

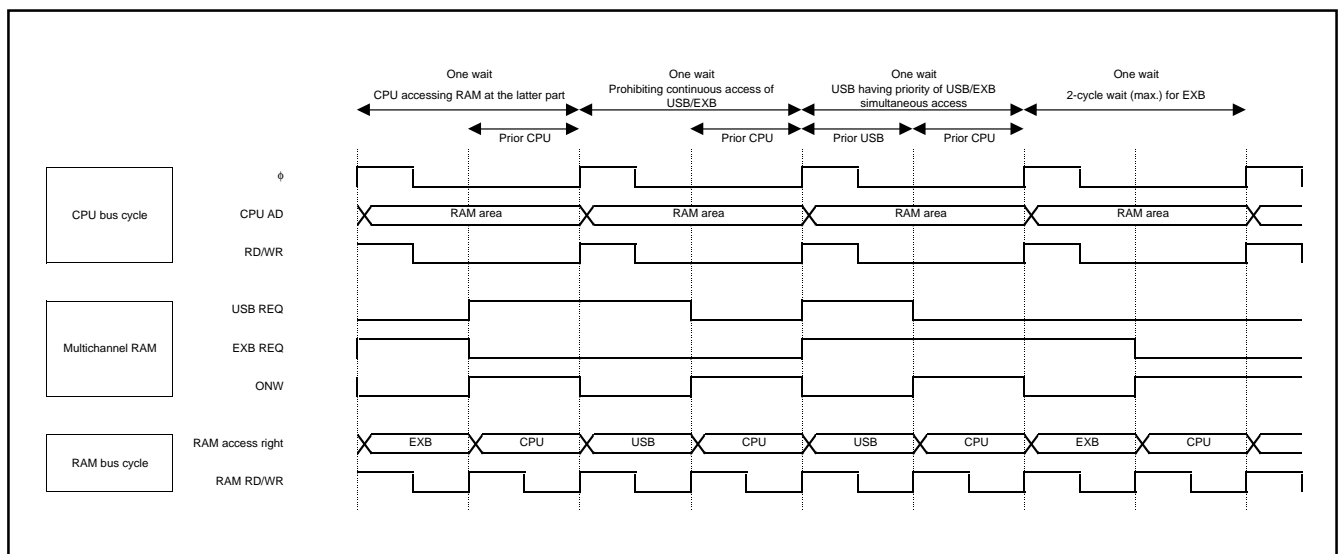


Fig. 99 Multichannel RAM timing diagram (one wait)

### Multichannel RAM Operation Example

The multichannel RAM operation example is shown below.  
This example shows the case that an external MCU uses the 38K0 group as a peripheral LSI (USB controller).

The following explains that the external MCU reads out the data which is received via the USB.

- ① The data which is received via the USB is written into the multichannel RAM.
- ② Receive completion is propagated to the CPU.
- ③ The external bus interface is activated owing to the CPU.
- ④ (1) The external bus interface sets the data which is read from the multichannel RAM into the internal data buffer.
- (2) The external MCU reads out the data bus buffer of the external bus interface.
- (3) The above operation is repeated by the number of the received bytes. After that, the data transfer is completed.

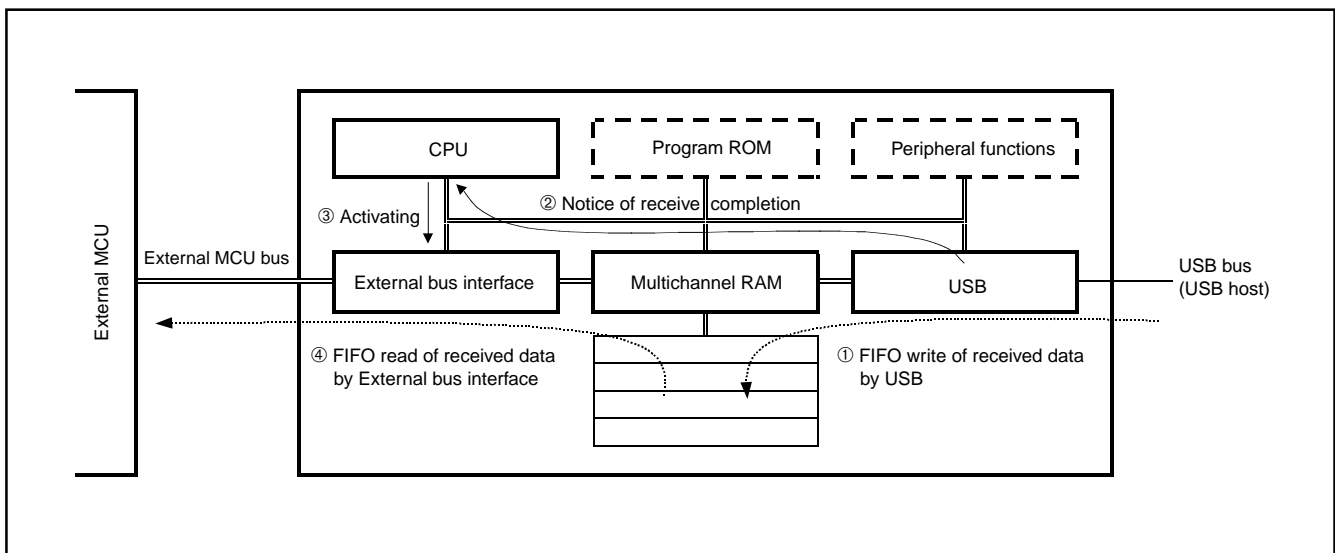


Fig. 100 Multichannel RAM operation example

## A-D CONVERTER

The functional blocks of the A-D converter are described below.

### [A-D Conversion Register 1, 2 (AD1, AD2)] 003716, 003816

The A-D conversion register is a read-only register that stores the result of an A-D conversion. When reading this register during an A-D conversion, the previous conversion result is read.

Bit 7 of the A-D conversion register 2 must be set to "0". Not only 10-bit reading but also only high-order 8-bit reading of conversion result can be performed by selecting the reading procedure of the A-D conversion registers 1, 2 after A-D conversion is completed (in Figure 102).

The 8-bit reading inclined to MSB is performed when reading the A-D converter register 1 after A-D conversion is started or reset; and when the A-D converter register 1 is read after reading the A-D converter register 2, the 8-bit reading inclined to LSB is performed.

### [A-D Control Register (ADCON)] 003616

The A-D control register controls the A-D conversion process. Bits 0 to 2 select a specific analog input pin. Bit 3 signals the completion of an A-D conversion. The value of this bit remains at "0" during an A-D conversion, and changes to "1" when an A-D conversion ends. Writing "0" to this bit starts the A-D conversion.

## Comparison Voltage Generator

The comparison voltage generator divides the voltage between VREF and AVSS into 1024, and that outputs the comparison voltage.

The A-D converter successively compares the comparison voltage Vref in each mode, dividing the VREF voltage (see below), with the input voltage.

#### • 10-bit reading

$$V_{ref} = \frac{V_{REF}}{1024} \times n \quad (n = 0-1023)$$

#### • 8-bit reading

$$V_{ref} = \frac{V_{REF}}{256} \times n \quad (n = 0-255)$$

## Channel Selector

The channel selector selects one of the input ports P17/AN7-P10/AN0.

## Comparator and Control Circuit

The comparator and control circuit compares an analog input voltage with the comparison voltage, and then stores the result in the A-D conversion registers 1, 2. When an A-D conversion is completed, the control circuit sets the AD conversion completion bit and the AD interrupt request bit to "1".

Note that because the comparator consists of a capacitor coupling, set f(system clock) to 500 kHz or more during an A-D conversion.

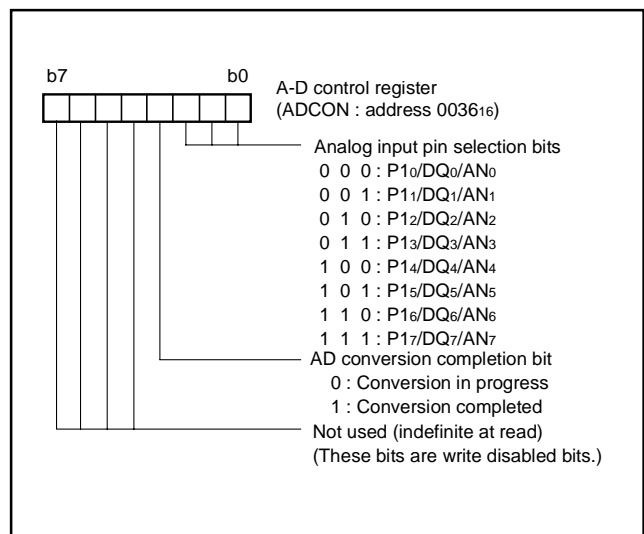


Fig. 101 Structure of A-D control register

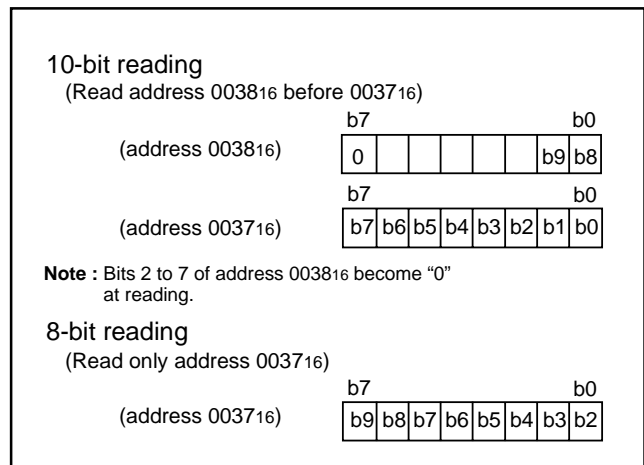


Fig. 102 10-bit A-D mode reading



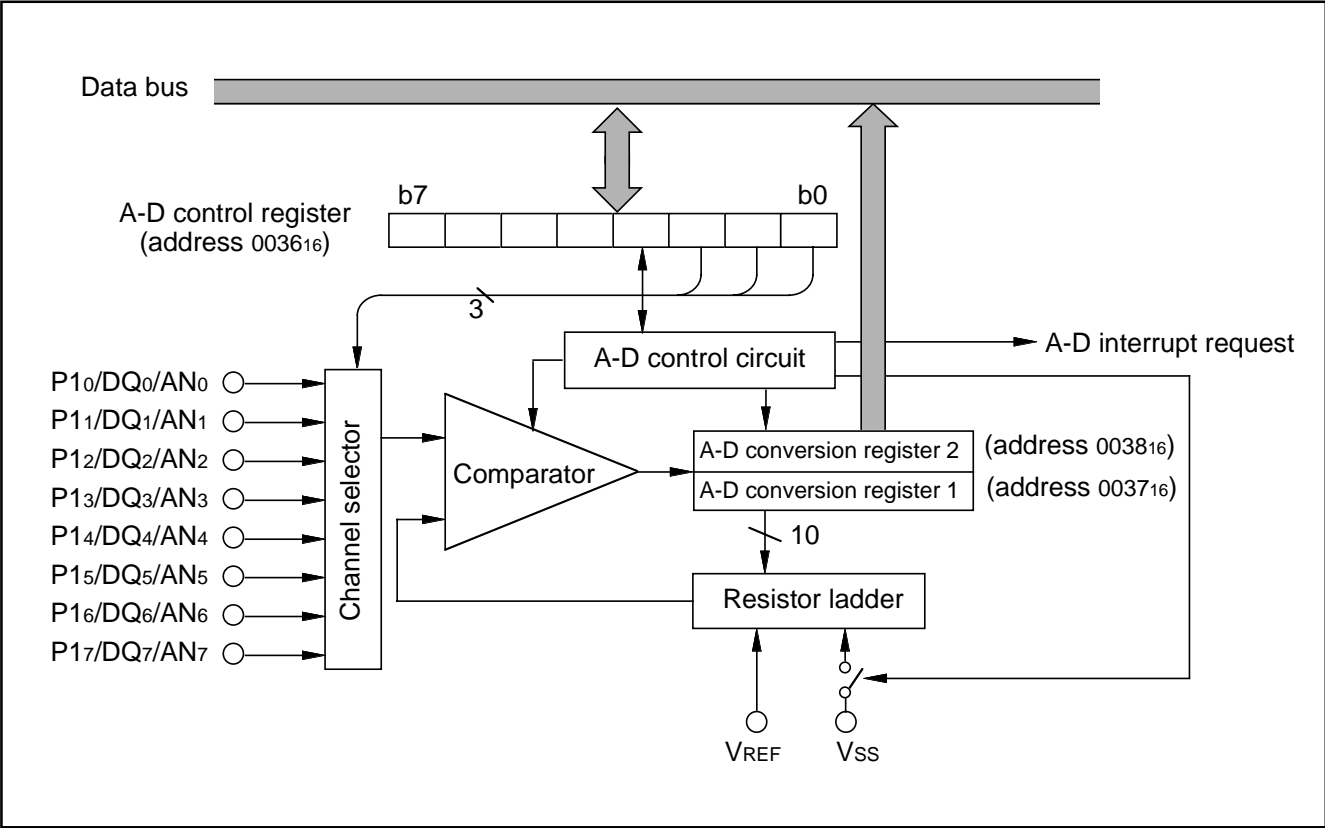


Fig. 103 A-D converter block diagram

## WATCHDOG TIMER

The watchdog timer gives a mean of returning to the reset status when a program cannot run on a normal loop (for example, because of a software run-away). The watchdog timer consists of an 8-bit watchdog timer L and an 8-bit watchdog timer H.

### Standard Operation of Watchdog Timer

When any data is not written into the watchdog timer control register (address 0039<sub>16</sub>) after resetting, the watchdog timer is in the stop state. The watchdog timer starts to count down by writing an optional value into the watchdog timer control register (address 0039<sub>16</sub>) and an internal reset occurs at an underflow of the watchdog timer H.

Accordingly, programming is usually performed so that writing to the watchdog timer control register (address 0039<sub>16</sub>) may be started before an underflow. When the watchdog timer control register (address 0039<sub>16</sub>) is read, the values of the high-order 6 bits of the watchdog timer H, STP instruction disable bit (bit 6), and watchdog timer H count source selection bit (bit 7) are read.

### Initial Value of Watchdog Timer

At reset or writing to the watchdog timer control register (address 0039<sub>16</sub>), each watchdog timer H and L is set to "FF<sub>16</sub>."

#### ● Watchdog timer H count source selection bit operation

Bit 7 of the watchdog timer control register (address 0039<sub>16</sub>) permits selecting a watchdog timer H count source. When this bit is set to "0", the count source becomes the underflow signal of watchdog timer L. The detection time is set to 131.072 ms at system clock 8 MHz frequency.

When this bit is set to "1", the count source becomes the system clock divided by 16. The detection time in this case is set to 512 μs at system clock 8 MHz frequency. This bit is cleared to "0" after resetting.

#### ● Operation of STP instruction disable bit

Bit 6 of the watchdog timer control register (address 0039<sub>16</sub>) permits disabling the STP instruction when the watchdog timer is in operation.

When this bit is "0", the STP instruction is enabled.

When this bit is "1", the STP instruction is disabled.

Once the STP instruction is executed, an internal reset occurs.

When this bit is set to "1", it cannot be rewritten to "0" by program. This bit is cleared to "0" after resetting.

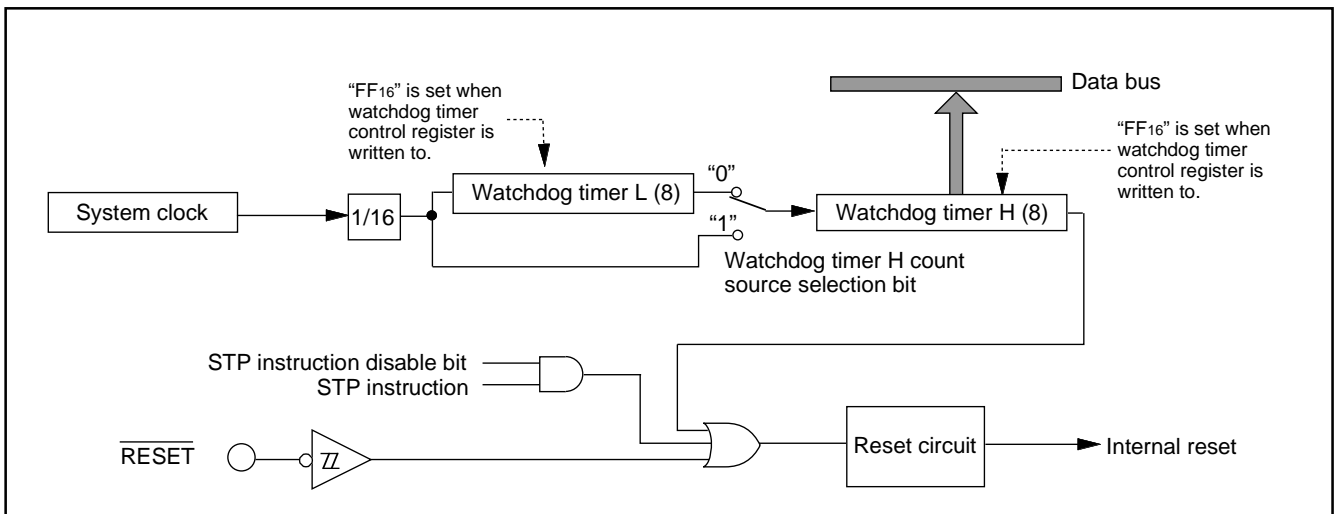


Fig. 104 Block diagram of Watchdog timer

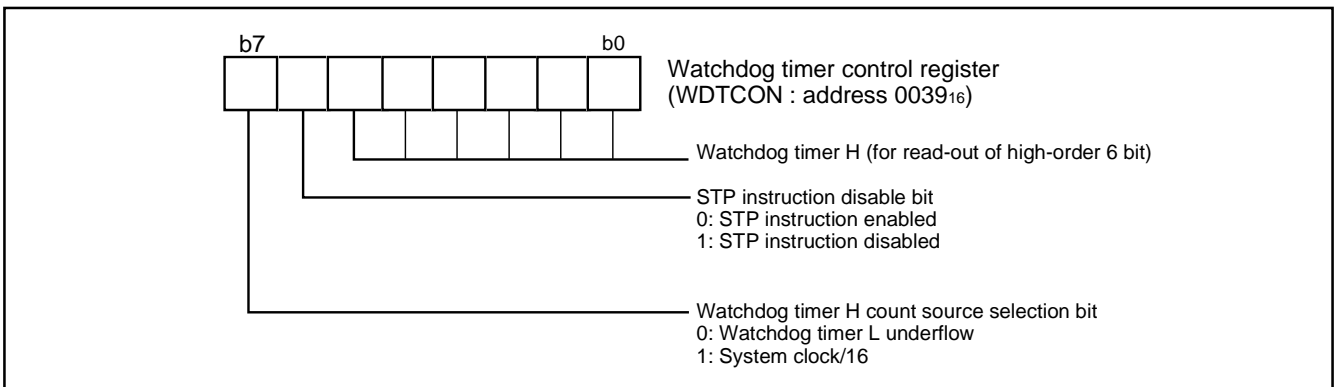


Fig. 105 Structure of Watchdog timer control register

### RESET CIRCUIT

To reset the microcomputer,  $\overline{\text{RESET}}$  pin should be held at an "L" level for 16 cycles or more of  $X_{IN}$ . Then the  $\overline{\text{RESET}}$  pin is returned to an "H" level (the power source voltage should be between 3.0 V and 5.25 V, and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address  $\text{FFFD}_{16}$  (high-order byte) and address  $\text{FFFC}_{16}$  (low-order byte). Make sure that the reset input voltage is 0.6 V for  $V_{CC}$  of 3.0 V.

