To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

38K0 Group

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

REJ03B0192-0300
Rev.3.00
Oct 05, 2006

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 Interface, 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.

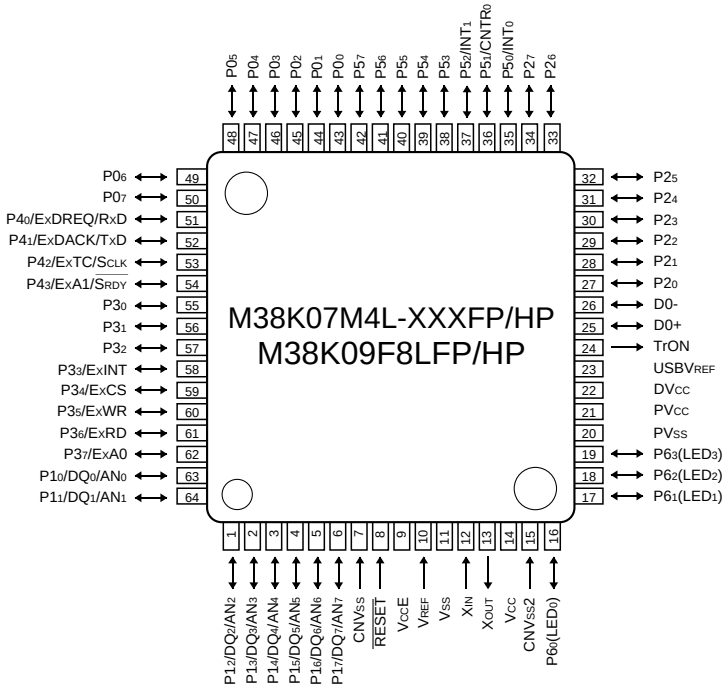
FEATURES

- 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 (Full-Speed USB2.0 specification) ..... 4 endpoints
- External bus interface ..... 8-bit × 1 channel
- Timers ..... 8-bit × 3
- Watchdog timer ..... 16-bit × 1
- Serial Interface
  Serial I/O (UART or Clock-synchronized) ..... 8-bit × 1
- A/D converter ..... 10-bit × 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 (L version)
  System clock/Internal clock division mode
  At 12 MHz/2-divide mode(ϕ = 6 MHz) ..... 4.00 to 5.25 V
  At 8 MHz/Through mode (ϕ = 8 MHz) ..... 4.00 to 5.25 V
  At 6 MHz/Through mode (ϕ = 6 MHz) ..... 3.00 to 5.25 V
- 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 ..... 30 mW (typ.)
  (at 6 MHz system clock, in through mode)
- Operating temperature range ..... –20 to 85°C
- Packages

FP ..... PLQP0064GA-A (64-pin 14 × 14 mm LQFP)
HP ..... PLQP0064KB-A (64-pin 10 × 10 mm LQFP)

PIN CONFIGURATION (TOP VIEW)



Package type : PLQP0064GA-A (64P6U-A)/PLQP0064KB-A (64P6Q-A)

Fig. 1 Pin configuration of 38K0 group

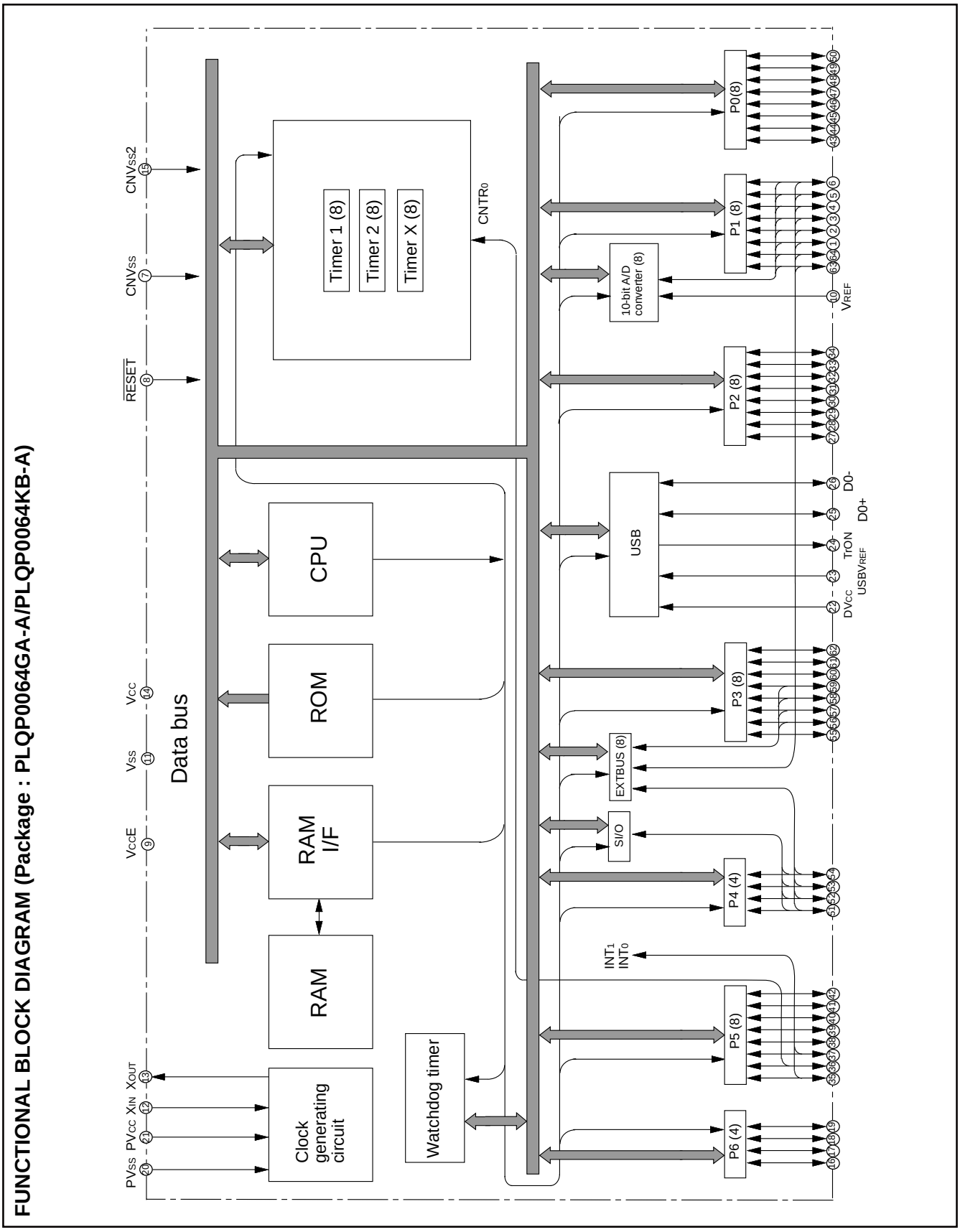


Fig. 2 Functional block diagram

PIN DESCRIPTION

Table 1. Pin description

Pin	Name	Function	Function except a port function
V_{CC}, V_{SS}	Power source	• Apply voltage of 3.0 V – 5.25 V (L version) to V_{CC}, and 0 V to V_{SS}.	
$V_{CC}E$	Analog power source	• Power source pin for ports P1, P3, P4 and analog circuit. Connect this pin to V_{CC}.	
CNV$_{SS}$	CNV$_{SS}$	• This pin controls the operation mode of the chip. Connect this pin to V_{SS}. In the flash memory mode, this pin becoems V_{PP} power source input pin.	
CNV$_{SS}$2	CNV$_{SS}$2	• This pin controls the operation mode of the chip. Connect this pin to V_{SS}.	
V_{REF}	Analog reference voltage input	• Reference voltage input pin for A/D converter.	
DV$_{CC}$ PV$_{CC}$, PV$_{SS}$	Analog power source	• Power source pin for analog circuit. • Connect the DV$_{CC}$ and PV$_{CC}$ pins to V_{CC}, and the PV$_{SS}$ pin to V_{SS}.	
$\overline{\text{RESET}}$	Reset input	• Reset input pin for active “L”	
X_{IN}	Clock input	• Input and output pins for the main clock generating circuit. • Connect a ceramic resonator or a quartz-crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillation frequency. •If an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open.	
X_{OUT}	Clock output		
USBV$_{REF}$	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 = 3.60 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 V_{CC} because use of the built-in USB reference voltage circuit is prohibited in this voltage range.	
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	
P0$_0$–P0$_7$	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)
P1$_0$/DQ$_0$/AN$_0$–P1$_7$/DQ$_7$/AN$_7$	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
P2$_0$–P2$_7$	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	
P3$_0$–P3$_2$	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	
P3$_3$/ExINT P3$_4$/ExCS P3$_5$/ExWR P3$_6$/ExRD P3$_7$/ExA0			• External bus interface function pins
P4$_0$/ExDREQ/RxD P4$_1$/ExDACK/TxD P4$_2$/ExTC/SCLK P4$_3$/ExA1/$\overline{\text{SRDY}}$	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
P5$_0$/INT$_0$	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
P5$_1$/CNTR$_0$			• Timer X funciton pin
P5$_2$/INT$_1$			• Interrupt input pin
P5$_3$–P5$_7$			
P6$_0$–P6$_3$	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

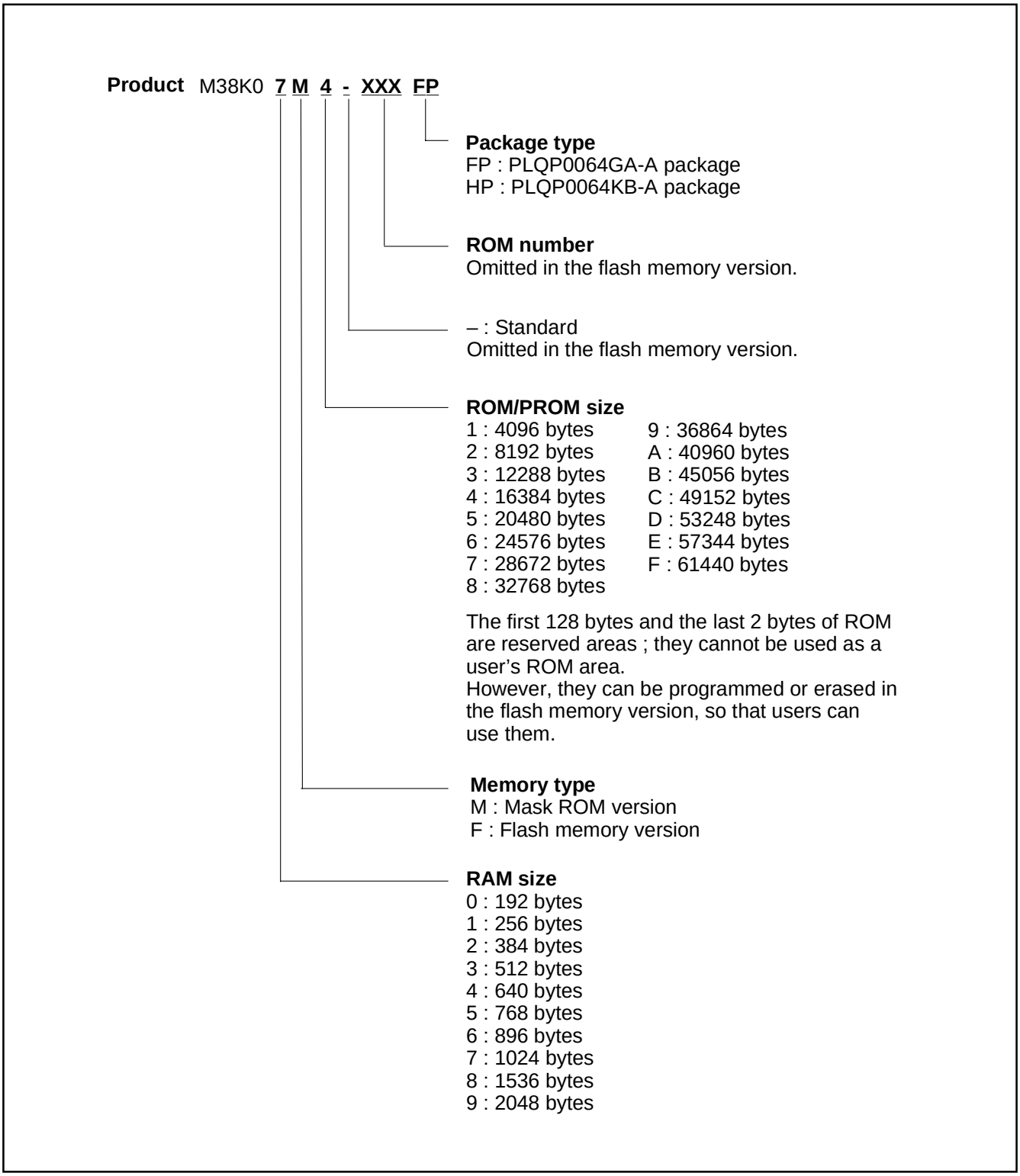


Fig. 3 Part numbering

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

PLQP0064GA-A ..... 0.8 mm-pitch plastic molded LQFP
PLQP0064KB-A ..... 0.5 mm-pitch plastic molded LQFP
100D0M ..... 0.65 mm-pitch metal seal PIGGY BACK

Memory Expansion Plan

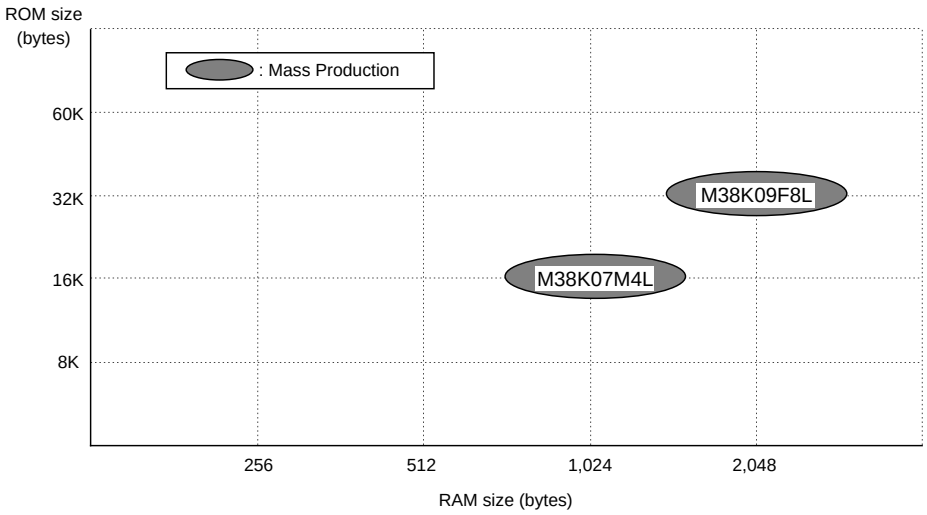


Fig. 4 Memory expansion plan

Currently products are listed below.

Table 2. List of 38K0 group products (L version) **As of October 2006**

Product	ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38K07M4L-XXXFP	16384 (16254)	1024	PLQP0064GA-A	Mask ROM version
M38K07M4L-XXXHP			PLQP0064KB-A	
M38K09F8LFP	32768 (32638)	2048	PLQP0064GA-A	Flash memory version
M38K09F8LHP			PLQP0064KB-A	
M38K09RFS	—	2048	100D0M	

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 “00_{16}”. If the stack page selection bit is “1”, the high-order 8 bits becomes “01_{16}”.

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 PC_H and PC_L. It is used to indicate the address of the next instruction to be executed.

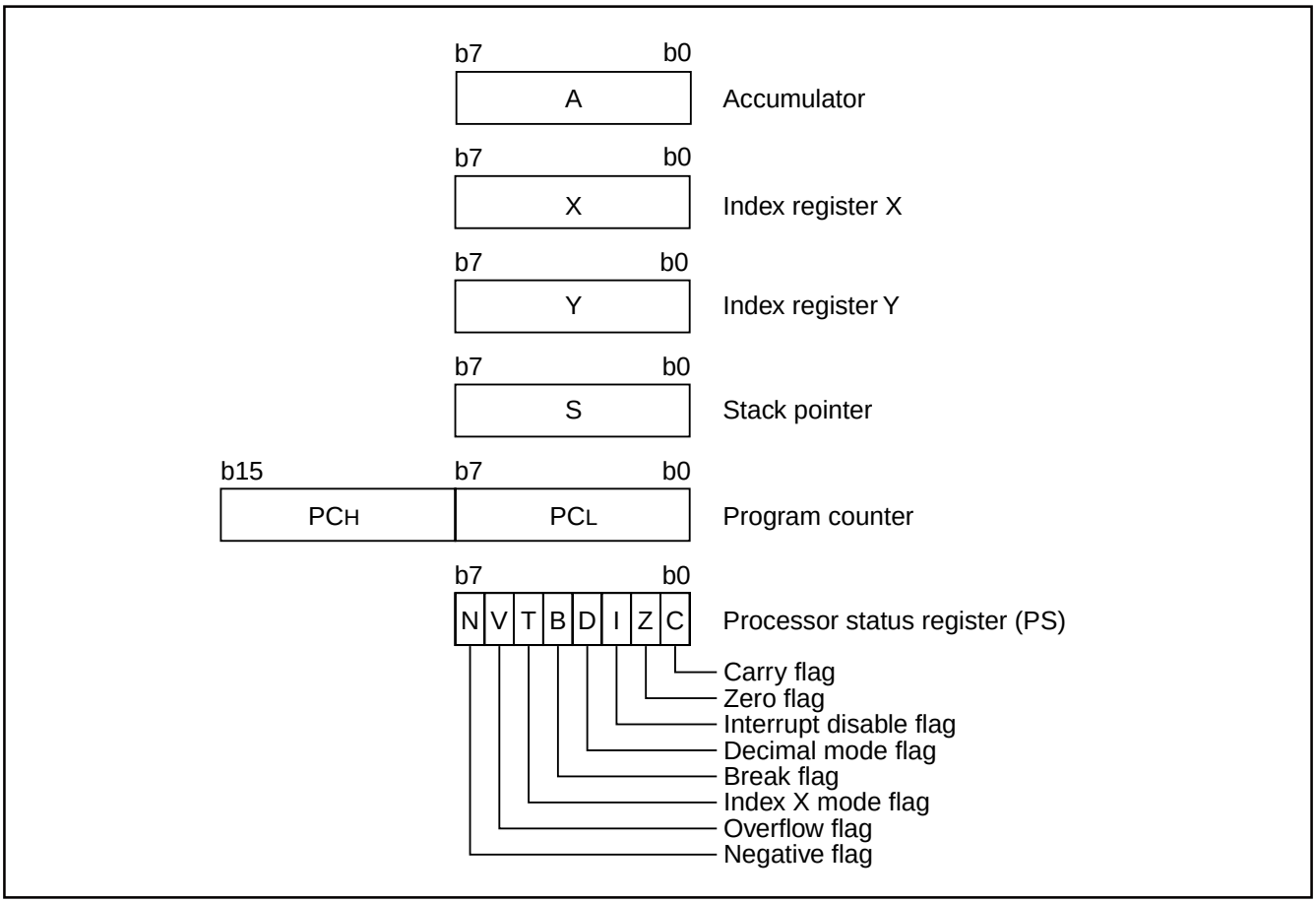


Fig. 5 740 Family CPU register structure

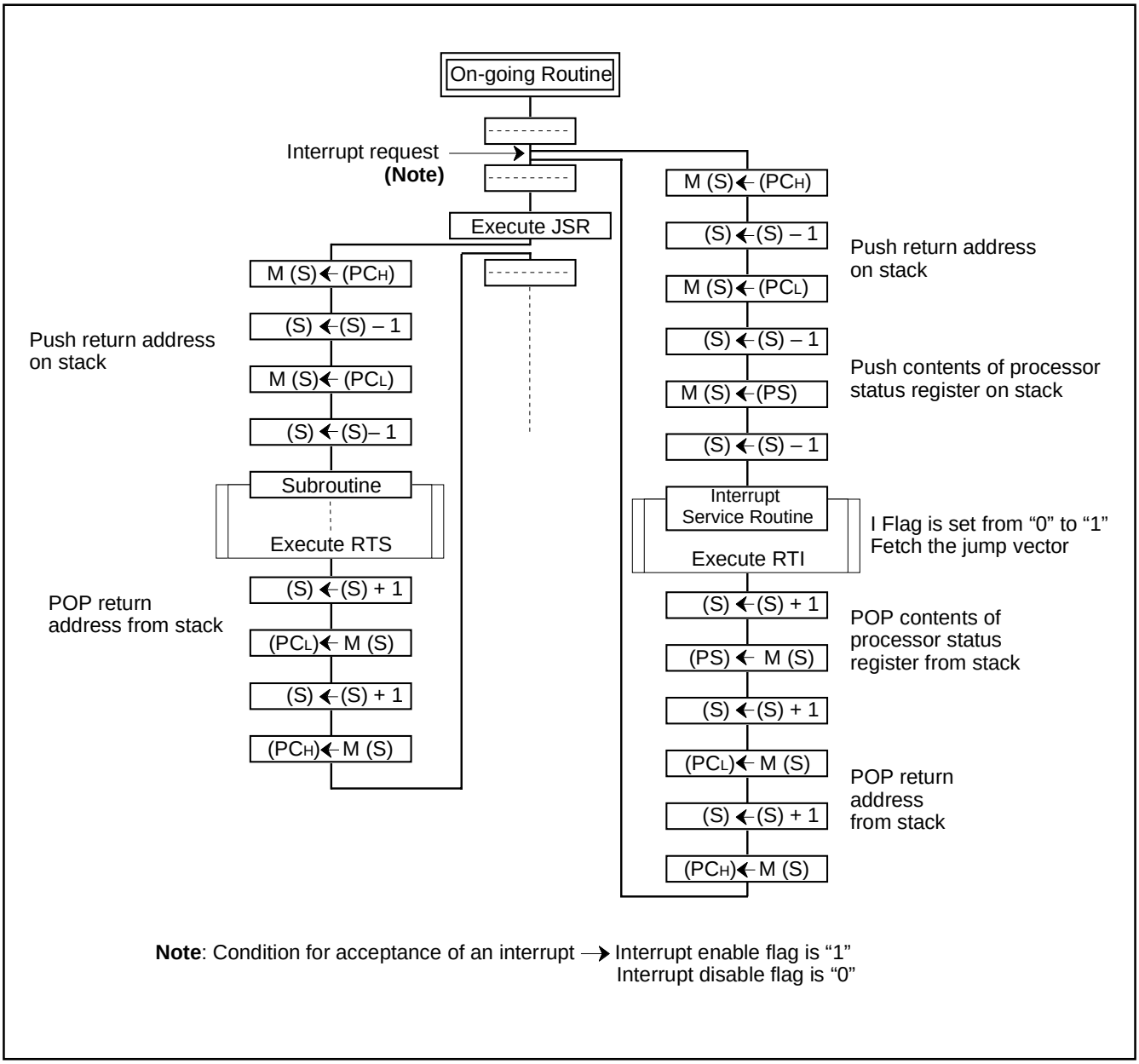


Fig. 6 Register push and pop at interrupt generation and subroutine call

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.

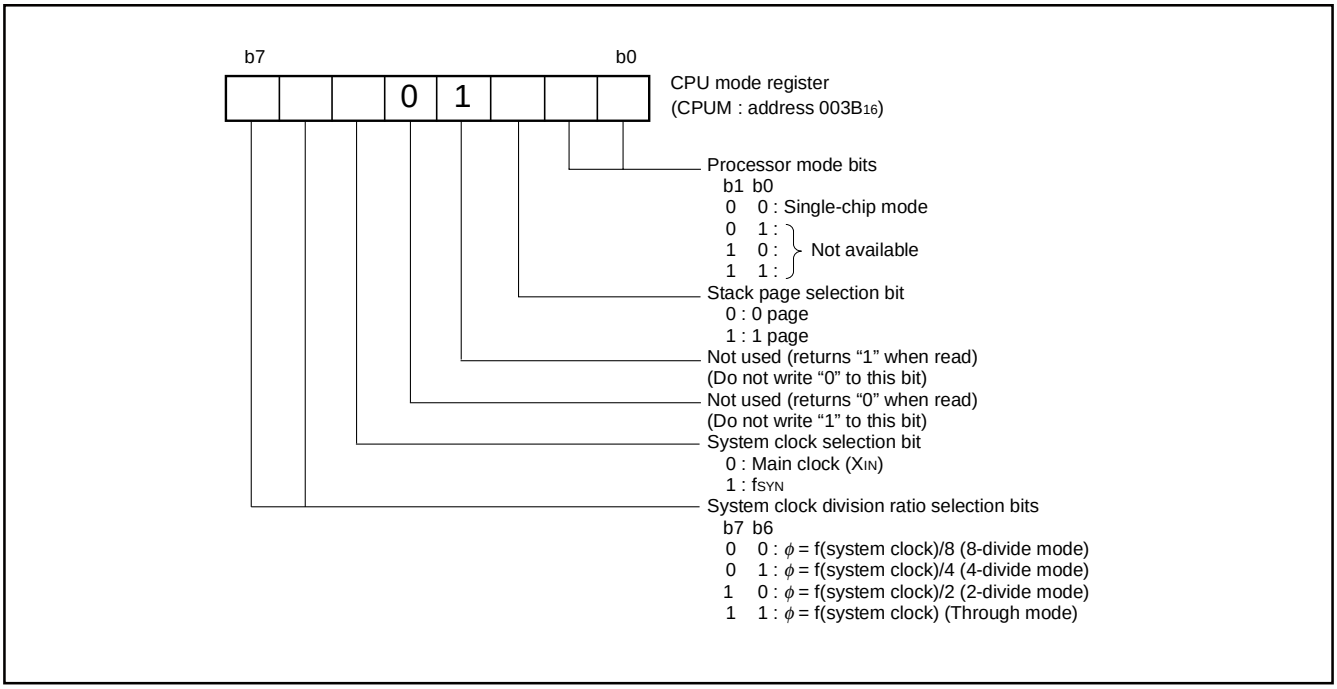


Fig. 7 Structure of CPU mode register

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_{16} to $00FF_{16}$ 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 $FF00_{16}$ to $FFFF_{16}$ 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.

RAM area

RAM size (bytes)	Address $XXXX_{16}$
192	$00FF_{16}$
256	$013F_{16}$
384	$01BF_{16}$
512	$023F_{16}$
640	$02BF_{16}$
768	$033F_{16}$
896	$03BF_{16}$
1024	$043F_{16}$
1536	$063F_{16}$
2048	$083F_{16}$

ROM area

ROM size (bytes)	Address $YYYY_{16}$	Address $ZZZZ_{16}$
4096	$F000_{16}$	$F080_{16}$
8192	$E000_{16}$	$E080_{16}$
12288	$D000_{16}$	$D080_{16}$
16384	$C000_{16}$	$C080_{16}$
20480	$B000_{16}$	$B080_{16}$
24576	$A000_{16}$	$A080_{16}$
28672	9000_{16}	9080_{16}
32768	8000_{16}	8080_{16}
36864	7000_{16}	7080_{16}
40960	6000_{16}	6080_{16}
45056	5000_{16}	5080_{16}
49152	4000_{16}	4080_{16}
53248	3000_{16}	3080_{16}
57344	2000_{16}	2080_{16}
61440	1000_{16}	1080_{16}

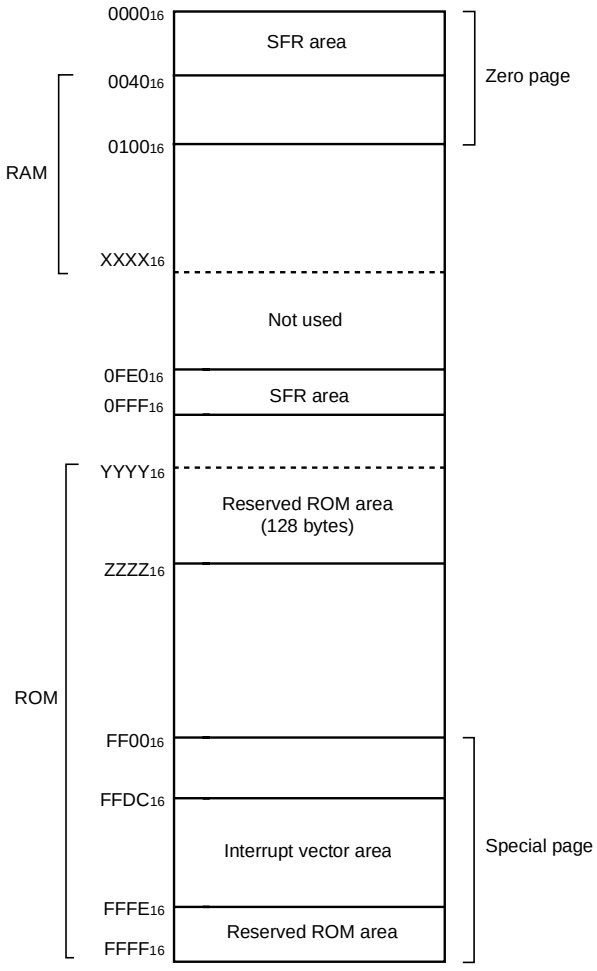


Fig. 8 Memory map diagram

0000₁₆	Port P0 (P0)
0001₁₆	Port P0 direction register (P0D)
0002₁₆	Port P1 (P1)
0003₁₆	Port P1 direction register (P1D)
0004₁₆	Port P2 (P2)
0005₁₆	Port P2 direction register (P2D)
0006₁₆	Port P3 (P3)
0007₁₆	Port P3 direction register (P3D)
0008₁₆	Port P4 (P4)
0009₁₆	Port P4 direction register (P4D)
000A₁₆	Port P5 (P5)
000B₁₆	Port P5 direction register (P5D)
000C₁₆	Port P6 (P6)
000D₁₆	Port P6 direction register (P6D)
000E₁₆	Reserved **(Note)**
000F₁₆	Reserved **(Note)**
0010₁₆	USB control register (USBCON)
0011₁₆	USB address enable register (USBAE)
0012₁₆	USB address 0 register (USBA0)
0013₁₆	USB address 1 register (USBA1)
0014₁₆	Frame number register Low (FNUML)
0015₁₆	Frame number register High (FNUMH)
0016₁₆	USB interrupt source enable register (USBICON)
0017₁₆	USB interrupt source register (USBIREQ)
0018₁₆	Endpoint index register (USBINDEX)
0019₁₆	Endpoint field register 1 (EPXXREG1)
001A₁₆	Endpoint field register 2 (EPXXREG2)
001B₁₆	Endpoint field register 3 (EPXXREG3)
001C₁₆	Endpoint field register 4 (EPXXREG4)
001D₁₆	Endpoint field register 5 (EPXXREG5)
001E₁₆	Endpoint field register 6 (EPXXREG6)
001F₁₆	Endpoint field register 7 (EPXXREG7)

0020₁₆	Prescaler 12 (PRE12)
0021₁₆	Timer 1 (T1)
0022₁₆	Timer 2 (T2)
0023₁₆	Timer X mode register (TM)
0024₁₆	Prescaler X (PREX)
0025₁₆	Timer X (TX)
0026₁₆	Transmit/Receive buffer register (TB/RB)
0027₁₆	Serial I/O status register (SIOSTS)
0028₁₆	Reserved **(Note)**
0029₁₆	Reserved **(Note)**
002A₁₆	Reserved **(Note)**
002B₁₆	Reserved **(Note)**
002C₁₆	Reserved **(Note)**
002D₁₆	Reserved **(Note)**
002E₁₆	Reserved **(Note)**
002F₁₆	Reserved **(Note)**
0030₁₆	EXB interrupt source enable register (EXBICON)
0031₁₆	EXB interrupt source register (EXBIREQ)
0032₁₆	Reserved **(Note)**
0033₁₆	EXB index register (EXBINDEX)
0034₁₆	EXB field register 1 (EXBREG1)
0035₁₆	EXB field register 2 (EXBREG2)
0036₁₆	AD control register (ADCON)
0037₁₆	AD conversion register 1 (AD1)
0038₁₆	AD conversion register 2 (AD2)
0039₁₆	Watchdog timer control register (WDTCON)
003A₁₆	Reserved **(Note)**
003B₁₆	CPU mode register (CPUM)
003C₁₆	Interrupt request register 1(IREQ1)
003D₁₆	Interrupt request register 2(IREQ2)
003E₁₆	Interrupt control register 1(ICON1)
003F₁₆	Interrupt control register 2(ICON2)

0FE0₁₆	Serial I/O control register (SIOCON)
0FE1₁₆	UART control register (UARTCON)
0FE2₁₆	Baud rate generator (BRG)
0FE3₁₆	Reserved **(Note)**
0FE4₁₆	Reserved **(Note)**
0FE5₁₆	Reserved **(Note)**
0FE6₁₆	Reserved **(Note)**
0FE7₁₆	Reserved **(Note)**
0FE8₁₆	Reserved **(Note)**
0FE9₁₆	Reserved **(Note)**
0FEA₁₆	Reserved **(Note)**
0FEB₁₆	Reserved **(Note)**
0FEC₁₆	Endpoint field register 8 (EPXXREG8)
0FED₁₆	Endpoint field register 9 (EPXXREG9)
0FEE₁₆	Reserved **(Note)**
0FEF₁₆	Reserved **(Note)**

0FF0₁₆	Port P0 pull-up control register (PULL0)
0FF1₁₆	Reserved **(Note)**
0FF2₁₆	Port P5 pull-up control register (PULL5)
0FF3₁₆	Interrupt edge selection register (INTEDGE)
0FF4₁₆	Reserved **(Note)**
0FF5₁₆	Reserved **(Note)**
0FF6₁₆	Reserved **(Note)**
0FF7₁₆	Reserved **(Note)**
0FF8₁₆	PLL control register (PLLCON)
0FF9₁₆	Reserved **(Note)**
0FFA₁₆	Reserved **(Note)**
0FFB₁₆	MISRG
0FFC₁₆	Reserved **(Note)**
0FFD₁₆	Reserved **(Note)**
0FFE₁₆	Flash memory control register (FMCR)
0FFF₁₆	Reserved **(Note)**

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.
P0$_0$–P0$_7$	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)
P1$_0$–P1$_7$	Port P1		CMOS compatible input level CMOS 3-state output (Power source is VccE)	A/D conversion input External bus interface funciton I/O	AD control register EXB control register	(2)
P2$_0$–P2$_7$	Port P2		CMOS compatible input level CMOS 3-state output	———	———	(3)
P3$_0$–P3$_2$	Port P3		CMOS/TTL compatible input level CMOS 3-state output (Power source is VccE)	———	———	(4)
P3$_3$/ExINT				External bus interface funciton output	EXB control register	(5)
P3$_4$/ExCS P3$_5$/ExWR P3$_6$/ExRD P3$_7$/ExA0				External bus interface funciton input	EXB control register	(6)
P4$_0$/RxD/ExDREQ	Port P4			Serial I/O input External bus interface funciton output	Serial I/O control register EXB control register	(7)
P4$_1$/TxD/ExDACK				Serial I/O output External bus interface funciton input	Serial I/O control register EXB control register	(8)
P4$_2$/SCLK/ExTC				Serial I/O I/O External bus interface funciton input	Serial I/O control register EXB control register	(9)
P4$_3$/$\overline{\text{SRDY}}$/ExA1				Serial I/O output External bus interface funciton input	Serial I/O control register EXB control register	(10)
P5$_0$/INT$_0$ P5$_2$/INT$_1$	Port P5		CMOS compatible input level CMOS 3-state output	External interrupt input	Port P5 pull-up control register Interrupt edge selection register	(11)
P5$_1$/CNTR$_0$				Timer X function I/O	Timer X mode register	(12)
P5$_3$–P5$_7$				———	———	(13)
P6$_0$–P6$_3$	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.