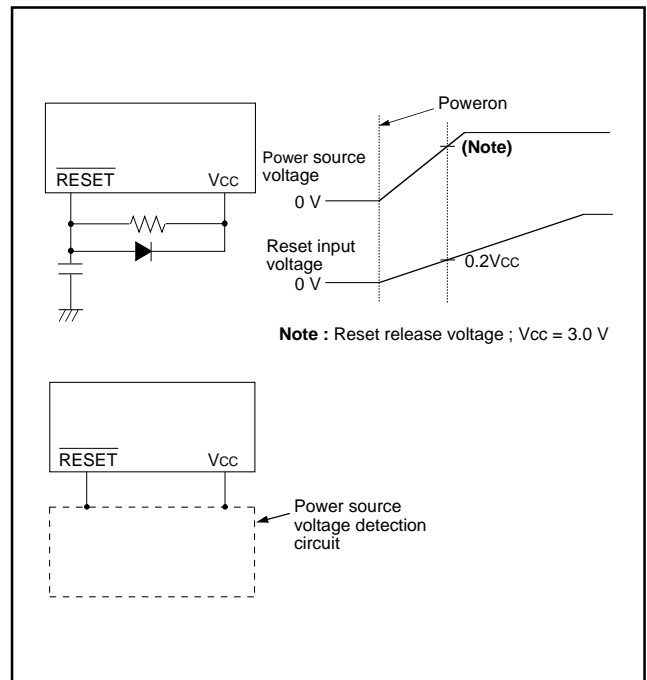


Fig. 106 Example of reset circuit

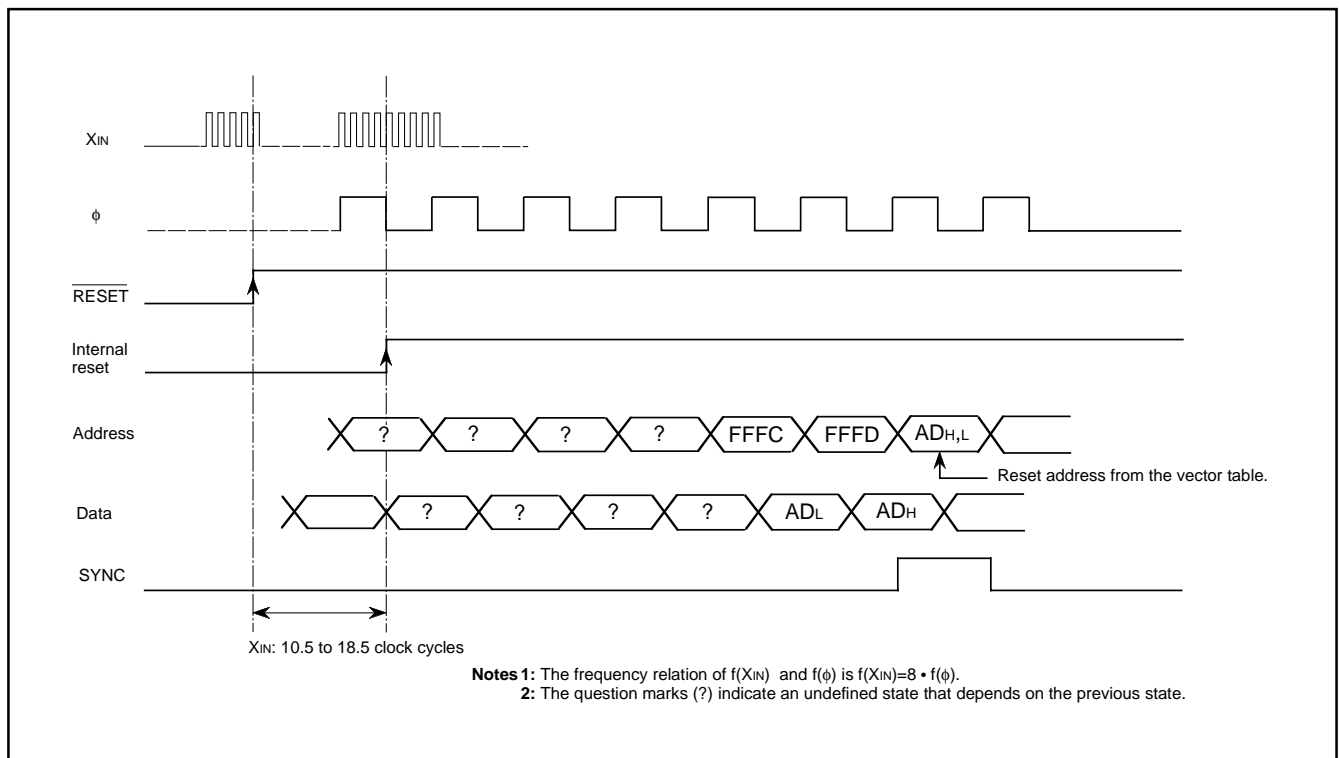


Fig. 107 Reset sequence

**PLL CIRCUIT (FREQUENCY SYNTHESIZER)**

The PLL circuit generates fvco (PLL output clock), which is required for fusb (USB clock) and fsyn (fusb division clock), from f(XIN) (external input reference clock). Figure 108 shows the PLL circuit block diagram.

It is possible to input 6 or 12 MHz clock from the externals as a standard clock input. When using the USB function, set the PLL operation mode selection bit so that fvco may be set to 48 MHz.

The PLL circuit operates by setting the PLL operation enable bit to "1". When supplying fvco to the USB block, wait for the oscillation stable time (1ms or less) of PLL before selecting fvco with the USB clock selection bit.

According to the setting of the USB clock division ratio selection bit, the division clock of fusb is supplied to fsyn. When using this clock as system clock, set the USB clock division ratio selection bit so that it may be set to 6 MHz, 8 MHz or 12 MHz. (However, using it only when fusb is 48MHz is recommended).

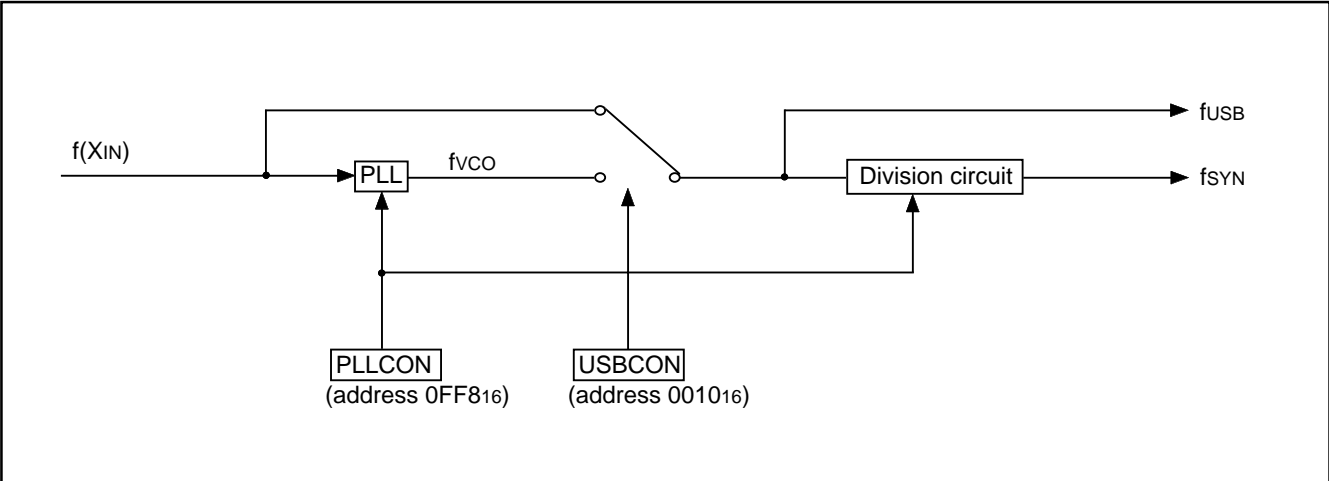


Fig. 108 Block diagram of PLL circuit

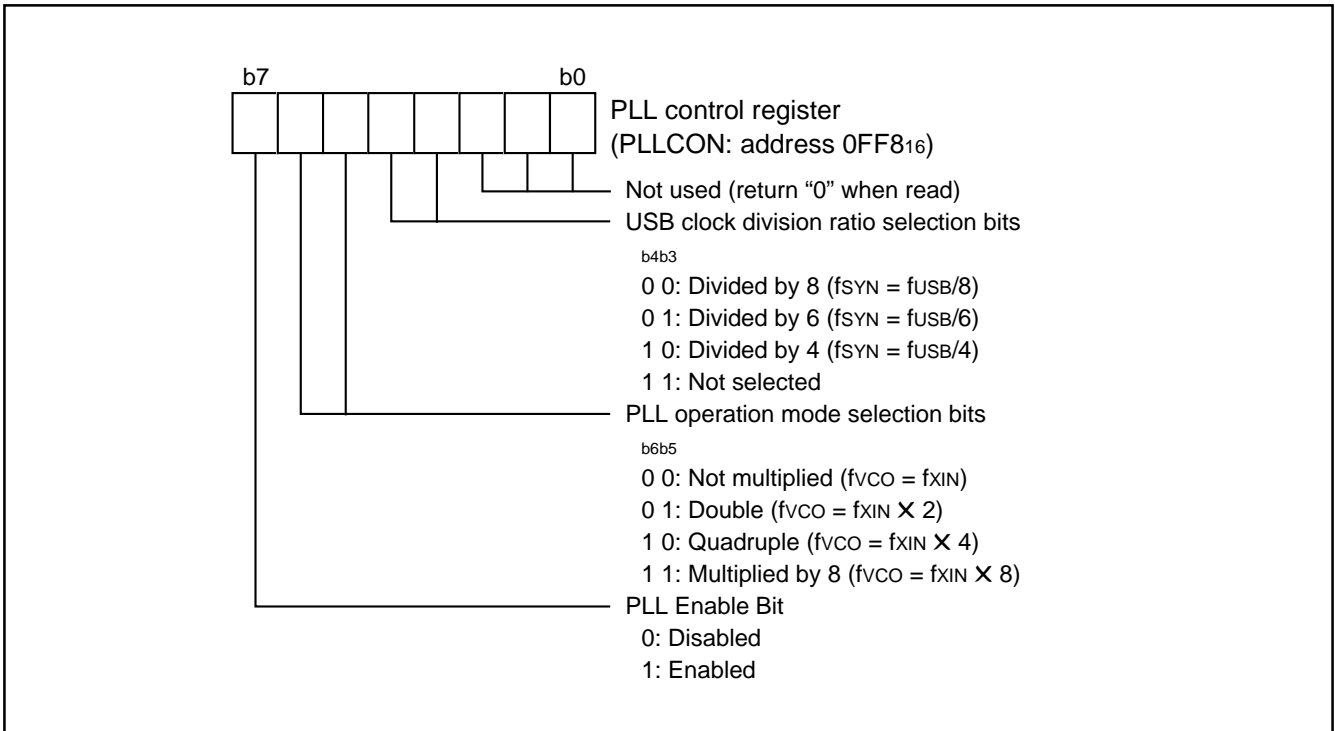


Fig. 109 Structure of PLL control register

## CLOCK GENERATING CIRCUIT

An oscillation circuit can be formed by connecting a resonator between X<sub>IN</sub> and X<sub>OUT</sub>. Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between X<sub>IN</sub> and X<sub>OUT</sub> since a feed-back resistor exists on-chip.

## Frequency Control

Either f<sub>SYN</sub> or f(X<sub>IN</sub>) can be selected as an internal system clock. Furthermore, the frequency of internal clock  $\phi$  can be selected by the system clock division ratio selection bit.

### (1) f<sub>SYN</sub> clock

f<sub>SYN</sub> clock is generated by the PLL circuit. f(X<sub>IN</sub>) or f<sub>VCO</sub> can be selected as an input clock. When using as an internal system clock, there is restriction on use. Refer to the clause of "PLL CIRCUIT".

### (2) f(X<sub>IN</sub>) clock

The frequency applied to the X<sub>IN</sub> pin is used as an internal system clock frequency.

## Oscillation Control

### (1) Stop mode

If the STP instruction is executed, the internal clock  $\phi$  stops at an "H" level, and the X<sub>IN</sub> oscillator stops. When the oscillation stabilizing time set after STP instruction released bit is "0," the prescaler 12 is set to "FF<sub>16</sub>" and timer 1 is set to "01<sub>16</sub>." When the oscillation stabilizing time set after STP instruction released bit is "1," set the sufficient time for oscillation of used oscillator to stabilize since nothing is set to the prescaler 12 and timer 1. X<sub>IN</sub> divided by 16 is compulsorily connected to the input of the prescaler 12. Oscillator restarts when an external interrupt (including USB resume interrupt) is received, but the internal clock  $\phi$  remains at "H" until timer 1 underflows. The internal clock  $\phi$  is not supplied until timer 1 underflows. Because the sufficient time is required for the oscillation to stabilize when a ceramic resonator etc. is used. When the oscillator is restarted by reset, apply "L" level to the  $\overline{\text{RESET}}$  pin until the oscillation is stable since a wait time will not be generated automatically.

### (2) Wait mode

If the WIT instruction is executed, the internal clock  $\phi$  stops at an "H" level, but the oscillator does not stop. The internal clock  $\phi$  restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

To ensure that the interrupts will be received to release the STP or WIT state, their interrupt enable bits must be set to "1" before executing of the STP or WIT instruction.

When releasing the STP state, the prescaler 12 and timer 1 will start counting the clock X<sub>IN</sub> divided by 16. Accordingly, set the timer 1 interrupt enable bit to "0" before executing the STP instruction.

### ■Note

When using the oscillation stabilizing time set after STP instruction released bit set to "1", evaluate time to stabilize oscillation of the used oscillator and set the value to the timer 1 and prescaler 12.



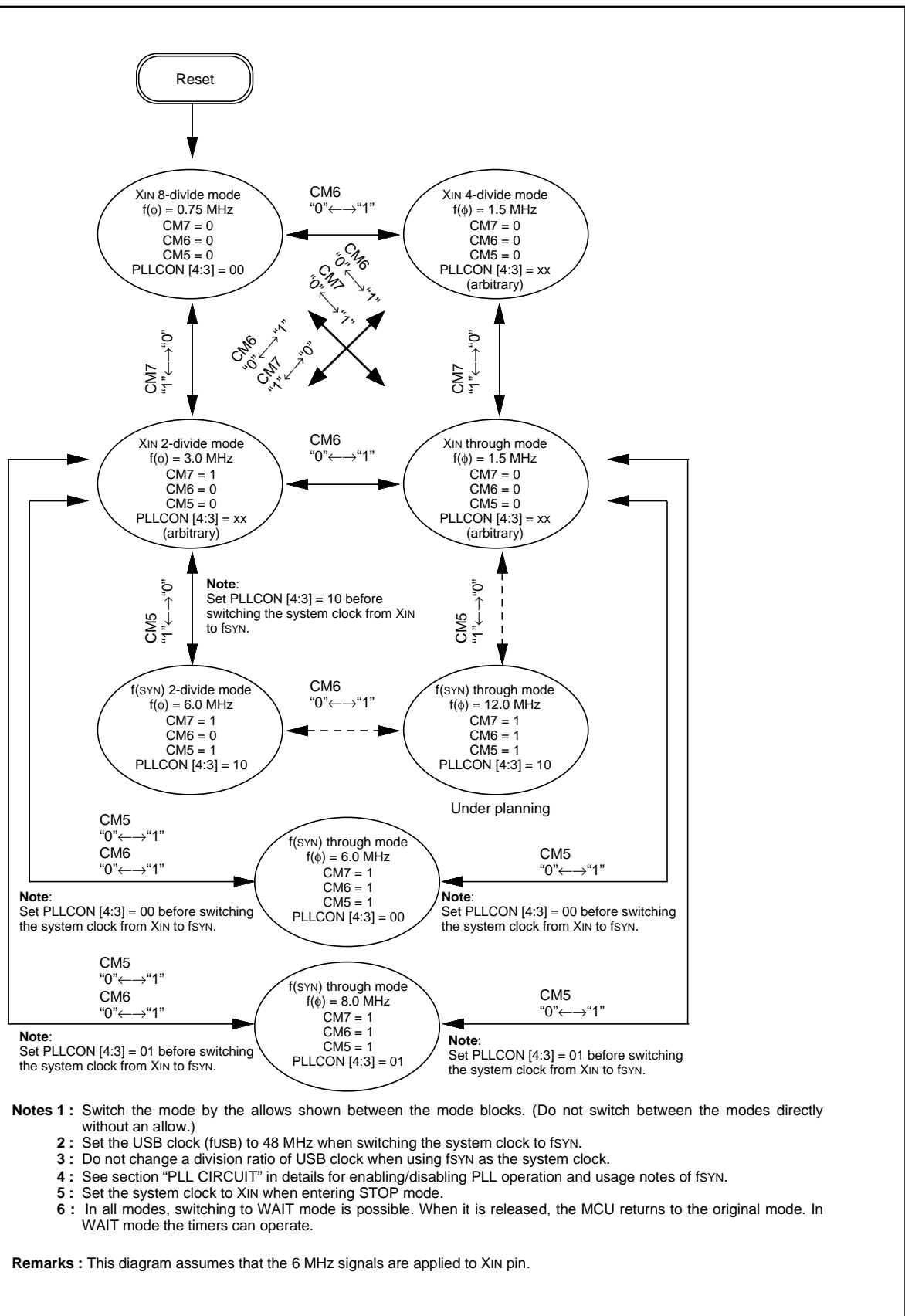


Fig. 114 State transitions of clock



## FLASH MEMORY MODE

The 38K0 group's flash memory version has an internal new DINOR (Divided bit line NOR) flash memory that can be rewritten with a single power source when Vcc is 4.5 to 5.25 V, and 2 power sources when Vcc is 3.0 to 4.5 V.

For this flash memory, three flash memory modes are available in which to read, program, and erase: the parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and the CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU).

## Summary

Table 8 lists the summary of the 38K0 group's flash memory version.

This flash memory version has some blocks on the flash memory as shown in Figure 115 and each block can be erased. The flash memory is divided into User ROM area and Boot ROM area.

In addition to the ordinary User ROM area to store the MCU operation control program, the flash memory has a Boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This Boot ROM area can be rewritten in only parallel I/O mode.

**Table 8 Summary of 38K0 group's flash memory version**

Item		Specifications
Power source voltage (Vcc)		3.00 – 5.25 V (Program and erase in 4.00 to 5.25 V of Vcc.) 3.00 – 4.00 V (Program and erase in 3.00 to 5.25 V of Vcc.)
Program/Erase VPP voltage (VPP)		4.50 – 5.25 V
Flash memory mode		3 modes; Flash memory can be manipulated as follows: •CPU rewrite mode: Manipulated by the Central Processing Unit (CPU). •Parallel I/O mode: Manipulated using an external programmer ( <b>Note 1</b> ) •Standard serial I/O mode: Manipulated using an external programmer ( <b>Note 1</b> )
Erase block division	User ROM area	1 block (32 Kbytes)
	Boot ROM area	1 block (4 Kbytes) ( <b>Note 2</b> )
Program method		Byte program
Erase method		Batch erasing
Program/Erase control method		Program/Erase control by software command
Number of commands		6 commands
Number of program/Erase times		100 times
Data retention period		10 years
ROM code protection		Available in parallel I/O mode and standard serial I/O mode

**Notes 1:** In the parallel I/O mode or the standard serial I/O mode, use the exclusive external equipment flash programmer which supports the 38K2 Group (flash memory version).

**2:** The Boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. This Boot ROM area can be rewritten in only parallel I/O mode.

**(1) CPU Rewrite Mode**

In CPU rewrite mode, the internal flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).

In CPU rewrite mode, only the User ROM area shown in Figure 115 can be rewritten; the Boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the User ROM area and each block area.

The control program for CPU rewrite mode can be stored in either User ROM or Boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM area to be executed before it can be executed.

**Microcomputer Mode and Boot Mode**

The control program for CPU rewrite 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 unusable.)

See Figure 115 for details about the Boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVss pin low. In this case, the CPU starts operating using the control program in the User ROM area.

When the microcomputer is reset by pulling the P16 ( $\overline{CE}$ ) pin high, the CNVss pin high, the CPU starts operating using the control program in the Boot ROM area. This mode is called the "Boot" mode.

**Block Address**

Block addresses refer to the maximum address of each block. These addresses are used in the block erase command.

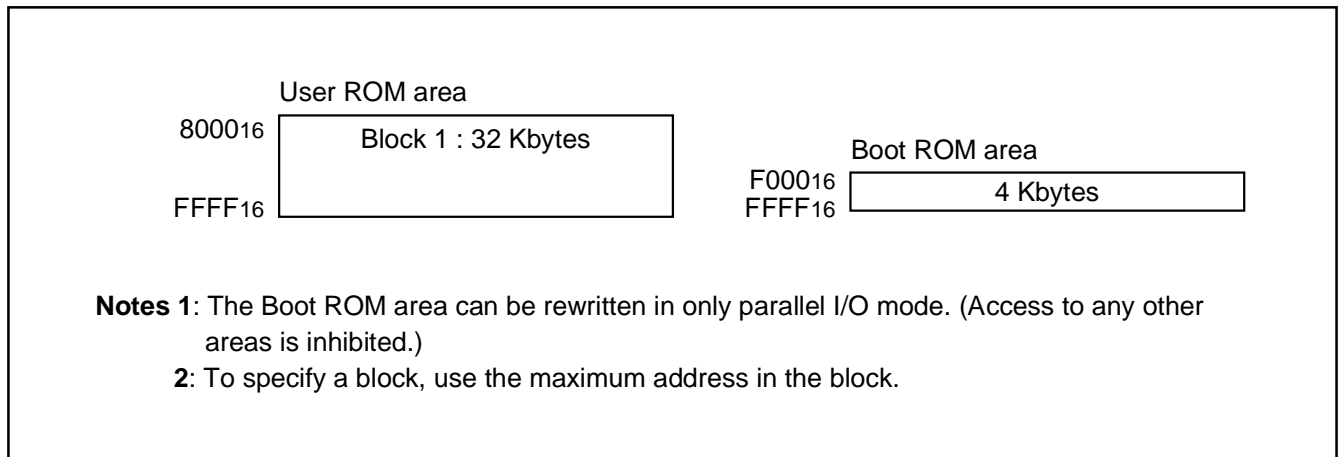


Fig. 115 Block diagram of built-in flash memory

### Outline Performance (CPU Rewrite Mode)

CPU rewrite mode is usable in the single-chip or Boot mode. The only User ROM area can be rewritten in CPU rewrite mode.

In CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. This rewrite control program must be transferred to a memory such as the internal RAM before it can be executed.

The MCU enters CPU rewrite mode by applying 4.50 V to 5.25 V to the CNVss pin and setting "1" to the CPU Rewrite Mode Select Bit (bit 1 of address 0FFE<sub>16</sub>). Software commands are accepted once the mode is entered.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

Figure 116 shows the flash memory control register.

Bit 0 is the RY/ $\overline{\text{BY}}$  status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is "0" (busy). Otherwise, it is "1" (ready). This is equivalent to the RY/ $\overline{\text{BY}}$  pin function in parallel I/O mode.

Bit 1 is the CPU Rewrite Mode Select Bit. When this bit is set to "1", the MCU enters CPU rewrite mode. Software commands are accepted once the mode is entered. In CPU rewrite mode, the

CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in a memory other than internal flash memory for write to bit 1. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. The bit can be set to "0" by only writing "0".

Bit 2 is the CPU Rewrite Mode Entry Flag. This flag indicates "1" in CPU rewrite mode, so that reading this flag can check whether CPU rewrite mode has been entered or not.

Bit 3 is the flash memory reset bit used to reset the control circuit of internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU Rewrite Mode Select Bit is "1", setting "1" for this bit resets the control circuit. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. To release the reset, it is necessary to set this bit to "0".

Bit 4 is the User Area/Boot Area Select Bit. When this bit is set to "1", Boot ROM area is accessed, and CPU rewrite mode in Boot ROM area is available. In Boot mode, this bit is set to "1" automatically. Reprogramming of this bit must be in a memory other than internal flash memory.

Figure 117 shows a flowchart for setting/releasing CPU rewrite mode.

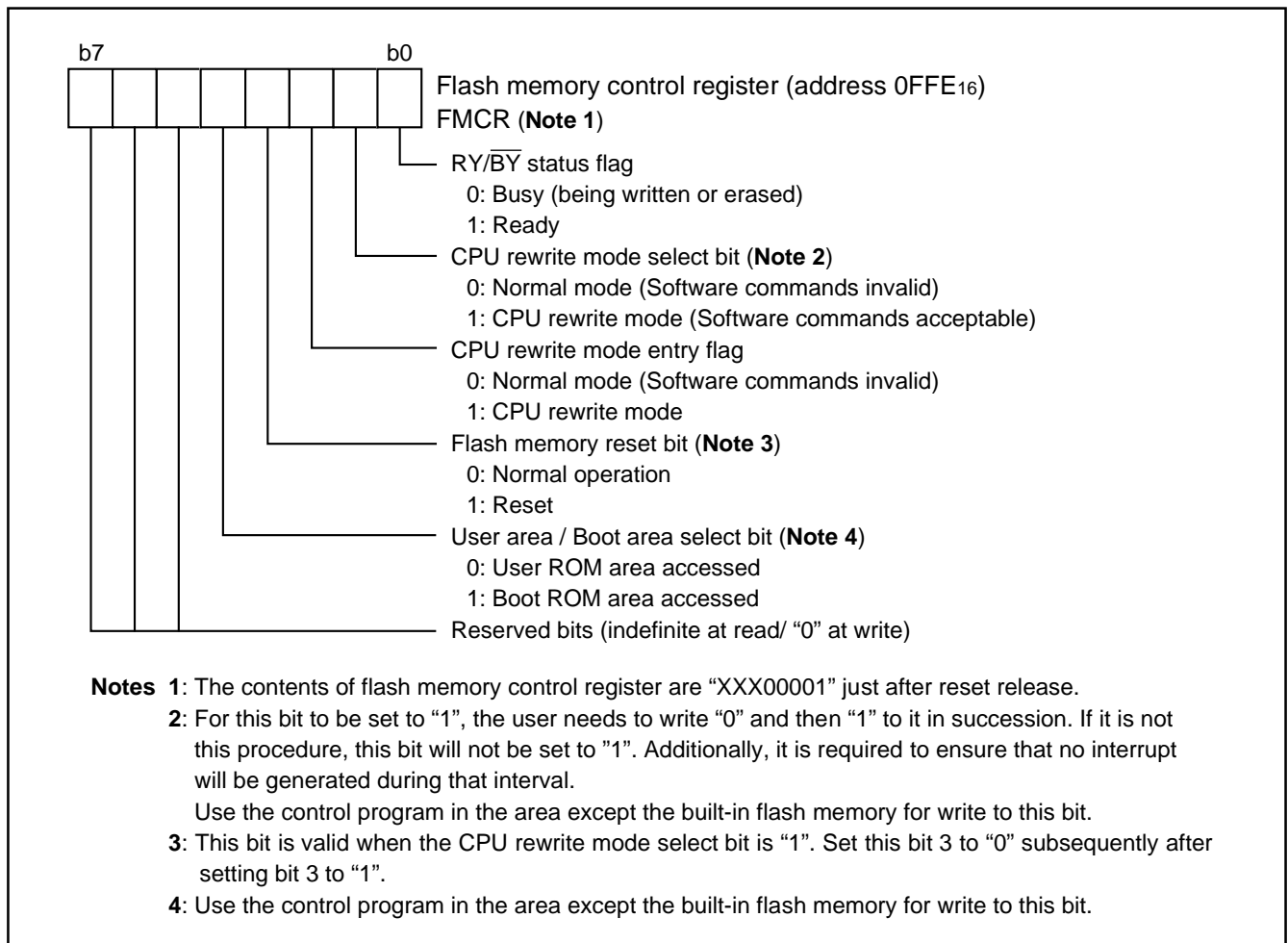


Fig. 116 Structure of flash memory control register

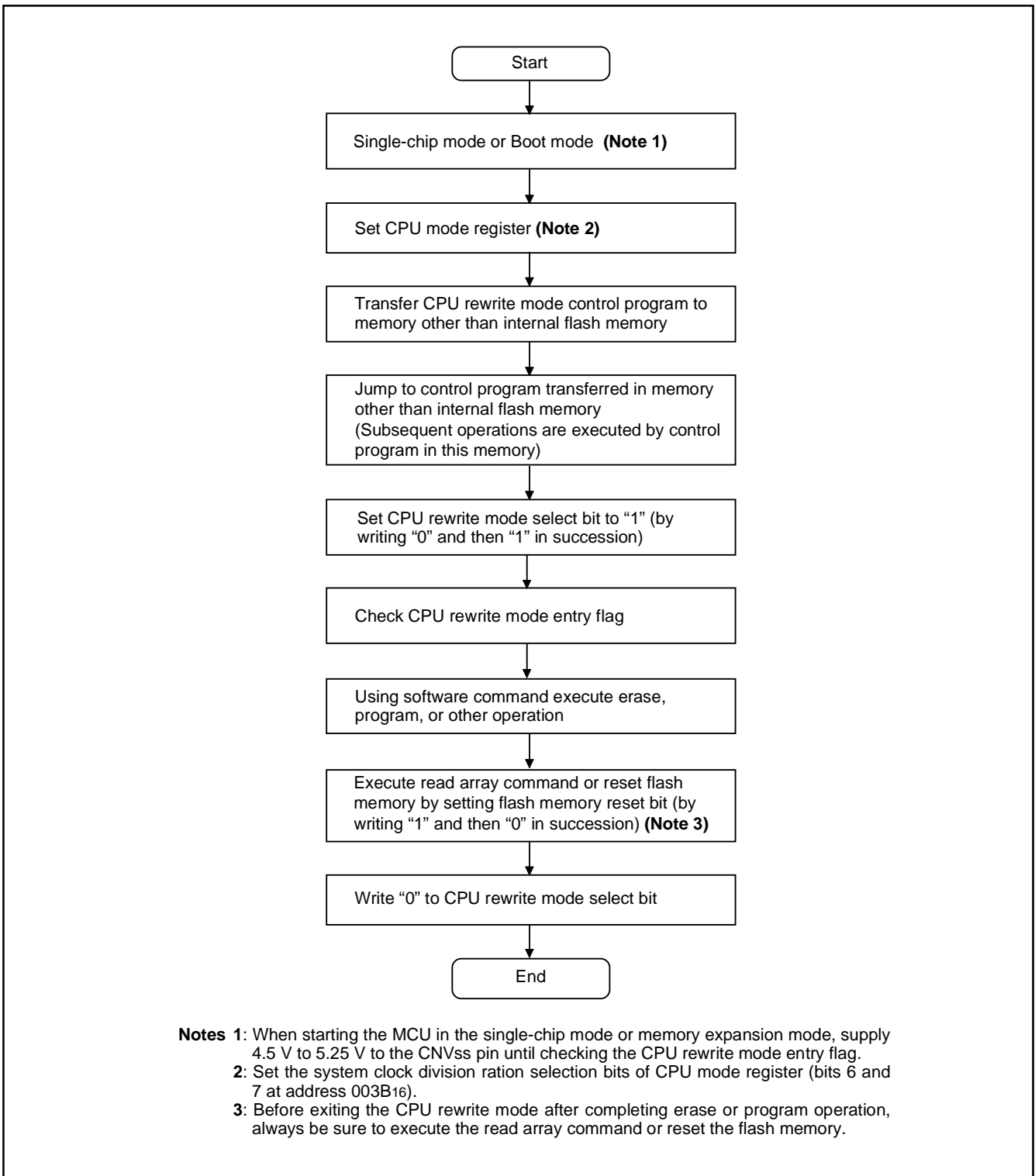


Fig. 117 CPU rewrite mode set/release flowchart

### Notes on CPU Rewrite Mode

Take the notes described below when rewriting the flash memory in CPU rewrite mode.

#### ●Operation speed

During CPU rewrite mode, set the internal clock  $\phi$  to 1.5 MHz or less using the system clock division ratio selection bits (bits 6 and 7 of address 003B<sub>16</sub>).

#### ●Instructions inhibited against use

The instructions which refer to the internal data of the flash memory cannot be used during CPU rewrite mode .

#### ●Interrupts inhibited against use

The interrupts cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory.

#### ●Watchdog timer

If the watchdog timer has been already activated, internal reset due to an underflow will not occur because the watchdog timer is surely cleared during program or erase.

#### ●Reset

Reset is always valid. The MCU is activated using the boot mode at release of reset in the condition of CNV<sub>SS</sub> = "H", so that the program will begin at the address which is stored in addresses FFFC<sub>16</sub> and FFFD<sub>16</sub> of the boot ROM area.

## Software Commands

Table 9 lists the software commands.

After setting the CPU Rewrite Mode Select Bit to "1", write a software command to specify an erase or program operation.

Each software command is explained below.

### ●Read Array Command (FF<sub>16</sub>)

The read array mode is entered by writing the command code "FF<sub>16</sub>" in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the contents of the specified address are read out at the data bus (D<sub>0</sub> to D<sub>7</sub>).

The read array mode is retained intact until another command is written.

### ●Read Status Register Command (70<sub>16</sub>)

When the command code "70<sub>16</sub>" is written in the first bus cycle, the contents of the status register are read out at the data bus (D<sub>0</sub> to D<sub>7</sub>) by a read in the second bus cycle.

The status register is explained in the next section.

### ●Clear Status Register Command (50<sub>16</sub>)

This command is used to clear the bits SR<sub>4</sub> and SR<sub>5</sub> of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50<sub>16</sub>" in the first bus cycle.

### ●Program Command (40<sub>16</sub>)

Program operation starts when the command code "40<sub>16</sub>" is written in the first bus cycle. Then, if the address and data to program are written in the 2nd bus cycle, the control circuit of flash memory (data programming and verification) will start a program.

Whether the write operation is completed can be confirmed by reading the status register or the RY/B $\bar{Y}$  Status Flag. When the program starts, the read status register mode is entered automatically and the contents of the status register is read at the data bus (D<sub>B0</sub> to D<sub>B7</sub>). The status register bit 7 (SR<sub>7</sub>) is set to "1" at the same time the write operation starts and is returned to "1" upon completion of the write operation. In this case, the read status register mode remains active until the read array command (FF<sub>16</sub>) is written.

During the program movement, The RY/B $\bar{Y}$  Status Flag of flash memory control register is set to "0". When the program completes, it becomes "1".

At program end, program results can be checked by reading the status register.

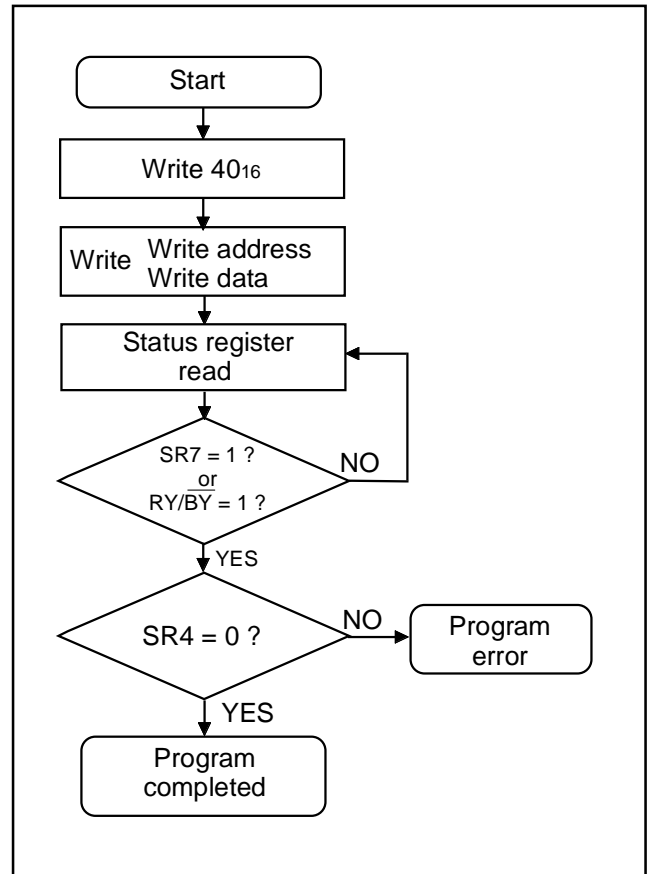


Fig. 118 Program flowchart

Table 9 List of software commands (CPU rewrite mode)

Command	Cycle number	First bus cycle			Second bus cycle		
		Mode	Address	Data (D <sub>0</sub> to D <sub>7</sub> )	Mode	Address	Data (D <sub>0</sub> to D <sub>7</sub> )
Read array	1	Write	X (Note 4)	FF <sub>16</sub>			
Read status register	2	Write	X	70 <sub>16</sub>	Read	X	SRD (Note 1)
Clear status register	1	Write	X	50 <sub>16</sub>			
Program	2	Write	X	40 <sub>16</sub>	Write	WA (Note 2)	WD (Note 2)
Erase all blocks	2	Write	X	20 <sub>16</sub>	Write	X	20 <sub>16</sub>
Block erase	2	Write	X	20 <sub>16</sub>	Write	BA (Note 3)	D0 <sub>16</sub>

Notes 1: SRD = Status Register Data

2: WA = Write Address, WD = Write Data

3: BA = Block Address to be erased (Input the maximum address of each block.)

4: X denotes a given address in the User ROM area .

●Erase All Blocks Command (20<sub>16</sub>/20<sub>16</sub>)

By writing the command code "20<sub>16</sub>" in the first bus cycle and the confirmation command code "20<sub>16</sub>" in the second bus cycle that follows, the operation of erase all blocks (erase and erase verify) starts.

Whether the erase all blocks command is terminated can be confirmed by reading the status register or the RY/ $\overline{\text{BY}}$  Status Flag of flash memory control register. When the erase all blocks operation starts, the read status register mode is entered automatically and the contents of the status register can be read out at the data bus (D<sub>0</sub> to D<sub>7</sub>). The status register bit 7 (SR7) is set to "0" at the same time the erase operation starts and is returned to "1" upon completion of the erase operation. In this case, the read status register mode remains active until the read array command (FF<sub>16</sub>) is written.

The RY/ $\overline{\text{BY}}$  Status Flag is "0" during erase operation and "1" when the erase operation is completed as is the status register bit 7.

After the erase all blocks end, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

●Block Erase Command (20<sub>16</sub>/D0<sub>16</sub>)

By writing the command code "20<sub>16</sub>" in the first bus cycle and the confirmation command code "D0<sub>16</sub>" and the block address in the second bus cycle that follows, the block erase (erase and erase verify) operation starts for the block address of the flash memory to be specified.

Whether the block erase operation is completed can be confirmed by reading the status register or the RY/ $\overline{\text{BY}}$  Status Flag of flash memory control register. At the same time the block erase operation starts, the read status register mode is automatically entered, so that the contents of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the block erase operation starts and is returned to "1" upon completion of the block erase operation. In this case, the read status register mode remains active until the read array command (FF<sub>16</sub>) is written.

The RY/ $\overline{\text{BY}}$  Status Flag is "0" during block erase operation and "1" when the block erase operation is completed as is the status register bit 7.

After the block erase ends, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

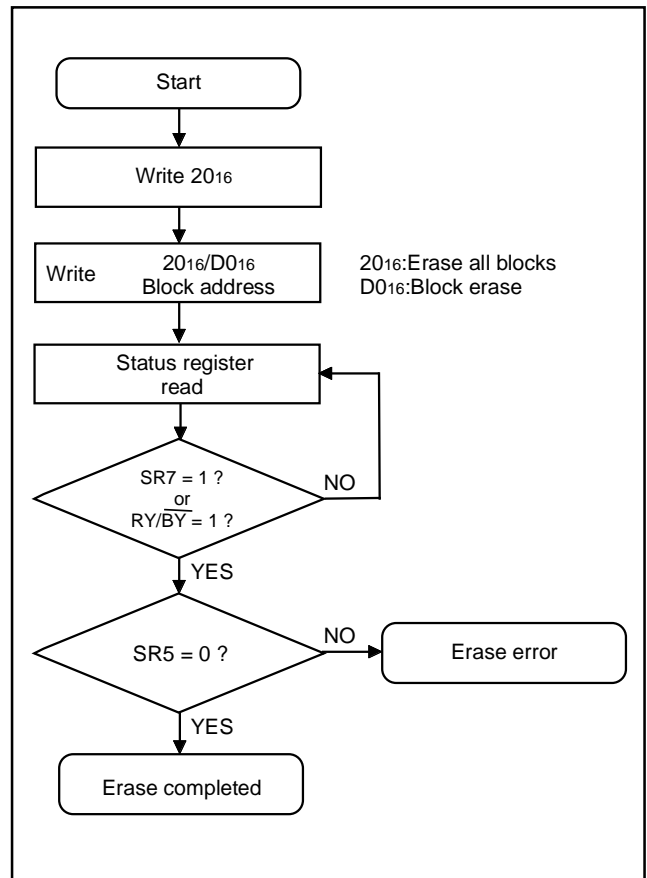


Fig. 119 Erase flowchart

### Status Register (SRD)

The status register shows the operating status of the flash memory and whether erase operations and programs ended successfully or in error. It can be read in the following ways:

- (1) By reading an arbitrary address from the User ROM area after writing the read status register command (70<sub>16</sub>)
- (2) By reading an arbitrary address from the User ROM area in the period from when the program starts or erase operation starts to when the read array command (FF<sub>16</sub>) is input.

Also, the status register can be cleared by writing the clear status register command (50<sub>16</sub>).

After reset, the status register is set to "80<sub>16</sub>".