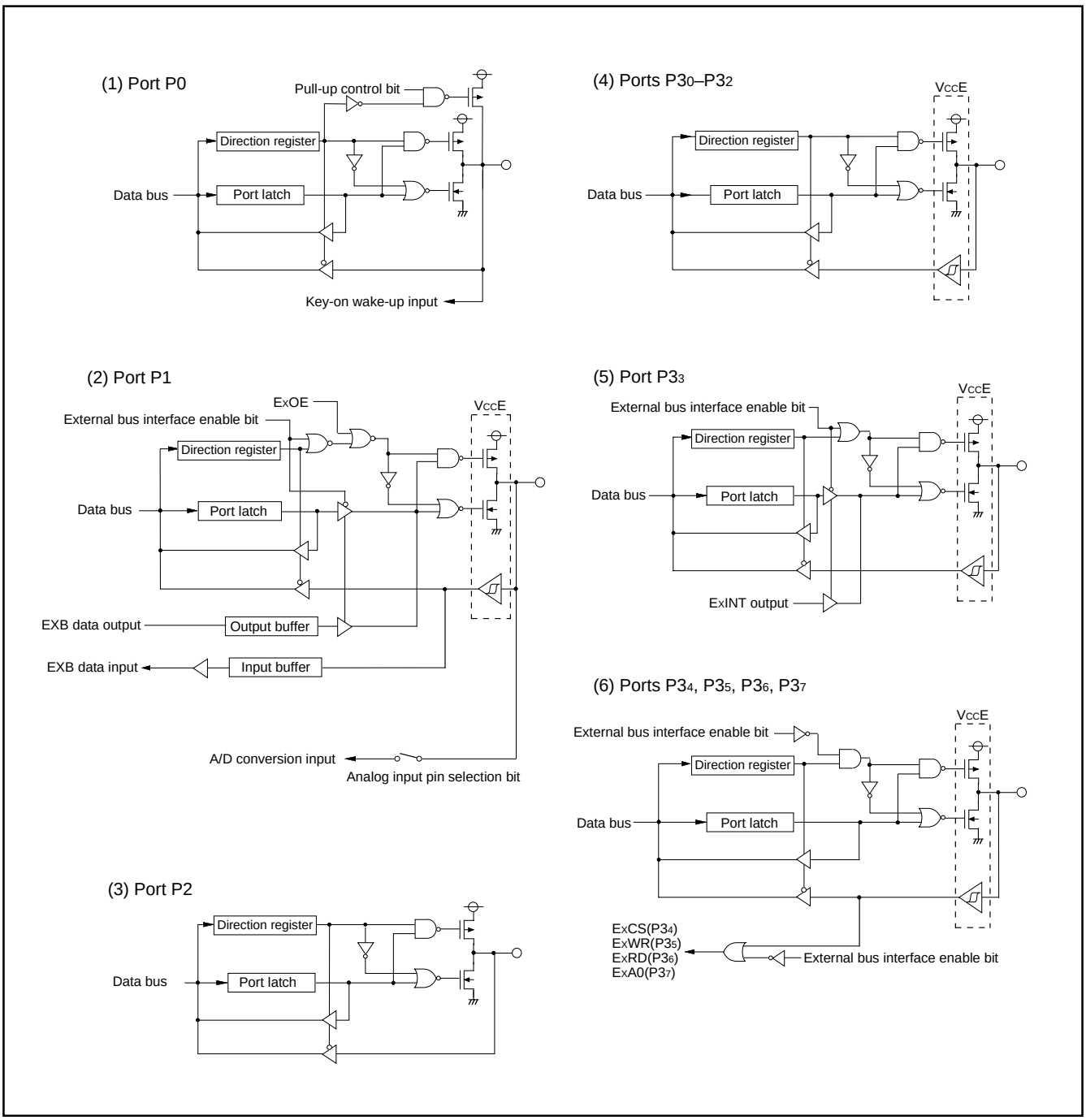


Fig. 10 Port block diagram (1)

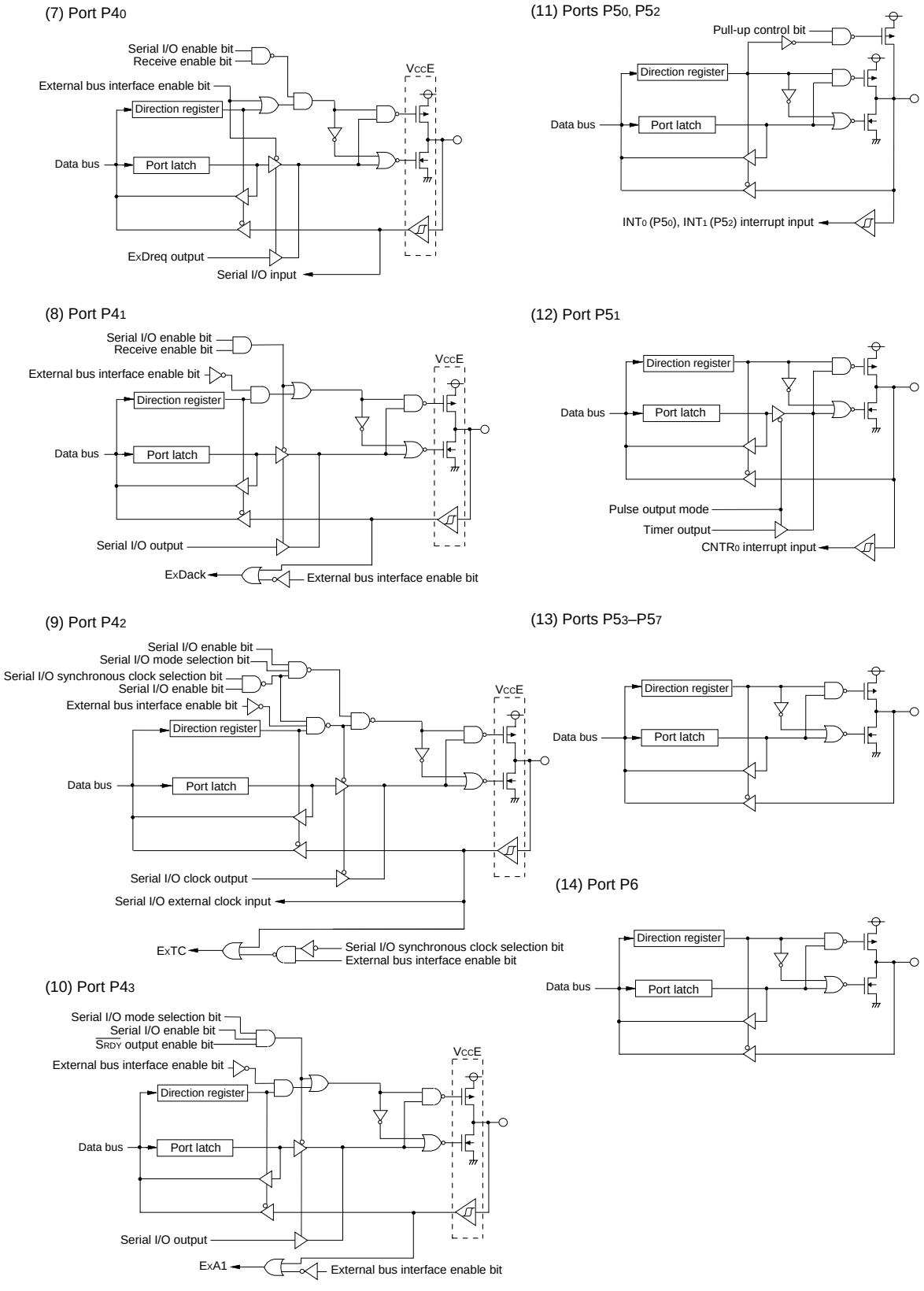


Fig. 11 Port block diagram (2)

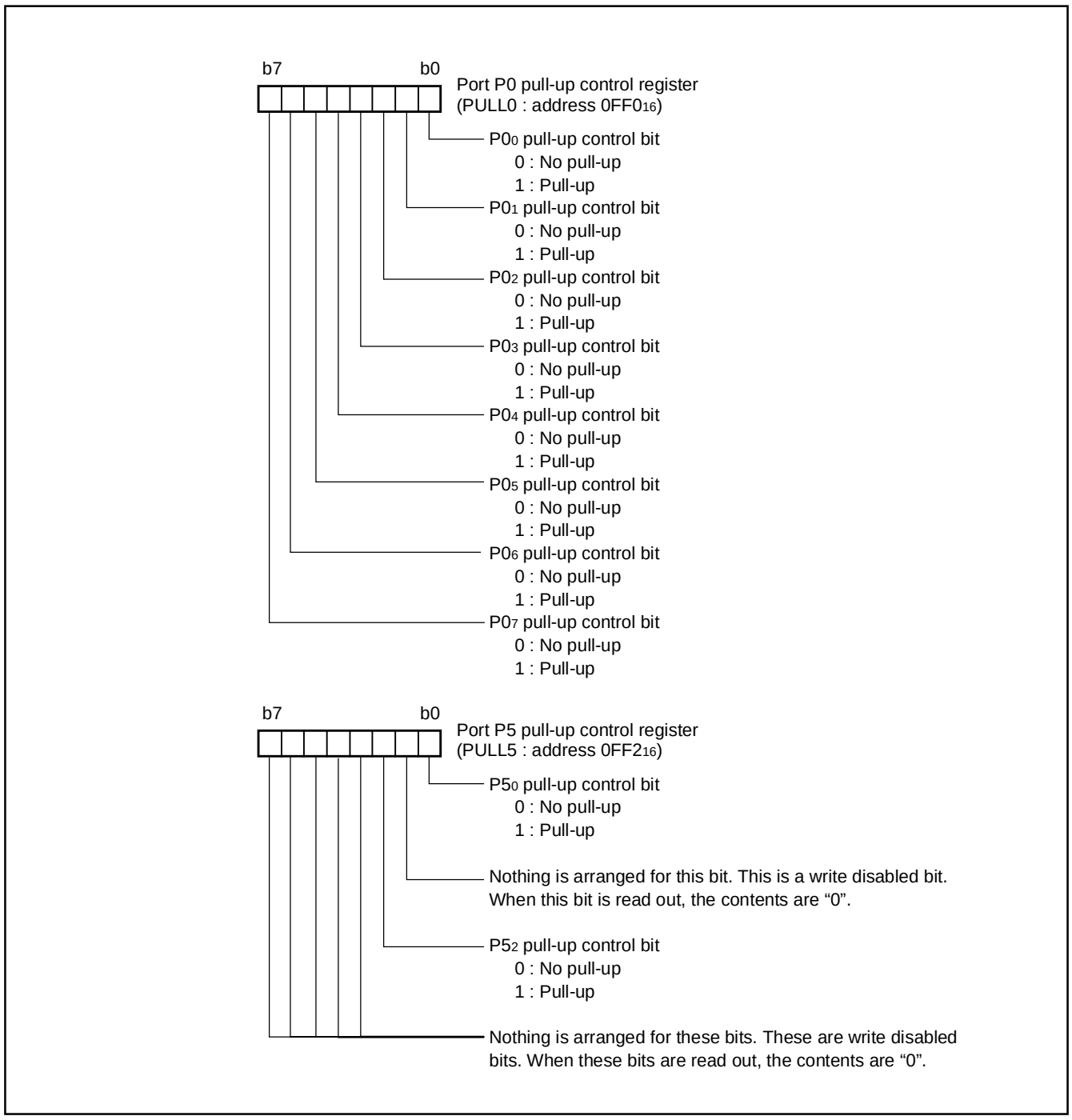


Fig. 12 Structure of port I/O-related registers

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 switching external interrupt active edge

Related register: Interrupt edge selection register (address $0FF3_{16}$), Timer X mode register (address 0023_{16})

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	$FFFD_{16}$	$FFFC_{16}$	At reset
USB bus reset	2	$FFFB_{16}$	$FFFA_{16}$	At detection of USB bus reset signal (2.5 μs interval SE0)
USB SOF	3	$FFF9_{16}$	$FFF8_{16}$	At detection of USB SOF signal
USB device	4	$FFF7_{16}$	$FFF6_{16}$	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	$FFF5_{16}$	$FFF4_{16}$	At completion of reception or transmission or at completion of DMA transmission
INT_0	6	$FFF3_{16}$	$FFF2_{16}$	At detection of either rising or falling edge of INT_0 input
Timer X	7	$FFF1_{16}$	$FFF0_{16}$	At timer X underflow
Timer 1	8	$FFEF_{16}$	$FFEE_{16}$	At timer 1 underflow
Timer 2	9	$FFED_{16}$	$FFEC_{16}$	At timer 2 underflow
INT_1	10	$FFEB_{16}$	$FFEA_{16}$	At detection of either rising or falling edge of INT_1 input
(Note 3)	—	$FFE9_{16}$	$FFE8_{16}$	**(Note 4)**
Serial I/O reception	11	$FFE7_{16}$	$FFE6_{16}$	At completion of serial I/O data reception
Serial I/O transmission	12	$FFE5_{16}$	$FFE4_{16}$	At completion of serial I/O data transmission
$CNTR_0$	13	$FFE3_{16}$	$FFE2_{16}$	At detection of either rising or falling edge of $CNTR_0$ input
Key-on wake up	14	$FFE1_{16}$	$FFE0_{16}$	At falling of conjunction of input level for port P0 (at input mode)
A/D conversion	15	$FFDF_{16}$	$FFDE_{16}$	At completion of A/D conversion
BRK instruction	16	$FFDD_{16}$	$FFDC_{16}$	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 $003F_{16}$) to “0”.

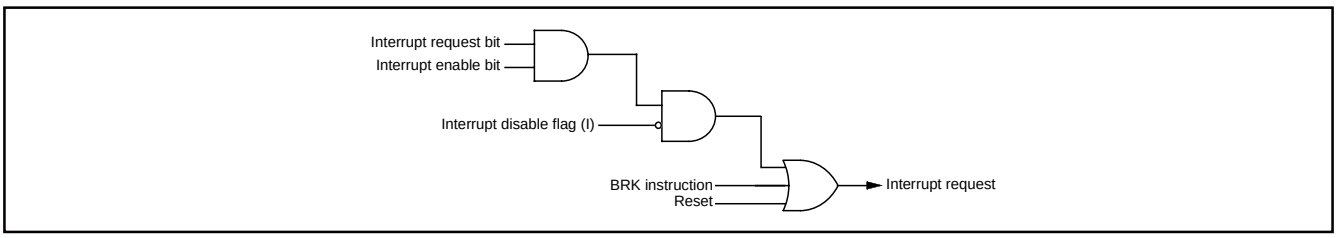


Fig. 13 Interrupt control

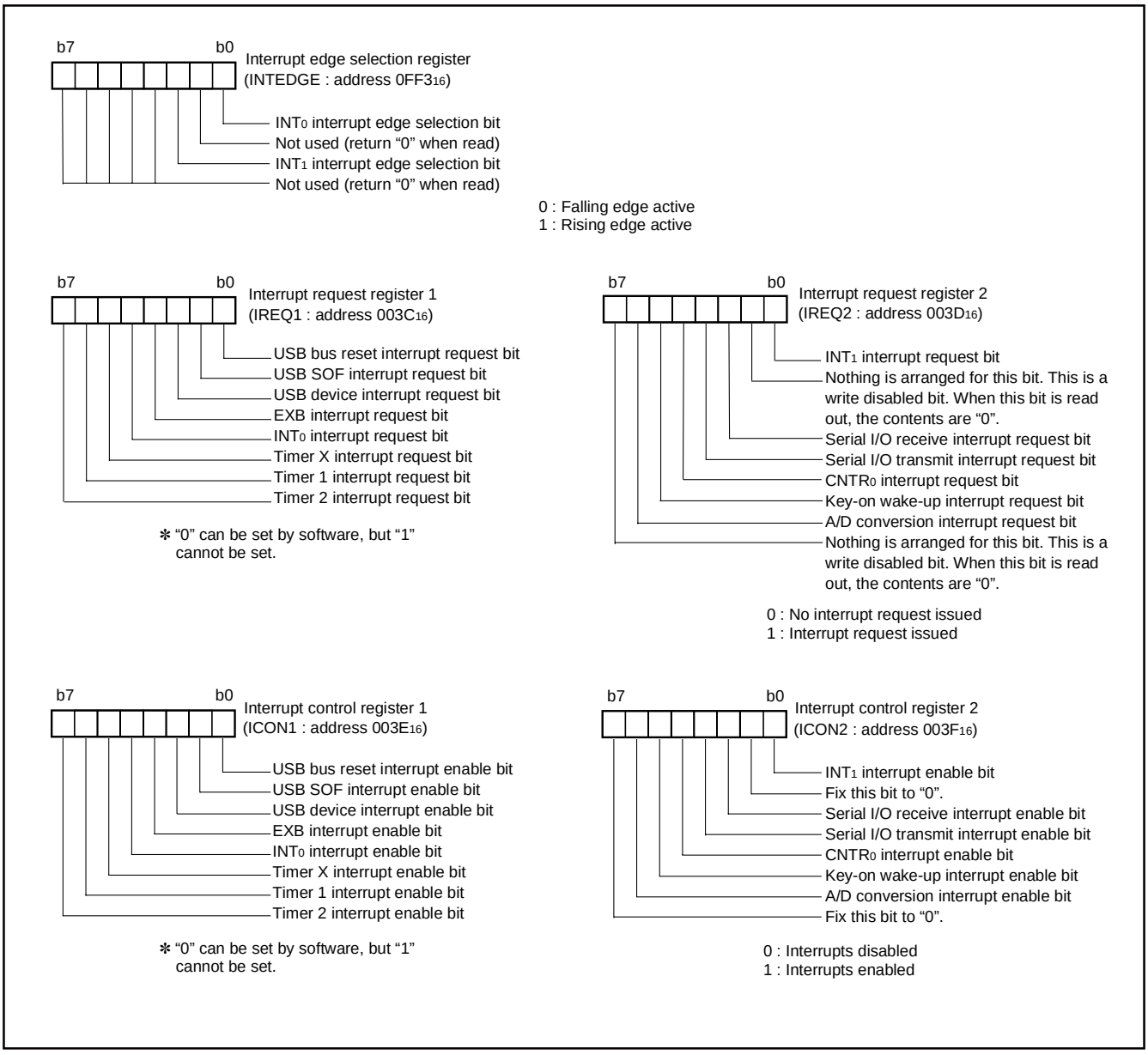


Fig. 14 Structure of interrupt-related registers

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 P0$_0$–P0$_3$.

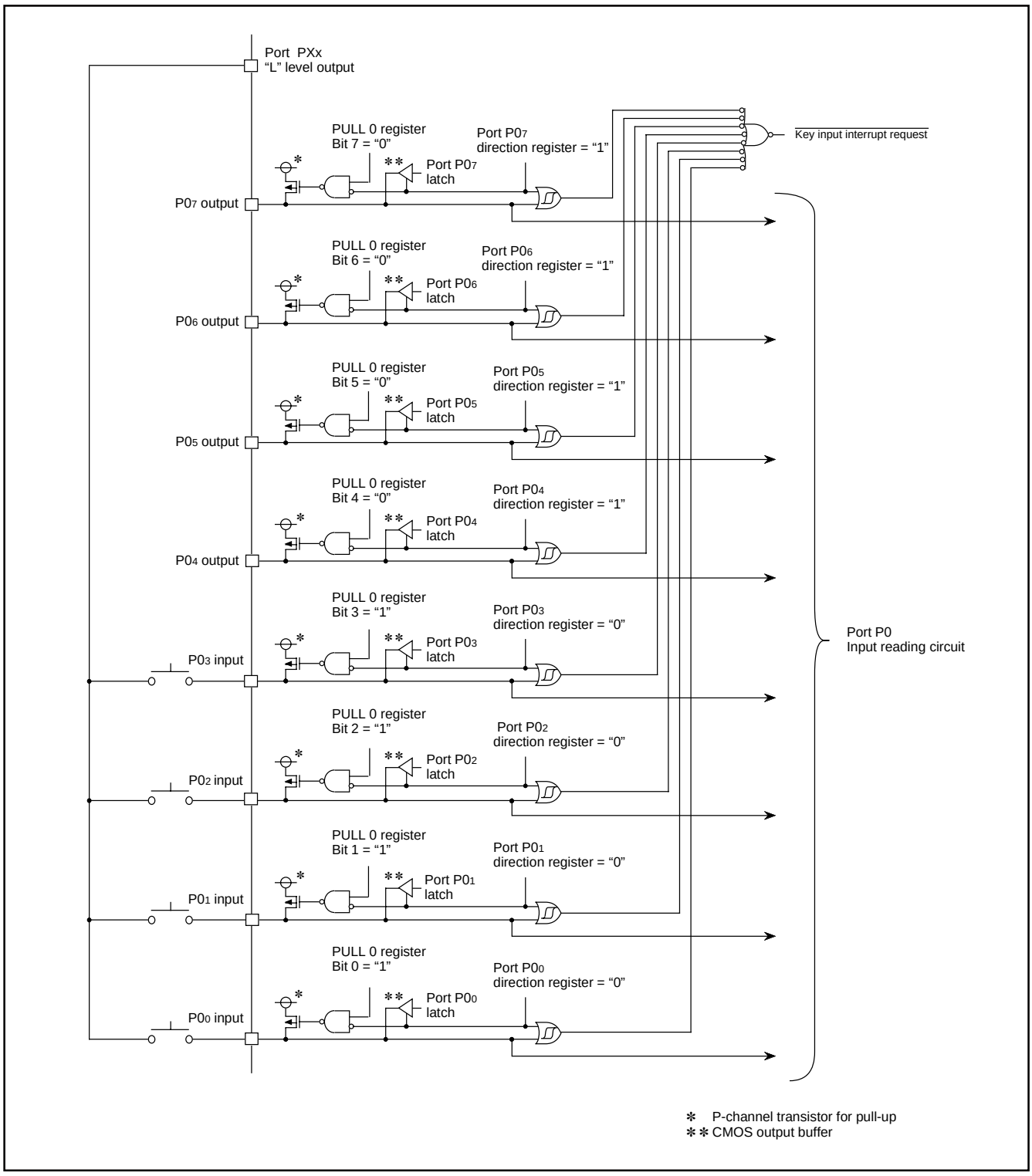


Fig. 15 Connection example when using key input interrupt and port P0 block diagram

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 "00$_{16}$", 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".

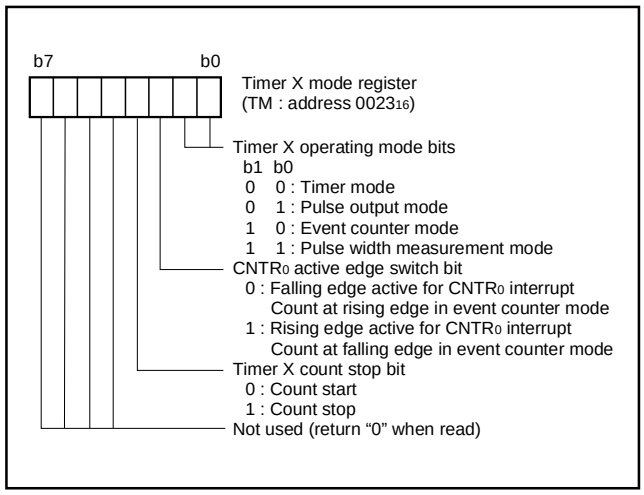


Fig. 16 Structure of timer X mode register

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 "00$_{16}$", the signal output from the CNTR$_0$ pin is inverted. If the CNTR$_0$ 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 P5$_1$ 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$_0$ pin.

When the CNTR$_0$ active edge selection bit is "0", the rising edge of the CNTR$_0$ pin is counted.

When the CNTR$_0$ active edge selection bit is "1", the falling edge of the CNTR$_0$ pin is counted.

(4) Pulse Width Measurement Mode

If the CNTR$_0$ active edge selection bit is "0", the timer counts the system clock divided by 16 while the CNTR$_0$ pin is at "H". If the CNTR$_0$ active edge selection bit is "1", the timer counts it while the CNTR$_0$ 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.

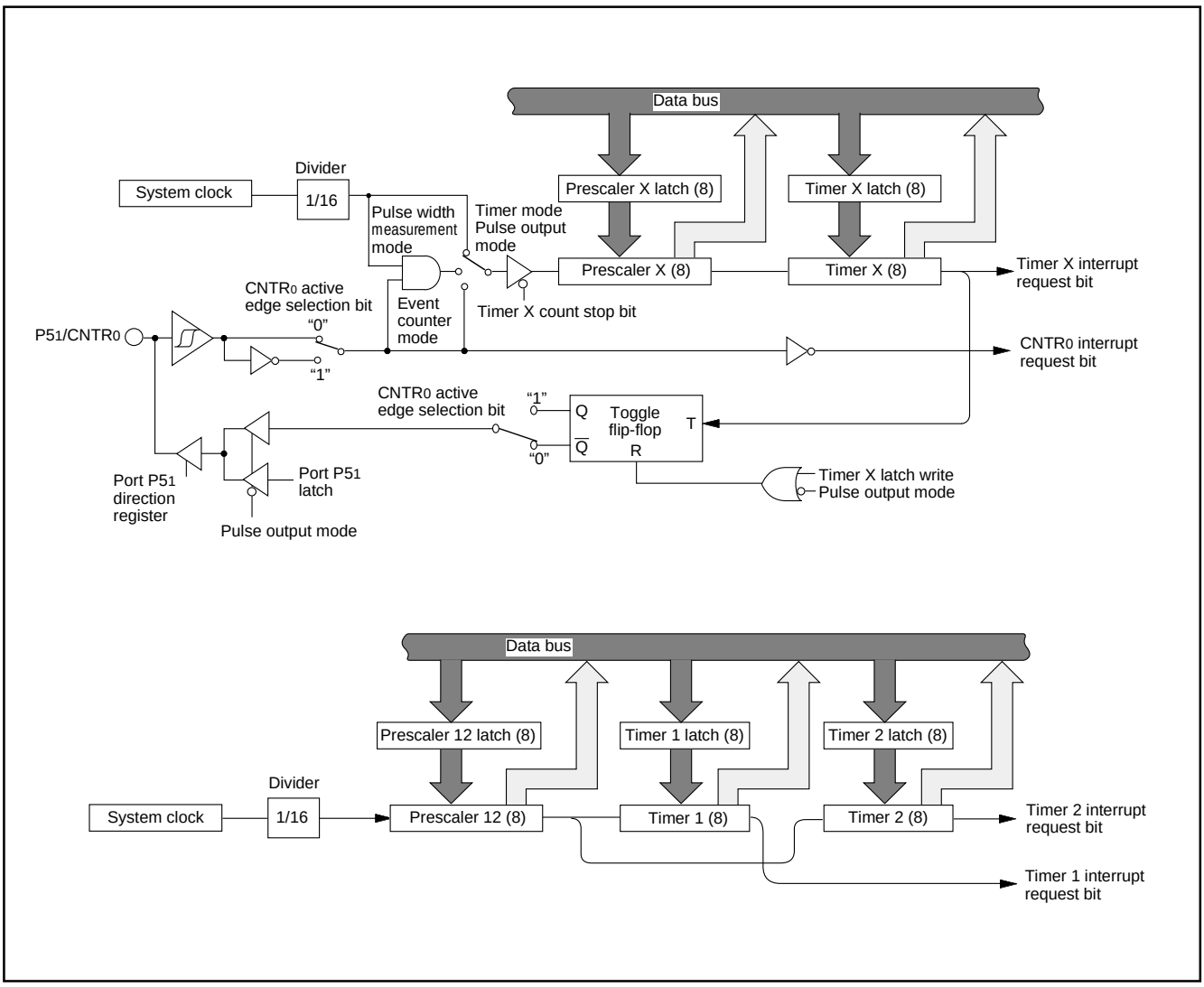


Fig. 17 Timer block diagram

SERIAL INTERFACE
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 $0FE0_{16}$) 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 Trancemit/Receive buffer register.

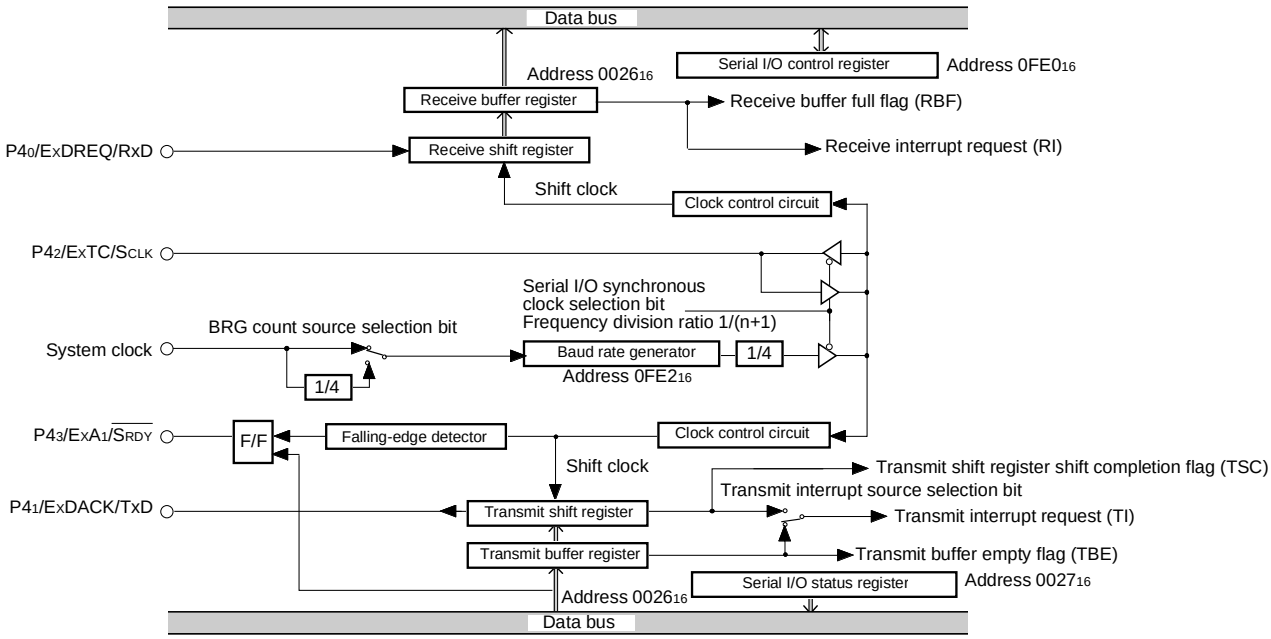
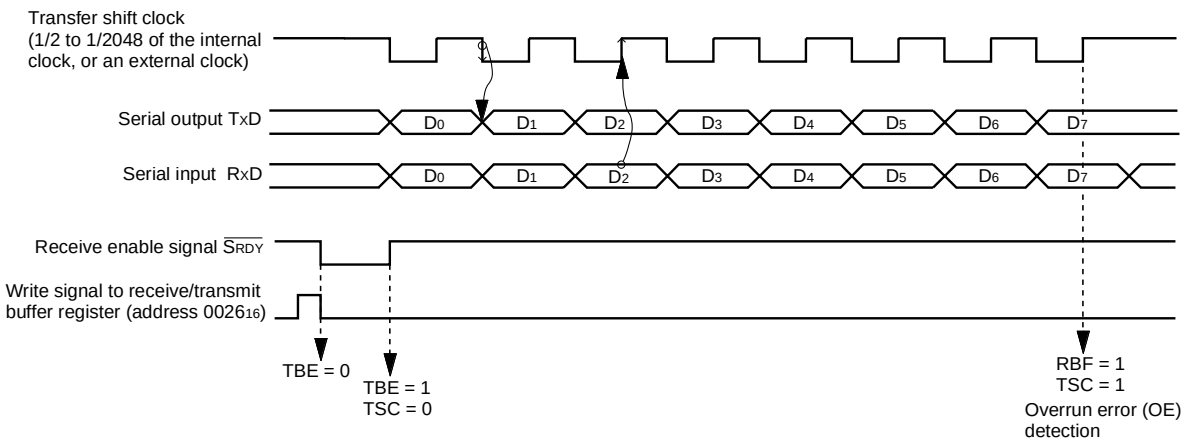


Fig. 18 Block diagram of clock synchronous serial I/O



Notes 1 : The transmit interrupt (TI) can be generated either when the transmit buffer register has emptied (TBE = 1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
2 : If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the TxD pin.
3 : The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes “1” .

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 register, 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.