Table 10 shows the status register. Each bit in this register is explained below.

#### •Sequencer status (SR7)

The sequencer status indicates the operating status of the flash memory. This bit is set to "0" (busy) during write or erase operation and is set to "1" when these operations ends.

After power-on, the sequencer status is set to "1" (ready).

#### •Erase status (SR5)

The erase status indicates the operating status of erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is set to "0".

#### •Program status (SR4)

The program status indicates the operating status of write operation. When a write error occurs, it is set to "1". The program status is set to "0" when it is cleared.

If "1" is written for any of the SR5 and SR4 bits, the program, erase all blocks, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50<sub>16</sub>) and clear the status register.

Table 10 Definition of each bit in status register

Each bit of SRD0 bits	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, it is possible to know the execution results of erase and program operations. Figure 120 shows a full status check flowchart and the action to be taken when each error occurs.

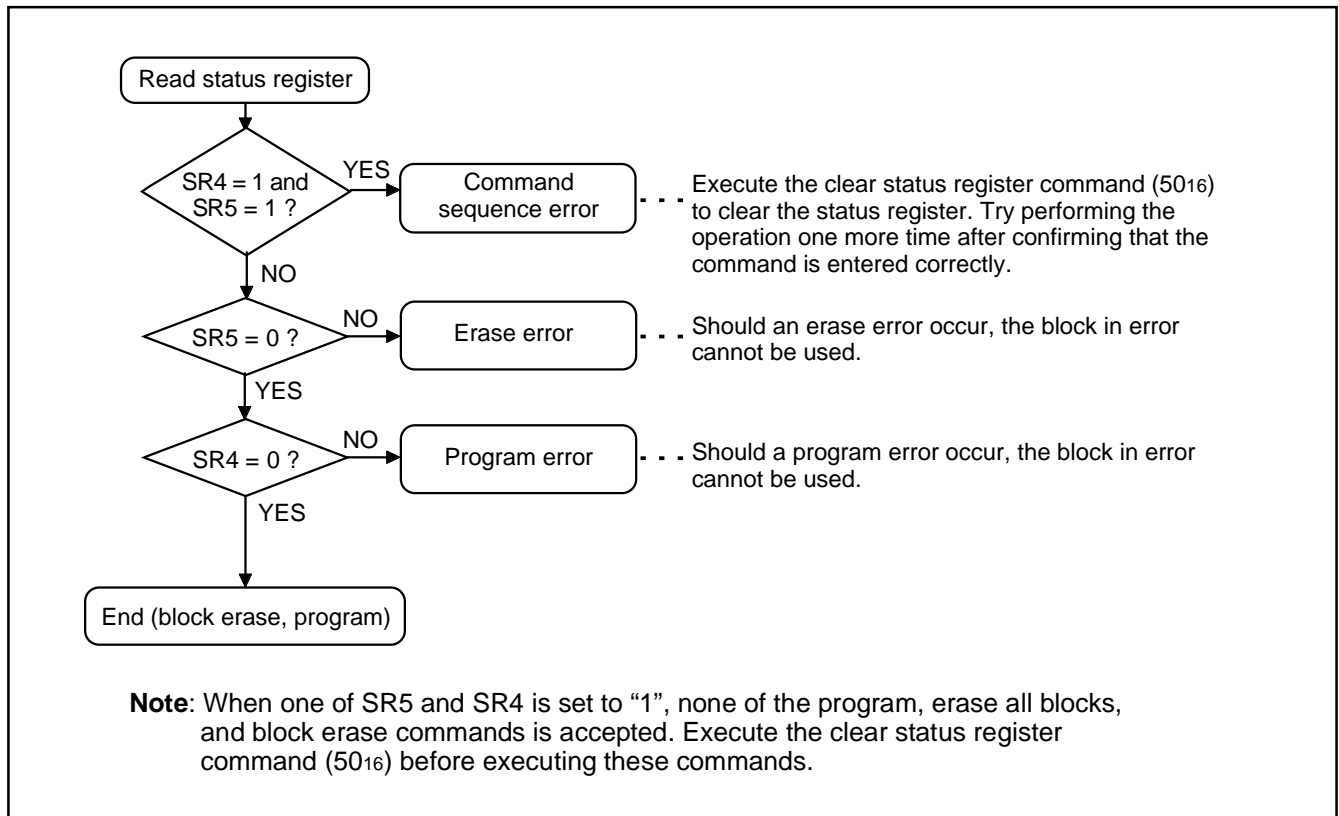


Fig. 120 Full status check flowchart and remedial procedure for errors

## Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of internal flash memory from being read out or rewritten easily, this MCU incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

### ●ROM Code Protect Function

The ROM code protect function is the function to inhibit reading out or modifying the contents of internal flash memory by using the ROM code protect control register (address FFDB<sub>16</sub>) in parallel I/O mode. Figure 121 shows the ROM code protect control register (address FFDB<sub>16</sub>). (This address exists in the User ROM area.)

If one or both of the pair of ROM Code Protect Bits is set to "0", the ROM code protect is turned on, so that the contents of internal flash memory are protected against readout and modification. The 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 an attempt is made to select both level 1 and level 2, level 2 is selected by default.

If both of the two ROM Code Protect Reset Bits are set to "00", the ROM code protect is turned off, so that the contents of internal flash memory can be read out or modified. Once the ROM code protect is turned on, the contents of the ROM Code Protect Reset Bits cannot be modified in parallel I/O mode. Use the serial I/O or CPU rewrite mode to rewrite the contents of the ROM Code Protect Reset Bits.

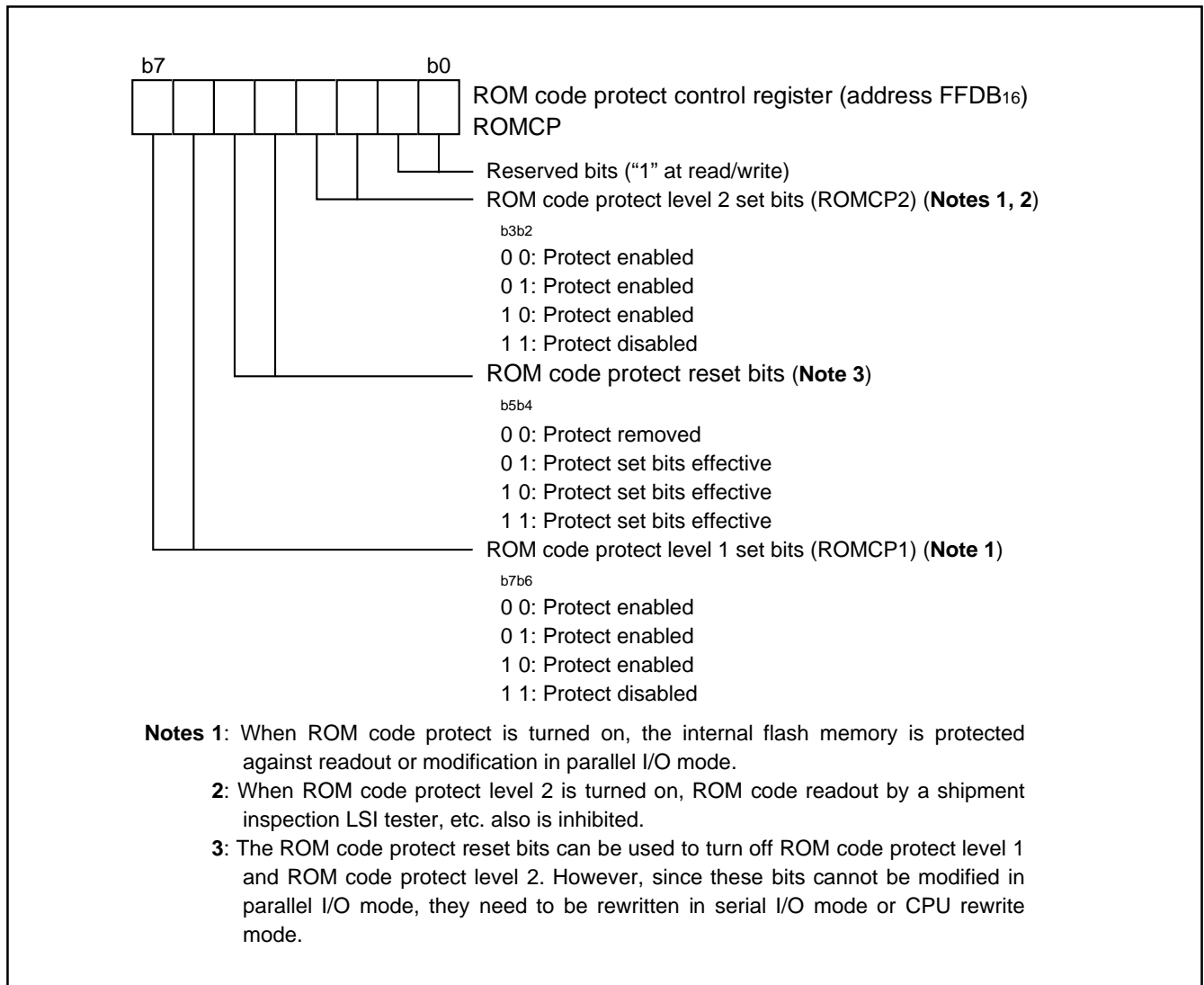


Fig. 121 Structure of ROM code protect control register

### ID Code Check Function

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the programmer is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the programmer are not accepted. The ID code consists of 8-bit data, and its areas are FFDA<sub>16</sub> to FFDA<sub>16</sub>. Write a program which has had the ID code preset at these addresses to the flash memory.

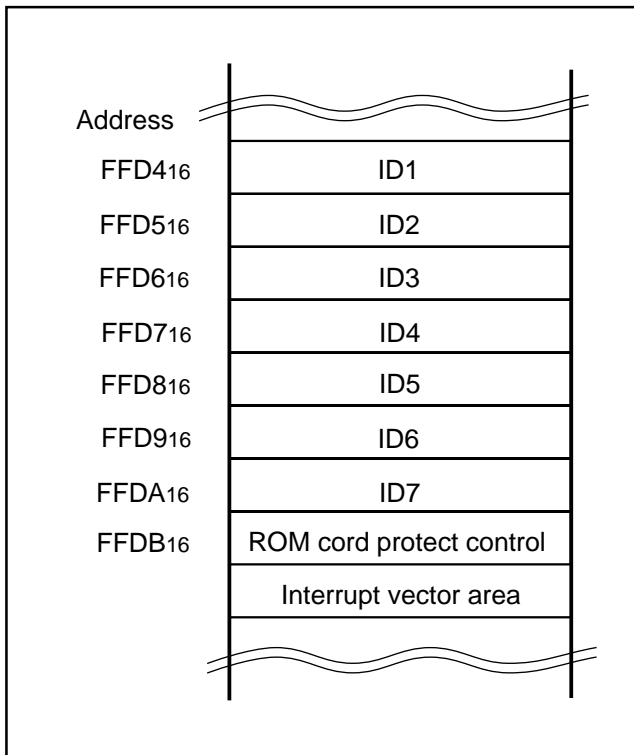


Fig. 122 ID code store addresses

## **(2) Parallel I/O Mode**

Parallel I/O mode is the mode which parallel output and input software command, address, and data required for the operations (read, program, erase, etc.) to a built-in flash memory. Use the exclusive external equipment flash programmer which supports the 38K0 Group (flash memory version). Refer to each programmer maker's handling manual for the details of the usage.

### **User ROM and Boot ROM Areas**

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 115 can be rewritten. Both areas of flash memory can be operated on in the same way.

The boot ROM area is 4 Kbytes in size. It is located at addresses F000<sub>16</sub> through FFFF<sub>16</sub>. 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, an erase block operation is applied to only one 4 Kbyte block.

The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the Mitsubishi factory. Therefore, using the device in standard serial I/O mode, you must perform program and block erase in the user ROM area.

### (3) Standard Serial I/O Mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is clock synchronized serial. This mode requires a purpose-specific peripheral unit. The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU rewrite mode), rewrite data input and so forth. The standard serial I/O mode is started by connecting "H" to the P16 (CE) pin and "H" to the P42 (SCLK) pin and "H" to the CNVSS (VPP) pin (apply 4.5 V to 5.25 V to Vpp from an external source), and releasing the reset operation. (In the ordinary microcomputer mode, set CNVSS pin to "L" level.)

This control program is written in the Boot ROM area when the product is shipped from Mitsubishi. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the Boot ROM area is rewritten in parallel I/O mode. Figure 123 shows the pin connections for the standard serial I/O mode.

In standard serial I/O mode, serial data I/O uses the four serial I/O pins SCLK, RxD, TxD and  $\overline{\text{SRDY}}$  (BUSY). The SCLK pin is the transfer clock input pin through which an external transfer clock is input. The TxD pin is for CMOS output. The  $\overline{\text{SRDY}}$  (BUSY) pin outputs "L" level when ready for reception and "H" level when reception starts.

Serial data I/O is transferred serially in 8-bit units.

In standard serial I/O mode, only the User ROM area shown in Figure 115 can be rewritten. The Boot ROM area cannot.

In standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, commands sent from the peripheral unit (programmer) are not accepted unless the ID code matches.

### Outline Performance (Standard Serial I/O Mode)

In standard serial I/O mode, software commands, addresses and data are input and output between the MCU and peripheral units (serial programmer, etc.) using 4-wire clock-synchronized serial I/O. In reception, software commands, addresses and program data are synchronized with the rise of the transfer clock that is input to the SCLK pin, and are then input to the MCU via the RxD pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD pin.

The TxD pin is for CMOS output. Transfer is in 8-bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the  $\overline{\text{SRDY}}$  (BUSY) pin is "H" level. Accordingly, always start the next transfer after the  $\overline{\text{SRDY}}$  (BUSY) pin is "L" level.

Also, data and status registers in a memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following explains software commands, status registers, etc.

Table 11 Description of pin function (Standard Serial I/O Mode)

Pin name	Signal name	I/O	Function
VCC,VSS	Power supply		Apply 3.00 to 5.25 V to the Vcc pin and 0 V to the Vss pin.
VCC <sub>E</sub>	Power supply		Connect this pin to Vcc.
CNV <sub>SS</sub>	V <sub>PP</sub>	I	Connect this pin to V <sub>PP</sub> (V <sub>PP</sub> = 4.50 to 5.25 V).
CNV <sub>SS2</sub>	CNV <sub>SS2</sub>	I	Connect this pin to Vss.
VREF	Analog reference voltage	I	Connect this pin to Vcc when not using.
DVCC, PVCC	Analog power supply		Connect this pin to Vcc.
PVSS	Analog power supply		Connect this pin to Vss.
RESET	Reset input	I	To reset, input "L" level for 20 cycles or longer clocks of $\phi$ .
XIN	Clock input	I	Connect a ceramic or crystal resonator between the XIN and XOUT pins. When entering an externally driven clock, enter it from XIN and leave XOUT open.
XOUT	Clock output	O	
USBVREF	USB reference voltage input	I	Connect this pin to Vcc when not using.
TrON	USB reference voltage output		Leave this pin open when not using.
D0+,D0-	USB upstream input	I/O	Input "L" level when not using.
P00 to P07	Input port P0	I	Input "L" or "H" level, or keep open.
P10 to P15	Input port P1	I	Input "L" or "H" level, or keep open.
P16	Input port P1	I	Input "L" or "H" level, or keep open. Input "H" level only at release of reset.
P17	Input port P1	I	Input "L" or "H" level, or keep open.
P20 to P27	Input port P2	I	Input "L" or "H" level, or keep open.
P30 to P37	Input port P3	I	Input "L" or "H" level, or keep open.
P40	RxD input	I	This is a serial data input pin.
P41	TxD output	O	This is a serial data output pin.
P42	SCLK input	I	This is a serial clock input pin. Input "H" level only at release of reset.
P43	BUSY output	O	This is a BUSY output pin.
P50 to P57	Input port P5	I	Input "L" or "H" level, or keep open.
P60 to P63	Input port P6	I	Input "L" or "H" level, or keep open.

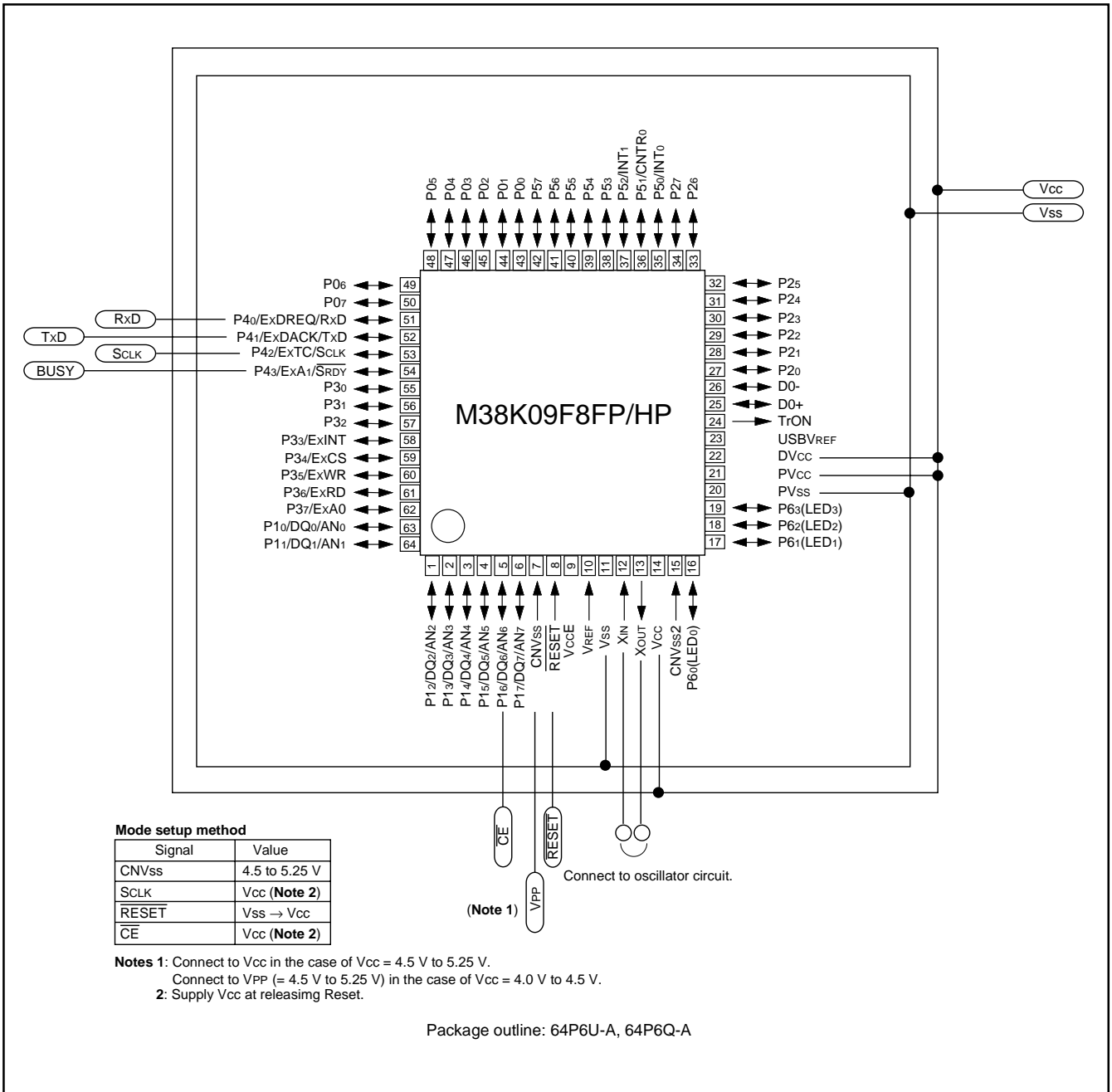


Fig. 123 Pin connection diagram in standard serial I/O mode (1)

## Software Commands

Table 12 lists software commands. In standard serial I/O mode, erase, program and read are controlled by transferring software commands via the RxD pin. Software commands are explained

here below. Basically, the software commands of the standard serial I/O mode are the same as that of the parallel I/O mode, but the block erase function is excluded, and 4 commands are added: ID check, download, version data output and Boot ROM area output functions.

**Table 12 Software commands (Standard serial I/O mode)**

Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte	.....	When ID is not verified
1 Page read	FF <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2 Page program	41 <sub>16</sub>	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3 Erase all blocks	A7 <sub>16</sub>	D0 <sub>16</sub>						Not acceptable
4 Read status register	70 <sub>16</sub>	SRD output	SRD1 output					Acceptable
5 Clear status register	50 <sub>16</sub>							Not acceptable
6 ID check function	F5 <sub>16</sub>	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
7 Download function	FA <sub>16</sub>	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
8 Version data output function	FB <sub>16</sub>	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
9 Boot ROM area output function	FC <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable

**Notes1:** Shading indicates transfer from the internal flash memory microcomputer to a programmer. All other data is transferred from a programmer to the internal flash memory microcomputer.

**2:** SRD refers to status register data. SRD1 refers to status register 1 data.

**3:** All commands can be accepted when the flash memory is totally blank.

**4:** Address low is A<sub>0</sub> to A<sub>7</sub>; Address middle is A<sub>8</sub>to A<sub>15</sub>; Address high is A<sub>16</sub>to A<sub>23</sub>. Address-high A<sub>16</sub>to A<sub>23</sub> are always "00<sub>16</sub>".



The contents of software commands are explained as follows.

●Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

- (1) Transfer the "FF16" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0 to D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first synchronized with the fall of the clock.

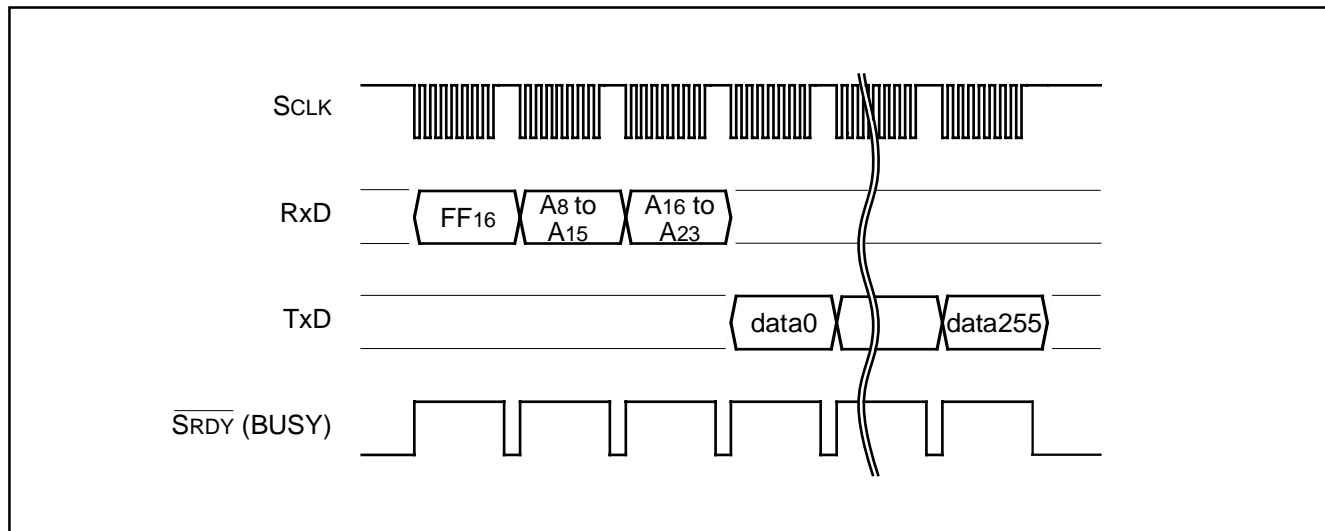


Fig. 124 Timing for page read

●Read Status Register Command

This command reads status information. When the "7016" command code is transferred with the 1st byte, the contents of the status register (SRD) with the 2nd byte and the contents of status register 1 (SRD1) with the 3rd byte are read.

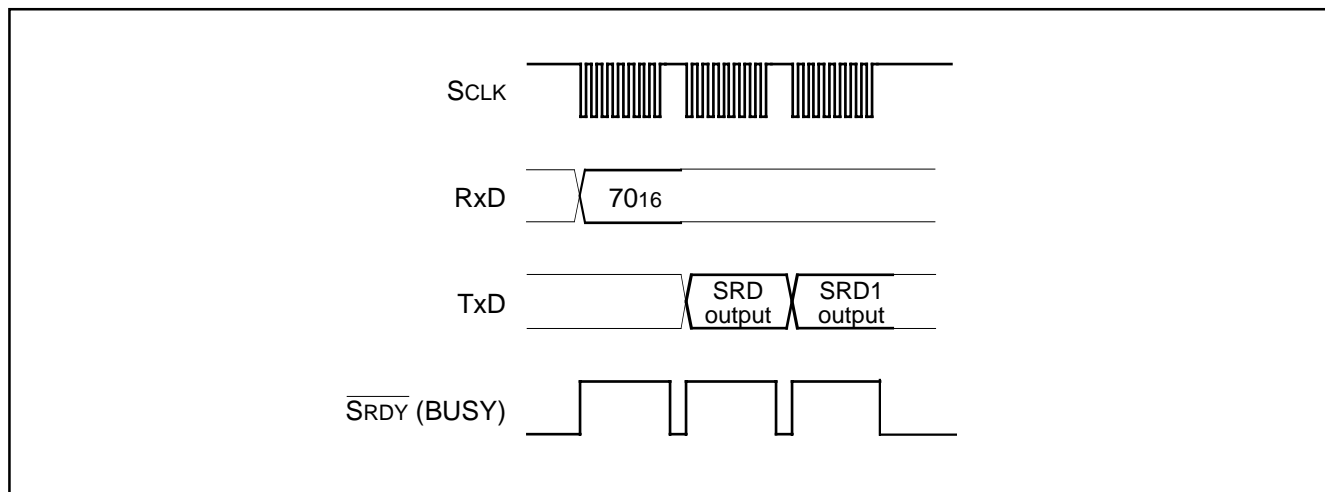


Fig. 125 Timing for reading status register

●Clear Status Register Command

This command clears the bits (SR3 to SR5) which are set when the status register operation ends in error. When the "5016" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the  $\overline{\text{SRDY}}$  (BUSY) signal changes from "H" to "L" level.

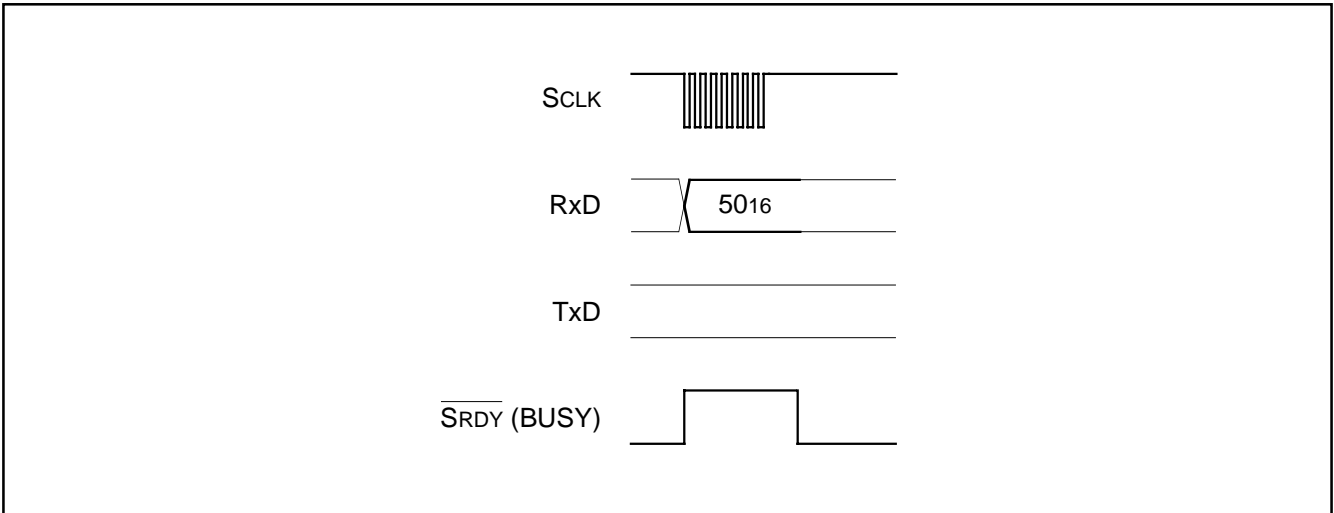


Fig. 126 Timing for clear status register

●Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

- (1) Transfer the "4116" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

- (3) From the 4th byte onward, as write data (D0 to D7) for the page (256 bytes) specified with addresses A8 to A23 is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the  $\overline{\text{SRDY}}$  (BUSY) signal changes from "H" to "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

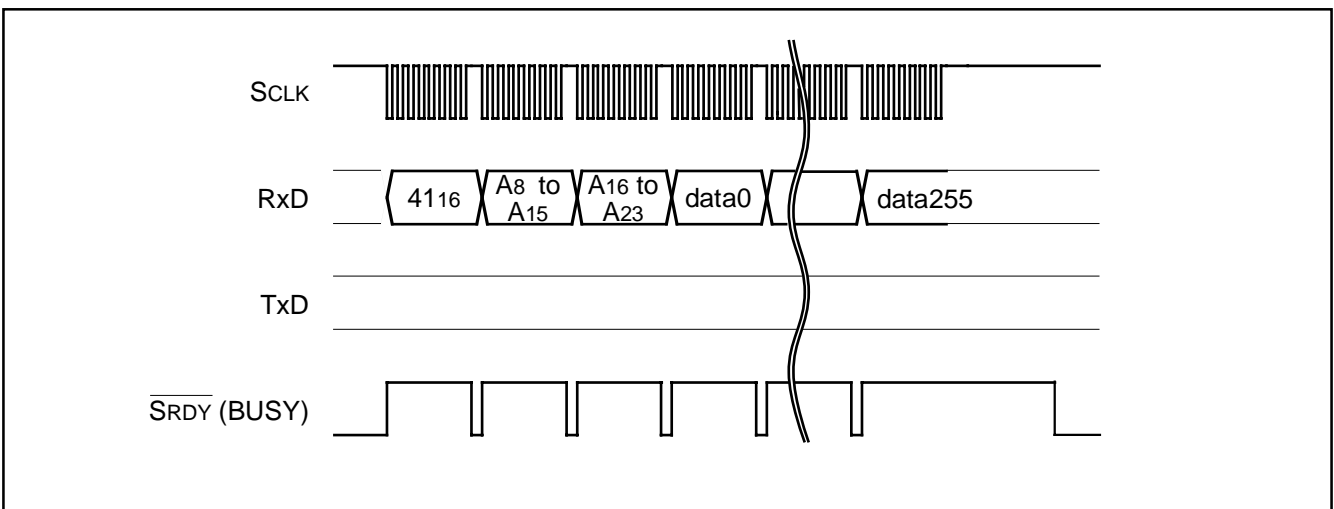


Fig. 127 Timing for page program

●Erase All Blocks Command

This command erases the contents of all blocks. Execute the erase all blocks command as explained here following.

- (1) Transfer the "A7<sub>16</sub>" command code with the 1st byte.
- (2) Transfer the verify command code "D0<sub>16</sub>" with the 2nd byte.

With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When erase all blocks end, the  $\overline{\text{SRDY}}$  (BUSY) signal changes from "H" to "L" level. The result of the erase operation can be known by reading the status register.

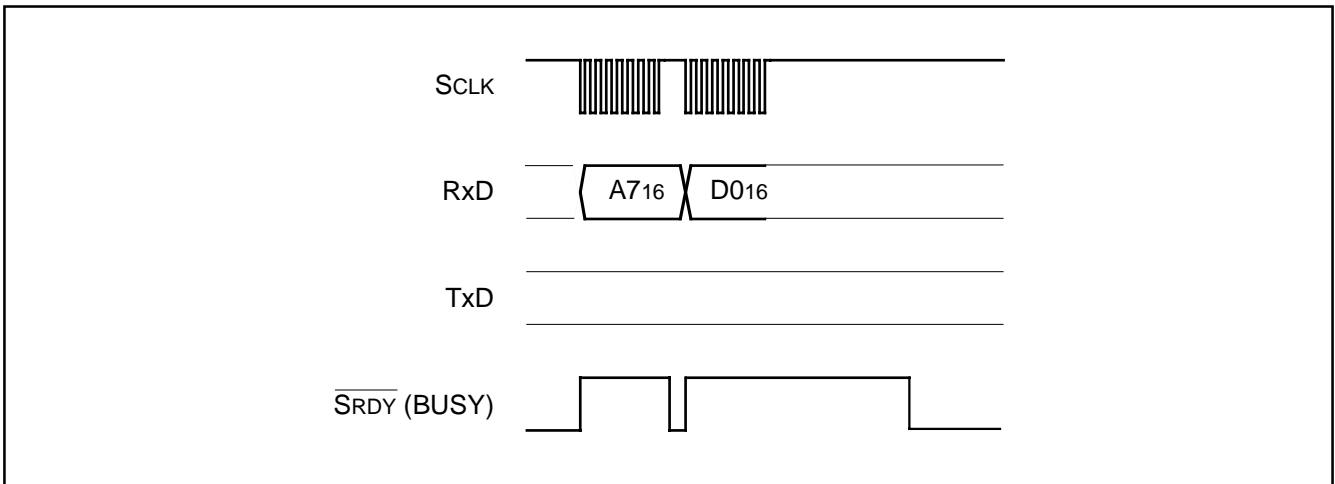


Fig. 128 Timing for erase all blocks

●Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the "FA16" command code with the 1st byte.
- (2) Transfer the program size with the 2nd and 3rd bytes.
- (3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
- (4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

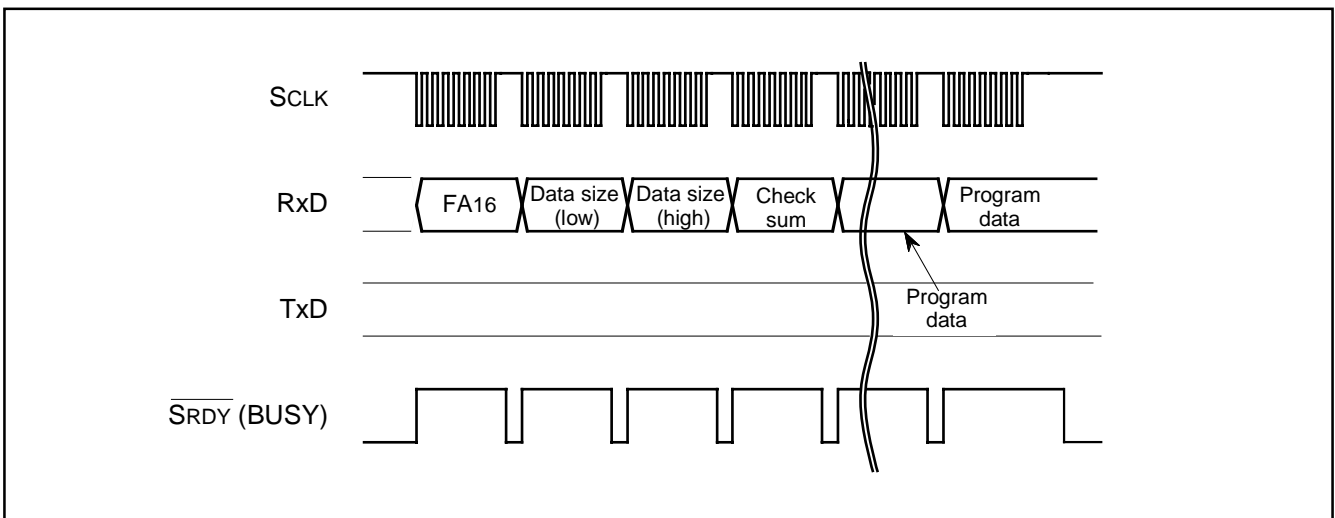


Fig. 129 Timing for download

●Version Information Output Command

This command outputs the version information of the control program stored in the Boot ROM area. Execute the version information output command as explained here following.

- (1) Transfer the "FB16" command code with the 1st byte.
  - (2) The version information will be output from the 2nd byte onward.
- This data is composed of 8 ASCII code characters.

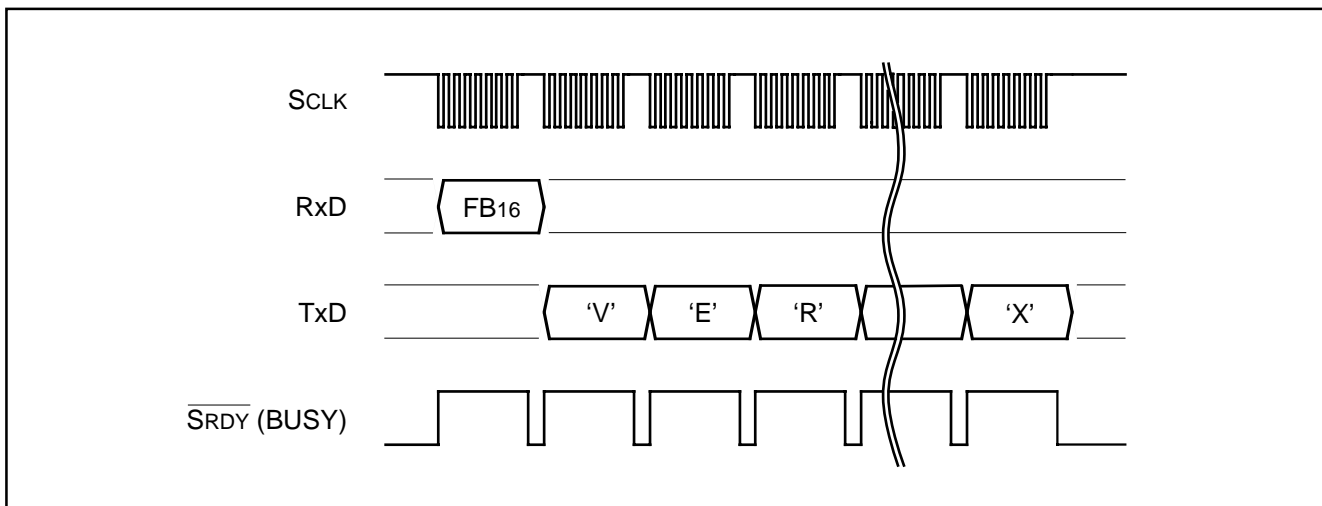


Fig. 130 Timing for version information output

●Boot ROM Area Output Command

This command reads the control program stored in the Boot ROM area in page (256 bytes) unit. Execute the Boot ROM area output command as explained here following.

- (1) Transfer the "FC16" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0 to D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first synchronized with the fall of the clock.

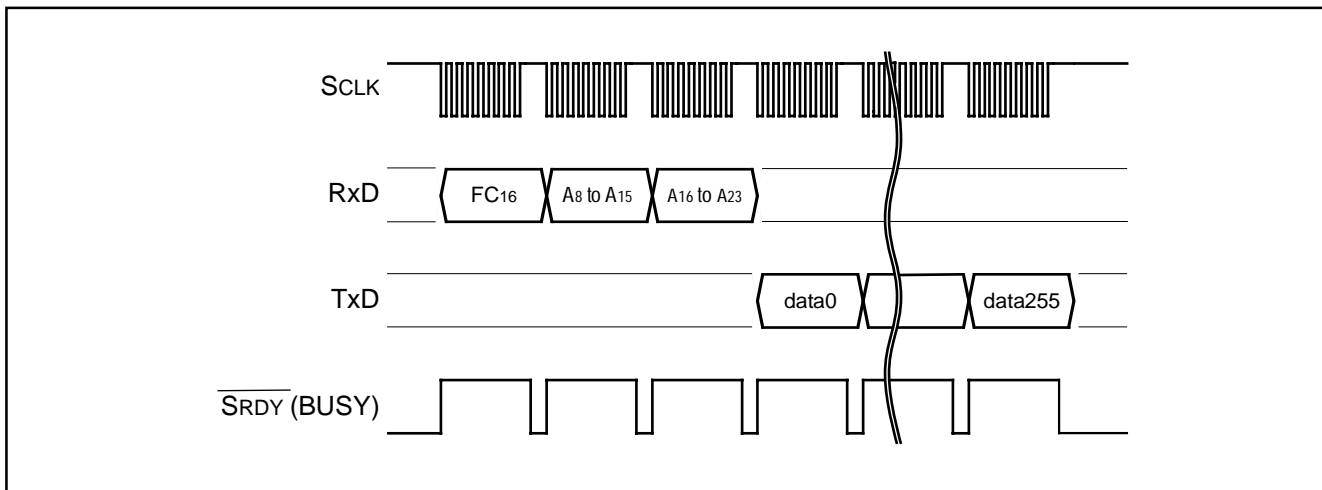


Fig. 131 Timing for Boot ROM area output

●ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

- (1) Transfer the "F5<sub>16</sub>" command code with the 1st byte.
- (2) Transfer addresses A<sub>0</sub> to A<sub>7</sub>, A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> ("00<sub>16</sub>") of the 1st byte of the ID code with the 2nd, 3rd and 4th respectively.
- (3) Transfer the number of data sets of the ID code with the 5th byte.
- (4) Transfer the ID code with the 6th byte onward, starting with the 1st byte of the code.