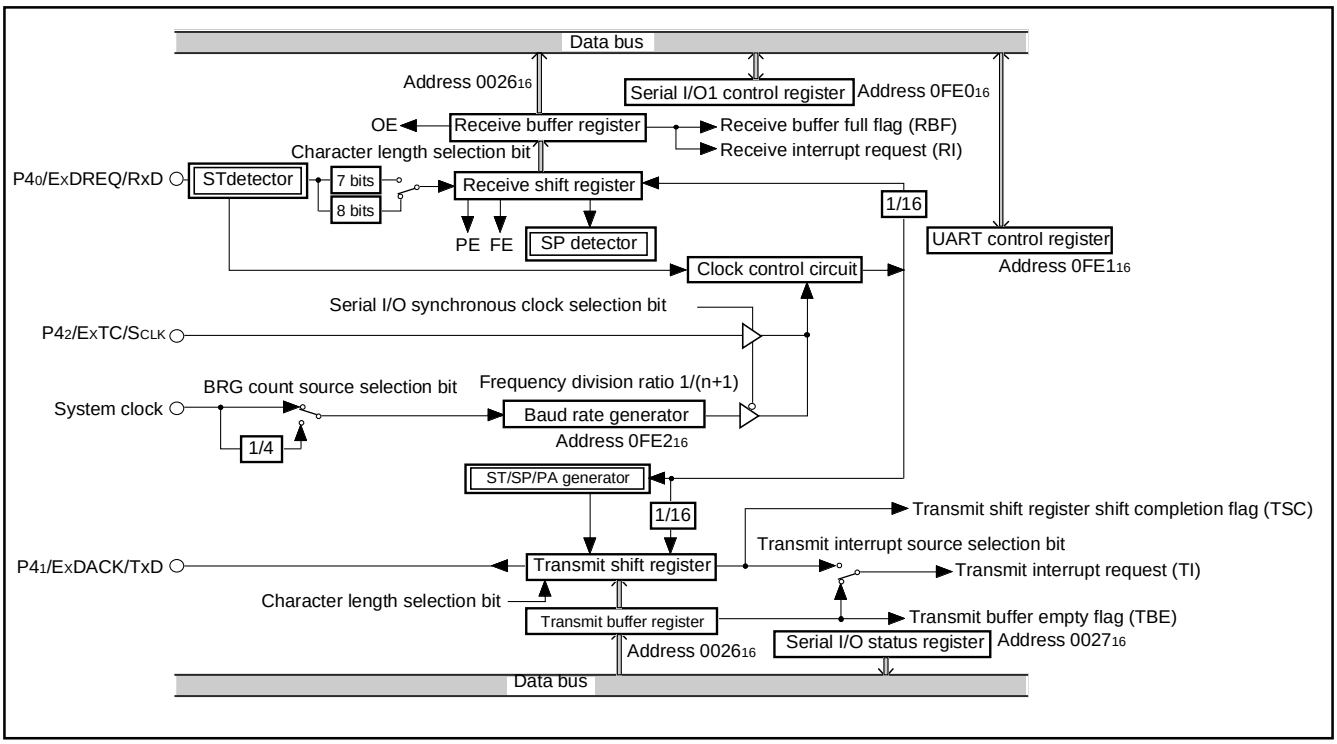
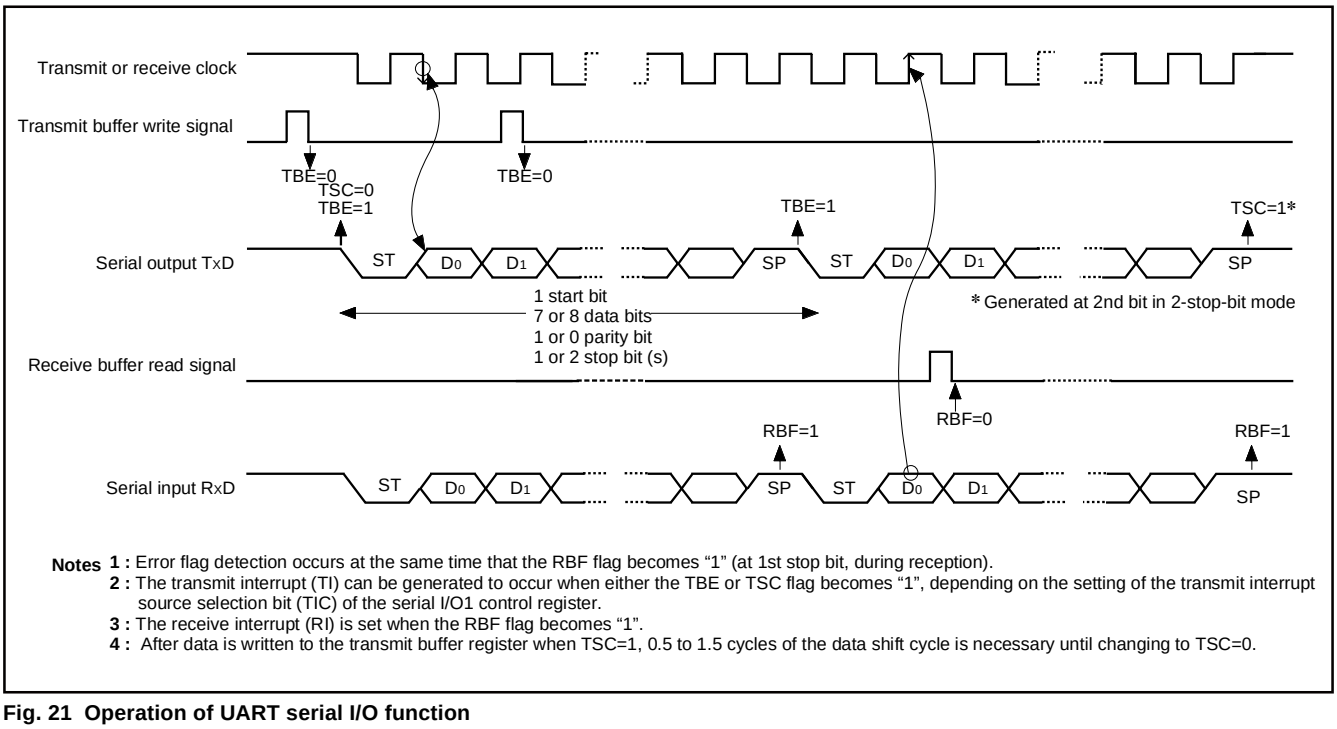


Fig. 20 Block diagram of UART serial I/O



Notes
1 : Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2 : The transmit interrupt (TI) can be generated to occur when either the TBE or TSC flag becomes “1”, depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
3 : The receive interrupt (RI) is set when the RBF flag becomes “1”.
4 : After data is written to the transmit buffer register when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

Fig. 21 Operation of UART serial I/O function

[Serial I/O Control Register (SIOCON)] 0FE0$_{16}$

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

[UART Control Register (UARTCON)] 0FE1$_{16}$

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$_{16}$

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$_{16}$

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$_{16}$

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 enalbed, take the following sequence.

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

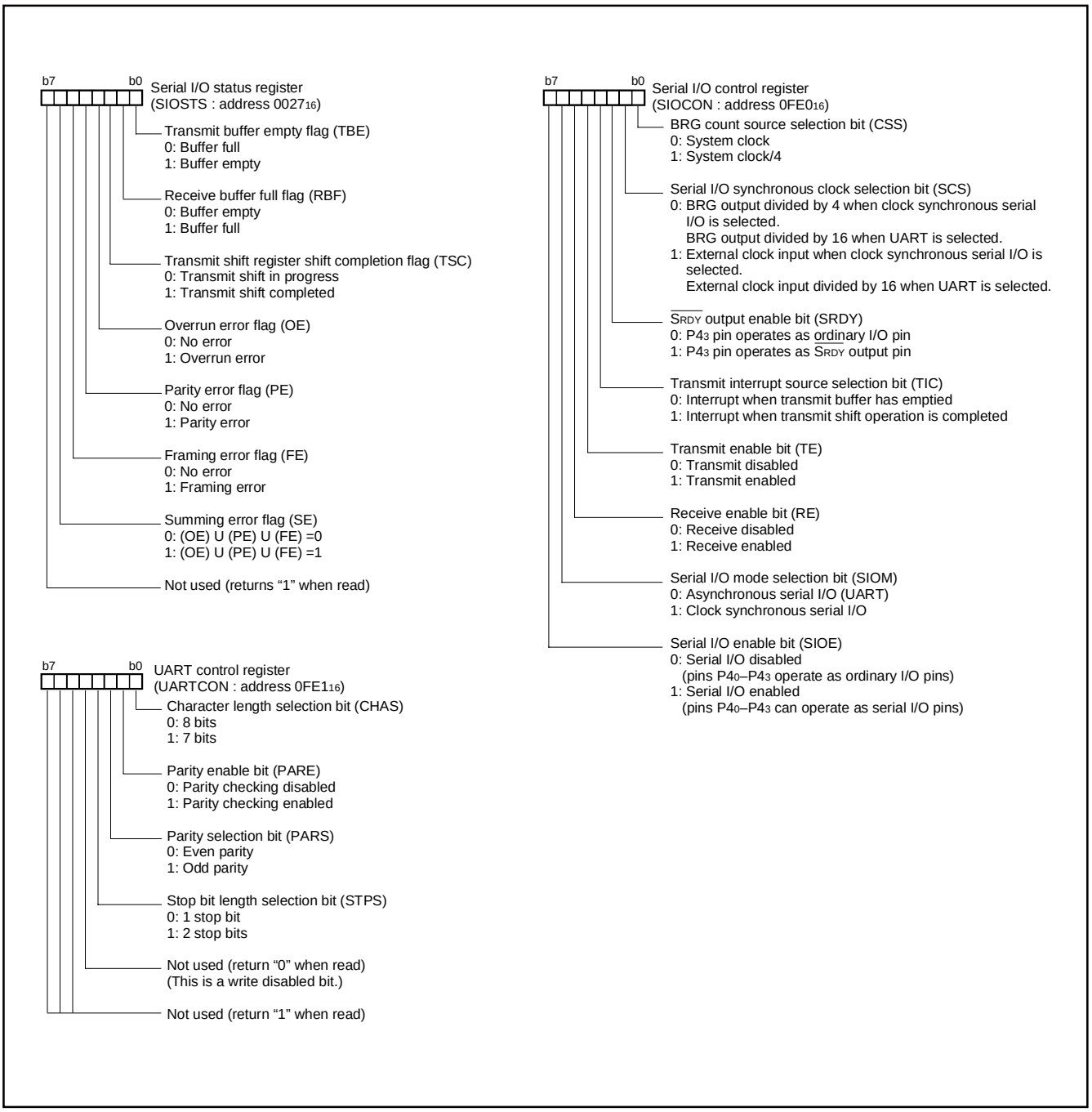


Fig. 22 Structure of serial I/O control registers

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 USB2.0's Full-Speed Transfer Mode (12 Mbps, equivalent to USB1.1). 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 38K0 group will be ready for any USB application that comes its way.

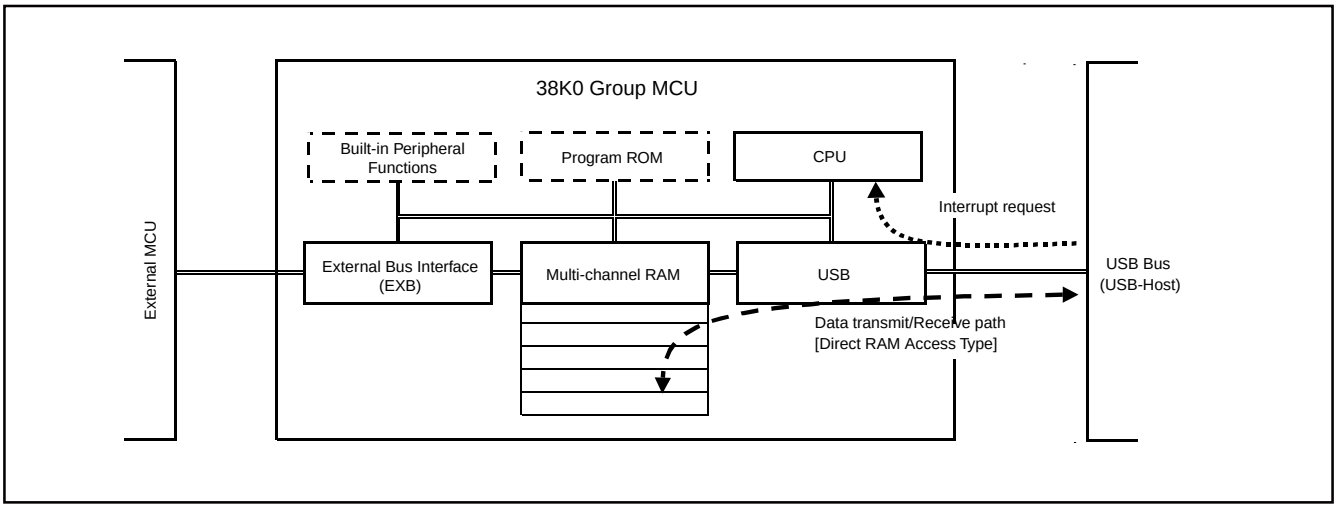


Fig. 23 USB function overview

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 ϕ:

When the USBFCC is operating at ϕ =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 ϕ = 6MHZ, 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)

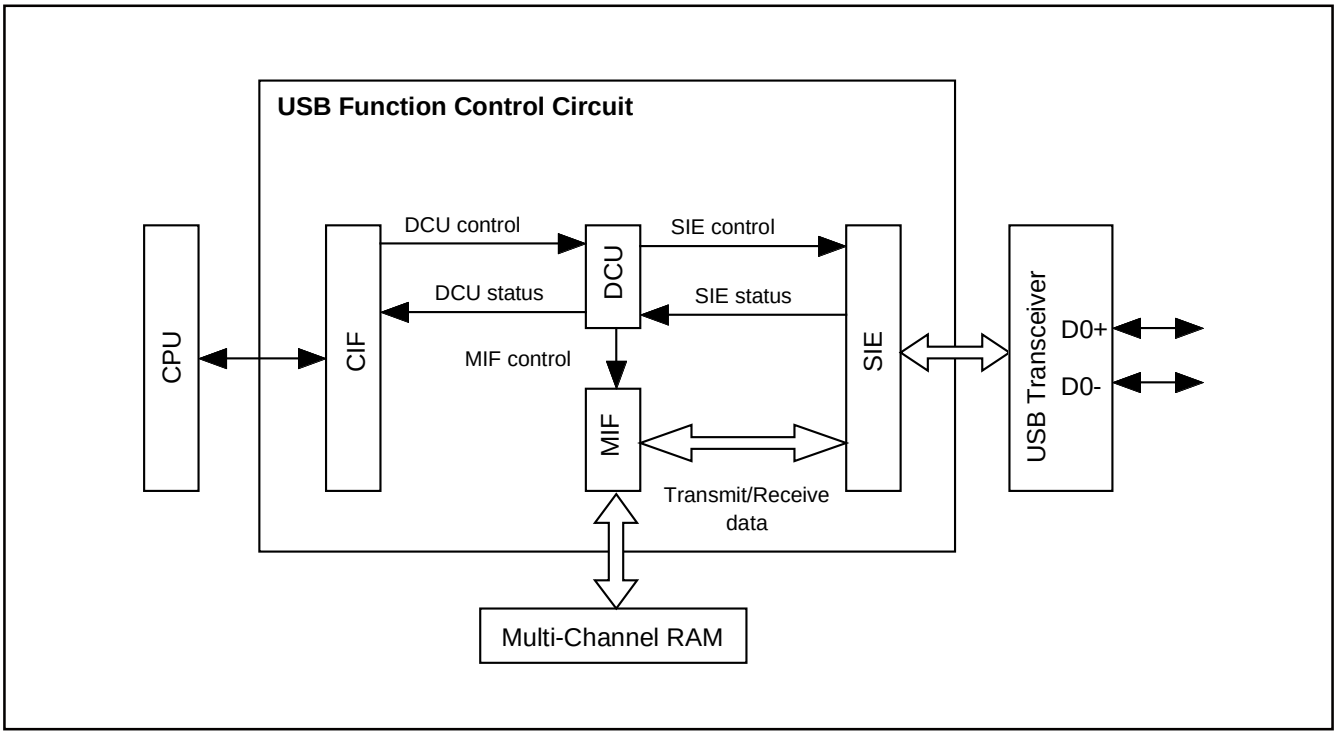


Fig. 24 USB Function Control Circuit (USBFCC) block diagram

(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.

USB Port External Circuit Configuration

The operation mode of the USB port driver circuit can be configured by USB control register (address 0010$_{16}$).
Figure 25 and Figure 26 show the USB port external circuit block diagram.

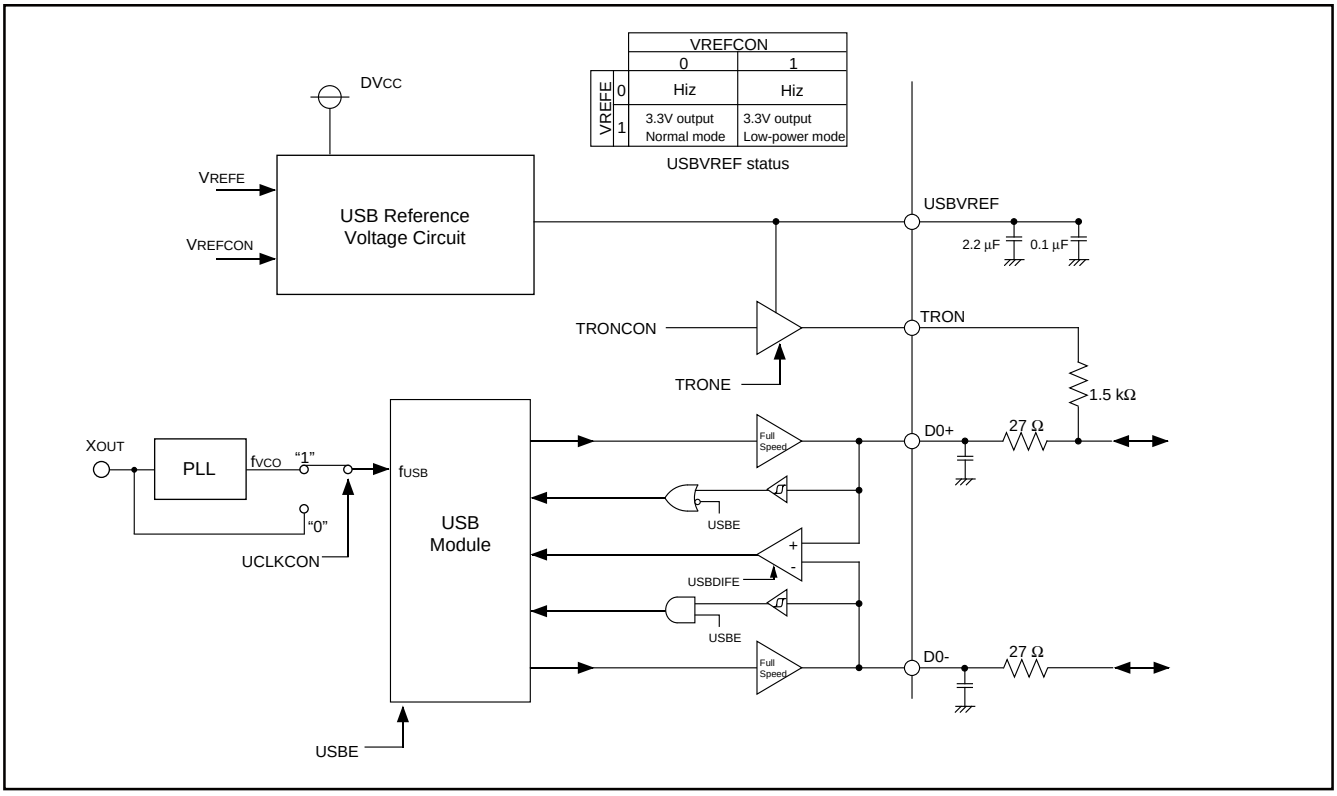


Fig. 25 USB port external circuit (D0+, D0-, USBVREF, TrON) block diagram (4.0V ≤ Vcc ≤ 5.25V)

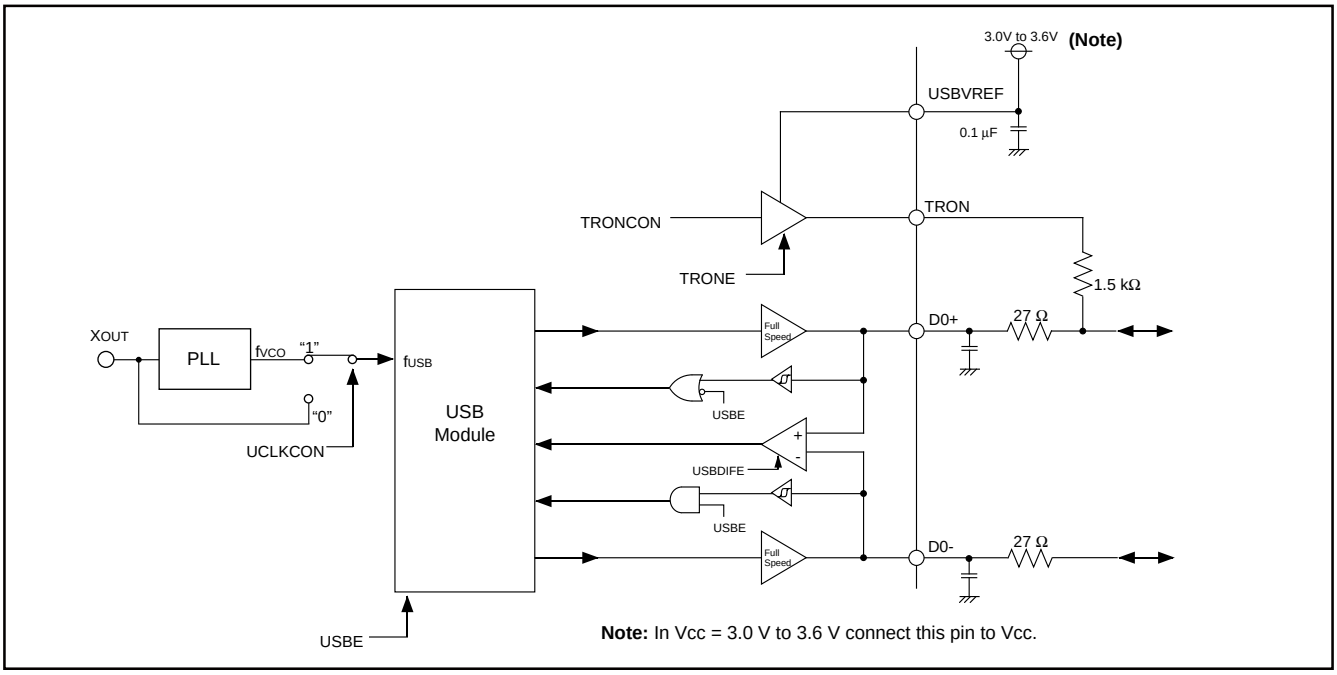


Fig. 26 USB port external circuit (D0+, D0-, USBVREF, TrON) block diagram (3.0V ≤ Vcc ≤ 4.0V)

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 0FED16) defines the beginning address of the buffer area (every 32 bytes) for each Endpoint. However, the only RAM area is configurable.

•00h [Address 000016], 01h [Address 002016]: Not configurable

•02h [Address 004016] to 1Fh [Address 03E016]: 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 001D16), 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 001916).

•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

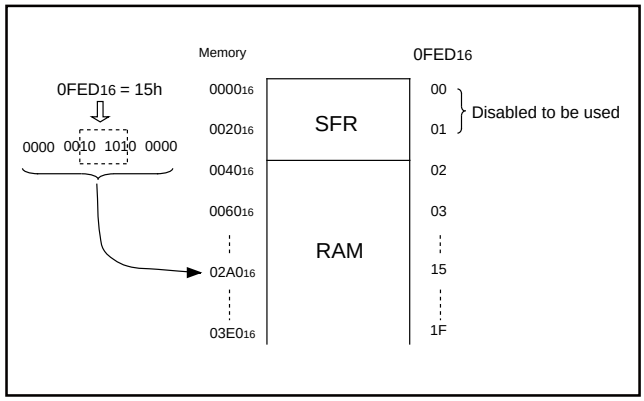


Fig. 27 Example setting of buffer area beginning address

•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 004016 to 03FF16.

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.

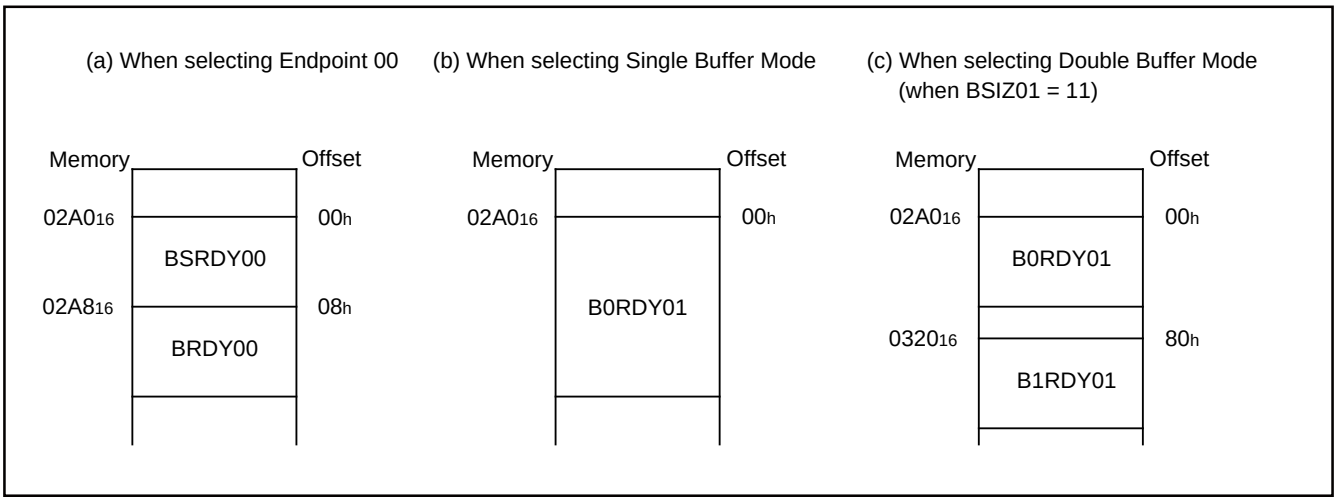


Fig. 28 Examples of interrupt source dependant buffer area offset address

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 µs 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.

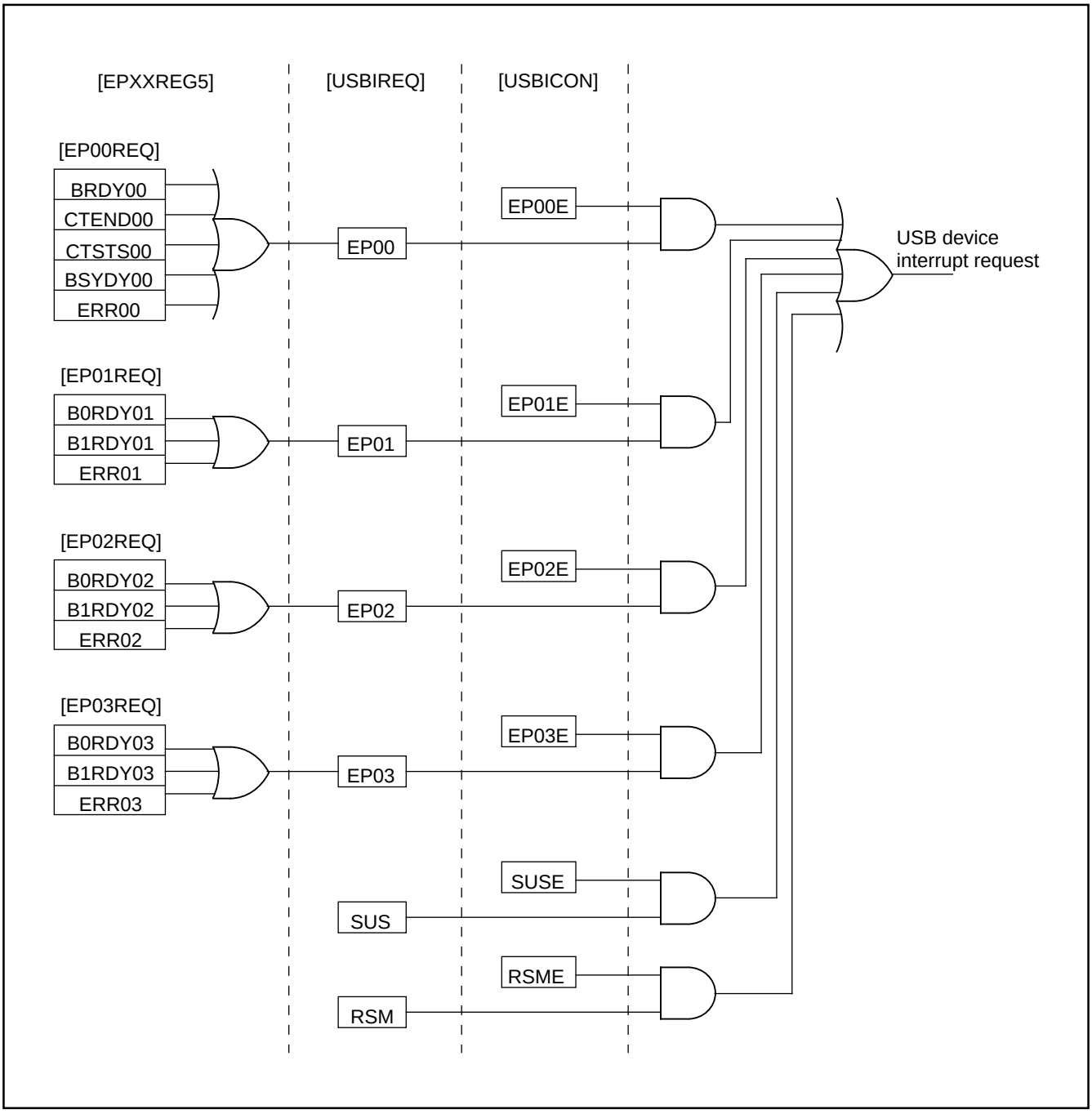


Fig. 29 USB device interrupt control

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
0010₁₆	USB control register	USBCON	USBE	UCLKCON	USBDIFE	VREFE	VREFCON	TRONE	TRONCON	WKUP
0011₁₆	USB Function enable register	USBAE								AD0E
0012₁₆	USB function address register	USBA0	USBADD0[6:0]							
0013₁₆										
0014₁₆	Frame number register Low	FNUML	FNUM[7:0]							
0015₁₆	Frame number register High	FNUMH						FNUM[10:8]		
0016₁₆	USB interrupt source enable register	USBICON	RSME	SUSE			EP03E	EP02E	EP01E	EP00E
0017₁₆	USB interrupt source register	USBIREQ	RSM	SUS			EP03	EP02	EP01	EP00
0018₁₆	Endpoint index register	USBINDEX							EPIDX[1:0]	
0019₁₆	Endpoint field register 1	EPXXREG1								
001A₁₆	Endpoint field register 2	EPXXREG2								
001B₁₆	Endpoint field register 3	EPXXREG3								
001C₁₆	Endpoint field register 4	EPXXREG4								
001D₁₆	Endpoint field register 5	EPXXREG5								
001E₁₆	Endpoint field register 6	EPXXREG6								
001F₁₆	Endpoint field register 7	EPXXREG7								
0FEC₁₆	Endpoint field register 8	EPXXREG8								
0FED₁₆	Endpoint field register 9	EPXXREG9								

(1) Endpoint 00

0019₁₆	EP00 stage register	EP00STG								SETUP00
001A₁₆	EP00 control register 1	EP00CON1							PID00[1:0]	
001B₁₆	EP00 control register 2	EP00CON2								BVAL00
001C₁₆	EP00 control register 3	EP00CON3								CTENDE00
001D₁₆	EP00 interrupt source register	EP00REQ				ERR00	BSRDY00	CTSTS00	CTEND00	BRDY00
001E₁₆	EP00 byte number register	EP00BYT					BBYT00[3:0]			
001F₁₆										
0FEC₁₆										
0FED₁₆	EP00 buffer area set register	EP00BUF						BADD00[4:0]		

(2) Endpoint 01

0019₁₆	EP01 set register	EP01CFG	TYP01[1:0]		DIR01	ITMD01	SQCL01	DBLB01	BSIZ01[1:0]	
001A₁₆	EP01 control register 1	EP01CON1							PID01[1:0]	
001B₁₆	EP01 control register 2	EP01CON2								B0VAL01
001C₁₆	EP01 control register 3	EP01CON3								B1VAL01
001D₁₆	EP01 interrupt source register	EP01REQ						ERR01	B1RDY01	B0RDY01
001E₁₆	EP01 byte number register 0	EP01BYT0					B0BYT01[6:0]			
001F₁₆	EP01 byte number register 1	EP01BYT1					B1BYT01[6:0]			
0FEC₁₆	EP01 MAX. packet size register	EP01MAX					MXPS01[6:0]			
0FED₁₆	EP01 buffer area set register	EP01BUF						BADD01[4:0]		

(3) Endpoint 02

0019₁₆	EP02 set register	EP02CFG	TYP02[1:0]		DIR02	ITMD02	SQCL02	DBLB02	BSIZ02[1:0]	
001A₁₆	EP02 control register 1	EP02CON1							PID02[1:0]	
001B₁₆	EP02 control register 2	EP02CON2								B0VAL02
001C₁₆	EP02 control register 3	EP02CON3								B1VAL02
001D₁₆	EP02 interrupt source register	EP02REQ						ERR02	B1RDY02	B0RDY02
001E₁₆	EP02 byte number register 0	EP02BYT0					B0BYT02[6:0]			
001F₁₆	EP02 byte number register 1	EP02BYT1					B1BYT02[6:0]			
0FEC₁₆	EP02 MAX. packet size register	EP02MAX					MXPS02[6:0]			
0FED₁₆	EP02 buffer area set register	EP02BUF						BADD02[4:0]		

(4) Endpoint 03

0019₁₆	EP03 set register	EP03CFG	TYP03[1:0]		DIR03	ITMD03	SQCL03	DBLB03	BSIZ03[1:0]	
001A₁₆	EP03 control register 1	EP03CON1							PID03[1:0]	
001B₁₆	EP03 control register 2	EP03CON2								B0VAL03
001C₁₆	EP03 control register 3	EP03CON3								B1VAL03
001D₁₆	EP03 interrupt source register	EP03REQ						ERR03	B1RDY03	B0RDY03
001E₁₆	EP03 byte number register 0	EP03BYT0					B0BYT03[6:0]			
001F₁₆	EP03 byte number register 1	EP03BYT1					B1BYT03[6:0]			
0FEC₁₆	EP03 MAX. packet size register	EP03MAX					MXPS03[6:0]			
0FED₁₆	EP03 buffer area set register	EP03BUF						BADD03[4:0]		

: Not used

Fig. 30 USB related registers

USB Related Registers

The USB related registers are shown below.

b7 b0

USB control register (USBCON) [address 0010_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
WKUP	Remote wakeup bit	0 : Returning to BUS idle state by writing "1" first and then "0". (Remote wakeup signal) 1 : K-state output	0	–	O	O
TRONCON	TrON output control bit	0 : "L" output mode (valid in TRONE = "1") 1 : "H" output mode (valid in TRONE = "1")	0	–	O	O
TRONE	TrON output enable bit	0 : TrON port output disabled (Hi-Z state) 1 : TrON port output enabled	0	–	O	O
VREFCON	USB reference voltage control bit	0 : Normal mode (valid in VREFE = "1") 1 : Low current mode (valid in VREFE = "1")	0	–	O	O
VREFE	USB reference voltage enable bit	0 : USB reference voltage circuit operation disabled 1 : USB reference voltage circuit operation enabled	0	–	O	O
USBDIFE	USB difference input enable bit	0 : Upstream-port difference input circuit operation disabled 1 : Upstream--port difference input circuit operation enabled	0	–	O	O
UCLKCON	USB clock select bit	0 : External oscillating clock f(X_{IN}) 1 : PLL circuit output clock f_{VCO}	0	–	O	O
USBE	USB module operation enable bit	0 : USB module reset 1 : USB module operation enabled	0	–	O	O