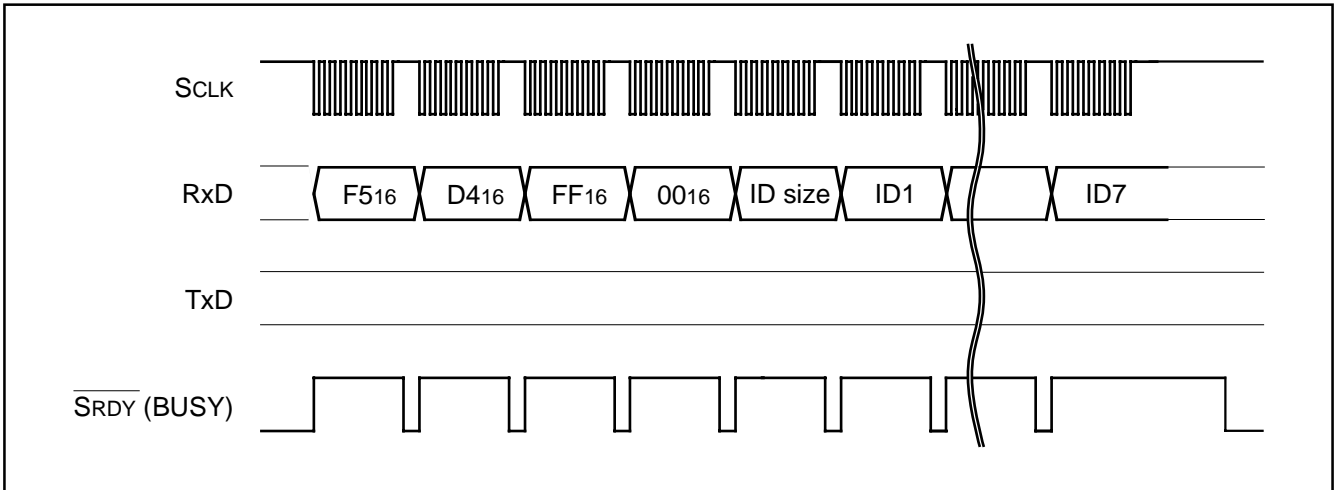


Fig. 132 Timing for ID check

●ID Code

When the flash memory is not blank, the ID code sent from the serial programmer and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the serial programmer is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses FFD4<sub>16</sub> to FFDA<sub>16</sub>. Write a program into the flash memory, which already has the ID code set for these addresses.

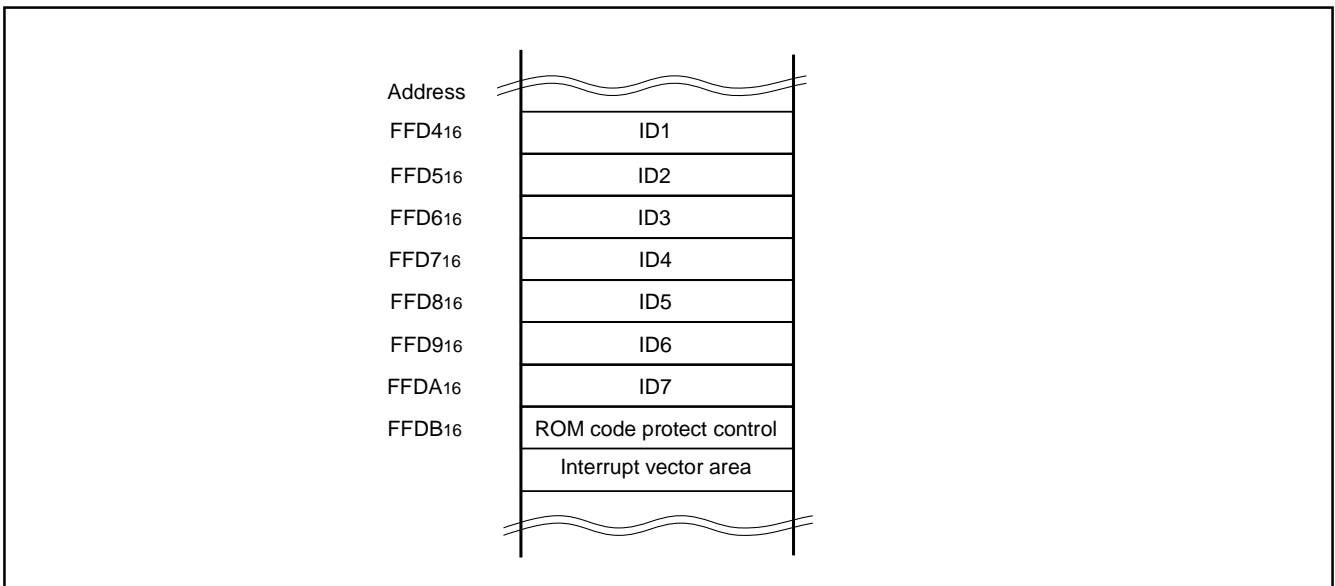


Fig. 133 ID code storage addresses

●**Status Register (SRD)**

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70<sub>16</sub>). Also, the status register is cleared by writing the clear status register command (50<sub>16</sub>).

Table 13 lists the definition of each status register bit. After releasing the reset, the status register becomes "80<sub>16</sub>".

•**Sequencer status (SR7)**

The sequencer status indicates the operating status of the flash memory.

After power-on and recover from deep power down mode, the sequencer status is set to "1" (ready).

This status bit is set to "0" (busy) during write or erase operation and is set to "1" upon completion of these operations.

•**Erase status (SR5)**

The erase status indicates the operating status of erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is set to "0".

•**Program status (SR4)**

The program status indicates the operating status of write operation. If a write error occurs, it is set to "1". When the program status is cleared, it is set to "0".

**Table 13 Status register (SRD)**

SRD0 bits	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	-	-

●**Status Register 1 (SRD1)**

The status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70<sub>16</sub>). Also, status register 1 is cleared by writing the clear status register command (50<sub>16</sub>).

Table 14 lists the definition of each status register 1 bit. This register becomes "00<sub>16</sub>" when power is turned on and the flag status is maintained even after the reset.

•**Boot update completed bit (SR15)**

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

•**Check sum consistency bit (SR12)**

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

•**ID check completed bits (SR11 and SR10)**

These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

•**Data reception time out (SR9)**

This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the MCU returns to the command wait state.

**Table 14 Status register 1 (SRD1)**

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not Update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 01 10 11	Not verified Verification mismatch Reserved Verified
SR9 (bit1)	Data reception time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-



**Full Status Check**

Results from executed erase and program operations can be known by running a full status check. Figure 124 shows a flow-chart of the full status check and explains how to remedy errors which occur.

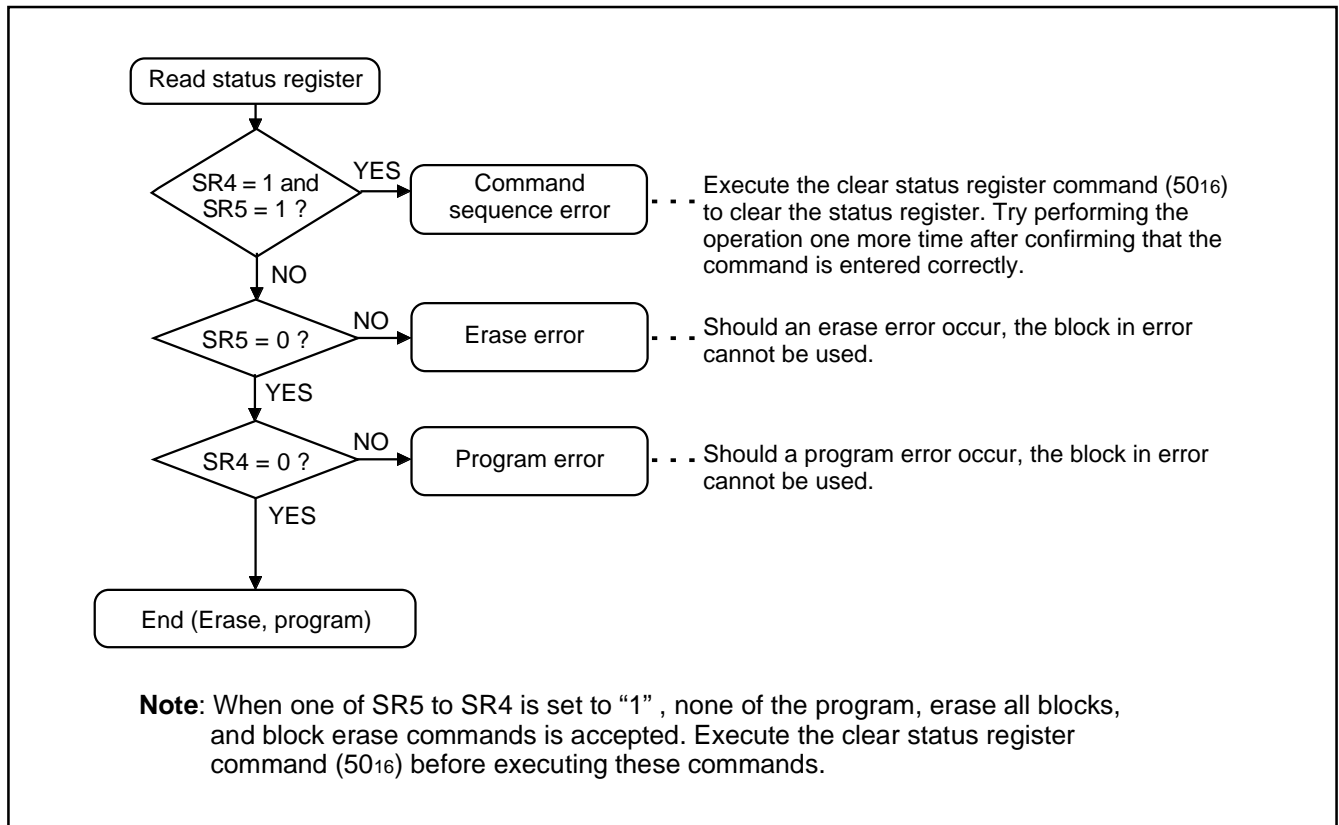


Fig. 134 Full status check flowchart and remedial procedure for errors

**Example Circuit Application for Standard Serial I/O Mode**

Figure 135 shows a circuit application for the standard serial I/O mode. Control pins will vary according to a programmer, therefore see a programmer manual for more information.

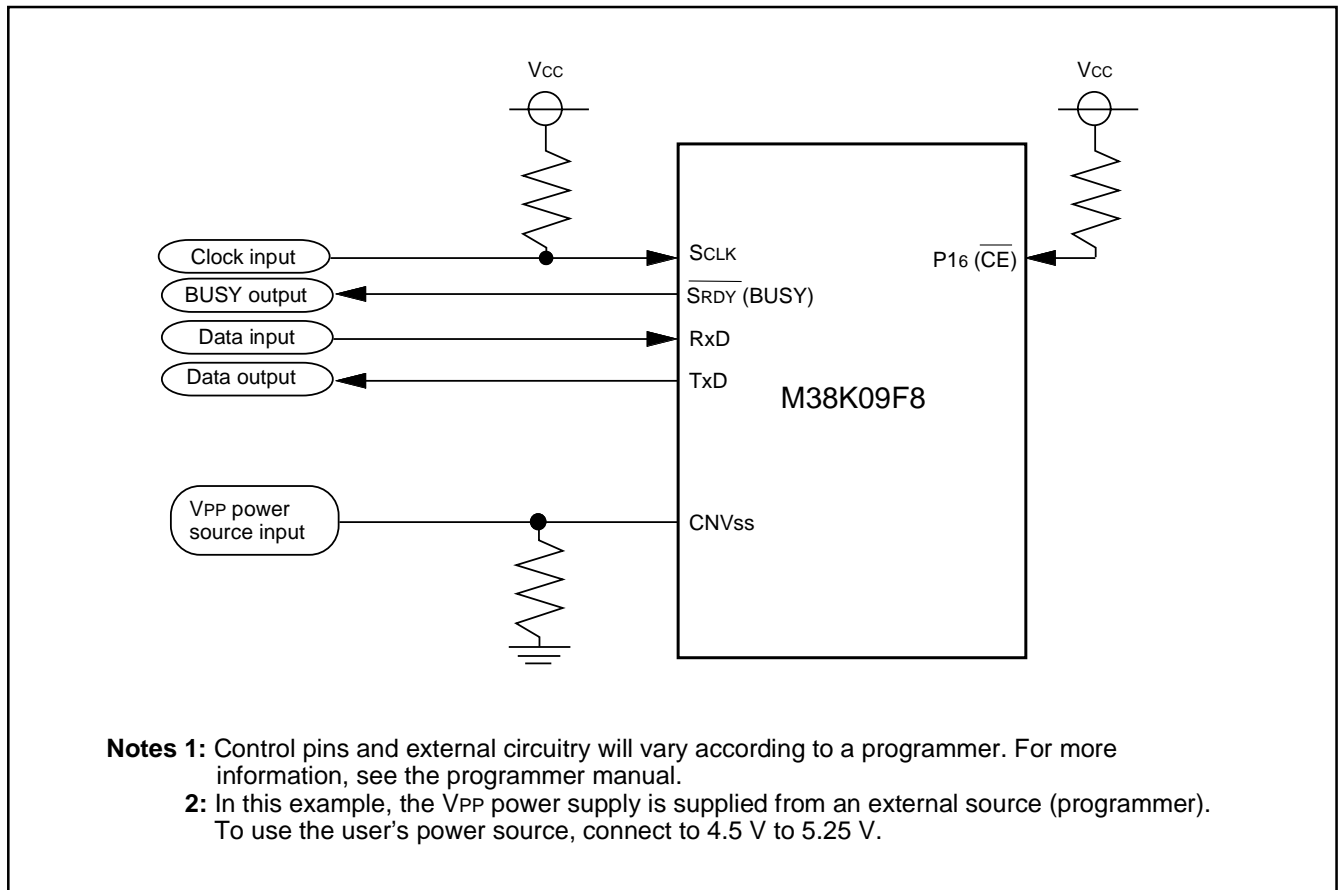


Fig. 135 Example circuit application for standard serial I/O mode

## NOTES ON PROGRAMMING

### Processor Status Register

The contents of the processor status register (PS) after a reset are undefined, except for the interrupt disable flag (I) which is "1." After a reset, initialize flags which affect program execution. In particular, it is essential to initialize the index X mode (T) and the decimal mode (D) flags because of their effect on calculations.

### Interrupts

The contents of the interrupt request bits do not change immediately after they have been written. After writing to an interrupt request register, execute at least one instruction before performing a BBC or BBS instruction.

### Decimal Calculations

- To calculate in decimal notation, set the decimal mode flag (D) to "1", then execute an ADC or SBC instruction. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.
- In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.

### Timers

- When n (0 to 255) is written to a timer latch, the frequency division ratio is  $1/(n+1)$ .
- When a count source of timer X is switched, stop a count of timer X.

### Multiplication and Division Instructions

- The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.
- The execution of these instructions does not change the contents of the processor status register.

### Ports

The contents of the port direction registers cannot be read. The following cannot be used:

- The data transfer instruction (LDA, etc.)
- The operation instruction when the index X mode flag (T) is "1"
- The addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instructions (ROR, CLB, or SEB, etc.) to a direction register.

Use instructions such as LDM and STA, etc., to set the port direction registers.

### A-D Converter

The comparator uses capacitive coupling amplifier whose charge will be lost if the clock frequency is too low.

Therefore, make sure that  $f(\text{system clock})$  in the middle/high-speed mode is at least on 500 kHz during an A-D conversion.

Do not execute the STP or WIT instruction during an A-D conversion.

### Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock  $\phi$  by the number of cycles needed to execute an instruction. However, When using the USB function or EXB function, an occurrence of one-wait due to the multichannel RAM will double an internal clock  $\phi$  cycle.

### Definition of A-D Conversion Accuracy

The A-D conversion accuracy is defined below (refer to Figure 136).

•Relative accuracy

① Zero transition voltage (V<sub>OT</sub>)

This means an analog input voltage when the actual A-D conversion output data changes from "0" to "1."

② Full-scale transition voltage (V<sub>FST</sub>)

This means an analog input voltage when the actual A-D conversion output data changes from "1023" to "1022."

③ Non-linearity error

This means a deviation from the line between V<sub>OT</sub> and V<sub>FST</sub> of a converted value between V<sub>OT</sub> and V<sub>FST</sub>.

④ Differential non-linearity error

This means a deviation from the input potential difference required to change a converted value between V<sub>OT</sub> and V<sub>FST</sub> by 1 LSB of the 1 LSB at the relative accuracy.

•Absolute accuracy

This means a deviation from the ideal characteristics between 0 to V<sub>REF</sub> of actual A-D conversion characteristics.

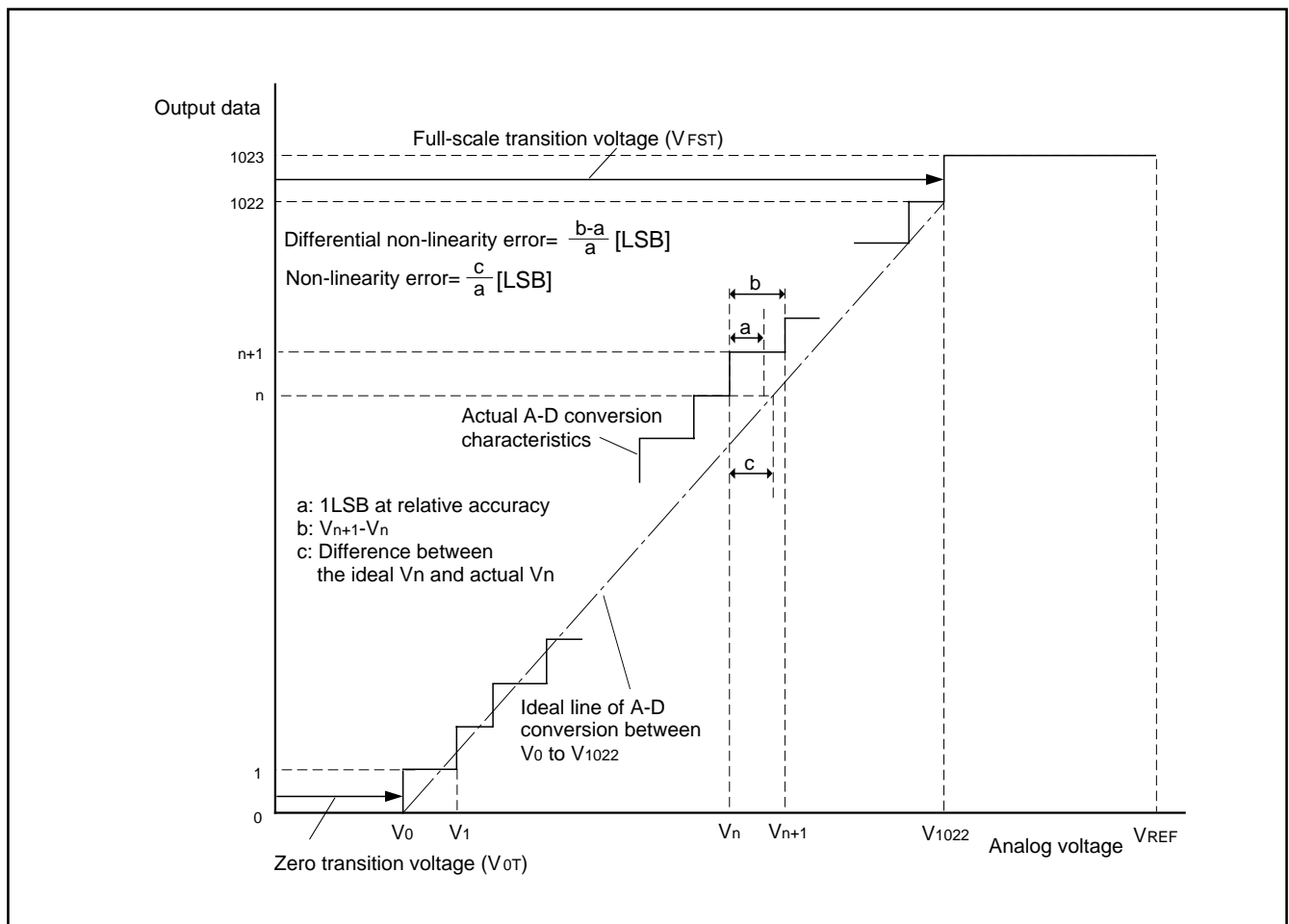


Fig. 136 Definition of A-D conversion accuracy

V<sub>n</sub>: Analog input voltage when the output data changes from "n" to "n + 1" (n = 0 to 1022)

• 1 LSB at relative accuracy  $\rightarrow \frac{V_{FST} - V_{OT}}{1022}$  (V)

• 1 LSB at absolute accuracy  $\rightarrow \frac{V_{REF}}{1024}$  (V)

## NOTES ON USAGE

### Handling of Power Source Pin

In order to avoid a latch-up occurrence, connect a capacitor suitable for high frequencies as bypass capacitor between power source pin (Vcc pin) and GND pin (Vss pin). Besides, connect the capacitor to as close as possible. For bypass capacitor which should not be located too far from the pins to be connected, a ceramic or electrolytic capacitor of 1.0  $\mu$ F is recommended.