–: State remaining

Fig. 31 Structure of USB control register

b7 b0

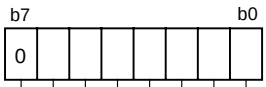
0 0 0 0 0 0 0

USB function enable register (USBAE) [address 0011_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
AD0E	USB function enable bit	0: USB function address register invalidated 1: USB function address register validated	0	–	O	O
b7:b1	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 32 Structure of USB function enable register

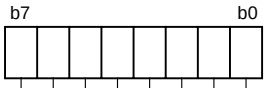


USB function address register (USBA0) [address 0012$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
USBADD0 [6:0]	USB function address bit	In AD0E = "0", this value changes after writing. In AD0E = "1", this value changes after completion of SET_ADDRESS control transferring.	0	0	O	O
b7	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

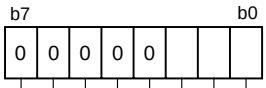
Fig. 33 Structure of USB function address register



Frame number register Low (FNUML) [address 0014$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
FNUM [7:0]	Frame number low bit	The frame number is updated at SOF reception.	In-definite	In-definite	O	×

Fig. 34 Structure of Frame number register Low

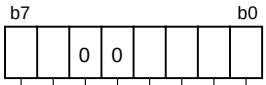


Frame number register High (FNUMH) [address 0015$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
FNUM [10:8]	Frame number high bit	The frame number is updated at SOF reception.	In-definite	In-definite	O	×
b7:b3	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 35 Structure of Frame number register High



USB interrupt source enable register (USBICON) [address 0016_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
EP00E	USB function/Endpoint 0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
EP01E	USB function/Endpoint 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
EP02E	USB function/Endpoint 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
EP03E	USB function/Endpoint 3 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
b5:b4	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
SUSE	Suspend interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
RSME	Resume interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O

–: State remaining

Fig. 36 Structure of USB interrupt source enable register

b7 b0

		0	0				

USB interrupt source register (USBIREQ) [address 0017_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
EP00	USB function/Endpoint 0 interrupt bit	This bit is set to “1” when any one of EP00 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP00 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	×
EP01	USB function/Endpoint 1 interrupt bit	This bit is set to “1” when any one of EP01 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP01 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	×
EP02	USB function/Endpoint 2 interrupt bit	This bit is set to “1” when any one of EP02 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP02 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	×
EP03	USB function/Endpoint 3 interrupt bit	This bit is set to “1” when any one of EP03 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP03 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	×
b5:b4	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
SUS	Suspend interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued This bit is set to “1” when detecting 3 ms or more of J-state, using USB clock (f_{USB}) at 48 MHz. “0” can be set by software, but “1” cannot be set.	0	0	O	O
RSM	Resume interrupt bit	This bit is set to “1” when the USB bus state changes from J-state to K-state or SE0 in the resume interrupt enable bit = “1”. It is also “1” in the condition of internal clock stopped. This bit is cleared to “0” by clearing the resume interrupt enable bit. Writing to this bit causes no state change.	0	0	O	×

–: State remaining

Fig.37 Structure of USB interrupt source register

b7 b0

0	0	0	0	0	0		

Endpoint index register (USBINDEX) [address 0018_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
EPIDX [1:0]	Endpoint index bit	b1 b0 0 0 : Endpoint 0 0 1 : Endpoint 1 1 0 : Endpoint 2 1 1 : Endpoint 3	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 38 Structure of Endpoint index register

(1) Endpoint 00

b7 b0

0	0	0	0	0	0	0	

EP00 stage register (EP00STG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
SETUP00	SETUP packet detection bit	This bit is set to “1” at reception of SETUP packet. Writing “0” to this bit clears this bit if the next SETUP token does not occur. Writing “1” to this bit causes no state change of the status flags.	1	1	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 39 Structure of EP00 stage register

b7 b0

0	0	0	0	0	0		

EP00 control register 1 (EP00CON1) [address $001A_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
PID00 [1:0]	Response PID bit	b1 b0 0 0 : NAK 0 1 : Automatic response (ACK, NAK, DATA0, DATA1) 1 X : STALL At occurrence of control transfer error: B1 is set to “1” by the hardware. At reception of SETUP token: B1 and b0 are cleared to “0” by the hardware.	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 40 Structure of EP00 control register 1

b7 b0

0	0	0	0	0	0	0	

EP00 control register 2 (EP00CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BVAL00	Buffer enable bit	0 : NAK transmission (SIE is disabled to read a buffer.) 1 : Transmitting/receiving data set state (SIE is possible to read from/write to a buffer.) At reception of SETUP token: This bit is cleared to “0” by the hardware.	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 41 Structure of EP00 control register 2

b7 b0

0	0	0	0	0	0	0	

EP00 control register 3 (EP00CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
CTENDE00	Control transfer completion enable bit	0 : NAK transmission in the status stage 1 : Control transfer completion enabled (SIE transmits NULL/ACK.) (valid in PID00 = "012") At reception of SETUP token: This bit is cleared to "0" by the hardware.	0	–	O	O
b7:b1	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 42 Structure of EP00 control register 3

b7 b0

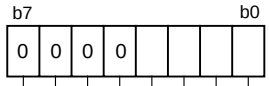
0	0	0					

EP00 interrupt source register (EP00REQ) [address 001D16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BRDY00	USB function/Endpoint 0 buffer ready interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when the buffer is ready state (enabled to be read/written) on USB function/Endpoint 0. "0" can be set by software, but "1" cannot be set.	0	0	O	O
CTEND00	USB function/Endpoint 0 control transfer completion interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when control transfer is completed (NULL/ACK transmission in the status stage) on USB function/Endpoint 0. "0" can be set by software, but "1" cannot be set.	0	0	O	O
CTSTS00	USB function/Endpoint 0 status stage transition interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when transition to status stage occurs in CTENDE00 = "0" (control transfer completion disabled) on USB function/Endpoint 0. "0" can be set by software, but "1" cannot be set. <Transition to status stage occurrence factor> At transfer of control write: When receiving IN-token in data stage (OUT) At transfer of control read: When receiving OUT-token in data stage (IN) At no data transfer: Nothing occurs.	0	0	O	O
BSRDY00	USB function/Endpoint 0 SETUP buffer ready interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when the exclusive buffer for SETUP is ready state (enabled to be read) on USB function/Endpoint 0. "0" can be set by software, but "1" cannot be set.	0	0	O	O
ERR00	USB function/Endpoint 0 error interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when control transfer error occurs on USB function/Endpoint 0. This bit is cleared to "0" by the hardware when receiving SETUP token. "0" can be set by software, but "1" cannot be set.	0	0	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 43 Structure of EP00 interrupt source register

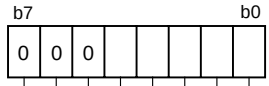


EP00 byte number register (EP00BYT) [address 001E16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BBYT00 [3:0]	Transmit/receive byte number bit	OUT : The received byte number is automatically set. IN : Set the transmitting byte number.	0	—	O	O
b7:b4	Not used	Write 0 when writing. 0 is read when reading.	—	—	O	O

—: State remaining

Fig. 44 Structure of EP00 byte number register



EP00 buffer area set register (EP00BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BADD00 [4:0]	EP00 beginning address set bit	Set the beginning address of EP00's buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040₁₆ 0 0 0 1 1 : 0060₁₆ ..... 1 1 1 1 0 : 03C0₁₆ 1 1 1 1 1 : 03E0₁₆	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 45 Structure of EP00 buffer area set register

(2) Endpoint 01

b7 b0

EP01 set register (EP01CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BSIZ01 [1:0]	Double buffer beginning address set bit	In double buffer mode set the beginning address of buffer 1 area, using a relative value for the beginning address of buffer 0. b1b0 0 0 = 8 bytes 0 1 = 16 bytes 1 0 = 64 bytes 1 1 = 128 bytes	0	–	O	O
DBLB01	Buffer mode select bit	0 : Single buffer mode 1 : Double buffer mode	0	–	O	O
SQCL01	Sequence toggle bit clear bit	0 : Toggle bit clear disabled 1 : Writing “1” clears the toggle bit and DATA0 is used as the next data PID. “0” is always read when reading.	0	–	O	O
ITMD01	Interrupt toggle mode select bit	0 : Normal mode 1 : Continuous toggle mode (valid at Interrupt IN transfer)	0	–	O	O
DIR01	Transfer direction bit	0 : OUT (Data is received from the host.) 1 : IN (Data is transmitted to the host.)	0	–	O	O
TYP01 [1:0]	Transfer type bite	b7b6 0 0 : Transfer disabled 0 1 : Bulk transfer 1 0 : Interrupt transfer 1 1 : Isochronous transfer	0	–	O	O

–: State remaining

Fig. 46 Structure of EP01 set register

b7 b0

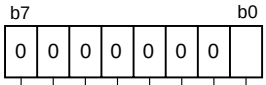
0 0 0 0 0 0

EP01 control register 1 (EP01CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
PID01 [1:0]	Response PID bit	b1 b0 0 0 : NAK 0 1 : Automatic response (ACK, NAK, DATA0, DATA1) 1 X : STALL At occurrence of over-max. packet size : B1 is set to “1” by the hardware.	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 47 Structure of EP01 control register 1

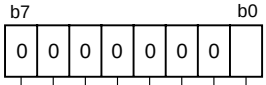


EP01 control register 2 (EP01CON2) [address 001B16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0VAL01	Buffer 0 enable bit	When the selected endpoint is IN, writing “1” to this bit makes the transmitting data a set state (SIE is possible to read). When the selected endpoint is OUT, writing “1” to this bit makes data reception possible (SIE is possible to write).	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 48 Structure of EP01 control register 2

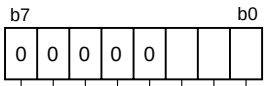


EP01 control register 3 (EP01CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B1VAL01	Buffer 1 enable bit	When the selected endpoint is IN, writing “1” to this bit makes the transmitting data a set state (SIE is possible to read). When the selected endpoint is OUT, writing “1” to this bit makes data reception possible (SIE is possible to write). In double buffer mode this bit is valid.	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

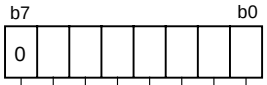
Fig. 49 Structure of EP01 control register 3



EP01 interrupt source register (EP01REQ) [address 001D16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0RDY01	USB function/Endpoint 1 buffer 0 ready interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when the buffer 0 is ready state (enabled to be read/written) on USB function/Endpoint 1. “0” can be set by software, but “1” cannot be set.	0	0	O	O
B1RDY01	USB function/Endpoint 1 buffer 1 ready interrupt bit	0: No interrupt request issued 1: Interrupt request issued In single buffer mode this bit is invalid. This bit is set to “1” when the buffer 1 is ready state (enabled to be read/written) on USB function/Endpoint 1 in double buffer mode. “0” can be set by software, but “1” cannot be set.	0	0	O	O
ERR01	USB function/Endpoint 1 error interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when STALL response occurs on USB function/Endpoint 1. “0” can be set by software, but “1” cannot be set.	0	0	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

Fig. 50 Structure of EP01 interrupt source register

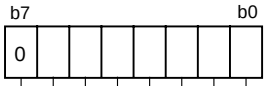


EP01 byte number register 0 (EP01BYT0) [address 001E16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0BYT01 [6:0]	IN : Transmit byte number bit	Single buffer mode: Set the transmitting byte number. Double buffer mode : Set the transmitting byte number of buffer 0.	0	–	O	O
	OUT : Receive byte number bit	Single buffer mode : The received byte number is automatically set. Double buffer mode : The received byte number of buffer 0 is automatically set.	0	–	O	×
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 51 Structure of EP01 byte number register 0

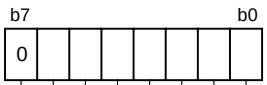


EP01 byte number register 1 (EP01BYT1) [address 001F16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B1BYT01 [6:0]	IN : Transmit byte number bit	Single buffer mode: These bits are invalid. Double buffer mode : Set the transmitting byte number of buffer 1.	0	–	O	O
	OUT : Receive byte number bit	Single buffer mode: These bits are invalid. Double buffer mode : The received byte number of buffer 1 is automatically set.	0	–	O	×
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 52 Structure of EP01 byte number register 1

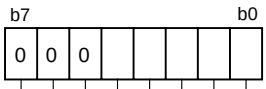


EP01 MAX. packet size register (EP01MAX) [address 0FEC16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
MXPS01 [6:0]	Max. packet size bit	IN : These bits are invalid. OUT : Set the maximum packet size.	0	–	O	O
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 53 Structure of EP01 MAX. packet size register



EP01 buffer area set register (EP01BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BADD01 [4:0]	EP01 beginning address set bit	Set the beginning address of EP01's buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040₁₆ 0 0 0 1 1 : 0060₁₆ ..... 1 1 1 1 0 : 03C0₁₆ 1 1 1 1 1 : 03E0₁₆	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 54 Structure of EP01 buffer area set register

(3) Endpoint 02

b7 b0

EP02 set register (EP02CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BSIZ02 [1:0]	Double buffer beginning address set bit	In double buffer mode set the beginning address of buffer 1 area, using a relative value for the beginning address of buffer 0. b1b0 0 0 = 8 bytes 0 1 = 16 bytes 1 0 = 64 bytes 1 1 = 128 bytes	0	–	O	O
DBLB02	Buffer mode select bit	0 : Single buffer mode 1 : Double buffer mode	0	–	O	O
SQCL02	Sequence toggle bit clear bit	0 : Toggle bit clear disabled 1 : Writing “1” clears the toggle bit and DATA0 is used as the next data PID. “0” is always read when reading.	0	–	O	O
ITMD02	Interrupt toggle mode select bit	0 : Normal mode 1 : Continuous toggle mode (valid at Interrupt IN transfer)	0	–	O	O
DIR02	Transfer direction bit	0 : OUT (Data is received from the host.) 1 : IN (Data is transmitted to the host.)	0	–	O	O
TYP02 [1:0]	Transfer type bite	b7b6 0 0 : Transfer disabled 0 1 : Bulk transfer 1 0 : Interrupt transfer 1 1 : Isochronous transfer	0	–	O	O

–: State remaining

Fig. 55 Structure of EP02 set register

b7 b0

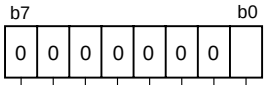
0 0 0 0 0 0

EP02 control register 1 (EP02CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
PID02 [1: 0]	Response PID bit	b1 b0 0 0 : NAK 0 1 : Automatic response (ACK, NAK, DATA0, DATA1) 1 X : STALL At occurrence of over-max. packet size : B1 is set to “1” by the hardware.	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 56 Structure of EP02 control register 1

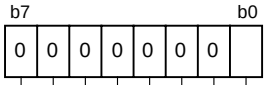


EP02 control register 2 (EP02CON2) [address 001B16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0VAL02	Buffer 0 enable bit	When the selected endpoint is IN, writing “1” to this bit makes the transmitting data a set state (SIE is possible to read). When the selected endpoint is OUT, writing “1” to this bit makes data reception possible (SIE is possible to write).	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 57 Structure of EP02 control register 2

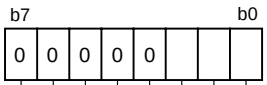


EP02 control register 3 (EP02CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B1VAL02	Buffer 1 enable bit	When the selected endpoint is IN, writing “1” to this bit makes the transmitting data a set state (SIE is possible to read). When the selected endpoint is OUT, writing “1” to this bit makes data reception possible (SIE is possible to write). In double buffer mode this bit is valid.	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

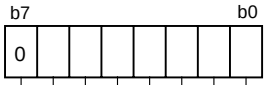
Fig. 58 Structure of EP02 control register 3



EP02 interrupt source register (EP02REQ) [address 001D16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0RDY02	USB function/Endpoint 2 buffer 0 ready interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued This bit is set to “1” when the buffer 0 is ready state (enabled to be read/written) on USB function/Endpoint 2. “0” can be set by software, but “1” cannot be set.	0	0	O	O
B1RDY02	USB function/Endpoint 2 buffer 1 ready interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued In single buffer mode this bit is invalid. This bit is set to “1” when the buffer 1 is ready state (enabled to be read/written) on USB function/Endpoint 2 in double buffer mode. “0” can be set by software, but “1” cannot be set.	0	0	O	O
ERR02	USB function/Endpoint 2 error interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued This bit is set to “1” when STALL response occurs on USB function/Endpoint 2. “0” can be set by software, but “1” cannot be set.	0	0	O	O
b7 to b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

Fig. 59 Structure of EP02 interrupt source register

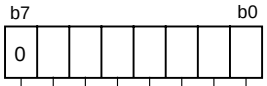


EP02 byte number register 0 (EP02BYT0) [address 001E16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0BYT02 [6:0]	IN : Transmit byte number bit	Single buffer mode: Set the transmitting byte number. Double buffer mode : Set the transmitting byte number of buffer 0.	0	–	O	O
	OUT : Receive byte number bit	Single buffer mode: The received byte number is automatically set. Double buffer mode : The received byte number of buffer 0 is automatically set.	0	–	O	×
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 60 Structure of EP02 byte number register 0

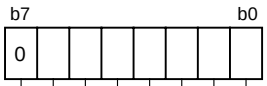


EP02 byte number register 1 (EP02BYT1) [address 001F16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B1BYT02 [6:0]	IN : Transmit byte number bit	Single buffer mode: These bits are invalid. Double buffer mode : Set the transmitting byte number of buffer 1.	0	–	O	O
	OUT : Receive byte number bit	Single buffer mode: These bits are invalid. Double buffer mode : The received byte number of buffer 1 is automatically set.	0	–	O	×
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 61 Structure of EP02 byte number register 1

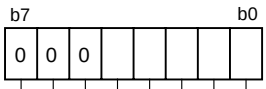


EP02 MAX. packet size register (EP02MAX) [address 0FEC16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
MXPS02 [6:0]	Max. packet size bit	IN : These bits are invalid. OUT : Set the maximum packet size.	0	–	O	O
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 62 Structure of EP02 MAX. packet size register



EP02 buffer area set register (EP02BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BADD02 [4:0]	EP02 beginning address set bit	Set the beginning address of EP02's buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040₁₆ 0 0 0 1 1 : 0060₁₆ ..... 1 1 1 1 0 : 03C0₁₆ 1 1 1 1 1 : 03E0₁₆	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 63 Structure of EP02 buffer area set register

(4) Endpoint 03

b7 b0

EP03 set register (EP03CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BSIZ03 [1:0]	Double buffer beginning address set bit	In double buffer mode set the beginning address of buffer 1 area, using a relative value for the beginning address of buffer 0. b1b0 0 0 = 8 bytes 0 1 = 16 bytes 1 0 = 64 bytes 1 1 = 128 bytes	0	–	O	O
DBLB03	Buffer mode select bit	0 : Single buffer mode 1 : Double buffer mode	0	–	O	O
SQCL03	Sequence toggle bit clear bit	0 : Toggle bit clear disabled 1 : Writing “1” clears the toggle bit and DATA0 is used as the next data PID. “0” is always read when reading.	0	–	O	O
ITMD03	Interrupt toggle mode select bit	0 : Normal mode 1 : Continuous toggle mode (valid at Interrupt IN transfer)	0	–	O	O
DIR03	Transfer direction bit	0 : OUT (Data is received from the host.) 1 : IN (Data is transmitted to the host.)	0	–	O	O
TYP03 [1:0]	Transfer type bit	b7b6 0 0 : Transfer disabled 0 1 : Bulk transfer 1 0 : Interrupt transfer 1 1 : Isochronous transfer	0	–	O	O

–: State remaining

Fig. 64 Structure of EP03 set register

b7 b0

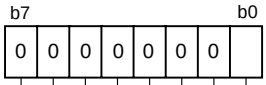
0 0 0 0 0 0

EP03 control register 1 (EP03CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
PID03 [1:0]	Response PID bit	b1 b0 0 0 : NAK 0 1 : Automatic response (ACK, NAK, DATA0, DATA1) 1 X : STALL At occurrence of over-max. packet size : B1 is set to “1” by the hardware.	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 65 Structure of EP03 control register 1

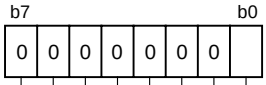


EP03 control register 2 (EP03CON2) [address 001B16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0VAL03	Buffer 0 enable bit	When the selected endpoint is IN, writing “1” to this bit makes the transmitting data a set state (SIE is possible to read). When the selected endpoint is OUT, writing “1” to this bit makes data reception possible (SIE is possible to write).	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 66 Structure of EP03 control register 2

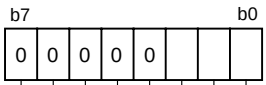


EP03 control register 3 (EP03CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B1VAL03	Buffer 1 enable bit	When the selected endpoint is IN, writing “1” to this bit makes the transmitting data a set state (SIE is possible to read). When the selected endpoint is OUT, writing “1” to this bit makes data reception possible (SIE is possible to write). In double buffer mode this bit is valid.	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

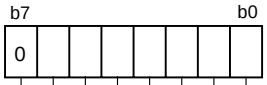
Fig. 67 Structure of EP03 control register 3



EP03 interrupt source register (EP03REQ) [address 001D16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0RDY03	USB function/Endpoint 3 buffer 0 ready interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued This bit is set to “1” when the buffer 0 is ready state (enabled to be read/written) on USB function/Endpoint 3. “0” can be set by software, but “1” cannot be set.	0	0	O	O
B1RDY03	USB function/Endpoint 3 buffer 1 ready interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued In single buffer mode this bit is invalid. This bit is set to “1” when the buffer 1 is ready state (enabled to be read/written) on USB function/Endpoint 3 in double buffer mode. “0” can be set by software, but “1” cannot be set.	0	0	O	O
ERR03	USB function/Endpoint 3 error interrupt bit	0 : No interrupt request issued 1 : Interrupt request issued This bit is set to “1” when STALL response occurs on USB function/Endpoint 3. “0” can be set by software, but “1” cannot be set.	0	0	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

Fig. 68 Structure of EP03 interrupt source register

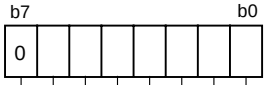


EP03 byte number register 0 (EP03BYT0) [address 001E16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0BYT03 [6:0]	IN : Transmit byte number bit	Single buffer mode: Set the transmitting byte number. Double buffer mode : Set the transmitting byte number of buffer 0.	0	–	O	O
	OUT : Receive byte number bit	Single buffer mode: The received byte number is automatically set. Double buffer mode : The received byte number of buffer 0 is automatically set.	0	–	O	×
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 69 Structure of EP03 byte number register 0

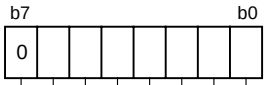


EP03 byte number register 1 (EP03BYT1) [address 001F16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B1BYT03 [6:0]	IN : Transmit byte number bit	Single buffer mode: These bits are invalid. Double buffer mode : Set the transmitting byte number of buffer 1.	0	–	O	O
	OUT : Receive byte number bit	Single buffer mode: These bits are invalid. Double buffer mode : The received byte number of buffer 1 is automatically set.	0	–	O	×
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 70 Structure of EP03 byte number register 1

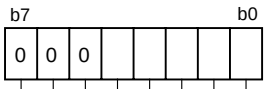


EP03 MAX. packet size register (EP03MAX) [address 0FEC16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
MXPS03 [6:0]	Max. packet size bit	IN : These bits are invalid. OUT : Set the maximum packet size.	0	–	O	O
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 71 Structure of EP03 MAX. packet size register



EP03 buffer area set register (EP03BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BADD03 [4:0]	EP03 beginning address set bit	Set the beginning address of EP03's buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040₁₆ 0 0 0 1 1 : 0060₁₆ ..... 1 1 1 1 0 : 03C0₁₆ 1 1 1 1 1 : 03E0₁₆	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 72 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.

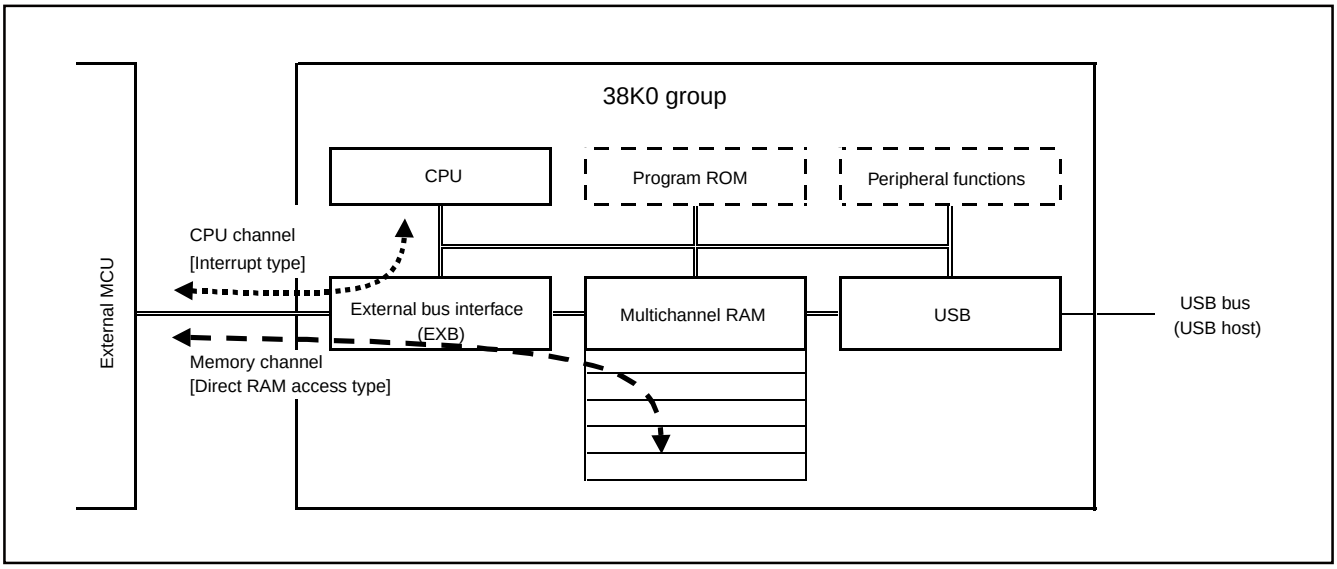


Fig. 73 External bus interface

●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 ϕ, 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 ϕ = 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 ϕ = 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.

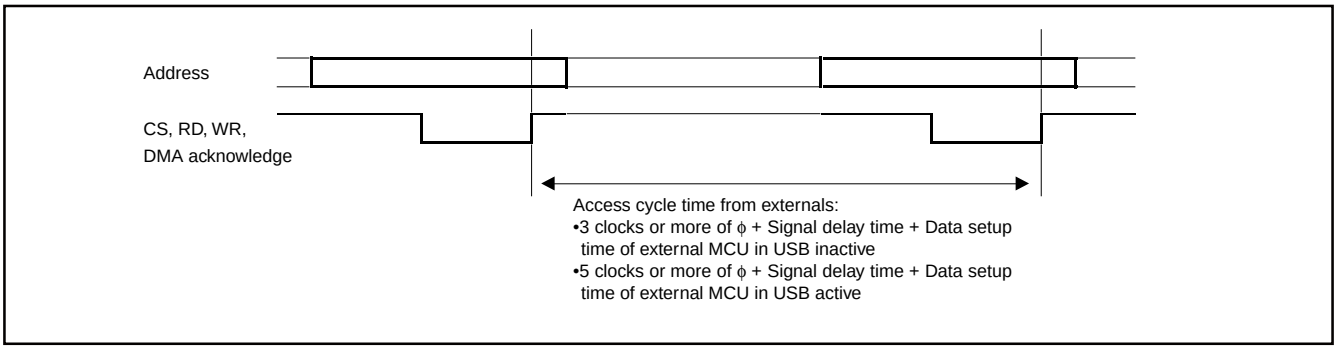


Fig. 74 Data transfer timing of memory channel

EXB Pin Assignment

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

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 (DQ_0 to DQ_7)
•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)

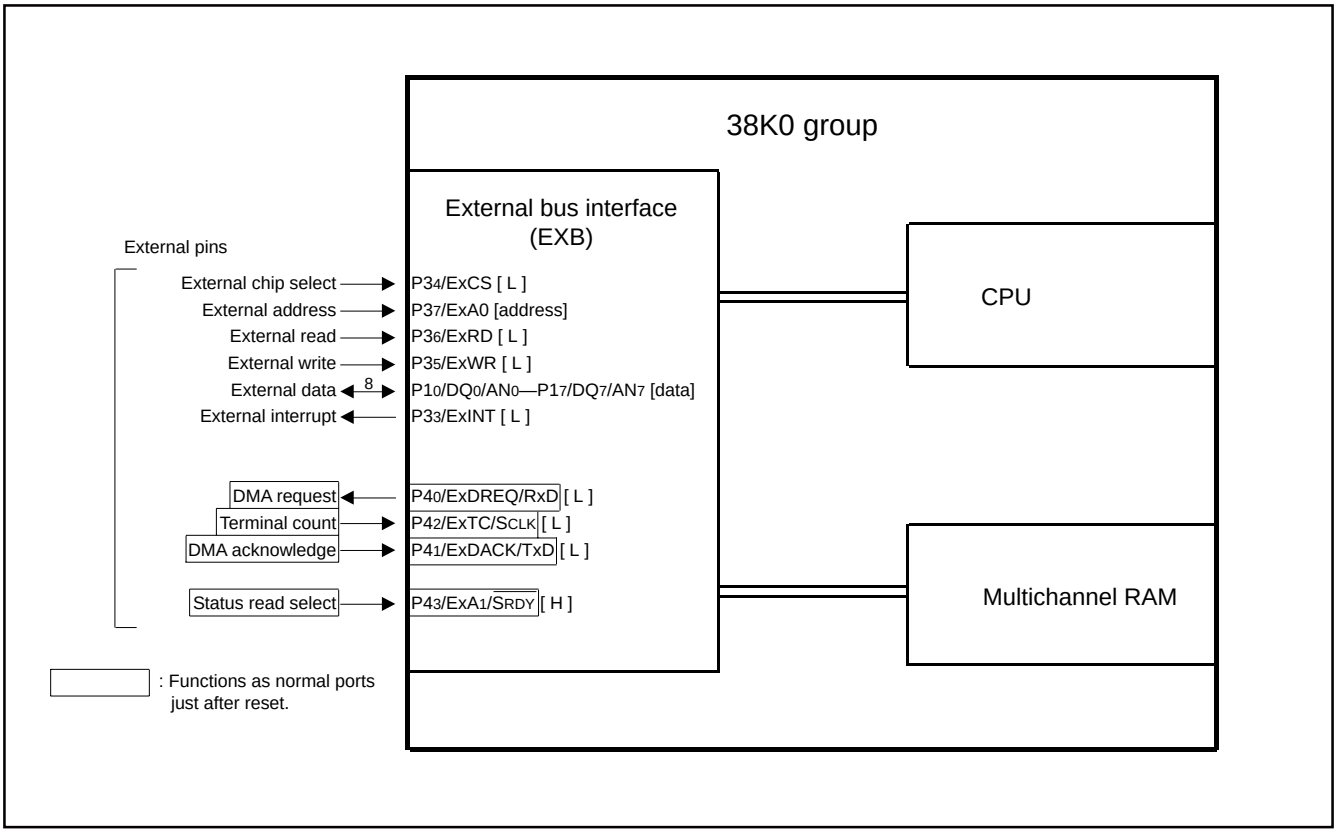


Fig. 75 External bus interface (EXB) pin assignment

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

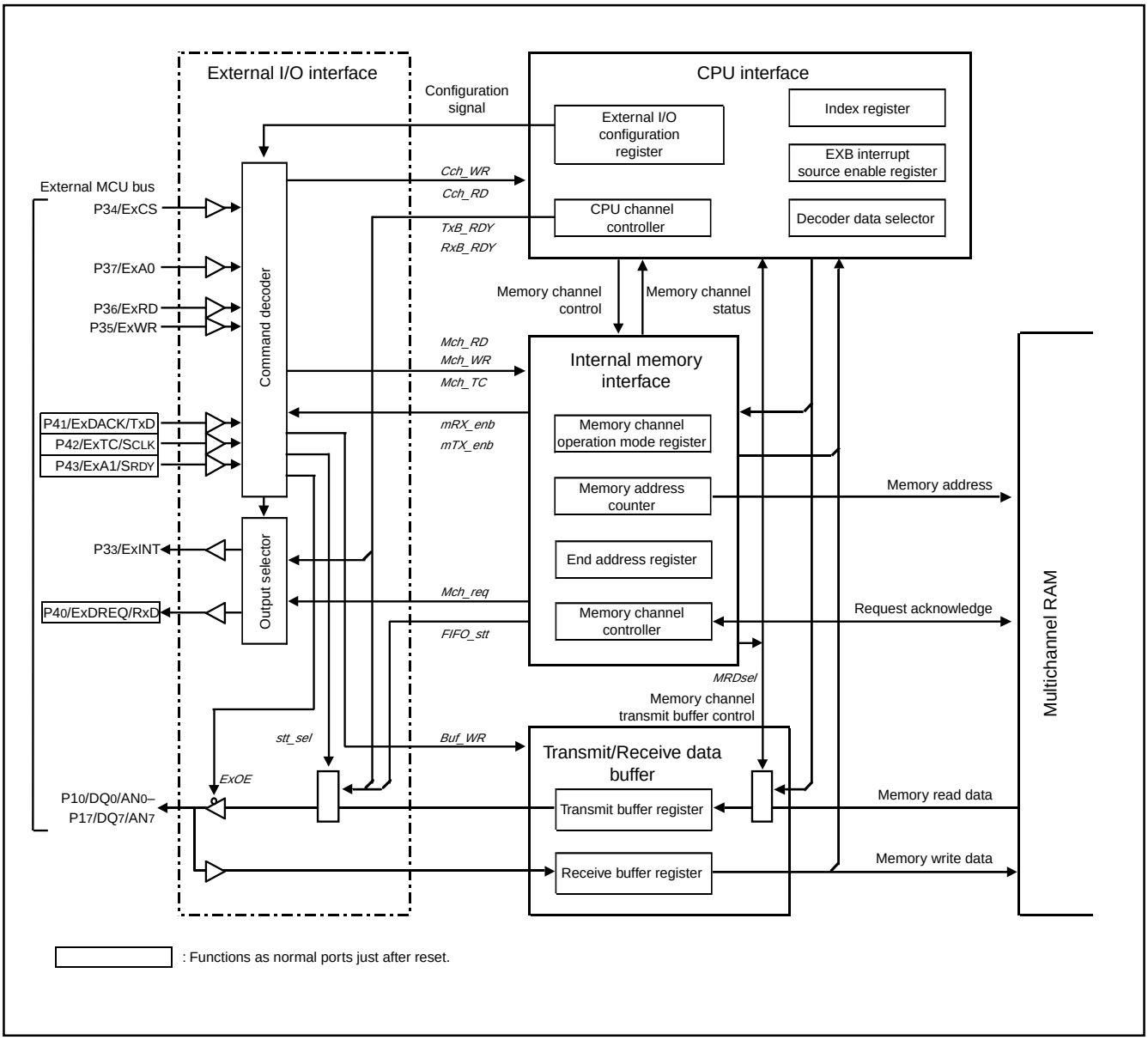


Fig. 76 Block diagram of external bus interface (EXB)

(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 $P3_3$/ExINT pin and $P4_0$/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 ..... $P3_4$/ExCS
- •Address ..... $P3_7$/ExA0
- •Data ..... $P1_0$/DQ_0/AN_0 to $P1_7$/DQ_7/AN_7
- •Read ..... $P3_6$/ExRD
- •Write ..... $P3_5$/ExWR
- •Interrupt request ..... $P3_3$/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 ..... $P4_0$/ExDREQ/RxD
- •DMA acknowledgment ..... $P4_1$/ExDACK/TxD
- •Terminal count ..... $P4_2$/ExTC/SCLK

It also has the status read select pin ($P4_3$/ExA1/$\overline{SRDY}$ 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 ..... $P4_3$/ExA1/$\overline{SRDY}$

●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 38K0 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.

●P3$_3$/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.

●P4$_0$/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 94 and 97.)

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 95 and 98.)

●P4$_1$/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.

●P4$_2$/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 “11$_2$”, 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 Figure 96.)

EXB Register List

The EXB register list is shown below.

Address	Register Name	SYMBOL	EXB SFR bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0030₁₆	EXB interrupt source enable register	EXBICON						MC_ENB	TXB_ENB	RXB_EMB
0031₁₆	EXB interrupt source register	EXBIREQ					MC_STS[1:0]		TXB_EMPTY	RXB_FULL
0033₁₆	EXB index register	EXBINDEX	0	0	0	0	0	INDEX[2:0]		
0034₁₆	Register window 1 (low)	EXBREG1	LOW_WIN[7:0]							
0035₁₆	Register window 2 (high)	EXBREG2	HIGH_WIN[7:0]							

: Not used

0 : "0" fixed

Fig. 77 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 0034₁₆ and 0035₁₆ as a register window depending on the value of EXB index register at address 0033₁₆.

Index	low high	Register Name	SYMBOL	EXB SFR bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00₁₆	low	External I/O configuration register	EXBCFGL				A1_CTR	INT_CTR[2:0]			EXB_CTR
	high		EXBCFGH				TC_CTR	DAK_CTR[1:0]		DRQ_CTR[1:0]	
01₁₆	low	Transmit/Receive buffer register	RXBUF/TXBUF	At CPU read : RXBUF[7:0] At CPU write : TXBUF[7:0]							
	high		—								
02₁₆	low	Memory channel operation mode register	MCHMOD						BURST	MC_DIR[1:0]	
	high		—								
03₁₆	low	Memory address counter	MEMADL	IM_A[7:0]							
	high		MEMADH	0	0	0	0	0	IM_A[10:8]		
04₁₆	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. 78 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.

b7 b0

0	0	0	0	0			

EXB interrupt source enable register (EXBICON) [address 0030_{16}] (**Note**)

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
RXB_ENB	CPU channel receive enable bit	0 : Operation disabled (Interrupt disabled) 1 : Operation enabled (Receive buffer full interrupt enabled)	0	–	O	O
TXB_ENB	CPU channel transmit enable bit	0 : Operation disabled (Interrupt disabled) 1 : Operation enabled (Transmit buffer empty interrupt enabled)	0	–	O	O
MC_ENB	Memory channel operation enable bit	0 : Operation disabled (Memory channel operation end interrupt disabled) 1 : Operation enabled (Memory channel operation end interrupt disabled)	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Note: Do not set each bit simultaneously.

Fig. 79 Structure of EXB interrupt source enable register

b7 b0

0	0	0	0				

EXB interrupt source register (EXBIREQ) [address 0031_{16}] (**Note 1**)

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
RXB_FULL	Receive buffer full bit	0 : Receive buffer empty 1 : Receive buffer full	0	0 (Note 3)	O	–
TXB_EMPTY	Transmit buffer empty bit	0 : Transmit buffer full 1 : Transmit buffer empty	0	0 (Note 4)	O	–
MC_STS [1:0] (**Note 2**)	Memory channel status bits	b3b2 0 0 : Memory channel operation stopped 0 1 : Memory channel being operating; No external access 1 0 : Memory channel being operating; External accessing 1 1 : Memory channel operation end; Memory channel operation end interrupt generated	0	0	O	–
b7:b4	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

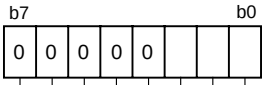
Notes 1: When the the ExA1 pin control bit of external I/O configuration register is “1”, the external MCU bus can read this register contents by setting the ExA1 pin to “H”.

2: The memory channel status bits indicate the status of memory channel. In MC_ENB = “0” these bits are always “00_2”. When the memory channel operation ends, these bits are set to “11_2” and the memory channel operation end interrupt is generated.
These bits can be read out during operation, so that it will show that whether the external MCU bus is accessing or not.

3: This bit is cleared to “0” when reading the transmit/receive buffer register in the CPU channel receive enable bit = “1” or when the CPU channel receive enable bit is “0”.

4: This bit is cleared to “0” when writing to the transmit/receive buffer register in the CPU channel transmit enable bit = “1” or when the CPU channel transmit enable bit is “0”.

Fig. 80 Structure of EXB interrupt source register

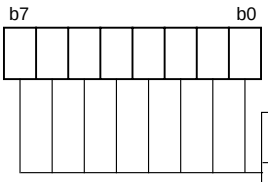


EXB index register (EXBINDEX) [address 0033$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
INDEX [2:0]	Index bits	The accessible register, using the register window, depends on these index bits contents as follows: b2b1b0 0 0 0 : External I/O configuration register 0 0 1 : Transmit/Receive buffer register 0 1 0 : Memory channel operation mode register 0 1 1 : Memory address counter 1 0 0 : End address register 1 0 1 : Do not set. 1 1 0 : Do not set. 1 1 1 : Do not set.	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

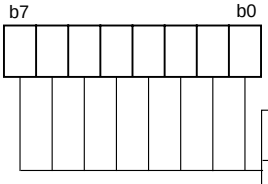
Fig. 81 Structure of EXB index register



Register window 1 (EXBREG1) [address 0034$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
LOW_WIN [7:0]	–	The accessible register, using this register window, depends on the EXB index register contents as follows: Index value “00$_{16}$” : External I/O configuration register “01$_{16}$” : Transmit/Receive buffer register “02$_{16}$” : Memory channel operation mode register “03$_{16}$” : Memory address counter “04$_{16}$” : End address register	In-definite	In-definite	O	O

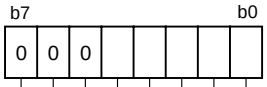
Fig. 82 Structure of Register window 1



Register window 2 (EXBREG2) [address 0035$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
HIGH_WIN [7:0]	–	The accessible register, using this register window, depends on the EXB index register contents as follows: Index value “00$_{16}$” : External I/O configuration register “01$_{16}$” : Transmit/Receive buffer register “02$_{16}$” : Memory channel operation mode register “03$_{16}$” : Memory address counter “04$_{16}$” : End address register	In-definite	In-definite	O	O

Fig. 83 Structure of Register window 2

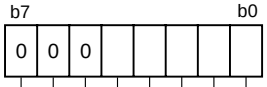


Index = 00_{16} : External I/O configuration register (EXBCFGL) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
EXB_CTR	EXB pin control bit (Pins $P1_0$ to $P1_7$, $P3_0$ to $P3_4$)	0 : Port 1 : EXB function pin	0	–	O	O
INT_CTR [2:0]	$P3_3$/ExINT pin control bit	Selects a signal of $P3_3$/ExINT pin. ON/OFF is programmed by each bit. An output logical sum of $P3_3$/ExINT pins set for ON are performed and it is output as an "L" active signal. b3b2b1 0 0 1 : RxB_RDY (RxBuf ready) output 0 1 0 : TxB_RDY (TxBuf ready) output 1 0 0 : Mch_req (Memory channel request) output Others : Do not set.	0	–	O	O
A1_CTR	$P4_3$/ExA1 pin control bit	0 : Port 1 : A1 input (used to read status)	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

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

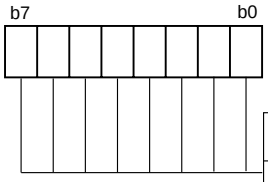


Index = 00_{16} : External I/O configuration register (EXBCFGH) [address 0035_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
DRQ_CTR [1:0]	$P4_0$/ExDREQ/RxD pin control bit	b1b0 0 0 : Port 0 1 : Do not set. 1 0 : ExDREQ function; RxB_RDY (RxBuf ready) output 1 1 : ExDREQ function; Mch_req (Memory channel request) output	0	–	O	O
DAK_CTR [1:0]	$P4_1$/ExDACK/TxD pin control bit	Specifies $P4_1$/ExDACK/TxD pin function. Selects which mode; requiring read or write signal, or not requiring it for use of DMA acknowledge function. b3b2 0 0 : Port 0 1 : Do not set. 1 0 : ExDACK function; DMA acknowledge input (Mode for read and write signals used together) 1 1 : ExDACK function; DMA acknowledge input (Mode for read and write signals not required)	0	–	O	O
TC_CTR	$P4_2$/ExTC/S_{CLK} pin control bit	0 : Port 1 : ExTC (terminal count) input	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 85 Index00[high]; Structure of External I/O configuration register

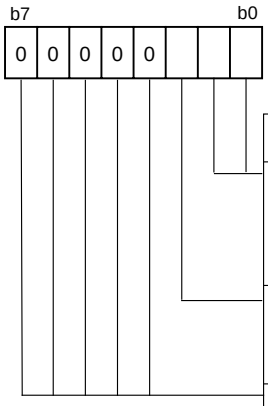


Index =01_{16} : Transmit/Receive buffer register (RXBUF/TXBUF) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
RXBUF/ TXBUF	–	The data received from an external bus is written here at the rise timing of external write signal. The data transmitted to an external bus is written here at the timing of internal CPU write or memory write.	0	–	O	O

The receive buffer register (RXBUF) contents can be read out by reading to this address with the CPU. The data which the CPU has written to this address is stored in the transmit buffer register (TXBUF).
However, do not perform write operation with the CPU to this address if the memory channel direction control bits of memory channel operation mode register is "10_2" (transmit mode) and the memory channel status bits of EXB interrupt source register are "01_2" or "10_2" (memory channel being operating).

Fig. 86 Index01[low]; Structure of Transmit/Receive buffer register

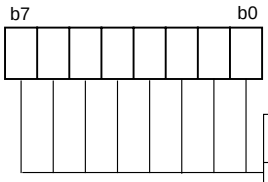


Index =02_{16} : Memory channel operation mode register (MCHMOD) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
MC_DIR [1:0]	Memory channel direction control bit	b1b0 0 0 : Operation disabled 0 1 : Receive mode 1 0 : Transmit mode 1 1 : Do not set.	0	–	O	O
BURST	Burst bit	0 : Cycle mode (each byte transfer according to assertion or negation) 1 : Burst mode (continuous transfer till the terminal count)	0	–	O	O
b7:b3	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

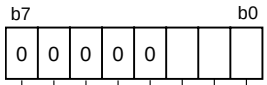
Fig. 87 Index02[low]; Structure of Memory channel operation mode register



Index = 03_{16} : Memory address counter (MEMADL) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
IM_A [7:0]	–	Register to set the low-order address of memory channel operation beginning. This contents are increased each time one memory access ends.	0	–	O	O

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

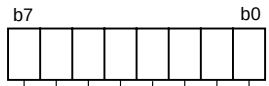


Index = 03_{16} : Memory address counter (MEMADH) [address 0035_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
IM_A [10:8]	–	Register to set the high-order address of memory channel operation start. This contents are increased each time one memory access ends.	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 89 Index03[high]; Structure of Memory address counter

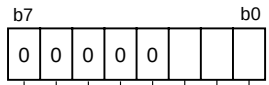


Index = 04_{16} : End address register (ENDADL) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
END_A [7:0]	–	Register to set the low-order address of memory channel operation end.	0	–	O	O

–: State remaining

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



Index = 04_{16} : End address register (ENDADH) [address 0035_{16}]

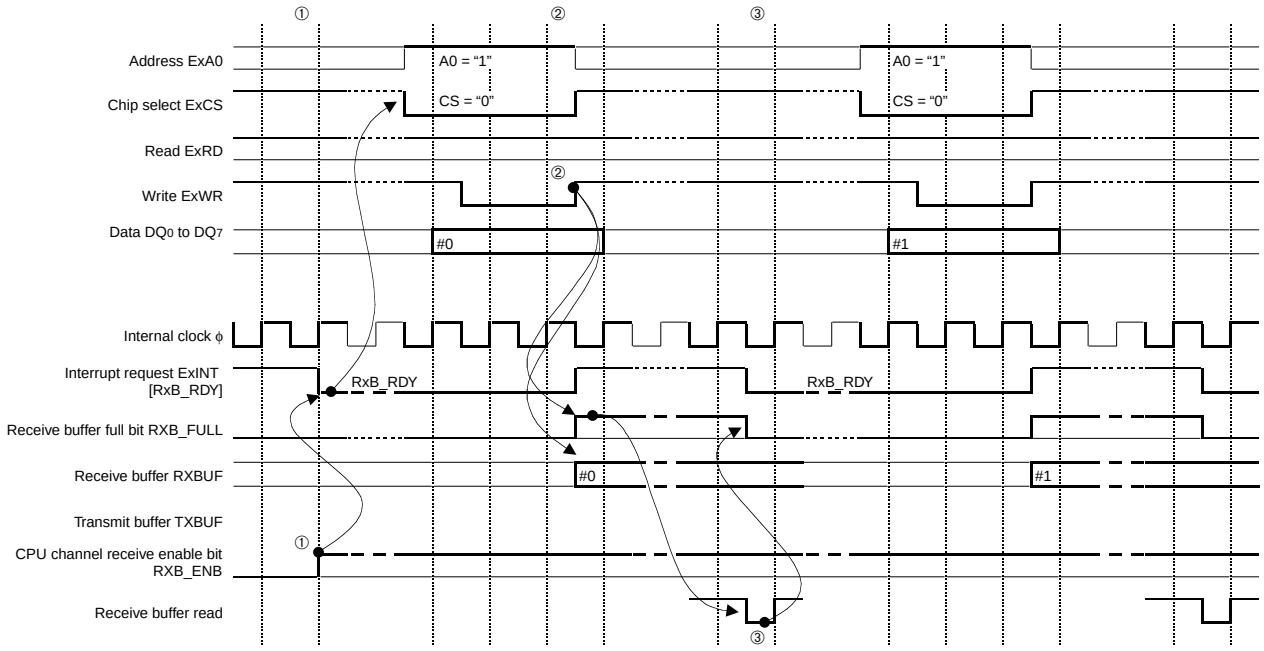
Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
END_A [10:8]	–	Register to set the high-order address of memory channel operation end.	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 91 Index04[high]; Structure of End address register

EXB Operation Timing Diagram
(1) CPU Channel Receiving Operation

CPU channel receiving operation is shown bellow.



<Initial setting>
External I/O configuration register INT_CTR[3:1] (P33/ExINT pin control) = 001₂ (RxB_RDY interrupt)

<Operation start>
EXB interrupt source enable register RXB_ENB (CPU channel receive enable) = "1" (Receive buffer full interrupt enabled)

① Writing the command for enabling operation makes RXB_RDY assertion and the P33/ExINT pin goes to "L".
If the CPU channel receive enable bit (RXB_ENB) is "0", both the receive buffer full bit (RXB_FULL) and the receive buffer ready signal (RxB_RDY) to an external are inactive.

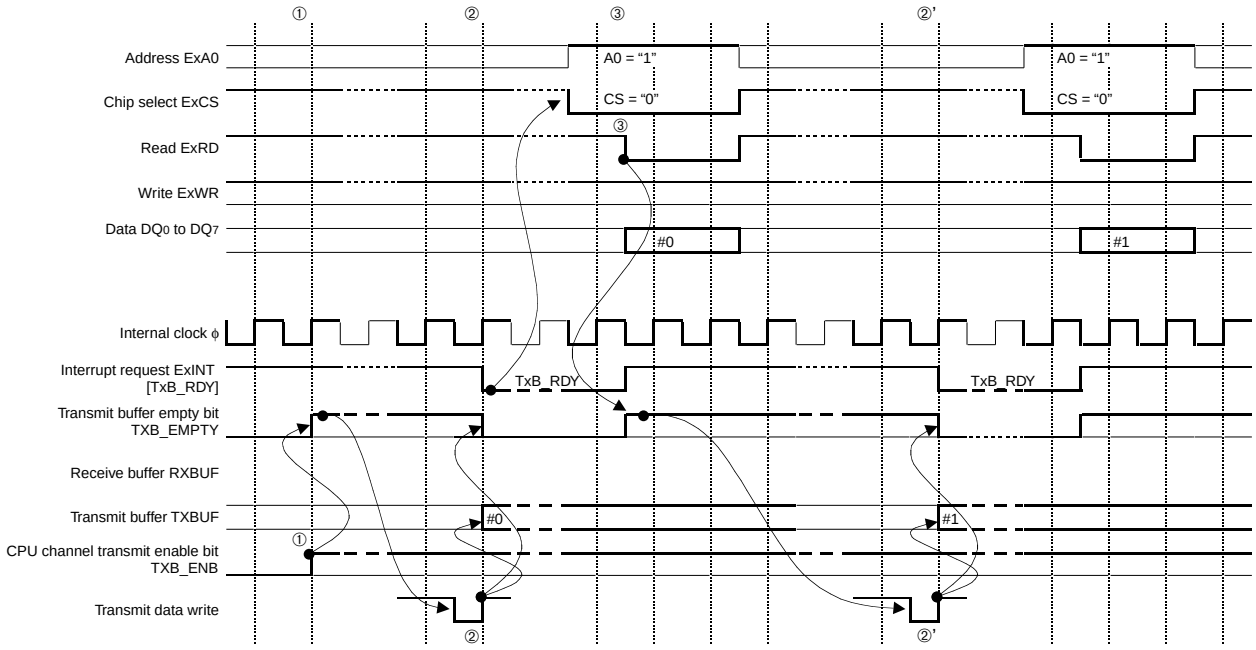
② When a write operation is performed from an external MCU bus in the condition of ExCS = "L" and WxA0 = "H", it will result in as follows:
- The data is written into the receive buffer (RXBUF)
- Negation of the receive buffer ready signal (RxB_RDY) to an external is made
- The RXB_FULL interrupt is generated.

③ When the CPU reads out the receive buffer (RXBUF) with an interrupt processing program, the receive buffer full bit (RXB_FULL) is cleared to "0".

Fig. 92 CPU channel receiving operation

(2) CPU Channel Transmitting Operation

CPU channel transmitting operation is shown bellow.



<Initial setting>
External I/O configuration register INT_CTR[3:1] (P3$_3$/ExINT pin control) = 010$_2$ (TxB_RDY interrupt)

<Operation start>
EXB interrupt source enable register TXB_ENB (CPU channel transmit enable) = "1" (Transmit buffer empty interrupt enabled)

① Writing the command for enabling operation generates TXB_EMPTY interrupt.
If the CPU channel transmit enable bit (TXB_ENB) is "0", both the transmit buffer empty bit (TXB_EMPTY) and the transmit buffer ready signal (TxB_RDY) to an external are inactive.

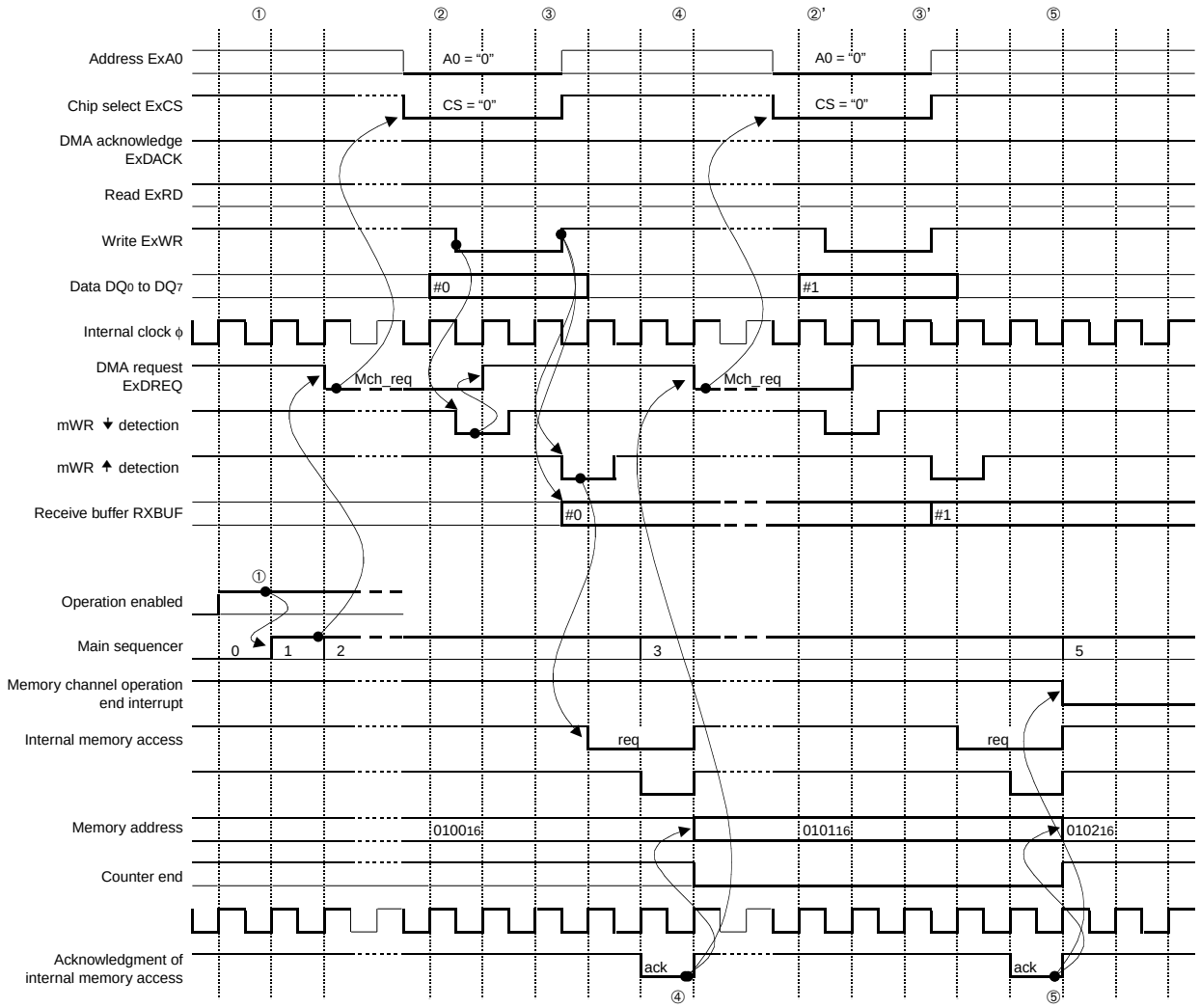
② When the CPU writes the data into the transmit buffer (TXBUF) with an interrupt processing program, the transmit buffer empty bit (TXB_EMPTY) is cleared to "0" and assertion of the transmit buffer ready signal (TxB_RDY) to an external is made.

③ When a read operation is performed from an external MCU bus in the condition of ExCS = "L" and ExA0 = "H", it will result in as follows:
- The contents of the transmit buffer (TXBUF) is read out
- The transmit buffer empty bit (TXB_EMPTY) is set to "1"
- Negation of the transmit buffer ready signal (TxB_RDY) to an external is made.

Fig. 93 CPU channel tranmitting operation

(3) Memory Channel Receiving Operation (1)- Cycle Mode

Memory channel receiving operation (1) is shown bellow.



<Initial setting>

External I/O configuration register	Set as necessary.
Memory channel operation mode register	MC_DIR[1:0] (Memory channel direction control) = 01_2 (Receive mode) Burst (burst) = "0" (Cycle mode)
Memory address counter	(Example) 0100_{16}
End address register	(Example) 0101_{16}

<Operation start command>

EXB interrupt source enable register	MC_ENB (Memory channel operation enable) = "1" (Operation start)

① In the memory channel receive mode when the command for enabling operation is written, operation starts (main sequencer starts) and assertion of the memory channel request which synchronized with a rise of ϕ is made.

② When the external MCU bus is in the condition of ExCS = "L" and ExA0 = "L" or a fall of ExWR is detected in the condition of ExDACK = "L", negation of the memory channel request which synchronized with a rise of ϕ is made.

③ When a rise of ExWR is detected, an internal memory access sequence which synchronized with a rise of ϕ is activated and a data is written in the internal memory within two clocks at a minimum.

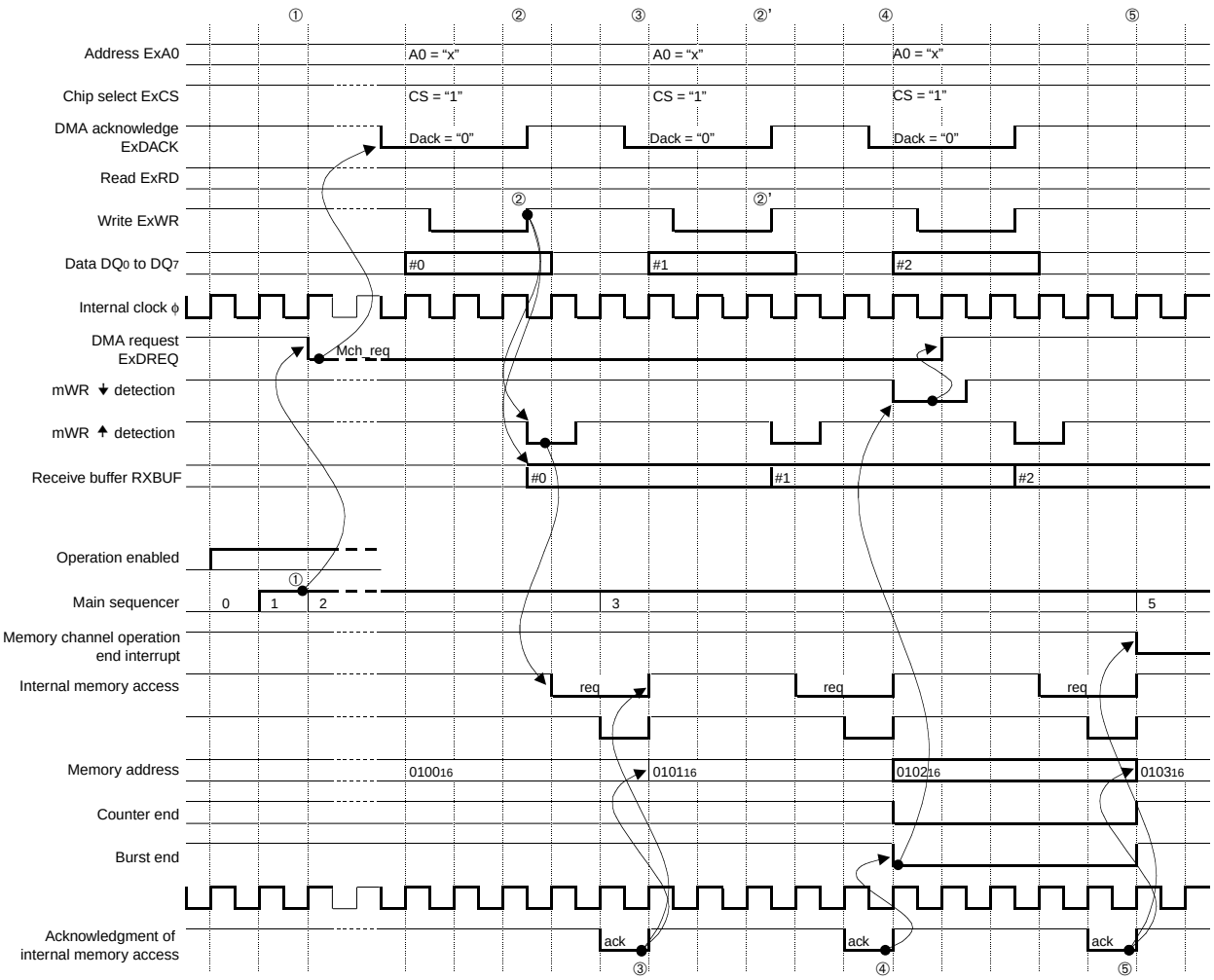
④ The memory address counter is increased simultaneously at write completion and assertion of the next memory channel request is made.

⑤ When the write operation to the end address has been completed, the memory address counter is increased, but assertion of the next memory channel request is not made and the memory channel operation end interrupt is generated.

Fig. 94 Memory channel receiving operation (1)

(4) Memory Channel Receiving Operation (2)- Burst Mode

Memory channel receiving operation (2) is shown bellow.



<Initial setting>

External I/O configuration register	Set as necessary.
Memory channel operation mode register	MC_DIR[1:0] (Memory channel direction control) = 01₂ (Receive mode) Burst (burst) = "1" (Burst mode)
Memory address counter	(Example) 0100₁₆
End address register	(Example) 0102₁₆

<Operation start command>

EXB interrupt source enable register	MC_ENB (Memory channel operation enable) = "1" (Operation start)

① In the memory channel receive mode when the command for enabling operation is written, assertion of the memory channel request which synchronized with a rise of ϕ is made.

② When a rise of ExWR is detected, an internal memory access sequence which synchronized with a rise of ϕ is activated and a data is written in the internal memory within two clocks at a minimum.

③ The memory address counter is increased simultaneously at the former data write completion.

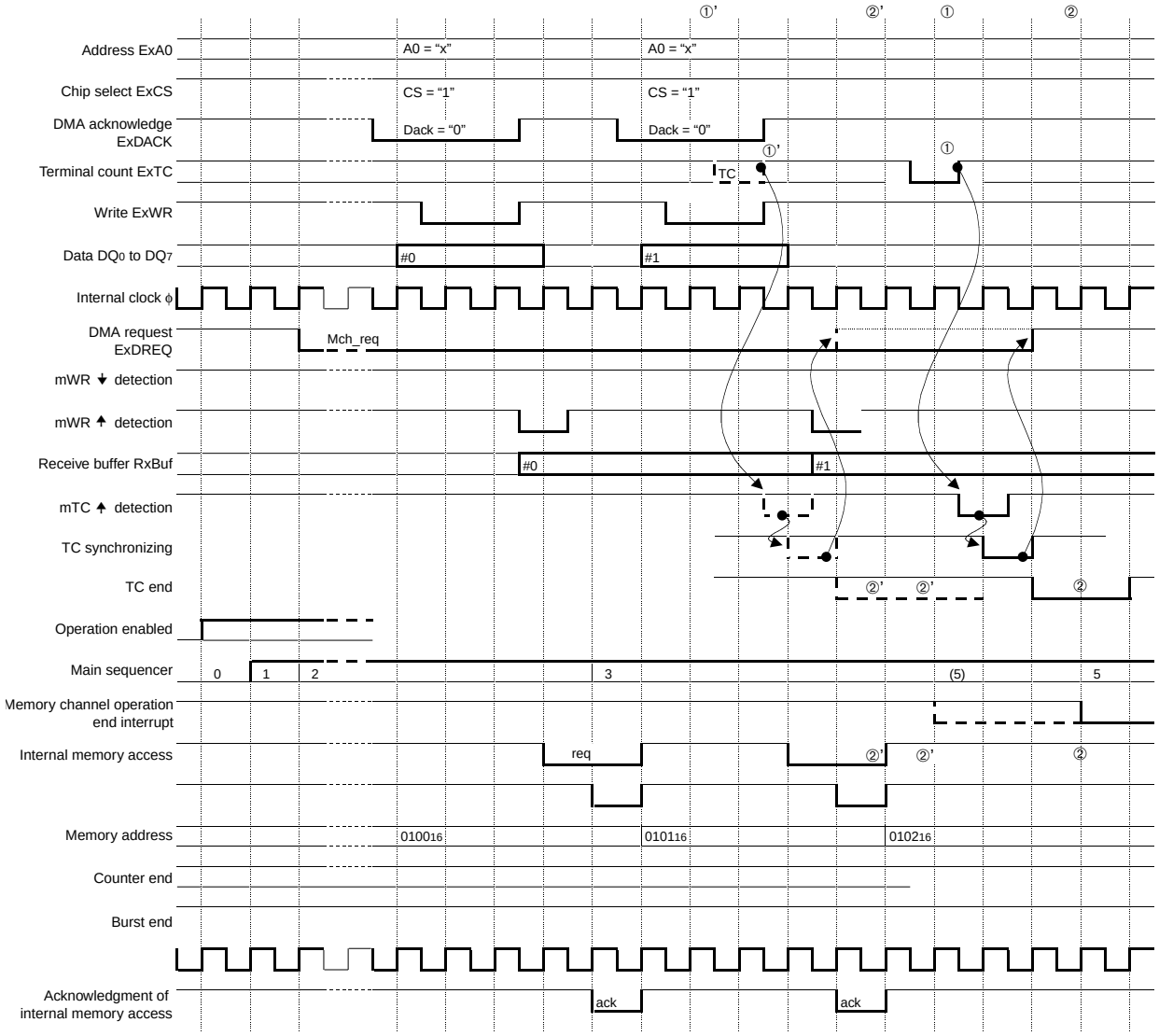
④ When the memory address counter reaches the end address, the detection circuit of external write signal (ExWR) operation is enabled and negation of the memory channel request which synchronized with the following ϕ is made.

⑤ When the write operation to the end address has been completed, the memory address counter is increased and the memory channel operation end interrupt is generated.

Fig. 95 Memory channel receiving operation (2)

(5) Memory Channel Receiving Operation (3)- Burst Mode (Terminal Count)

Memory channel receiving operation (3) is shown bellow.