### USB Port Pins (D0+, D0-) Treatment

•The USB specification requires a driver-impedance 28 to 44  $\Omega$ . In order to meet the USB specification impedance requirements, connect a resistor (27  $\Omega$  recommended) in series to the USB port pins.

In addition, in order to reduce the ringing and control the falling/rising timing and a crossover point, connect a capacitor between the USB port pins and the Vss pin if necessary.

The values and structure of those peripheral elements depend on the impedance characteristics and the layout of the printed circuit board. Accordingly, evaluate your system and observe waveforms before actual use and decide use of elements and the values of resistors and capacitors.

•Make sure the USB D+/D- lines do not cross any other wires. Keep a large GND area to protect the USB lines. Also, make sure you use a USB specification compliant connector for the connection.

### USBVREF pin Treatment (Noise Elimination)

•Connect a capacitor between the USBVREF pin and the Vss pin. The capacitor should have a 2.2  $\mu$ F capacitor (electrolytic capacitor) and a 0.1  $\mu$ F capacitor (ceramic type capacitor) connected in parallel.

•In Vcc = 3.0 to 3.6 V operation, connect the USBVREF pin directly to the Vcc pin in order to supply power to the USB port circuit. In addition, you will need to disable the built-in USB reference voltage circuit in this operation (set bit 4 of the USB control register to "0".) If you are using the bus powered supply in this condition, the DC-DC converter must be placed outside the MCU.

•In Vcc = 4.00 to 5.25 V operation, do not connect the external DC-DC converter to the USBVREF pin. Use the built-in USB reference voltage circuit.

### Flash Memory Version

The CNVss pin is connected to the internal memory circuit block by a low-ohmic resistance, since it has the multiplexed function to be a programmable power source pin (VPP pin) as well.

To improve the noise reduction, connect a track between CNVss pin and Vss pin or Vcc pin with 1 to 10 k $\Omega$  resistance.

The mask ROM version track of CNVss pin has no operational interference even if it is connected to Vss pin or Vcc pin via a resistor.

### Electric Characteristic Differences Between Mask ROM and Flash Memory Version MCUs

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and Flash Memory version MCUs due to the difference in the manufacturing processes.

When manufacturing an application system with the Flash Memory version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

### DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

1. Mask ROM Order Confirmation Form\*
2. Mark Specification Form\*
3. Data to be written to ROM, in EPROM form (three identical copies) or one floppy disk.

\* Mask ROM Confirmation Forms/Mark Specification Forms  
<http://www.infocom.maec.co.jp/>

**ELECTRICAL CHARACTERISTICS**

**Absolute Maximum Ratings**

**Table 15 Absolute maximum ratings**

Symbol	Parameter	Conditions	Ratings	Unit	
V <sub>CC</sub>	Power source voltage	All voltages are based on V <sub>SS</sub> . Output transistors are cut off.	-0.3 to 6.5	V	
AV <sub>CC</sub>	Analog power source voltage V <sub>CC</sub> E, V <sub>REF</sub> , PV <sub>CC</sub> , DV <sub>CC</sub> , USBV <sub>REF</sub>		-0.3 to V <sub>CC</sub> + 0.3	V	
V <sub>I</sub>	Input voltage P00-P07, P10-P17, P20-P27, P30-P37, P40-P43, P50-P57, P60-P63		-0.3 to V <sub>CC</sub> + 0.3	V	
V <sub>I</sub>	Input voltage RESET, X <sub>IN</sub> , CNV <sub>SS</sub> 2		-0.3 to V <sub>CC</sub> + 0.3	V	
V <sub>I</sub>	Input voltage CNV <sub>SS</sub>		Mask ROM version	-0.3 to V <sub>CC</sub> + 0.3	V
			Flash memory version	-0.3 to 6.5	V
V <sub>I</sub>	Input voltage D0+, D0-		-0.5 to 3.8	V	
V <sub>O</sub>	Output voltage P00-P07, P10-P17, P20-P27, P30-P37, P40-P43, P50-P57, P60-P63, X <sub>OUT</sub>		-0.3 to V <sub>CC</sub> + 0.3	V	
V <sub>O</sub>	Output voltage D0+, D0-, TrON	-0.5 to 3.8	V		
P <sub>d</sub>	Power dissipation (Note)	T <sub>a</sub> = 25°C	500	mW	
T <sub>opr</sub>	Operating temperature	MCU operating	-20 to 85	°C	
		In flash memory mode (For flash memory version)	25±5	°C	
T <sub>stg</sub>	Storage temperature		-40 to 125	°C	

**Note:** The maximum rating value depends on not only the MCU's power dissipation but the heat consumption characteristics of the package.

**Recommended Operating Conditions**

**Table 16 Recommended operating conditions** ( $V_{cc} = 3.00$  to  $5.25$  V,  $V_{ss} = 0$  V,  $T_a = -20$  to  $85^\circ\text{C}$ , unless otherwise noted)

Symbol	Parameter		Limits			Unit	
			Min.	Typ.	Max.		
Vcc	Power source voltage	Vcc	System clock 12 MHz (2-/4-/8-divide mode)	4.00	5.00	5.25	V
			System clock 8 MHz	4.00	5.00	5.25	V
			System clock 6 MHz	3.00	5.00	5.25	V
AVcc	Analog power source voltage	PVcc, DVcc		Vcc		V	
AVcc	Analog power source voltage	VcCE		Vcc		V	
VREF	Analog reference voltage	VREF	2.0		Vcc	V	
VREF	Analog reference voltage	USBVREF	Vcc = 3.6 to 4.0 V	3.0		3.6	V
			Vcc = 3.0 to 3.6 V	3.0		Vcc	V
Vss	Power source voltage	Vss		0		V	
AVss	Analog power source voltage	PVss		0		V	
VIH	"H" input voltage	P00–P07, P20–P27, P50–P57, P60–P63	0.8Vcc		Vcc	V	
		P10–P17, P30–P37, P40–P43	0.8VcCE		VcCE	V	
VIH	"H" input voltage	RESET, XIN, CNVss, CNVss2	0.8Vcc		Vcc	V	
VIH	"H" input voltage	D0+, D0-	2.0		3.6	V	
VIL	"L" input voltage	P00–P07, P20–P27, P50–P57, P60–P63	0		0.2Vcc	V	
		P10–P17, P30–P37, P40–P43	0		0.2VcCE	V	
VIL	"L" input voltage	RESET, XIN, CNVss, CNVss2	0		0.2Vcc	V	
VIL	"L" input voltage	D0+, D0-	0		0.8	V	

**Table 17 Recommended operating conditions** ( $V_{cc} = 3.00$  to  $5.25$  V,  $V_{ss} = 0$  V,  $T_a = -20$  to  $85^\circ\text{C}$ , unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$\Sigma I_{OH}(\text{peak})$	"H" total peak output current ( <b>Note 1</b> )	P00–P07, P20–P27, P50–P57, P60–P63			–80	mA
$\Sigma I_{OH}(\text{peak})$	"H" total peak output current ( <b>Note 1</b> )	P10–P17, P30–P37, P40–P43			–80	mA
$\Sigma I_{OL}(\text{peak})$	"L" total peak output current ( <b>Note 1</b> )	P00–P07, P20–P27, P50–P57			80	mA
$\Sigma I_{OL}(\text{peak})$	"L" total peak output current ( <b>Note 1</b> )	P60–P63			80	mA
$\Sigma I_{OL}(\text{peak})$	"L" total peak output current ( <b>Note 1</b> )	P10–P17, P30–P37, P40–P43			80	mA
$\Sigma I_{OH}(\text{avg})$	"H" total average output current ( <b>Note 1</b> )	P00–P07, P20–P27, P50–P57, P60–P63			–40	mA
$\Sigma I_{OH}(\text{avg})$	"H" total average output current ( <b>Note 1</b> )	P10–P17, P30–P37, P40–P43			–40	mA
$\Sigma I_{OL}(\text{avg})$	"L" total average output current ( <b>Note 1</b> )	P00–P07, P20–P27, P50–P57			40	mA
$\Sigma I_{OL}(\text{avg})$	"L" total average output current ( <b>Note 1</b> )	P60–P63			40	mA
$\Sigma I_{OL}(\text{avg})$	"L" total average output current ( <b>Note 1</b> )	P10–P17, P30–P37, P40–P43			40	mA
$I_{OH}(\text{peak})$	"H" peak output current ( <b>Note 2</b> )	P00–P07, P20–P27, P50–P57, P60–P63			–10	mA
$I_{OH}(\text{peak})$	"H" peak output current ( <b>Note 2</b> )	P10–P17, P30–P37, P40–P43			–10	mA
$I_{OL}(\text{peak})$	"L" peak output current ( <b>Note 2</b> )	P00–P07, P20–P27, P50–P57			10	mA
$I_{OL}(\text{peak})$	"L" peak output current ( <b>Note 2</b> )	P60–P63			20	mA
$I_{OL}(\text{peak})$	"L" peak output current ( <b>Note 2</b> )	P10–P17, P30–P37, P40–P43			10	mA
$I_{OH}(\text{avg})$	"H" average output current ( <b>Note 3</b> )	P00–P07, P20–P27, P50–P57, P60–P63			–5	mA
$I_{OH}(\text{avg})$	"H" average output current ( <b>Note 3</b> )	P10–P17, P30–P37, P40–P43			–5	mA
$I_{OL}(\text{avg})$	"L" average output current ( <b>Note 3</b> )	P00–P07, P20–P27, P50–P57			5	mA
$I_{OL}(\text{avg})$	"L" average output current ( <b>Note 3</b> )	P60–P63			10	mA
$I_{OL}(\text{avg})$	"L" average output current ( <b>Note 3</b> )	P10–P17, P30–P37, P40–P43			5	mA
$f(X_{IN})$	Main clock input oscillation frequency ( <b>Note 4</b> )	$V_{cc} = 4.00$ to $5.25$ V	6		12	MHz
		$V_{cc} = 3.00$ to $4.00$ V	6		6	MHz
$f(X_{IN})$ or $f(SYN)$	System clock frequency	$V_{cc} = 4.00$ to $5.25$ V	6		12	MHz
		$V_{cc} = 3.00$ to $4.00$ V	6		6	MHz
$f(\phi)$	$\phi$ frequency	$V_{cc} = 4.00$ to $5.25$ V			8	MHz
		$V_{cc} = 3.00$ to $4.00$ V			6	MHz

**Notes 1:** The total peak output current is the absolute value of the peak currents flowing through all the applicable ports. The total average output current is the average value of the absolute value of the currents measured over 100 ms flowing through all the applicable ports.

**2:** The peak output current is the absolute value of the peak current flowing in each port.

**3:** The average output current is the average value of the absolute value of the currents measured over 100 ms.

**4:** The duty of oscillation frequency is 50 %.



## Electrical Characteristics

**Table 18 Electrical characteristics (1)** ( $V_{CC} = 3.00$  to  $5.25$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85^\circ\text{C}$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VOH	"H" output voltage P00-P07, P20-P27, P50-P57, P60-P63	IOH = -10 mA (VCC = 4.00 to 5.25 V)	VCC-2.0			V
		IOH = -1 mA	VCC-1.0			V
VOH	"H" output voltage P10-P17, P30-P37, P40-P43	IOH = -10 mA (VCC E = 4.00 to 5.25 V)	VCC E-2.0			V
		IOH = -1 mA	VCC E-1.0			V
VOH	"H" output voltage D0+, D0-	D+ and D- pins pull- down with 0 V via a resistor of 15 kΩ ± 5 %	2.8		3.6	V
VOL	"L" output voltage P00-P07, P20-P27, P50-P57	IOL = 10 mA (VCC = 4.00 to 5.25 V)			2.0	V
		IOL = 1 mA			1.0	V
VOL	"L" output voltage P60-P63	IOL = 20 mA (VCC = 4.00 to 5.25 V)			2.0	V
		IOL = 1 mA			1.0	V
VOL	"L" output voltage P10-P17, P30-P37, P40-P43	IOL = 10 mA (VCC E = 4.00 to 5.25 V)			2.0	V
		IOL = 1 mA (VCC E = 3.00 to 5.25 V)			1.0	V
VOL	"L" output voltage D0+, D0-	D+ and D- pins pull-up with 3.6 V via a resistor of 1.5 kΩ ± 5 %	0		0.3	V
VT+–VT-	Hysteresis CNTR0, INT0, INT1			0.6		V
VT+–VT-	Hysteresis P10/DQ0–P17/DQ7, P30–P32, P33/ExINT, P34/ExCS, P35/ExWR, P36/ExRD, P37/ ExA0, P40/ExDREQ/RxD, P41/ExDACK/ TxD, P42/ExTC/SCLK, P43/ExA1/SRDY			0.6		V
VT+–VT	Hysteresis D0+, D0-			0.25		V
VT+–VT-	Hysteresis $\overline{\text{RESET}}$			0.5		V
IiH	"H" input current P00-P07, P20-P27, P50-P57, P60-P63	Vi = VCC (Pull-ups "off")			5.0	μA
IiH	"H" input current P10-P17, P30-P37, P40-P43	Vi = VCC E			5.0	μA
IiH	"H" input current $\overline{\text{RESET}}$ , CNVss	Vi = VCC			5.0	μA
IiH	"H" input current XIN	Vi = VCC		4.0		μA
IiL	"L" input current P00-P07, P20-P27, P50-P57, P60-P63	Vi = VSS (Pull-ups "off")			-5.0	μA
IiL	"L" input current P10-P17, P30-P37, P40-P43	Vi = VSS			-5.0	μA
IiL	"L" input current $\overline{\text{RESET}}$ , CNVss, CNVss2	Vi = VSS			-5.0	μA
IiL	"L" input current XIN	Vi = VSS		-4.0		μA
		Vi = VSS (VCC = 4.00 to 5.25 V)	-20.0	-60.0	-120.0	μA
IiL	"L" input current P00-P07, P50, P52 (Pull-ups "on")	Vi = VSS	-20.0			μA
		Vi = VSS	-10.0			μA
VRAM	RAM hold voltage	When clock is stopped	2.00		5.25	V

**Table 19 Electrical characteristics (2)** ( $V_{CC} = 3.00$  to  $5.25$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85^\circ\text{C}$ , unless otherwise noted)

Symbol	Parameter	Test conditions			Limits			Unit	
					Min.	Typ.	Max.		
I <sub>CC</sub>	Power source current (Output transistor is isolated.)	Normal mode (Note 1)	$V_{CC} = 4.00$ to $5.25$ V	$f(X_{IN}) = \text{system clock} = 12$ MHz, $\phi = 6$ MHz, USB reference voltage circuit enabled		23.5	60	mA	
				$f(X_{IN}) = 12$ MHz, System clock = $\phi = 8$ MHz, USB reference voltage circuit enabled		24.5	60	mA	
				$f(X_{IN}) = 6$ MHz, System clock = $\phi = 8$ MHz, USB reference voltage circuit enabled		24.0	60	mA	
				$f(X_{IN}) = \text{system clock} = \phi = 6$ MHz, USB reference voltage circuit enabled		22.0	60	mA	
				$V_{CC} = 3.00$ to $4.00$ V	$f(X_{IN}) = \text{system clock} = \phi = 6$ MHz, USB reference voltage circuit disabled			35	mA
				$V_{CC} = 3.00$ to $3.60$ V	$f(X_{IN}) = \text{system clock} = \phi = 6$ MHz, USB reference voltage circuit disabled		13.0	30	mA
			Wait mode (Note 2)	$V_{CC} = 4.00$ to $5.25$ V	$f(X_{IN}) = 12$ MHz, System clock = $\phi = 8$ MHz, USB reference voltage circuit enabled		6.0		mA
		$V_{CC} = 3.00$ to $4.00$ V		$f(X_{IN}) = \text{system clock} = \phi = 6$ MHz, USB reference voltage circuit disabled		2.0		mA	
			Stop mode (Note 3)	$V_{CC} = 4.00$ to $5.25$ V	USB reference voltage circuit enabled Low current mode		125.0	250	$\mu\text{A}$
				$V_{CC} = 3.00$ to $5.25$ V	USB reference voltage circuit disabled $T_a = 25^\circ\text{C}$		0.1		$\mu\text{A}$
	$V_{CC} = 3.00$ to $5.25$ V	USB reference voltage circuit disabled $T_a = 85^\circ\text{C}$				10	$\mu\text{A}$		

<Test conditions>

**Notes 1:** Operating in single-chip mode

- Clock input from X<sub>IN</sub> pin (X<sub>OUT</sub> oscillator stopped)
- f<sub>USB</sub> = 48 MHz
- All USB difference-input circuits enabled
- Leaving I/O pins open
- Operating functions: PLL circuit, CPU, Timers

**2:** Operating in single-chip mode with Wait mode

- Clock input from X<sub>IN</sub> pin (X<sub>OUT</sub> oscillator stopped)
- f<sub>USB</sub> = 48 MHz
- All USB difference-input circuits enabled
- Leaving I/O pins open
- Operating functions: PLL circuit, Timers, USB receiving
- Disabled functions: CPU

**3:** Operating in single-chip mode with Stop mode

- Oscillation stopped
- All USB difference-input circuits disabled
- Leaving I/O pins open

**Table 20 A-D Converter characteristics** ( $V_{CC} = 3.00$  to  $5.25$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85^\circ\text{C}$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				10	Bits
—	Linearity error	$T_a = 25^\circ\text{C}$			$\pm 3$	LSB
—	Differential nonlinear error	$T_a = 25^\circ\text{C}$			$\pm 1.5$	LSB
VOT	Zero transition voltage	$V_{CC} = V_{REF} = 5.12$ V	0	15	35	mV
VFST	Full scale transition voltage	$V_{CC} = V_{REF} = 5.12$ V	5105	5125	5150	mV
tCONV	Conversion time				122	$t_c(XIN)$ or $t_c(fSYN)$
RLADDER	Ladder resistor			35		k $\Omega$
IVREF	Reference power source input current	A-D converter operating; $V_{REF} = 5.0$ V	50	150	200	$\mu\text{A}$
		A-D converter not operating; $V_{REF} = 5.0$ V			5	
I <sub>I(AD)</sub>	A-D port input current				5.0	$\mu\text{A}$

Timing Requirements

**Table 21 Timing requirements (1)** (V<sub>CC</sub> = 4.00 to 5.25 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>w</sub> (RESET)	Reset input "L" pulse width	2			μs
t <sub>c</sub> (XIN)	Main clock input cycle time	83			ns
t <sub>WH</sub> (XIN)	Main clock input "H" pulse width	35			ns
t <sub>WL</sub> (XIN)	Main clock input "L" pulse width	35			ns
t <sub>c</sub> (CNTR)	CNTR <sub>0</sub> input cycle time	200			ns
t <sub>WH</sub> (CNTR)	CNTR <sub>0</sub> input "H" pulse width	80			ns
t <sub>WL</sub> (CNTR)	CNTR <sub>0</sub> input "L" pulse width	80			ns
t <sub>WH</sub> (INT)	INT <sub>0</sub> , INT <sub>1</sub> input "H" pulse width	80			ns
t <sub>WL</sub> (INT)	INT <sub>0</sub> , INT <sub>1</sub> input "L" pulse width	80			ns
t <sub>c</sub> (SCLK)	Serial I/O clock input cycle time (Note)	800			ns
t <sub>WH</sub> (SCLK)	Serial I/O clock input "H" pulse width (Note)	370			ns
t <sub>WL</sub> (SCLK)	Serial I/O clock input "L" pulse width (Note)	370			ns
t <sub>su</sub> (RxD-SCLK)	Serial I/O input set up time	220			ns
t <sub>h</sub> (SCLK-RxD)	Serial I/O input hold time	100			ns

**Note:** These limits are the rating values in the clock synchronous mode, bit 6 of address 0FE0<sub>16</sub> = "1". In the UART mode, bit 6 of address 0FE0<sub>16</sub> = "0"; the rating values are set to one fourth.