<Initial setting>

External I/O configuration register	Set as necessary.
Memory channel operation mode register	MC_DIR[1:0] (Memory channel direction control) = 01_2 (Receive mode)
	Burst (burst) = "1" (Burst mode)
Memory address counter	(Example) 0100_{16}
End address register	(Example) 0107_{16}

<Operation start command>

EXB interrupt source enable register	MC_ENB (Memory channel operation enable) = "1" (Operation start)

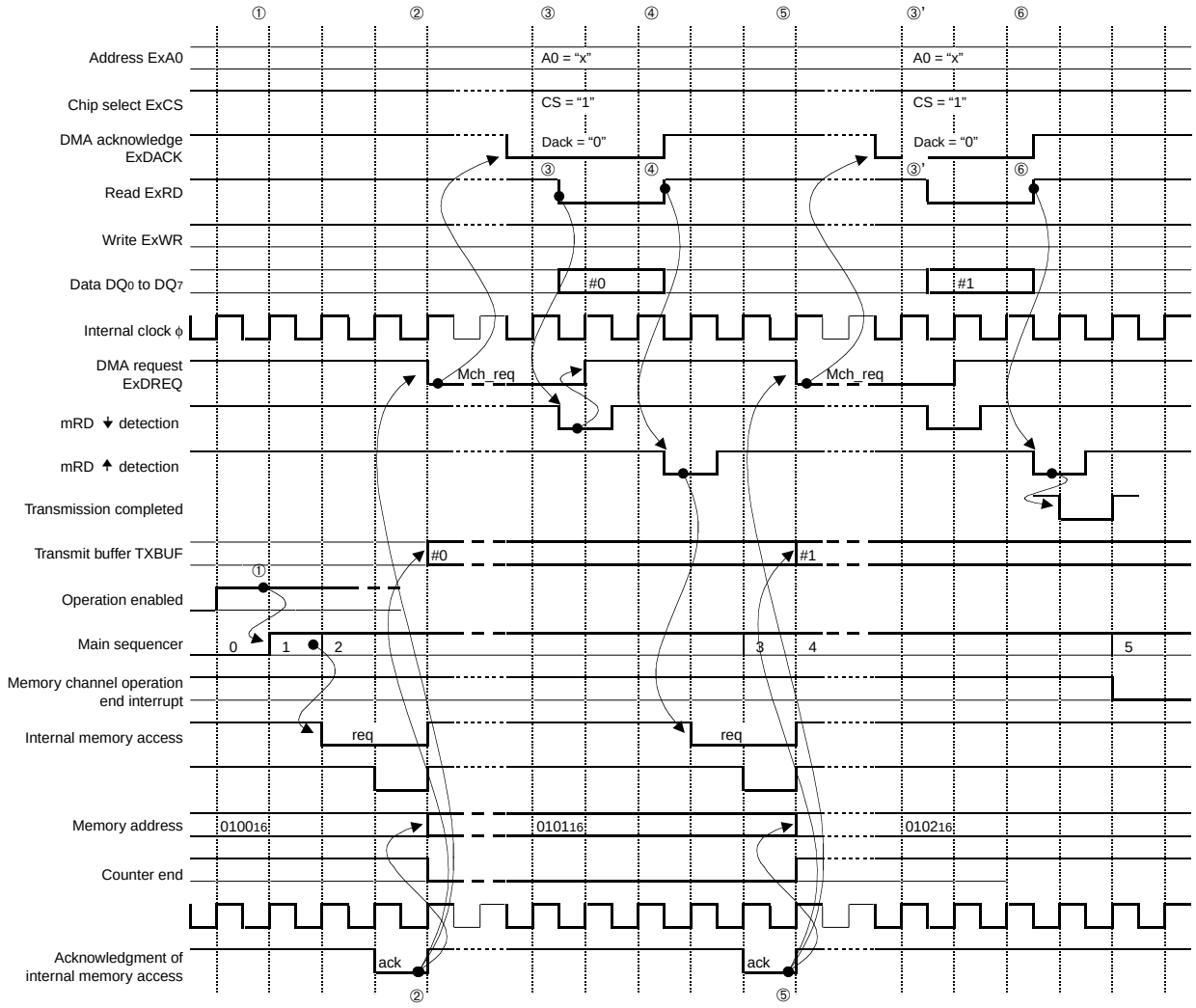
① When a rise of TC is detected, negation of the memory channel request which synchronized with a rise of ϕ is made.

② When the write operation to the end address has been completed, the memory channel operation end interrupt is generated.

Fig. 96 Memory channel receiving operation (3)

(6) Memory Channel Transmitting Operation
(1)-Cycle Mode

Memory channel transmitting operation (1) is shown bellow.



<Initial setting>

External I/O configuration register	Set as necessary.
Memory channel operation mode register	MC_DIR[1:0] (Memory channel direction control) = 10_2 (Transmit mode) Burst (burst) = "0" (Cycle mode)
Memory address counter	(Example) 0100_{16}
End address register	(Example) 0101_{16}

<Operation start command>

EXB interrupt source enable register	MC_ENB (Memory channel operation enable) = "1" (Operation start)

① In the memory channel transmit mode when the command for enabling operation is written, operation starts (main sequencer starts) and an internal memory access sequence which synchronized with a rise of ϕ is activated.

② A data is read out from the internal memory within two clocks at a minimum and this data is stored in the transmit buffer (TXBUF). The memory address counter is simultaneously increased and assertion of the memory channel request is made.

③ When the external MCU bus is in the condition of ExCS = "L" and ExA0 = "L" or a fall of ExRD is detected in the condition of ExDACK = "L", negation of the memory channel request which synchronized with a rise of ϕ is made.

④ When a rise of ExRD is detected, an internal memory access sequence which synchronized with a rise of ϕ is activated.

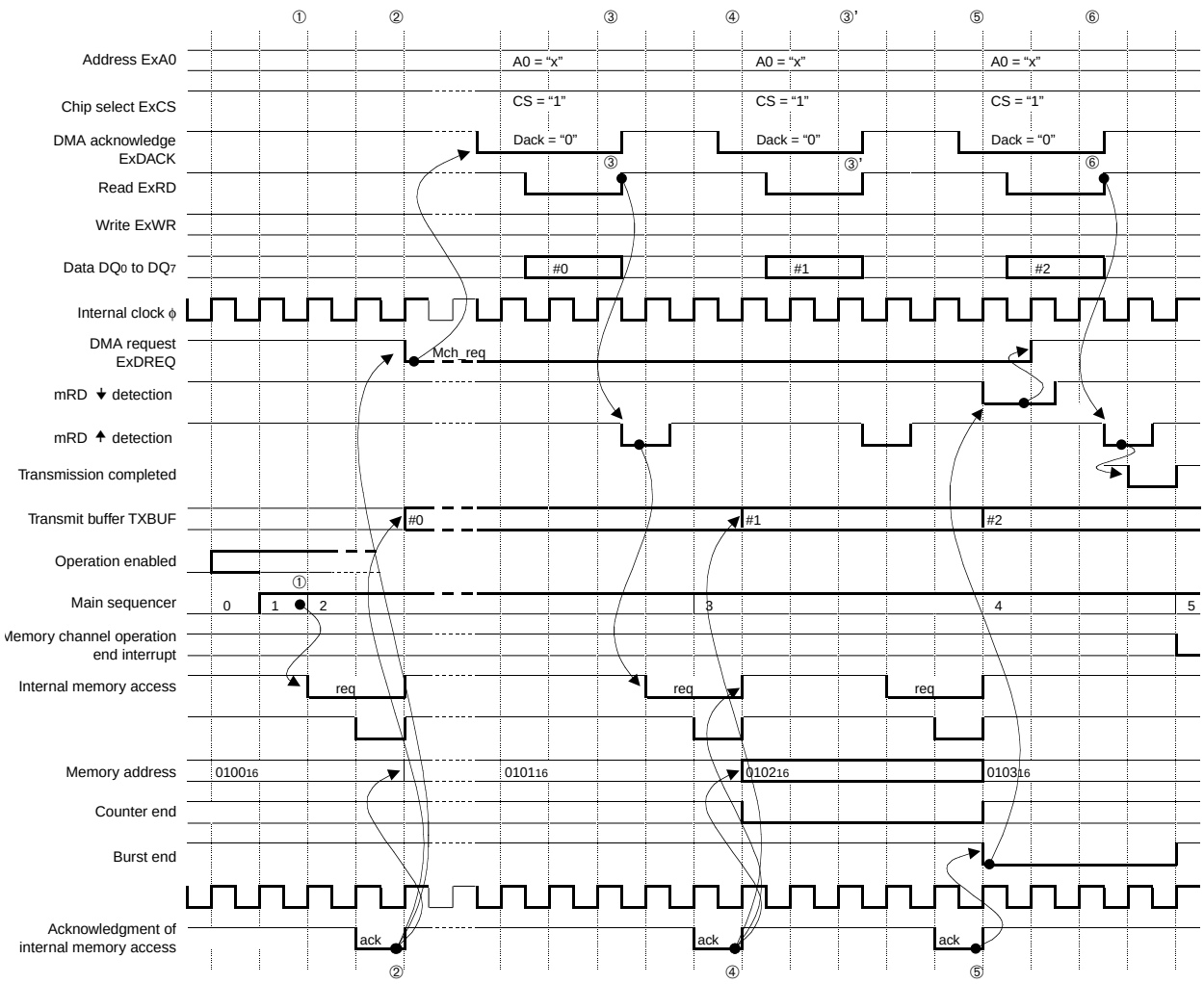
⑤ A data is read out from the internal memory within two clocks at a minimum and this data is stored in the transmit buffer (TXBUF). The memory address counter is simultaneously increased and assertion of the memory channel request is made.
When the read operation from the end address has been completed, the transition to the status to wait the memory channel operation end occurs.

⑥ When a rise of ExRD is detected, the memory channel operation sequence ends and the memory channel operation end interrupt is generated.

Fig. 97 Memory channel tranmitting operation (1)

(7) Memory Channel Transmitting Operation (2)-Burst Mode

Memory channel transmitting operation (2) is shown bellow.



<Initial setting>

External I/O configuration register	Set as necessary.
Memory channel operation mode register	MC_DIR[1:0] (Memory channel direction control) = 10_2 (Transmit mode)
	Burst (burst) = "1" (Burst mode)
Memory address counter	(Example) 0100_{16}
End address register	(Example) 0102_{16}

<Operation start command>

EXB interrupt source enable register	MC_ENB (Memory channel operation enable) = "1" (Operation start)

① In the memory channel transmit mode when the command for enabling operation is written, an internal memory access sequence which synchronized with a rise of ϕ is activated.

② A data is read out from the internal memory within two clocks at a minimum and this data is stored in the transmit buffer (TXBUF). The memory address counter is simultaneously increased and assertion of the memory channel request is made.

③ When a rise of ExRD is detected, an internal memory access sequence which synchronized with a rise of ϕ is activated.

④ A data is read out from the internal memory within two clocks at a minimum and this data is stored in the transmit buffer (TXBUF). The memory address counter is simultaneously increased.

⑤ When the read operation from the end address has been completed, the detection circuit of external read signal (ExRD) operation is enabled and negation of the memory channel request which synchronized with the following ϕ is made.

⑥ When a rise of ExRD is detected, the memory channel operation sequence ends and the memory channel operation end interrupt is generated.

Fig. 98 Memory channel tranmitting operation (2)

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 ϕ. The USB transfer rate is about 1.5 Mbytes/second. Access to the multichannel RAM is performed at every about 5.3 clocks in ϕ = 8 MHz, or at every about 4 clocks in ϕ = 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 ϕ.

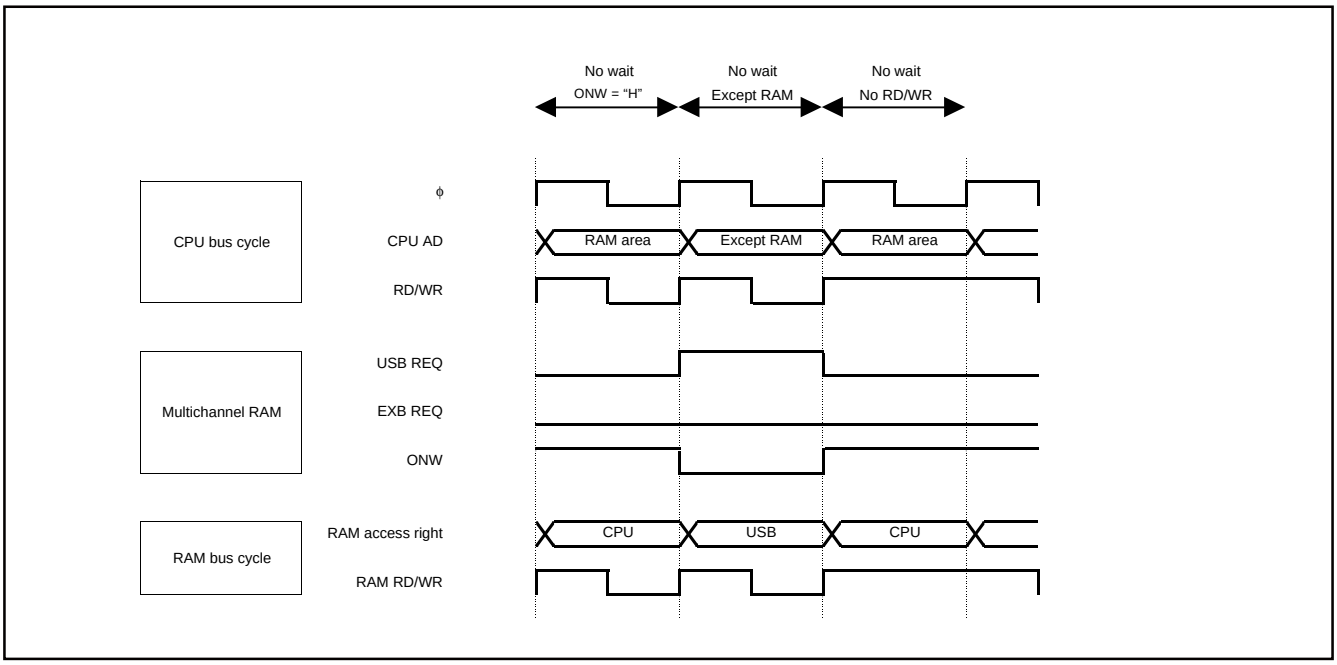


Fig. 99 Multichannel RAM timing diagram (no wait)

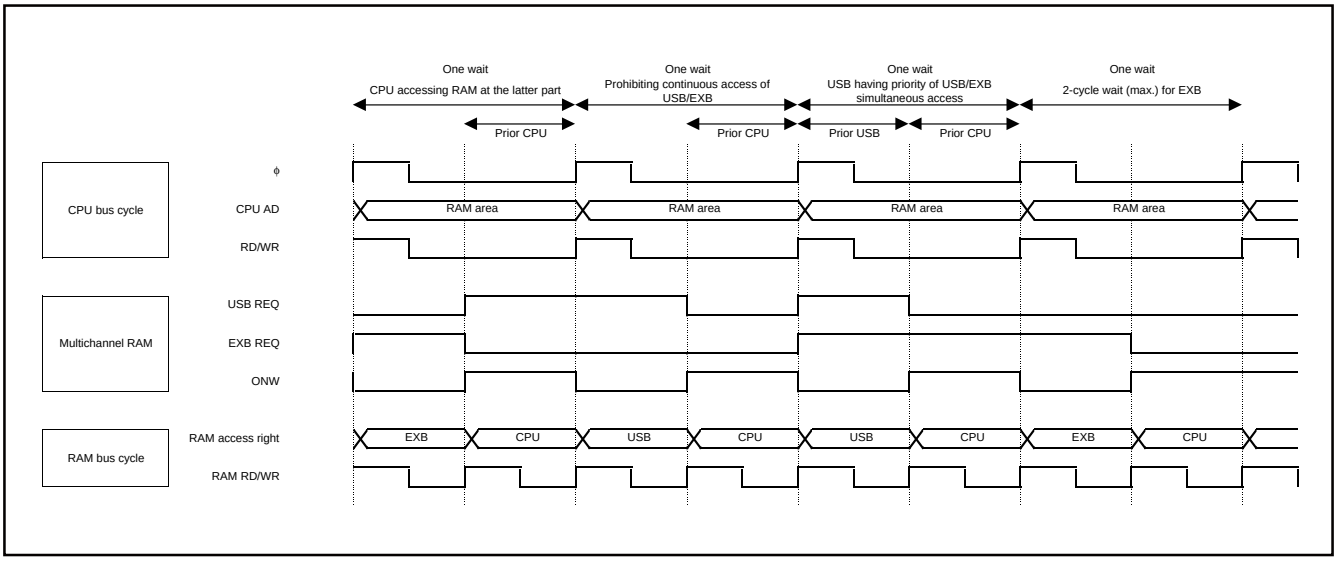


Fig. 100 Multichannel RAM timing diagram (one wait)

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.

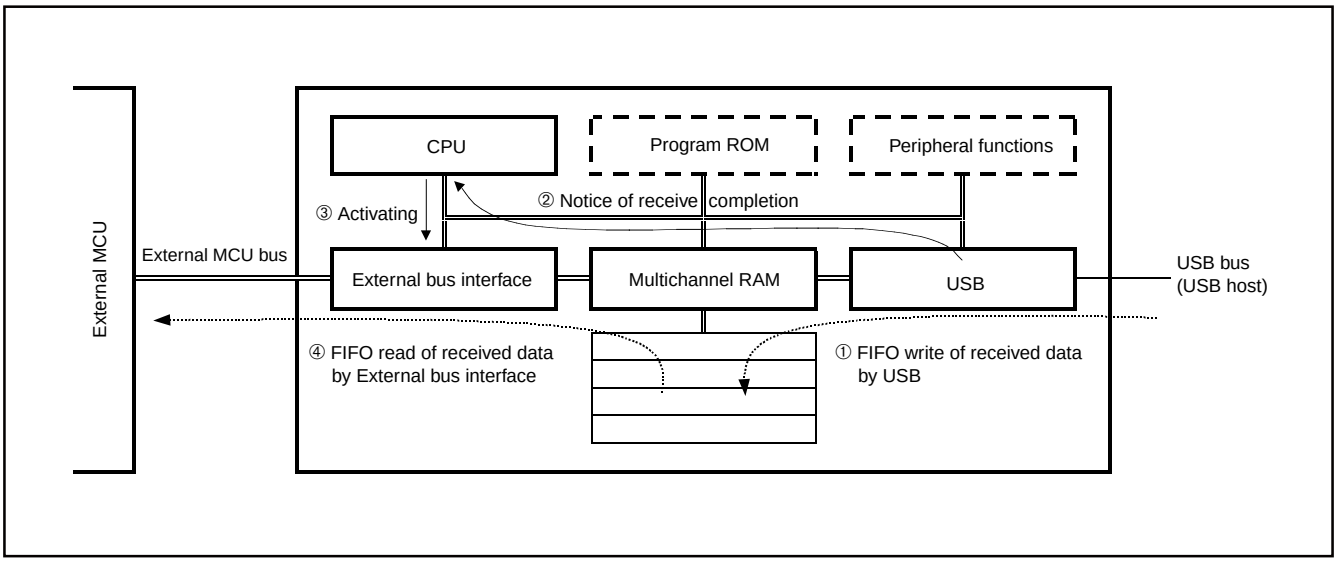


Fig. 101 Multichannel RAM operation example

A/D CONVERTER

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

[AD Conversion Register 1, 2 (AD1, AD2)] 0037$_{16}$, 0038$_{16}$

The AD 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 AD 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 AD conversion registers 1, 2 after A/D conversion is completed (in Figure 103).

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

[AD Control Register (ADCON)] 0036$_{16}$

The AD 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 V_{REF} and AV_{SS} into 1024, and that outputs the comparison voltage.

The A/D converter successively compares the comparison voltage V_{ref} in each mode, dividing the V_{REF} voltage (see below), with the input voltage.

• 10-bit reading

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

• 8-bit reading

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

Channel Selector

The channel selector selects one of the input ports P1$_7$/AN$_7$–P1$_0$/AN$_0$.

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 AD 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.

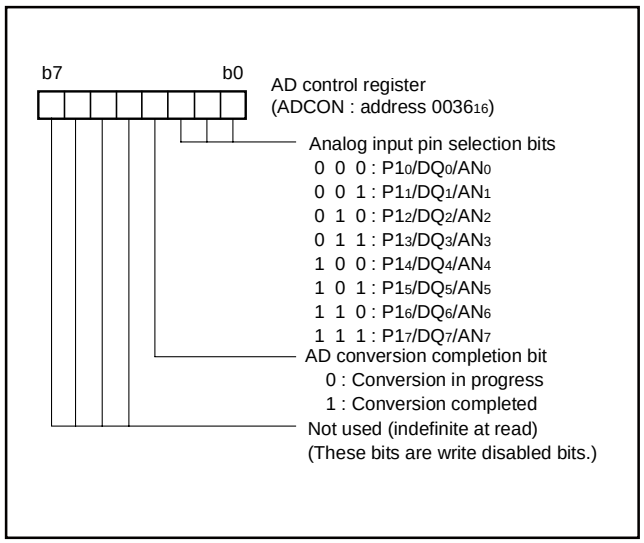


Fig. 102 Structure of AD control register

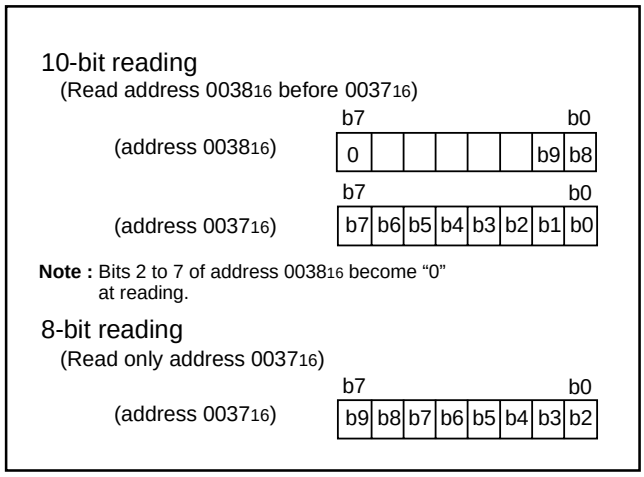


Fig. 103 10-bit A/D mode reading

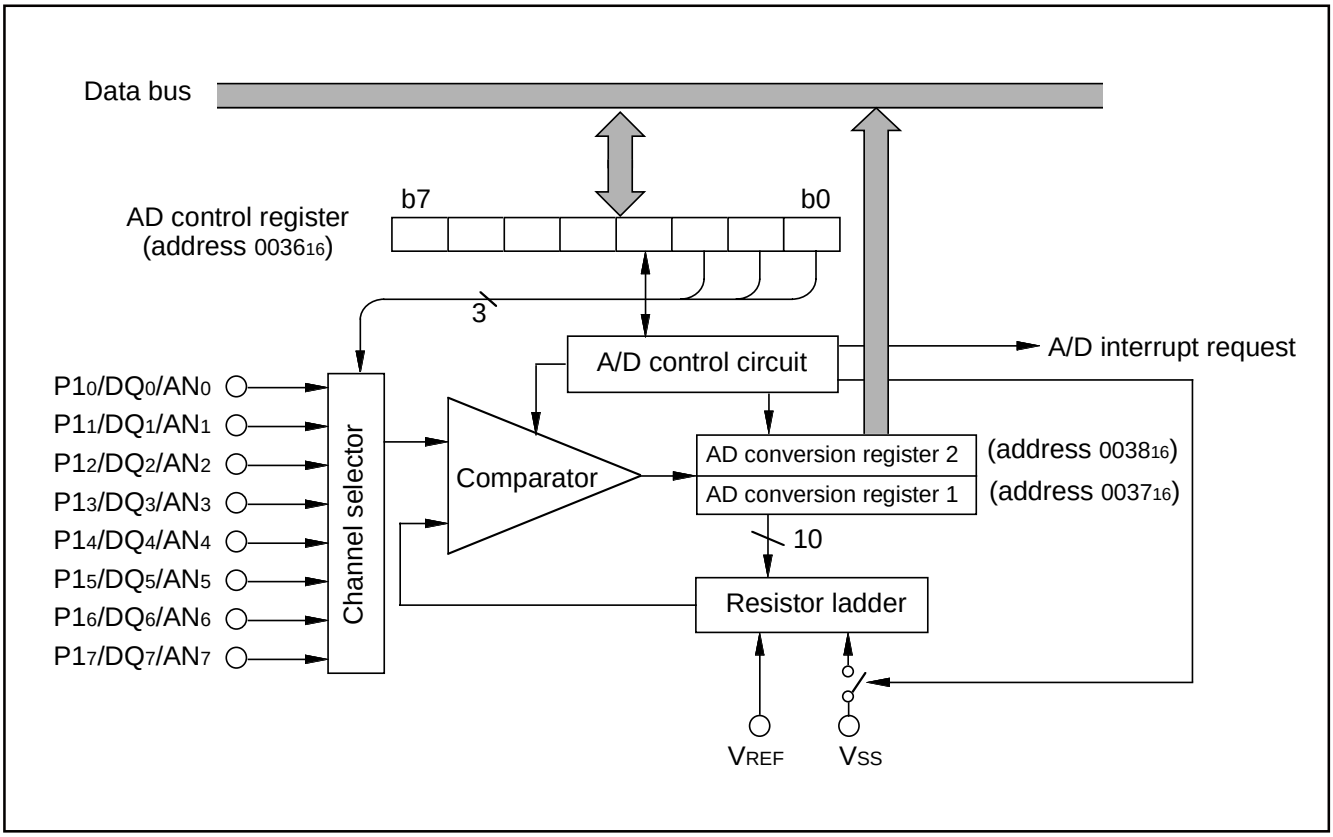


Fig. 104 A/D converter block diagram

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_{16}) 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_{16}) 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_{16}) may be started before an underflow. When the watchdog timer control register (address 0039_{16}) 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_{16}), each watchdog timer H and L is set to "FF_{16}."

●Watchdog timer H count source selection bit operation

Bit 7 of the watchdog timer control register (address 0039_{16}) 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_{16}) 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.

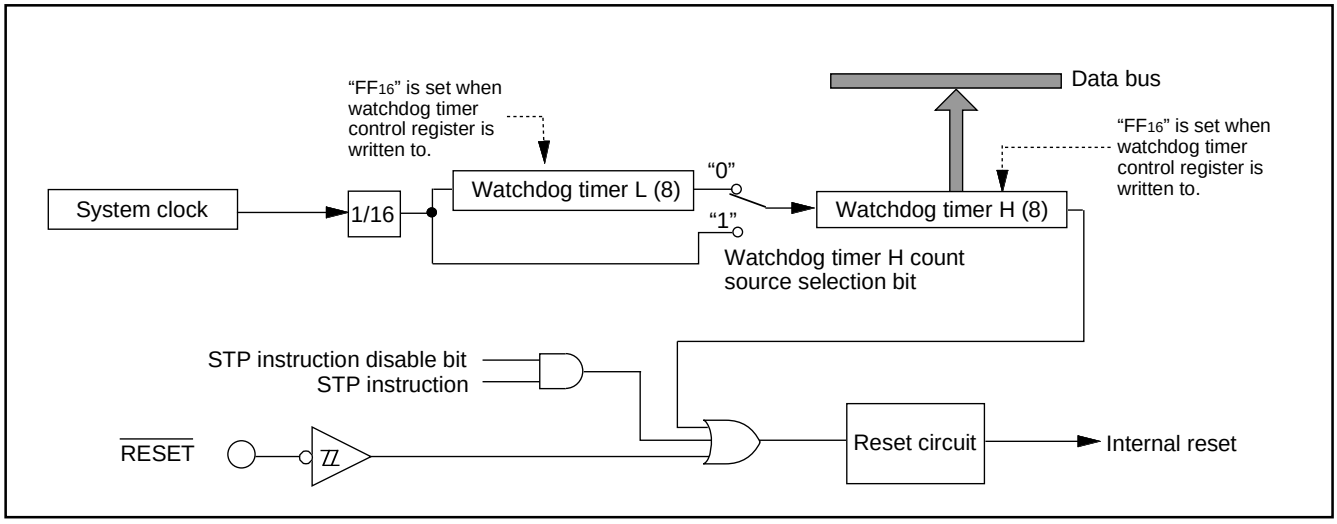


Fig. 105 Block diagram of Watchdog timer

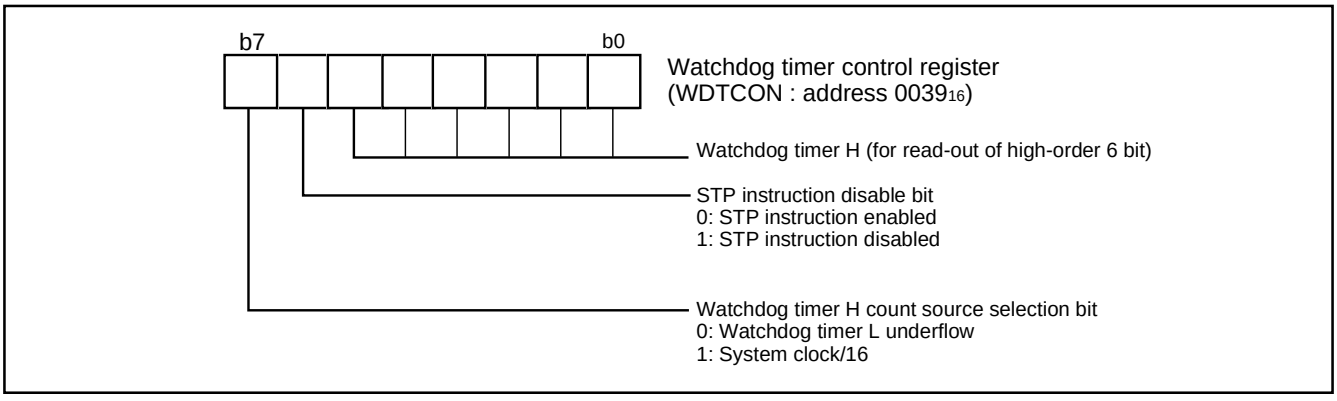


Fig. 106 Structure of Watchdog timer control register

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 for L version, and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address contained in address FFFD16 (high-order byte) and address FFFC16 (low-order byte). Make sure that the reset input voltage is under 0.6 V for V_{CC} of 3.0 V (L version).

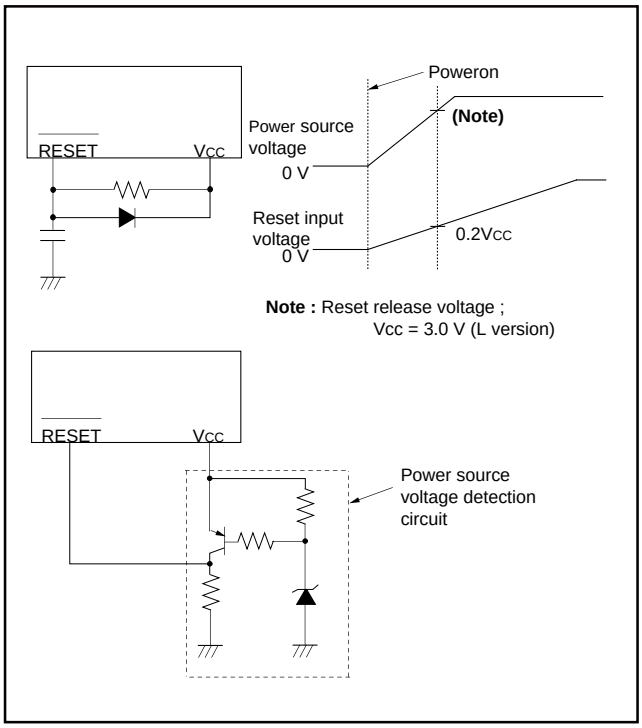


Fig. 107 Example of reset circuit

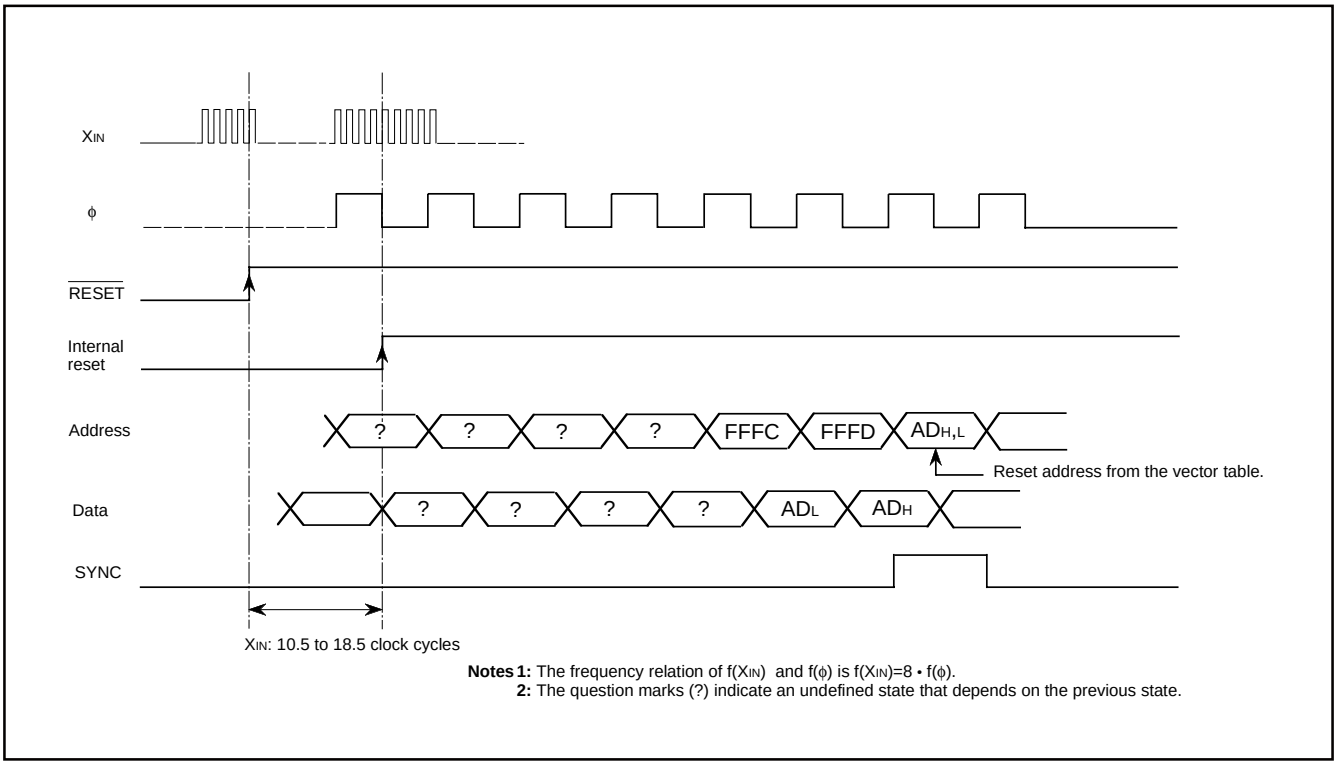


Fig. 108 Reset sequence

PLL CIRCUIT (FREQUENCY SYNTHESIZER)

The PLL circuit generates f_{VCO} (PLL output clock), which is required for f_{USB} (USB clock) and f_{SYN} (f_{USB} division clock), from $f(X_{IN})$ (external input reference clock). Figure 109 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 f_{VCO} to the USB block, wait for the oscillation stable time (1ms or less) of PLL before selecting f_{VCO} with the USB clock selection bit.

According to the setting of the USB clock division ratio selection bit, the division clock of f_{USB} is supplied to f_{SYN}. 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 f_{USB} is 48MHz is recommended).

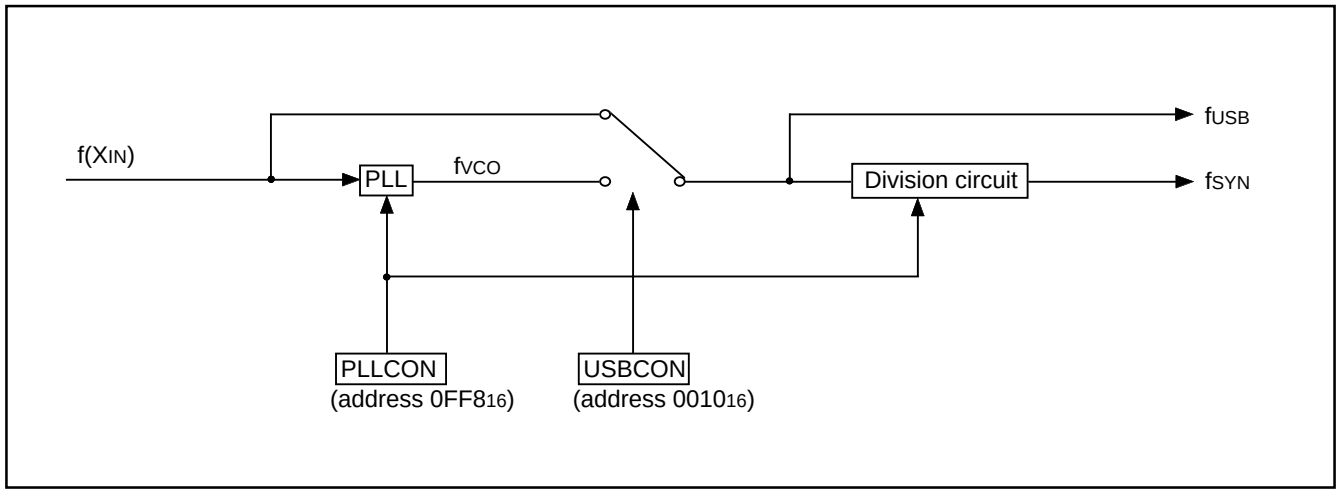


Fig. 109 Block diagram of PLL circuit

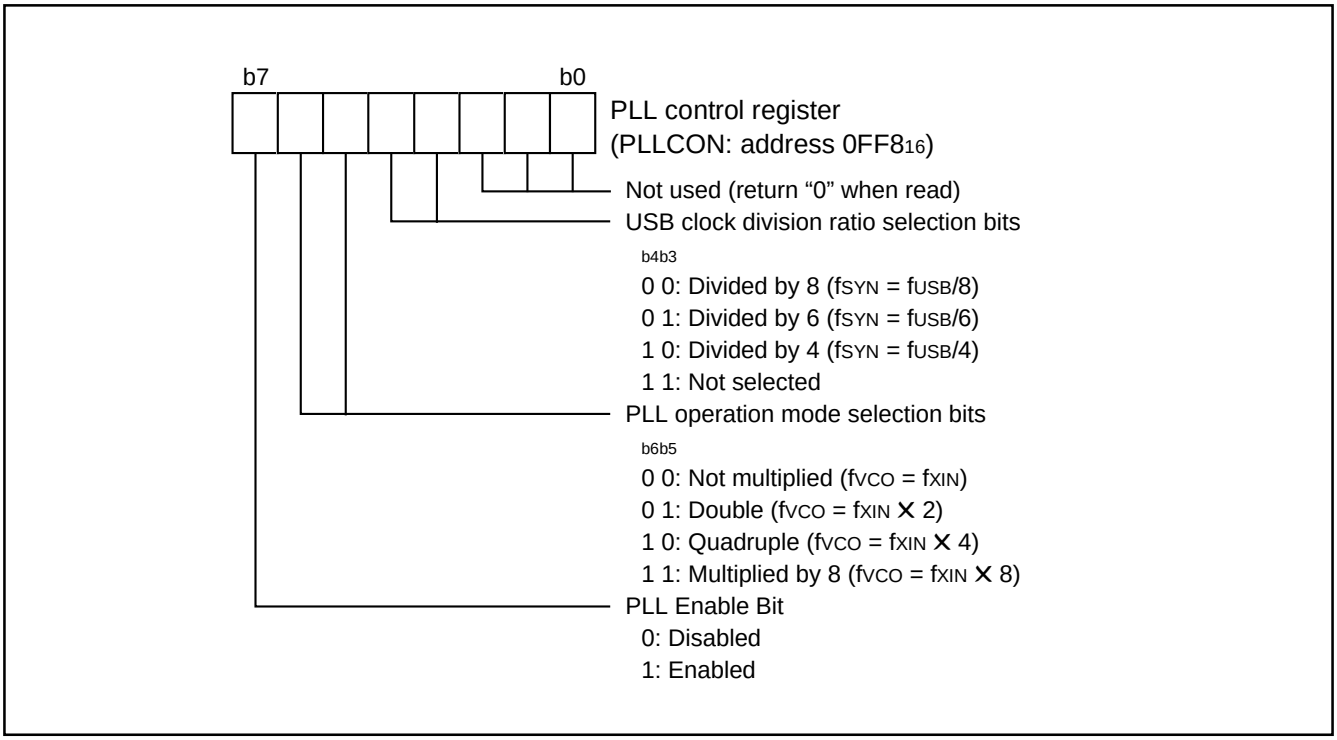


Fig. 110 Structure of PLL control register

CLOCK GENERATING CIRCUIT

An oscillation circuit can be formed by connecting a resonator between XIN and XOUT. Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between XIN and XOUT since a feed-back resistor exists on-chip. (An external feed-back resistor may be needed depending on conditions.)

Frequency Control

Either fSYN or f(XIN) can be selected as an internal system clock. Furthermore, the frequency of internal clock ϕ can be selected by the system clock division ratio selection bit.

(1) fSYN clock

fSYN clock is generated by the PLL circuit. f(XIN) or fVCO 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(XIN) clock

The frequency applied to the XIN pin is used as an internal system clock frequency.

Oscillation Control

(1) Stop mode

If the STP instruction is executed, the internal clock ϕ stops at an "H" level, and the XIN oscillator stops. When the oscillation stabilizing time set after STP instruction released bit is "0," the prescaler 12 is set to "FF16" and timer 1 is set to "0116." 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. XIN 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 ϕ remains at "H" until timer 1 underflows. The internal clock ϕ 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

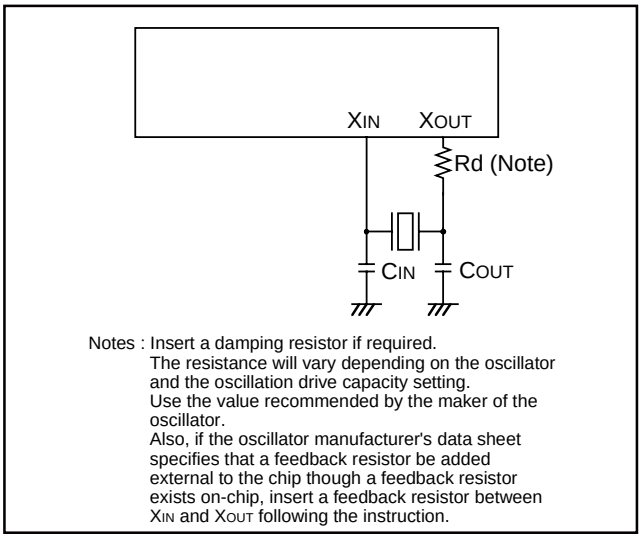
If the WIT instruction is executed, the internal clock ϕ stops at an "H" level, but the oscillator does not stop. The internal clock ϕ restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

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 XIN 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.



Notes : Insert a damping resistor if required.
The resistance will vary depending on the oscillator and the oscillation drive capacity setting.
Use the value recommended by the maker of the oscillator.
Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip though a feedback resistor exists on-chip, insert a feedback resistor between XIN and XOUT following the instruction.

Fig. 111 Ceramic resonator or quartz-crystal oscilltor circuit

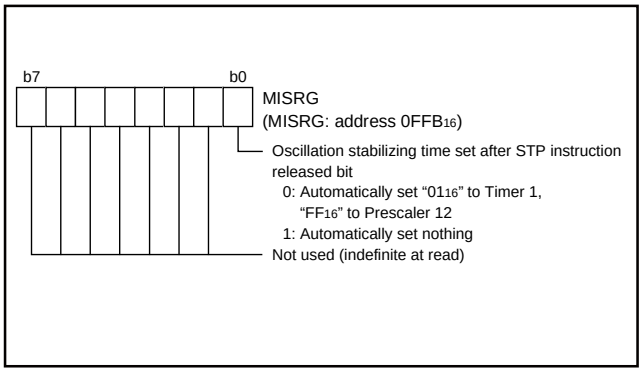


Fig. 113 Structure of MISRG

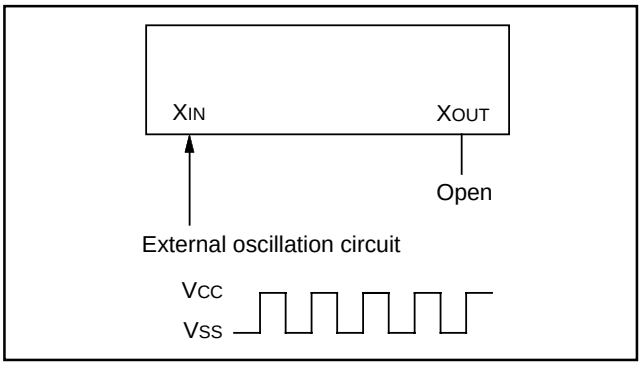


Fig. 112 External clock input circuit

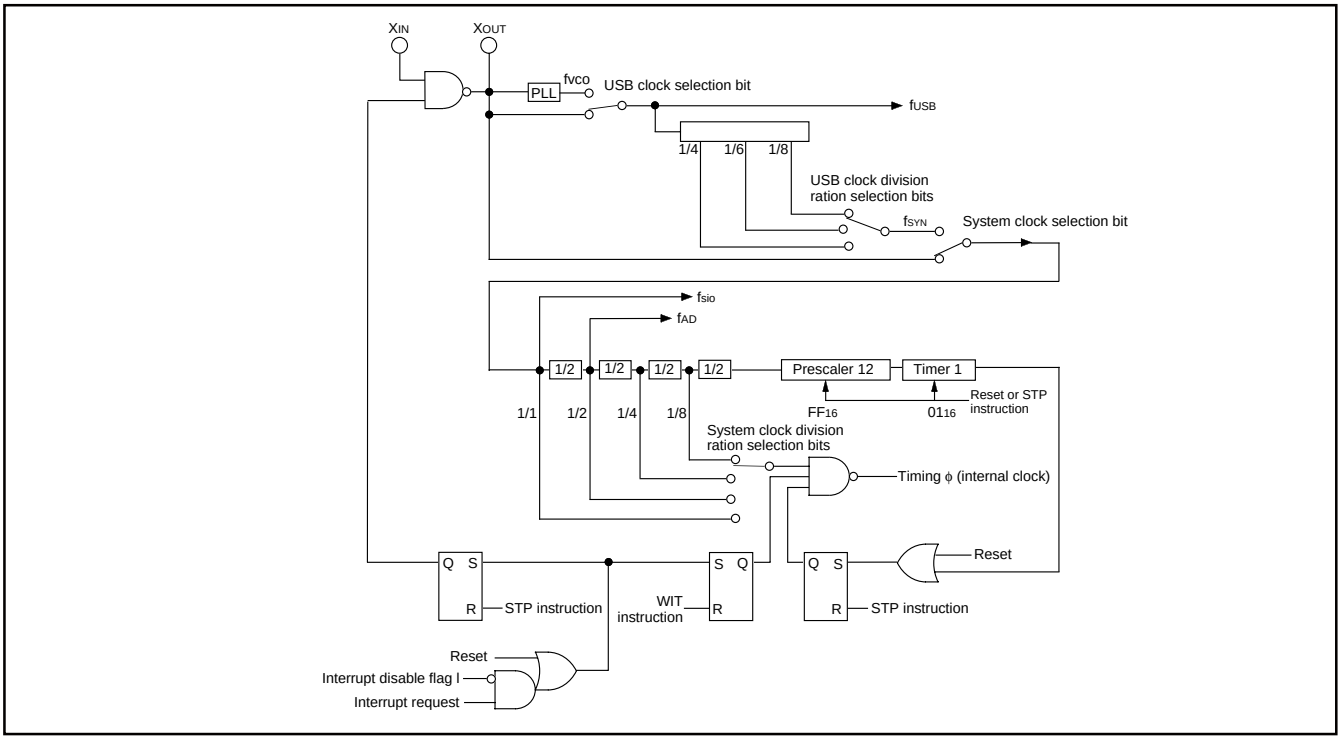
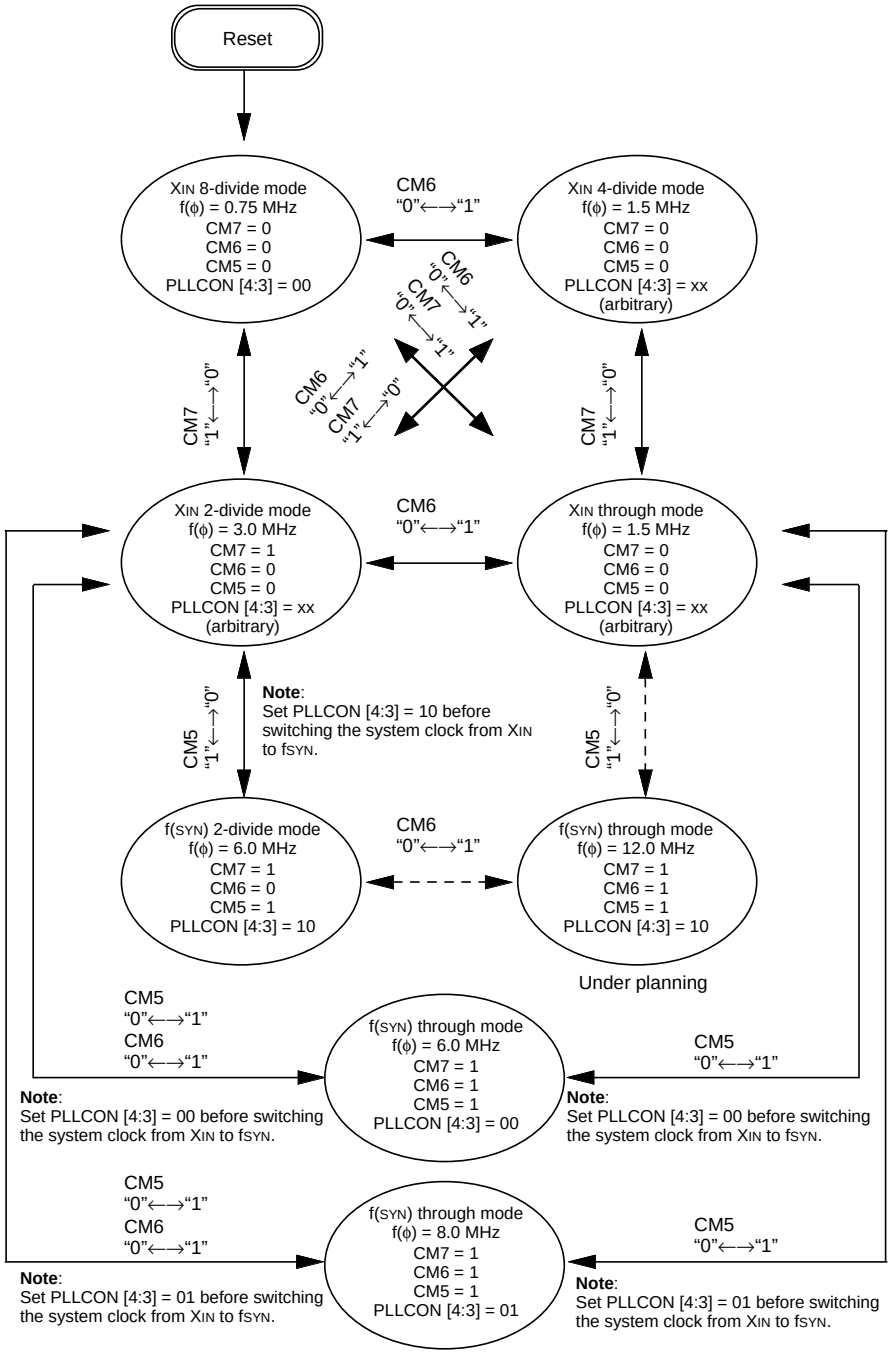


Fig. 114 System clock generating circuit block diagram (single-chip mode)



Notes 1 : Switch the mode by the allows shown between the mode blocks. (Do not switch between the modes directly without an allow.)
2 : Set the USB clock (fUSB) to 48 MHz when switching the system clock to fSYN.
3 : Do not change a division ratio of USB clock when using fSYN as the system clock.
4 : See section "PLL CIRCUIT" in details for enabling/disabling PLL operation and usage notes of fSYN.
5 : Set the system clock to XIN when entering STOP mode.
6 : In all modes, switching to WAIT mode is possible. When it is released, the MCU returns to the original mode. In WAIT mode the timers can operate.

Remarks : This diagram assumes that the 6 MHz signals are applied to XIN pin.

Fig. 115 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 116 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 (L version) (Program and erase in 4.00 to 5.25 V of Vcc.)
		3.00 – 4.00 V (L version) (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 (**Note 1**) •Standard serial I/O mode: Manipulated using an external programmer (**Note 1**)
Erase block division	User ROM area	1 block (32 Kbytes)
	Boot ROM area	1 block (4 Kbytes) (**Note 2**)
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 38K0 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 116 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 116 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.

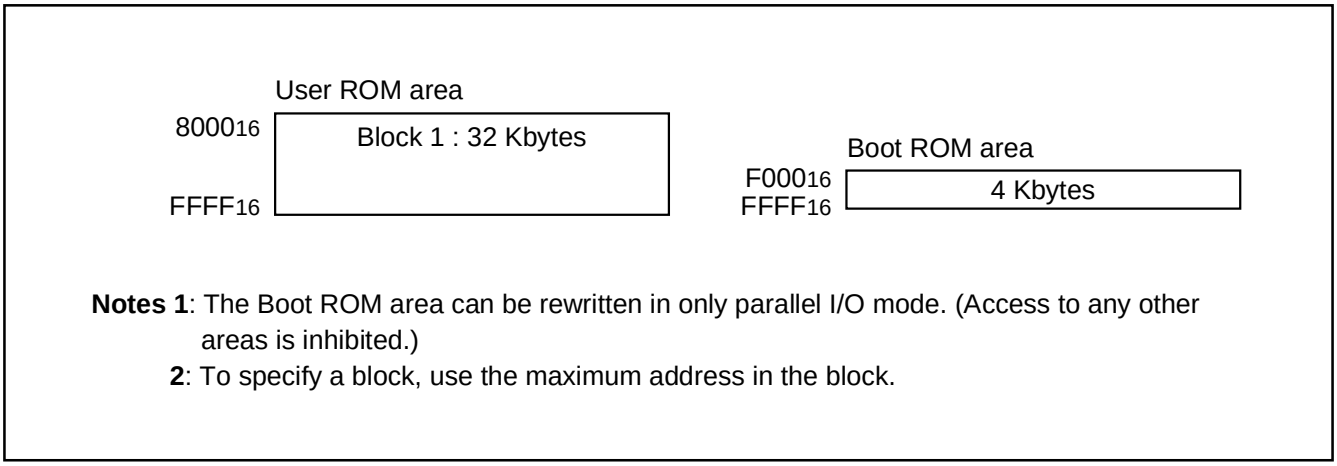


Fig. 116 Block diagram of built-in flash memory

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 0FFE16). 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 117 shows the flash memory control register.

Bit 0 is the RY/$\overline{BY}$ 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/$\overline{BY}$ pin function in parallel I/O mode.

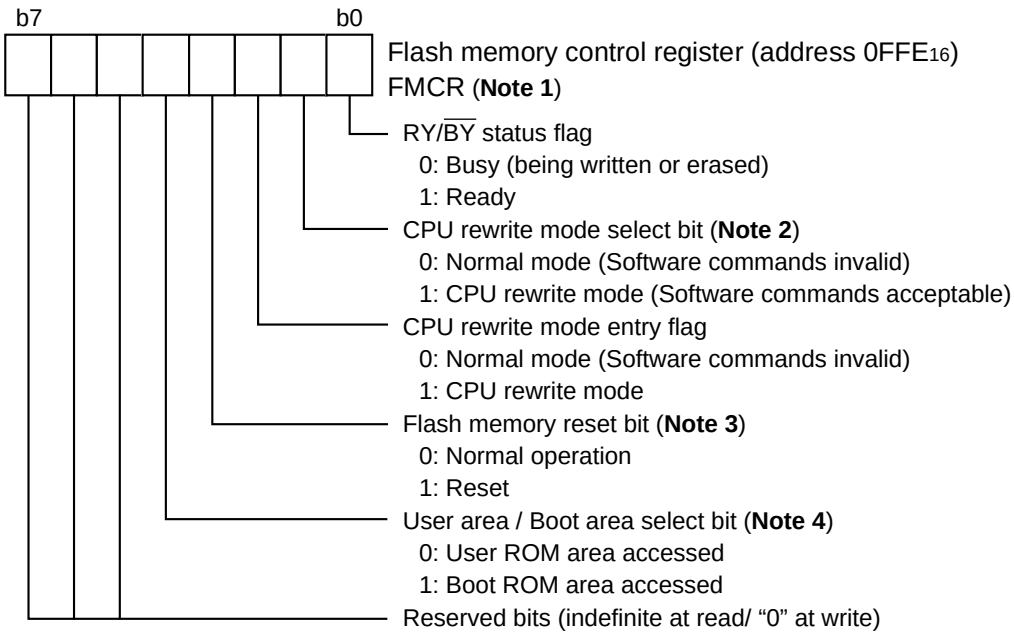
Bit 1 is the CPU Rewrite Mode Select Bit. When this bit is set to "1", the MCU enters CPU rewrite mode. Software commands are accepted once the mode is entered. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in a memory other 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".

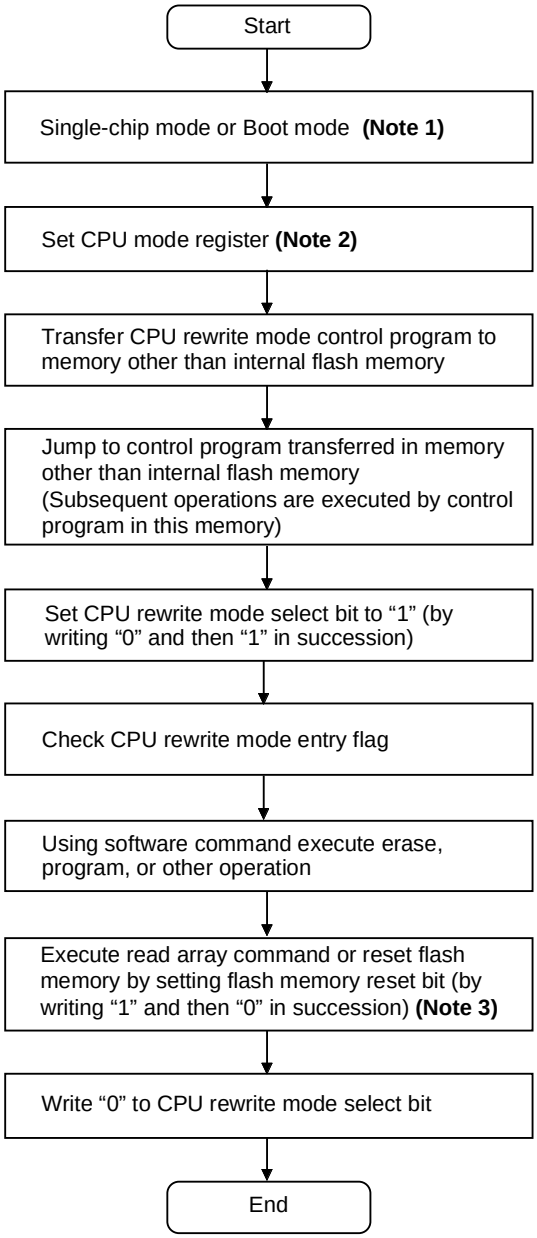
Bit 4 is the User Area/Boot Area Select Bit. When this bit is set to "1", Boot ROM area is accessed, and CPU rewrite mode in Boot ROM area is available. In Boot mode, this bit is set to "1" automatically. Reprogramming of this bit must be in a memory other than internal flash memory.

Figure 118 shows a flowchart for setting/releasing CPU rewrite mode.



Notes **1**: The contents of flash memory control register are "XXX00001" just after reset release.
2: For this bit to be set to "1", the user needs to write "0" and then "1" to it in succession. If it is not this procedure, this bit will not be set to "1". Additionally, it is required to ensure that no interrupt will be generated during that interval.
Use the control program in the area except the built-in flash memory for write to this bit.
3: This bit is valid when the CPU rewrite mode select bit is "1". Set this bit 3 to "0" subsequently after setting bit 3 to "1".
4: Use the control program in the area except the built-in flash memory for write to this bit.

Fig. 117 Structure of flash memory control register



Notes 1: When starting the MCU in the single-chip mode or memory expansion mode, supply 4.5 V to 5.25 V to the CNVss pin until checking the CPU rewrite mode entry flag.
2: Set the system clock division ration selection bits of CPU mode register (bits 6 and 7 at address $003B_{16}$).
3: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute the read array command or reset the flash memory.

Fig. 118 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 ϕ to 1.5 MHz or less using the system clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

●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 CNVss = “H”, so that the program will begin at the address which is stored in addresses $FFFC_{16}$ and $FFFD_{16}$ 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_{16})

The read array mode is entered by writing the command code "FF_{16}" 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_0 to D_7).
The read array mode is retained intact until another command is written.

●Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the first bus cycle, the contents of the status register are read out at the data bus (D_0 to D_7) by a read in the second bus cycle.
The status register is explained in the next section.

●Clear Status Register Command (50_{16})

This command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50_{16}" in the first bus cycle.

●Program Command (40_{16})

Program operation starts when the command code "40_{16}" 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/$\overline{BY}$ 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 (DB_0 to DB_7). The status register bit 7 (SR7) is set to "0" 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_{16}) is written.

During the program movement, The RY/$\overline{BY}$ 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.

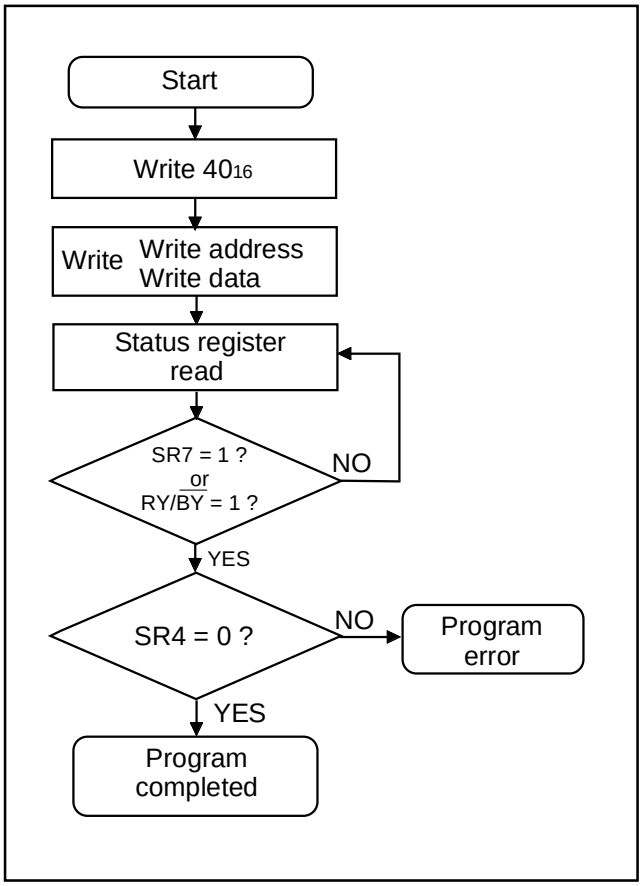


Fig. 119 Program flowchart

Table 9 List of software commands (CPU rewrite mode)

Command	Cycle number	First bus cycle			Second bus cycle		
		Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	1	Write	X **(Note 4)**	FF_{16}			
Read status register	2	Write	X	70_{16}	Read	X	SRD **(Note 1)**
Clear status register	1	Write	X	50_{16}			
Program	2	Write	X	40_{16}	Write	WA **(Note 2)**	WD **(Note 2)**
Erase all blocks	2	Write	X	20_{16}	Write	X	20_{16}
Block erase	2	Write	X	20_{16}	Write	BA **(Note 3)**	$D0_{16}$

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_{16}/20_{16})

By writing the command code "20_{16}" in the first bus cycle and the confirmation command code "20_{16}" 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/$\overline{BY}$ 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_0 to D_7). 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_{16}) is written.

The RY/$\overline{BY}$ 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_{16}/$D0_{16}$)

By writing the command code "20_{16}" in the first bus cycle and the confirmation command code "$D0_{16}$" 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/$\overline{BY}$ 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_{16}) is written.

The RY/$\overline{BY}$ 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.

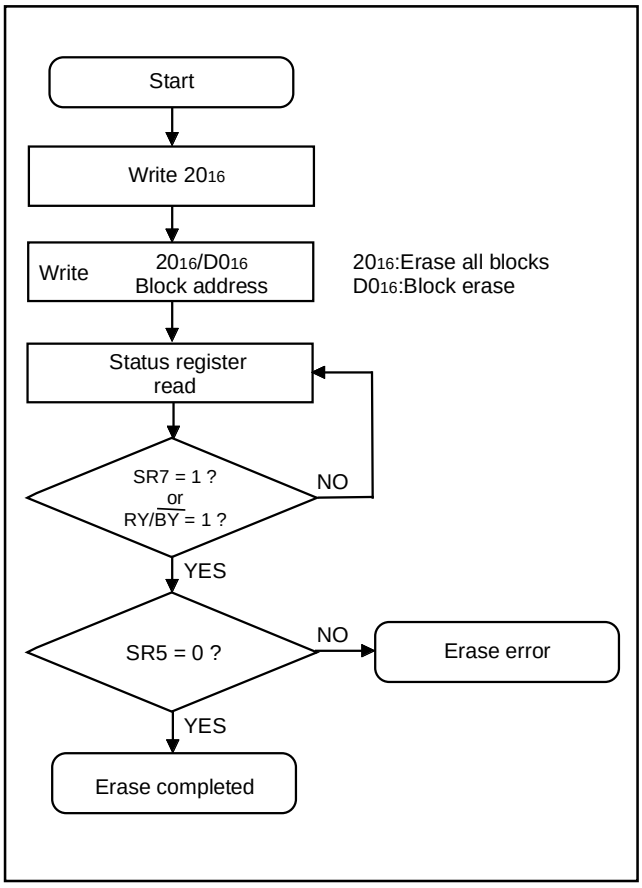


Fig. 120 Erase flowchart

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_{16})
(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_{16}) is input.

Also, the status register can be cleared by writing the clear status register command (50_{16}).
After reset, the status register is set to "80_{16}".
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.

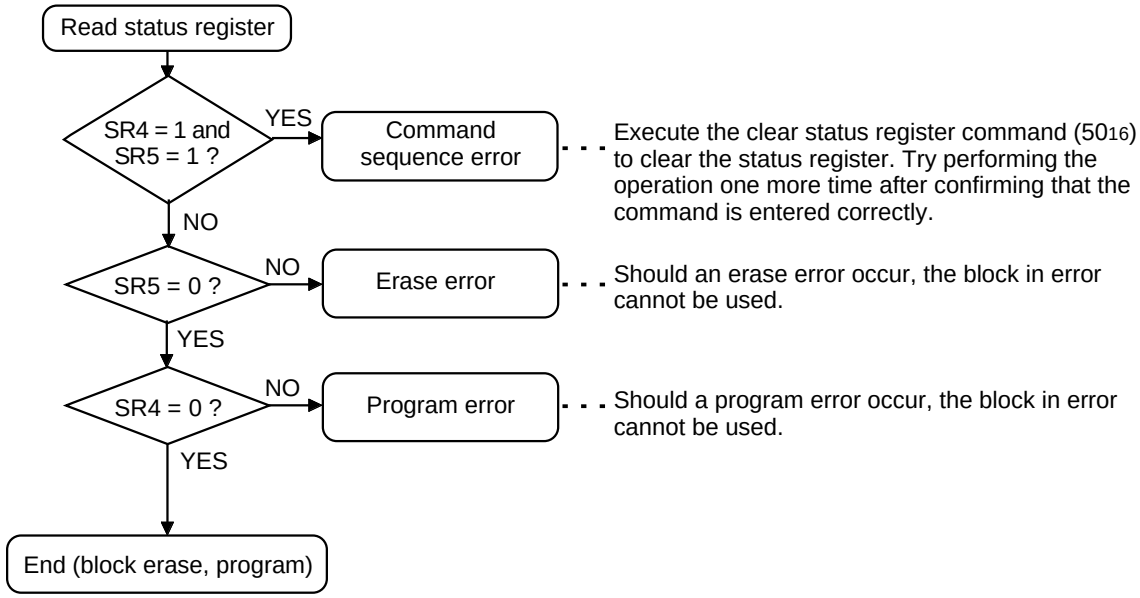
If "1" is written for any of the SR5 and SR4 bits, the program, erase all blocks, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.

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 121 shows a full status check flowchart and the action to be taken when each error occurs.



Note: When one of SR5 and SR4 is set to “1”, none of the program, erase all blocks, and block erase commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Fig. 121 Full status check flowchart and remedial procedure for errors

Functions To Inhibit Rewriting Flash Memory Version

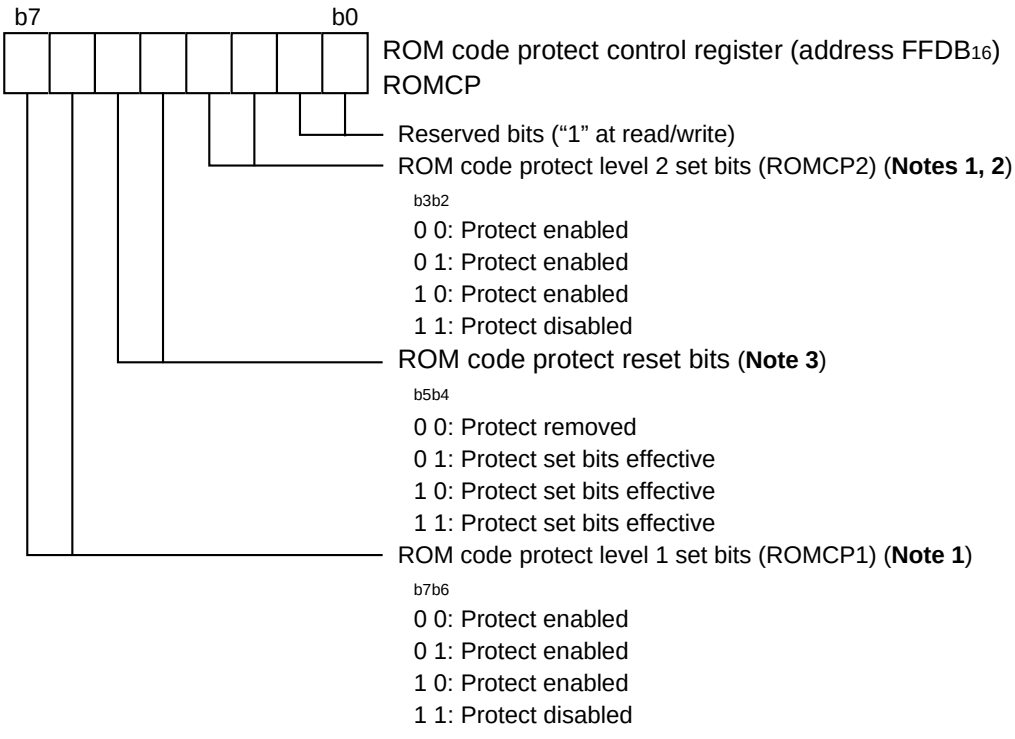
To prevent the contents of internal flash memory from being read out or rewritten easily, this MCU incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

●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 FFDB16) in parallel I/O mode. Figure 122 shows the ROM code protect control register (address FFDB16). (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.