**Table 22 Timing requirements (2)** (V<sub>CC</sub> = 3.00 to 4.00 V, V<sub>SS</sub> = 0 V, T<sub>a</sub> = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>w</sub> (RESET)	Reset input "L" pulse width	2			μs
t <sub>c</sub> (XIN)	Main clock input cycle time	166			ns
t <sub>WH</sub> (XIN)	Main clock input "H" pulse width	70			ns
t <sub>WL</sub> (XIN)	Main clock input "L" pulse width	70			ns
t <sub>c</sub> (CNTR)	CNTR <sub>0</sub> input cycle time	500			ns
t <sub>WH</sub> (CNTR)	CNTR <sub>0</sub> input "H" pulse width	230			ns
t <sub>WL</sub> (CNTR)	CNTR <sub>0</sub> input "L" pulse width	230			ns
t <sub>WH</sub> (INT)	INT <sub>0</sub> , INT <sub>1</sub> input "H" pulse width	230			ns
t <sub>WL</sub> (INT)	INT <sub>0</sub> , INT <sub>1</sub> input "L" pulse width	230			ns
t <sub>c</sub> (SCLK)	Serial I/O clock input cycle time (Note)	2000			ns
t <sub>WH</sub> (SCLK)	Serial I/O clock input "H" pulse width (Note)	950			ns
t <sub>WL</sub> (SCLK)	Serial I/O clock input "L" pulse width (Note)	950			ns
t <sub>su</sub> (RxD-SCLK)	Serial I/O input set up time	400			ns
t <sub>h</sub> (SCLK-RxD)	Serial I/O input hold time	200			ns

**Note:** These limits are the rating values in the clock synchronous mode, bit 6 of address 0FE0<sub>16</sub> = "1". In the UART mode, bit 6 of address 0FE0<sub>16</sub> = "0"; the rating values are set to one fourth.

## Switching Characteristics

**Table 23 Switching characteristics (1)** ( $V_{CC} = 4.00$  to  $5.25$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

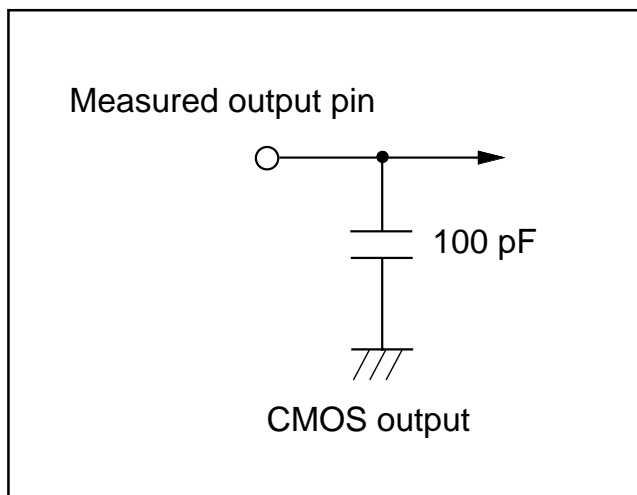
Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>WH</sub> (SCLK)	Serial I/O clock output "H" pulse width	t <sub>c</sub> (SCLK)/2-30			ns
t <sub>WL</sub> (SCLK)	Serial I/O clock output "L" pulse width	t <sub>c</sub> (SCLK)/2-30			ns
t <sub>d</sub> (SCLK-TxD)	Serial I/O output delay time			140	ns
t <sub>v</sub> (SCLK-TxD)	Serial I/O output valid time	-30			ns
t <sub>r</sub> (SCLK)	Serial I/O clock output rising time			30	ns
t <sub>f</sub> (SCLK)	Serial I/O clock output falling time			30	ns
t <sub>r</sub> (CMOS)	CMOS output rising time ( <b>Note</b> )			30	ns
t <sub>f</sub> (CMOS)	CMOS output falling time ( <b>Note</b> )			30	ns

**Notes:** Pins XOUT, D0+ and D0- are excluded.

**Table 24 Switching characteristics (2)** ( $V_{CC} = 3.00$  to  $4.00$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>WH</sub> (SCLK)	Serial I/O clock output "H" pulse width	t <sub>c</sub> (SCLK)/2-50			ns
t <sub>WL</sub> (SCLK)	Serial I/O clock output "L" pulse width	t <sub>c</sub> (SCLK)/2-50			ns
t <sub>d</sub> (SCLK-TxD)	Serial I/O output delay time			350	ns
t <sub>v</sub> (SCLK-TxD)	Serial I/O output valid time	-30			ns
t <sub>r</sub> (SCLK)	Serial I/O clock output rising time			50	ns
t <sub>f</sub> (SCLK)	Serial I/O clock output falling time			50	ns
t <sub>r</sub> (CMOS)	CMOS output rising time ( <b>Note</b> )			50	ns
t <sub>f</sub> (CMOS)	CMOS output falling time ( <b>Note</b> )			50	ns

**Notes:** Pins XOUT, D0+ and D0- are excluded.



**Fig. 137** Output switching characteristics measurement circuit

**Table 25 Switching characteristics (USB ports)** ( $V_{CC} = 3.00$  to  $5.25$  V,  $V_{SS} = 0$  V,  $T_a = -20$  to  $85$  °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$t_{fr}(D+/D-)$	USB full-speed output rising time	$CL = 50$ pF	4		20	ns
$t_{ff}(D+/D-)$	USB full-speed output falling time	$CL = 50$ pF	4		20	ns
$t_{frfm}(D+/D-)$	USB full-speed ports rising/falling ratio	$t_{fr}(D+/D-)/t_{ff}(D+/D-)$	90		111.11	%
$V_{crs}(D+/D-)$	USB output signal cross-over voltage		1.3		2.0	V

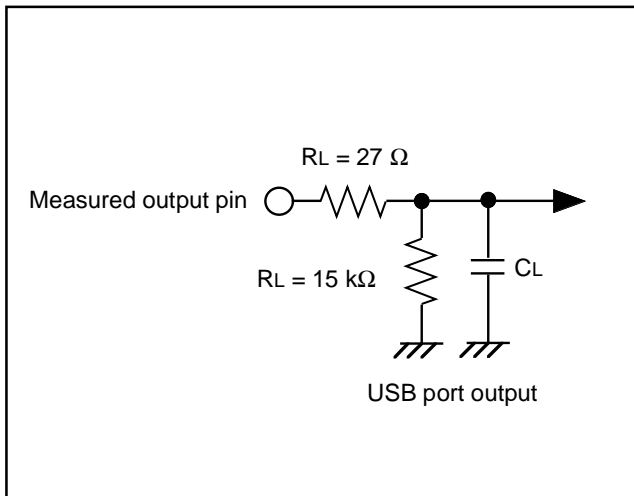


Fig. 138 USB output switching characteristics measurement circuit (1) for D0-

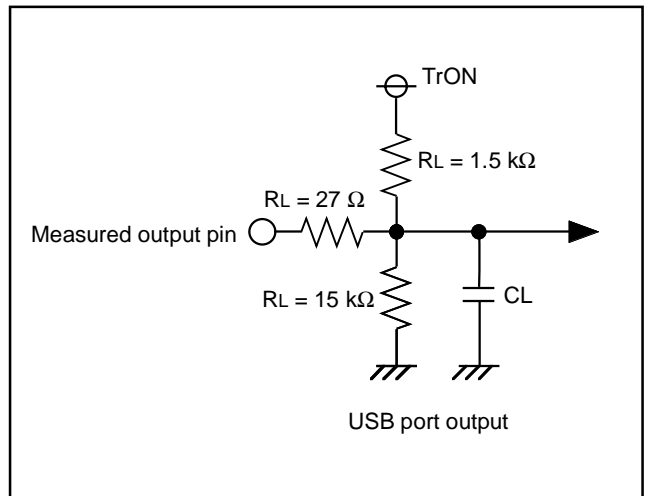


Fig. 139 USB output switching characteristics measurement circuit (2) for D0+

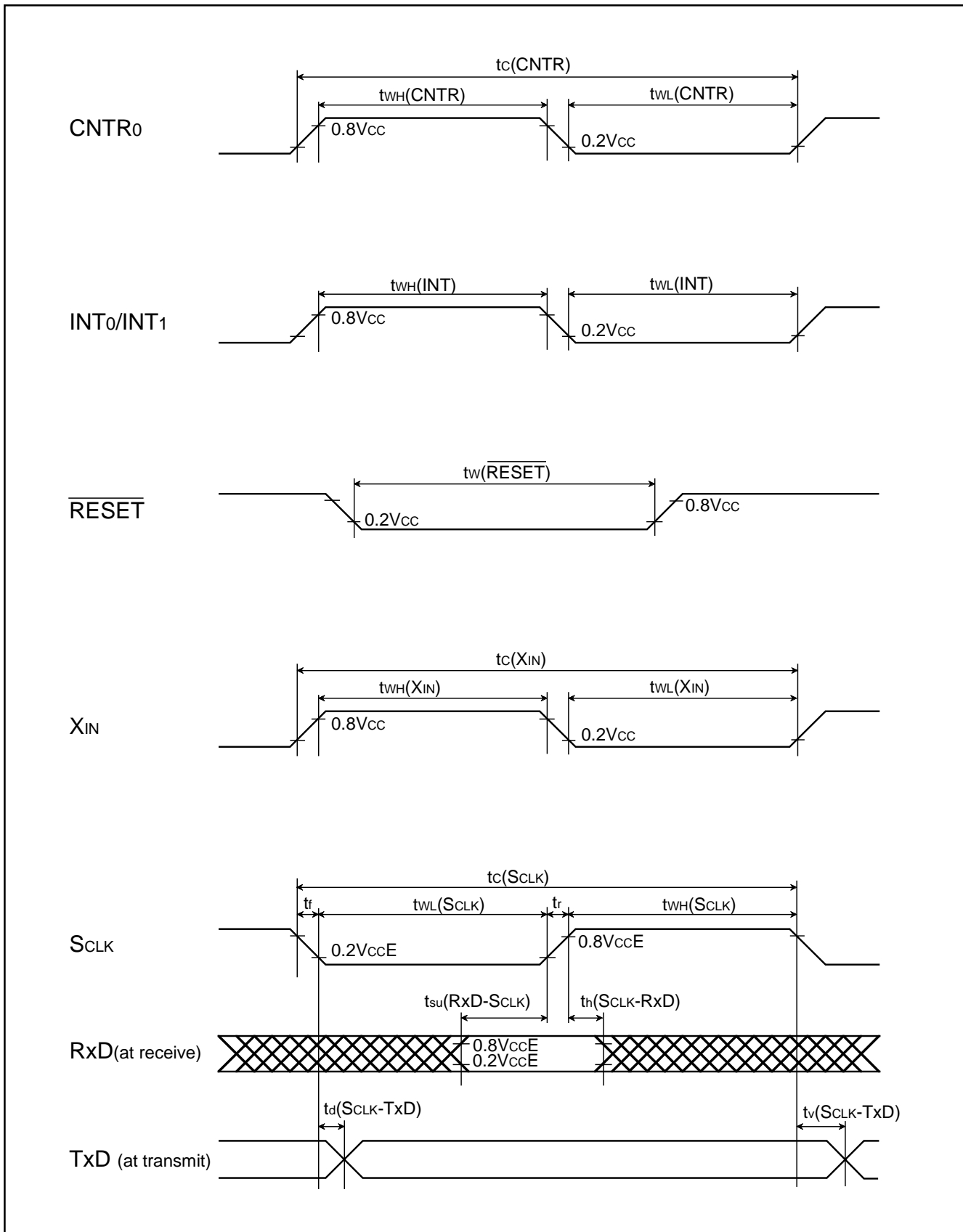


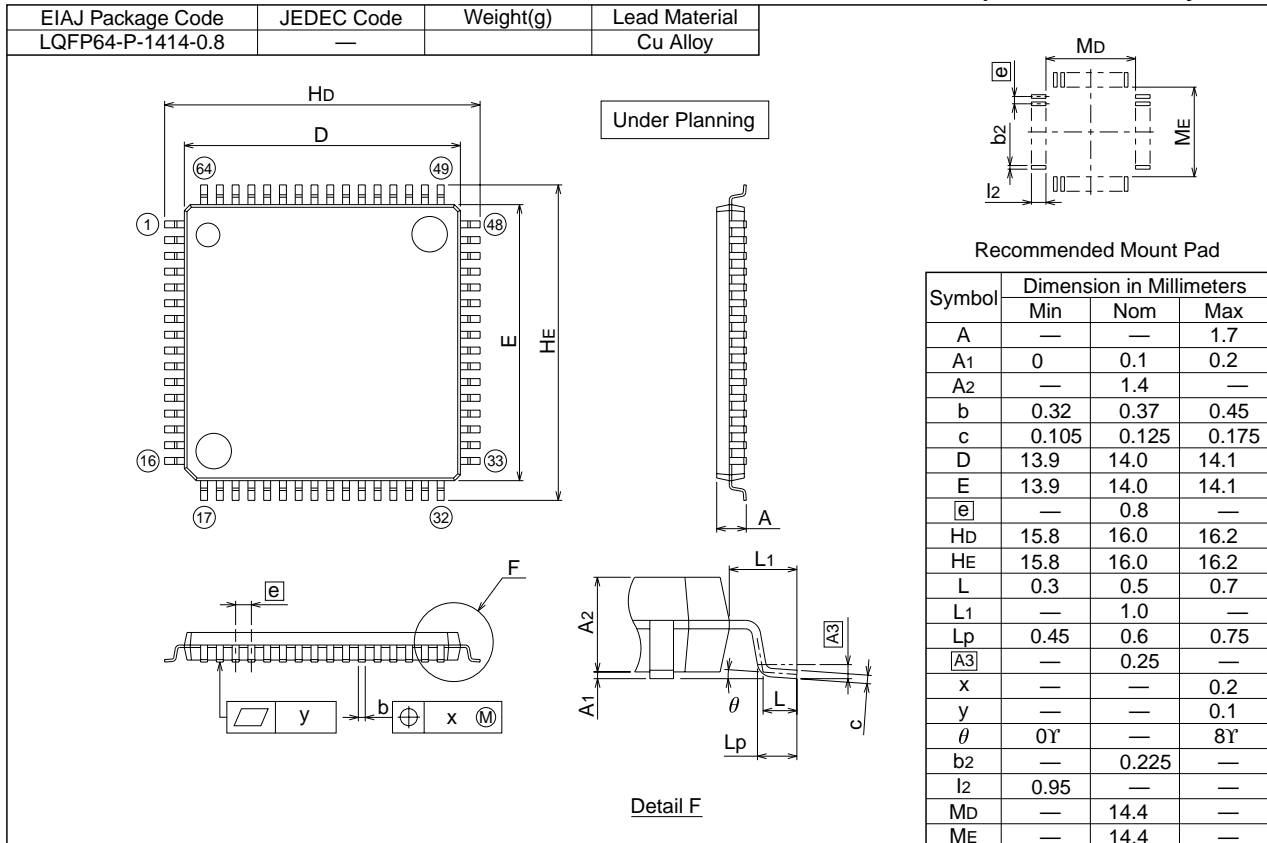
Fig. 140 Timing chart

**PACKAGE OUTLINE**

**64P6U-A**

(MMP)

Plastic 64pin 14X14mm body LQFP

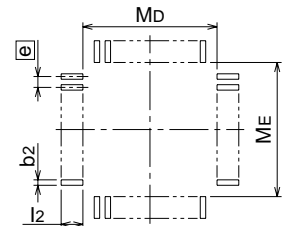
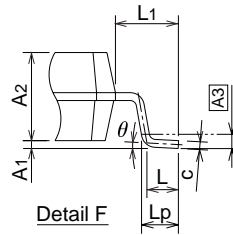
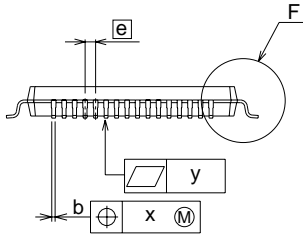
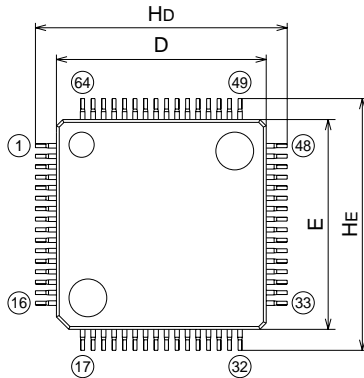




**64P6Q-A** (MMP)

**Plastic 64pin 10X10mm body LQFP**

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP64-P-1010-0.50	-	-	Cu Alloy



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	-	-	1.7
A1	0	0.1	0.2
A2	-	1.4	-
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	9.9	10.0	10.1
E	9.9	10.0	10.1
e	-	0.5	-
Hd	11.8	12.0	12.2
HE	11.8	12.0	12.2
L	0.3	0.5	0.7
L1	-	1.0	-
Lp	0.45	0.6	0.75
A3	-	0.25	-
x	-	-	0.08
y	-	-	0.1
theta	0°	-	10°
b2	-	0.225	-
l2	1.0	-	-
MD	-	10.4	-
ME	-	10.4	-



Keep safety first in your circuit designs!

- Mitsubishi Electric Corporation 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 non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor 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 Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation 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.
- 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 Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (<http://www.mitsubishichips.com>).
- 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. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor 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.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- 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.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

REVISION HISTORY

38K0 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.0	7/19/01		First edition issued
2.0	3/05/02	All pages	The symbol "PRELIMINARY" is deleted from the header.
		P. 1	Some Features are revised: Power source voltage, Power dissipation, Operating temperature range. Fig.1: The design of top view is revised.
		P. 3	Table 1: The Function of Vcc, VccE and USBVREF is revised.
		P. 5	100D0M package is added. Table 2: The product M38K09RFS is added.
		P. 9	Fig. 7: The description of system clock division ratio selection bits is revised.
		P. 25–30	The explanations from pages 25 to 30 are added.
		P. 32	Fig. 31: The Function is revised.
		P. 49	Fig. 69: The Function is revised.
		P. 57	Fig. 76: Bit name of EXBIREQ. is revised:
		P. 58	Fig. 78: Note is added. Fig. 79: Bit attributes are revised.
		P. 60	Fig. 84: Register symbol is revised.
		P. 72	The explanations of A-D converter are revised.
		P. 75	The voltages regarding RESET is revised.
		P. 76	Some explanations of PLL CIRCUIT including the clock frequency is revised.
		P. 80	Fig. 114 is added.
		P. 81	The explanations of FLASH MEMORY MODE and Table 8 are revised.
		P. 82	The explanations of Microcomputer Mode and Boot Mode, and Fig.115 are revised.
		P. 85	The explanations of Operation speed are revised.
		P. 92	The explanations of (2) Parallel I/O Mode are revised.
		P. 93	The explanations of (3) Standard Serial I/O Mode are revised.
		P. 94	Table 11: The Function of Vcc, VccE, CNVss, P10 to P15, P16 and P17 is revised.
		P. 95	Fig. 123: The descriptions of CE and SCLK are added.
		P. 106	Fig. 135: P16 (CE) is added.
		P. 107	The explanations of Instruction Execution Time are revised.
		P. 108	The explanations of Definition of A-D Conversion Accuracy is added.
		P. 109	The explanations are added: USB Port Pins, USBVREF pin Treatment and Electric Characteristic Differences Between Mask ROM and Flash Memory Version MCUs.

REVISION HISTORY

38K0 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
2.0	3/05/02	P. 110	Table 15: Operating temperature is revised.
		P. 111	Table 16: Measuring conditions, Power source voltage Vcc and Analog power source voltage VccE are revised. Analog power source voltage USBVREF is added.
		P. 112	Table 17: Measuring conditions, f(XIN) and Notes 1 and 2 are revised. [f(XIN) or f(SYN)] and f( $\phi$ ) are added.
		P. 113	Table 18: Measuring conditions and some of VOH, VOL, VT+–VT- and IIL are revised or added.
		P. 114	Table 19: The information are revised.
		P. 115	Table 20: Measuring conditions and IVREF are revised.
		P. 116 to 118	Tables 21 to 25: The information are revised or added.
		P. 118	Figures 138 and 139 are added.
		P. 119	Fig. 140 is revised.