Notes 1: When ROM code protect is turned on, the internal flash memory is protected against readout or modification in parallel I/O mode.

2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited.

3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be modified in parallel I/O mode, they need to be rewritten in serial I/O mode or CPU rewrite mode.

Fig. 122 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 FFD4$_{16}$ to FFDA$_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.

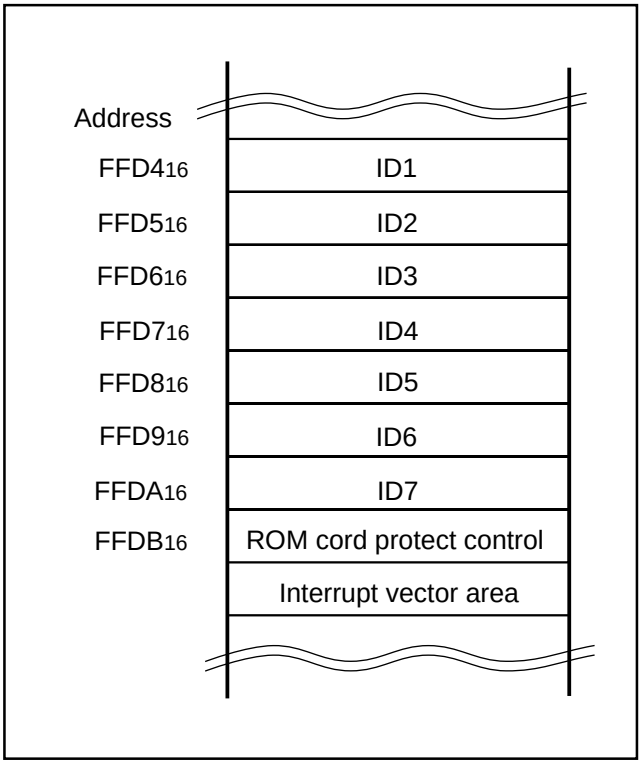


Fig. 123 ID code store addresses

(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 116 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 $F000_{16}$ through $FFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)

In the Boot ROM area, 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 ($\overline{CE}$) 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 Renesas Technology Corp.. 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 124 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 $\overline{SRDY}$ (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 $\overline{SRDY}$ (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 116 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 programer, 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.

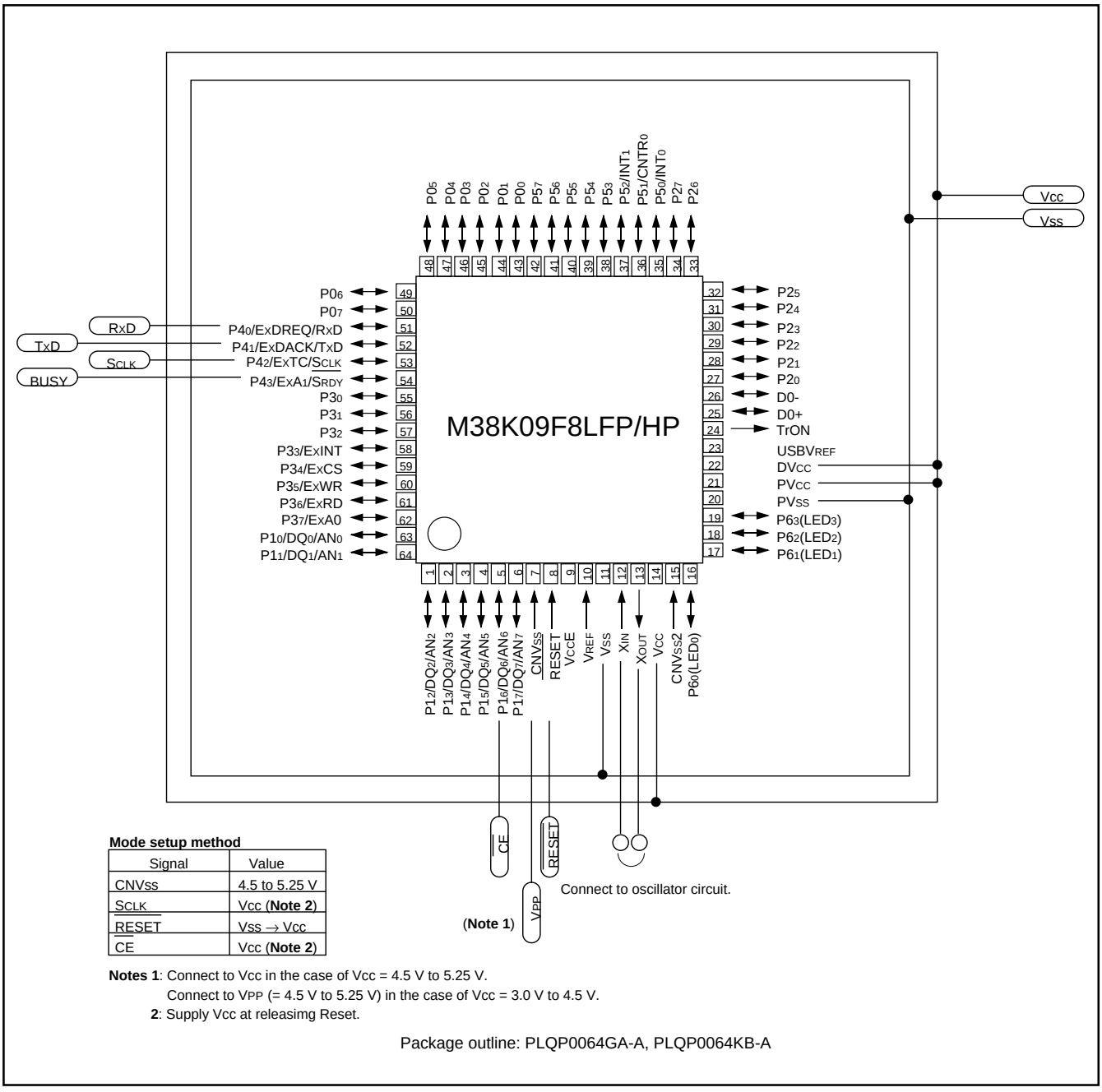
The TxD pin is for CMOS output. Transfer is in 8-bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the $\overline{SRDY}$ (BUSY) pin is “H” level. Accordingly, always start the next transfer after the $\overline{SRDY}$ (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 (L version) to the Vcc pin and 0 V to the Vss pin.
VCCE	Power supply		Connect this pin to Vcc.
CNVSS	VPP	I	Connect this pin to VPP (VPP = 4.50 to 5.25 V).
CNVSS2	CNVSS2	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.
$\overline{\text{RESET}}$	Reset input	I	To reset, input “L” level for 20 cycles or longer clocks of ϕ.
XIN	Clock input	I	Connect a ceramic or crystal resonator between the XIN and XOUT pins. When entering an externally drived clock, enter it from XIN and leave XOUT open.
XOUT	Clock output	O	
USBVREF	USB reference voltage input		Connect this pin to Vcc when not using.
TrON	USB reference voltage output	O	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.



Mode setup method

Signal	Value
CNVss	4.5 to 5.25 V
SCLK	Vcc (**Note 2**)
RESET	Vss → Vcc
CE	Vcc (**Note 2**)

Notes 1: Connect to Vcc in the case of Vcc = 4.5 V to 5.25 V.
Connect to VPP (= 4.5 V to 5.25 V) in the case of Vcc = 3.0 V to 4.5 V.
2: Supply Vcc at releasimg Reset.

Package outline: PLQP0064GA-A, PLQP0064KB-A

Fig. 124 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	FF16	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	4116	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Erase all blocks	A716	D016						Not acceptable
4	Read status register	7016	SRD output	SRD1 output					Acceptable
5	Clear status register	5016							Not acceptable
6	ID check function	F516	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
7	Download function	FA16	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
8	Version data output function	FB16	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	FC16	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_0 to A_7; Address middle is A_8 to A_{15}; Address high is A_{16} to A_{23}. Address-high A_{16} to A_{23} are always "00_{16}".

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.

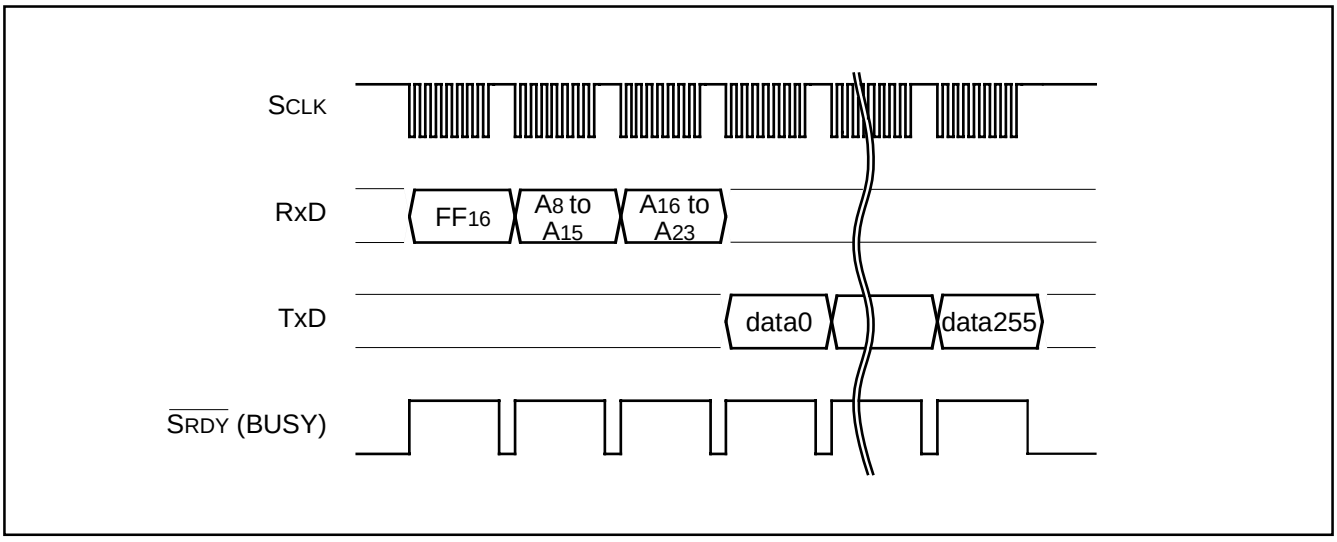


Fig. 125 Timing for page read

●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.

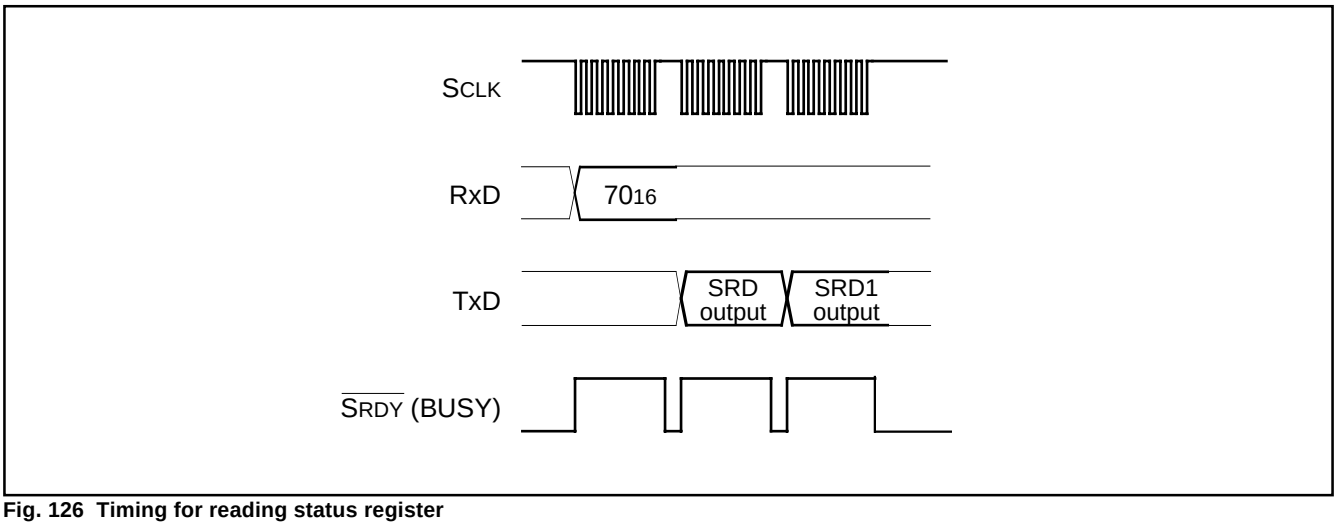


Fig. 126 Timing for reading status register

●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 "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the $\overline{SRDY}$ (BUSY) signal changes from "H" to "L" level.

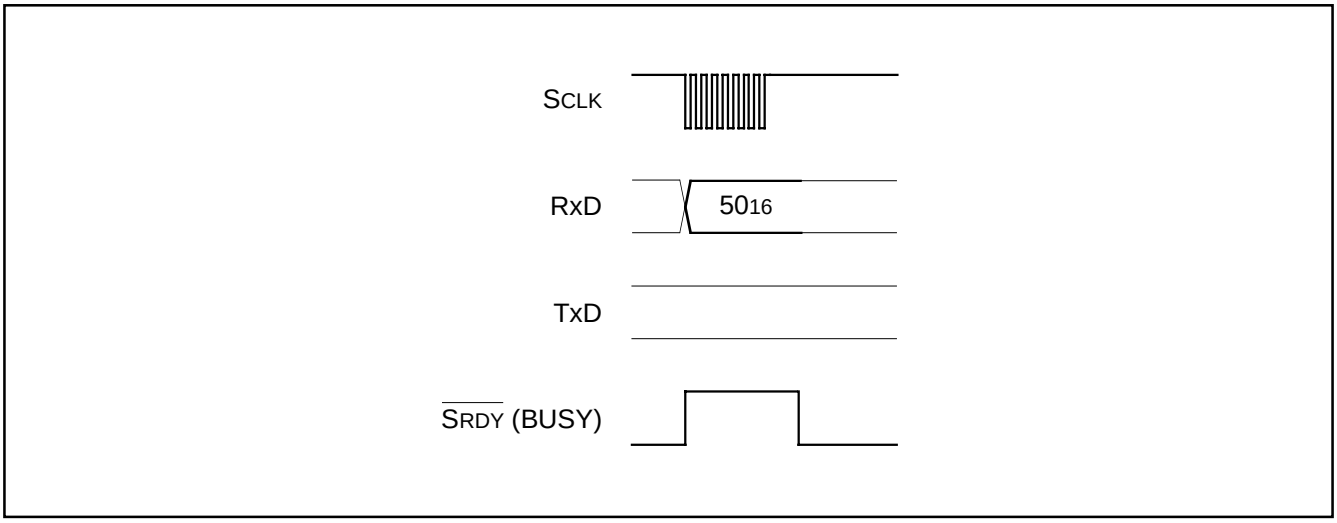


Fig. 127 Timing for clear status register

●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 "41_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D_0 to D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the $\overline{SRDY}$ (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.

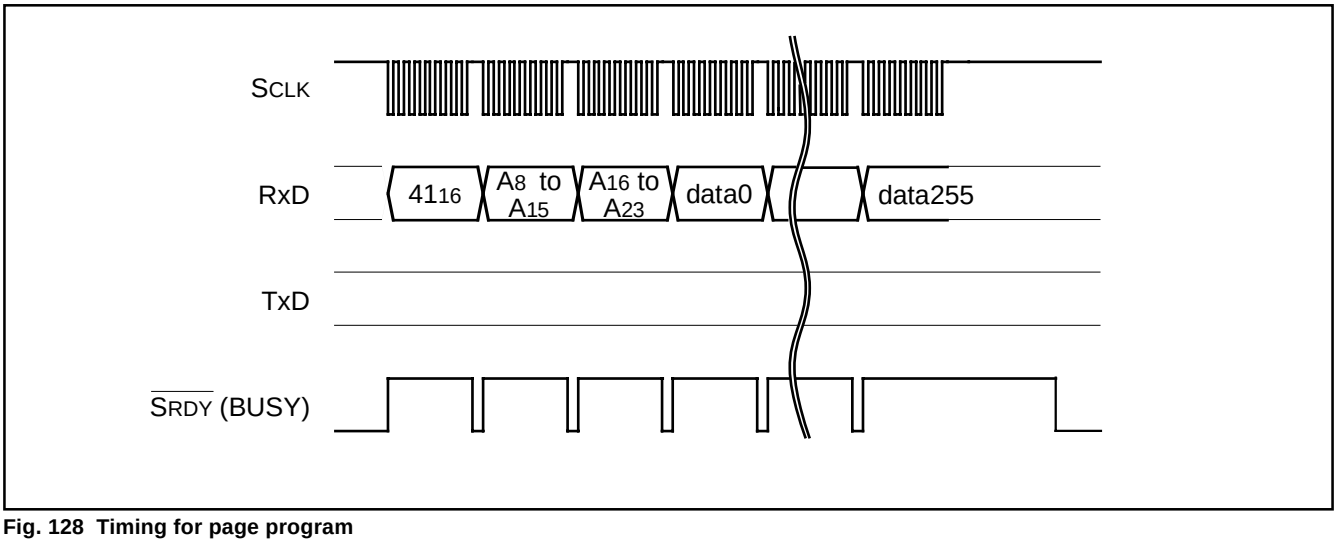


Fig. 128 Timing for page program

●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$_{16}$" command code with the 1st byte.

(2) Transfer the verify command code "D0$_{16}$" 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.

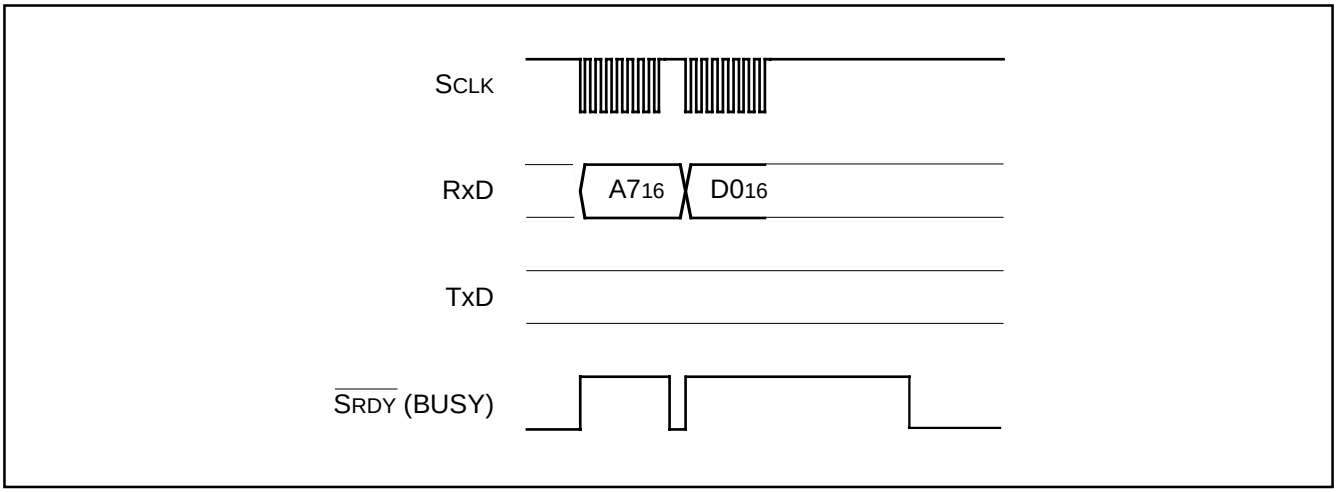


Fig. 129 Timing for erase all blocks

●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.

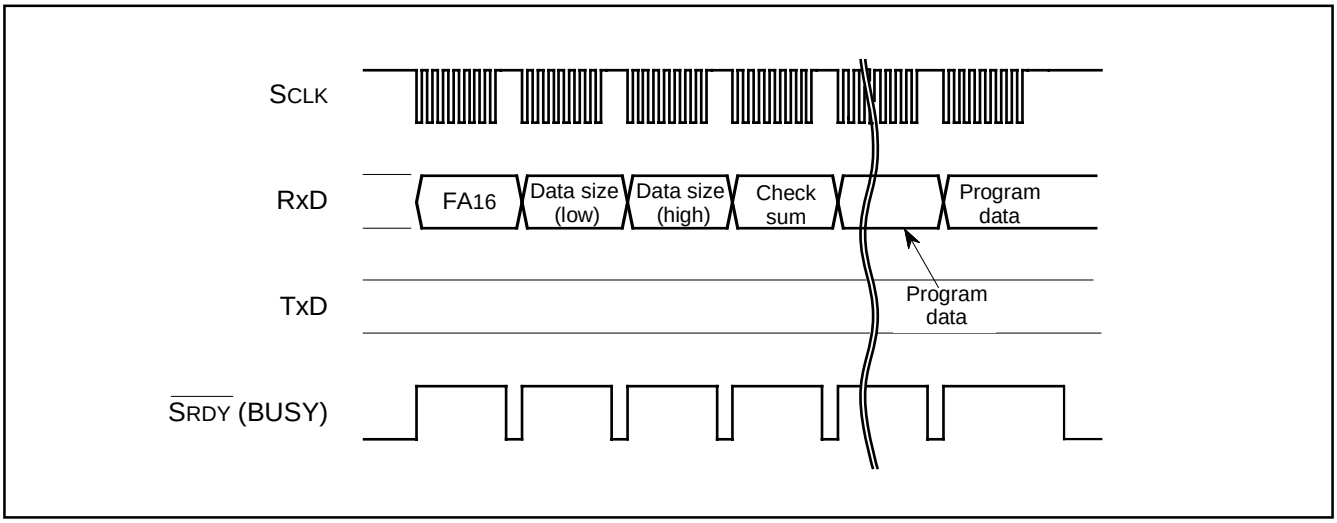


Fig. 130 Timing for download

●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.

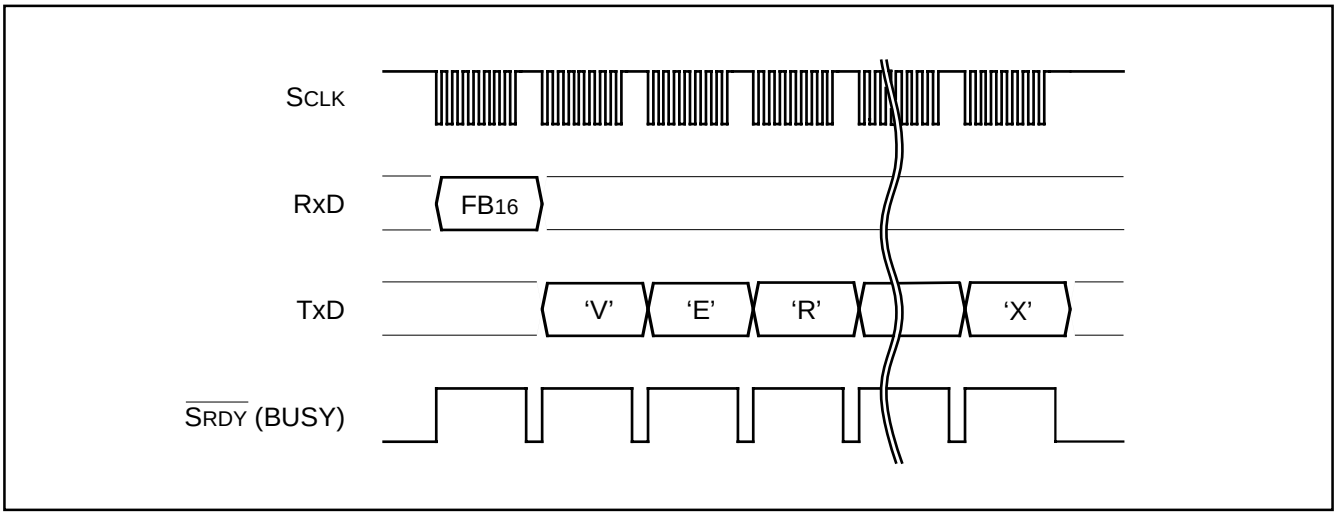


Fig. 131 Timing for version information output

●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.

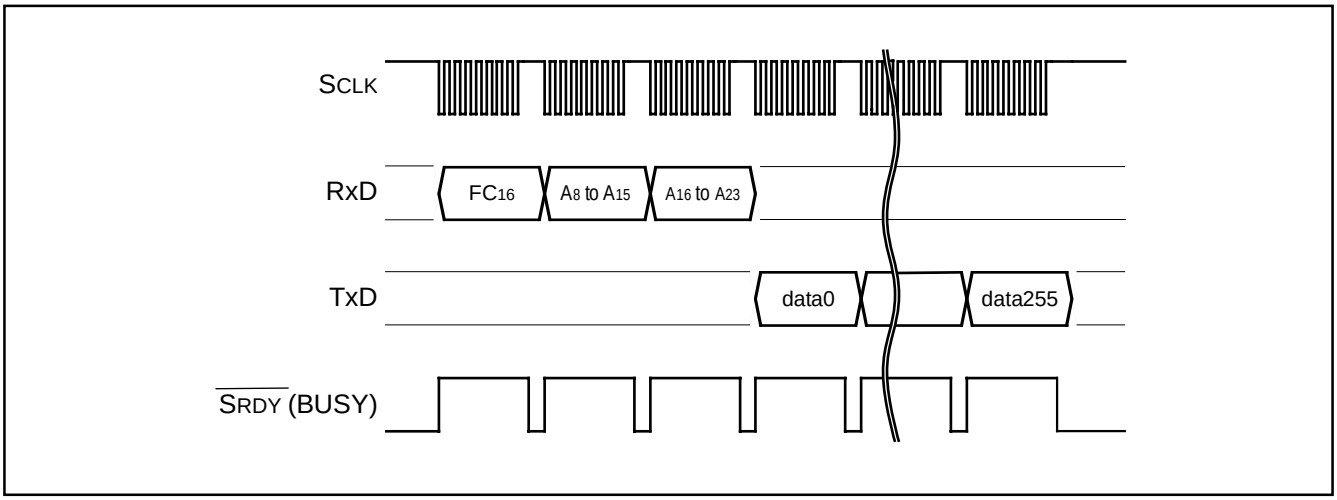


Fig. 132 Timing for Boot ROM area output

●ID Check

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

(1) Transfer the "F5$_{16}$" command code with the 1st byte.
(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ ("00$_{16}$") 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.

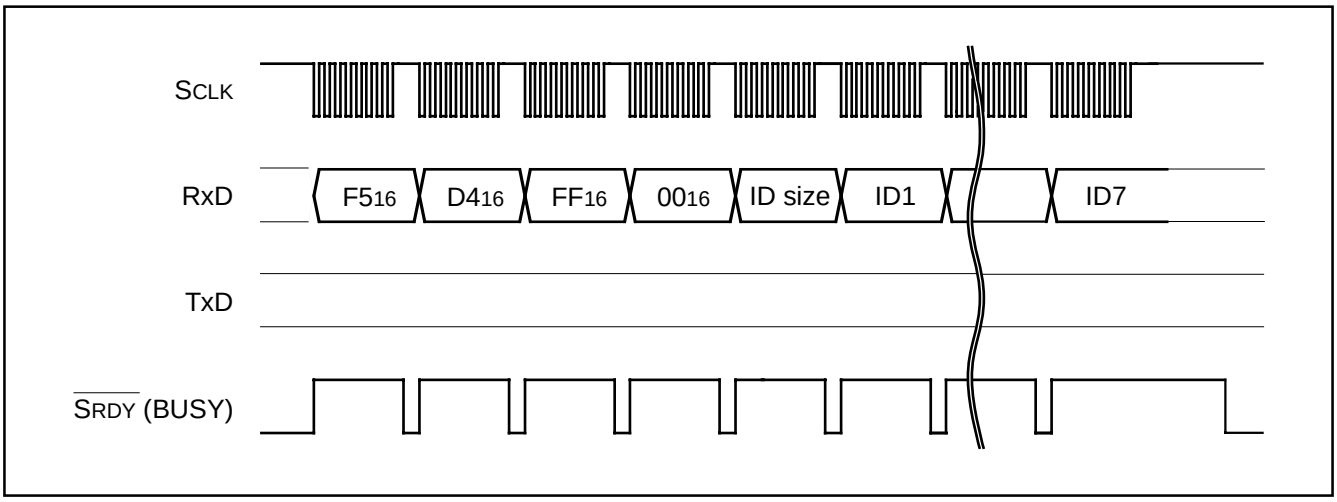


Fig. 133 Timing for ID check

●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$_{16}$ to FFDA$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

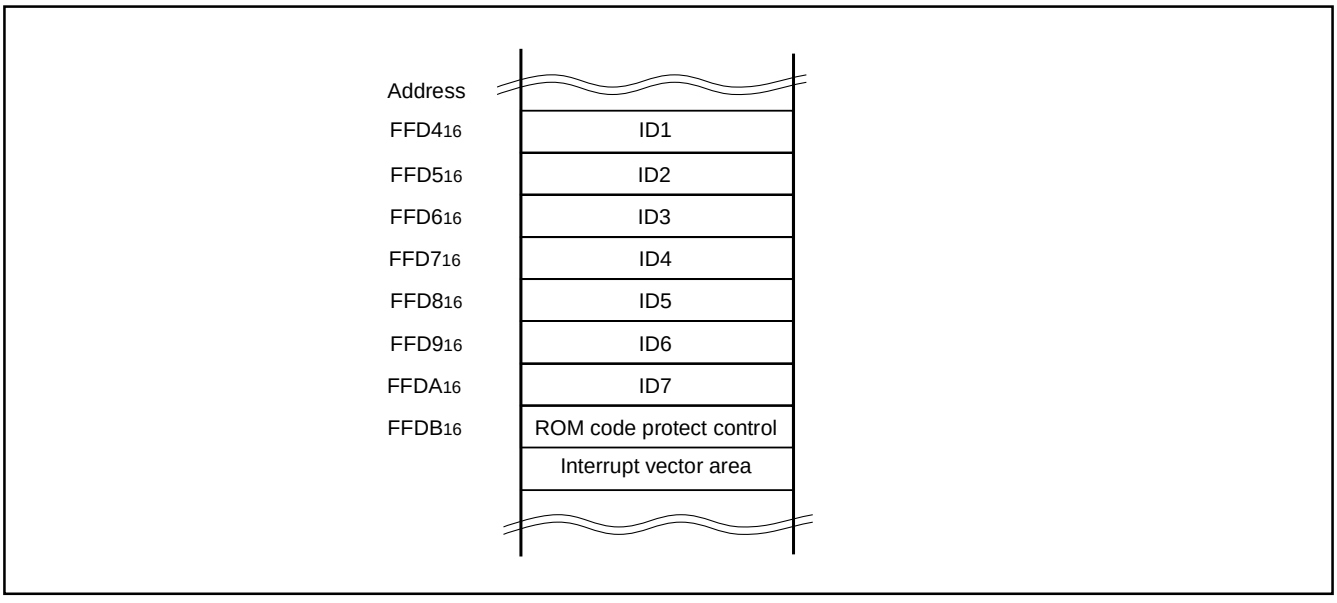


Fig. 134 ID code storage addresses

●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_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}).

Table 13 lists the definition of each status register bit. After releasing the reset, the status register becomes "80_{16}".

•Sequencer status (SR7)

The sequencer status indicates the operating status of the 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_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).

Table 14 lists the definition of each status register 1 bit. This register becomes "00_{16}" 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)

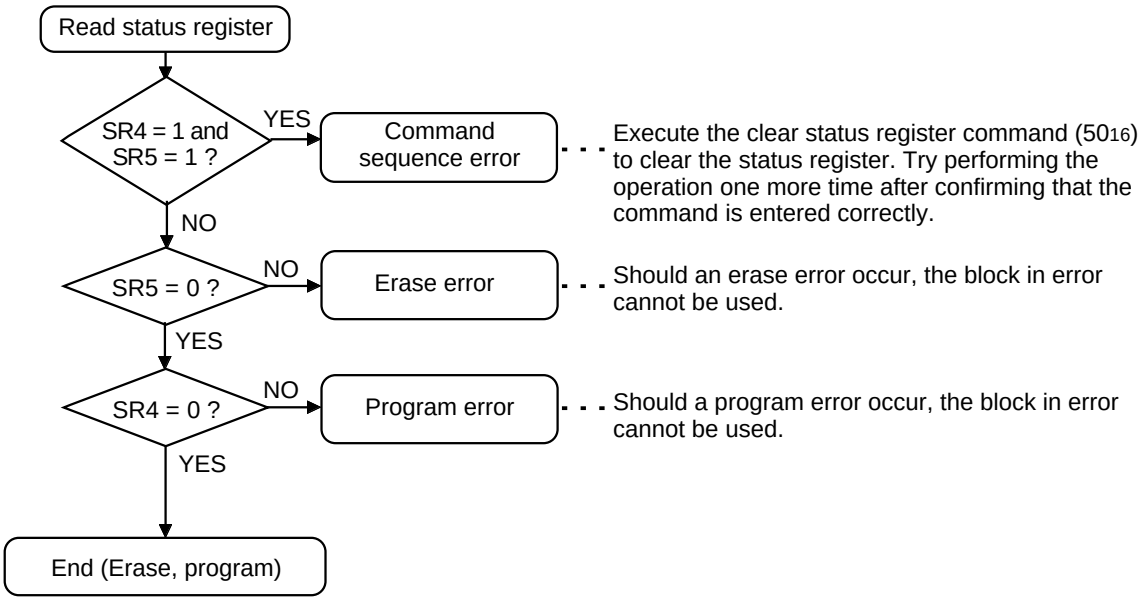
This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the MCU returns to the command wait state.

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 Not verified 01 Verification mismatch 10 Reserved 11 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 135 shows a flowchart of the full status check and explains how to remedy errors which occur.

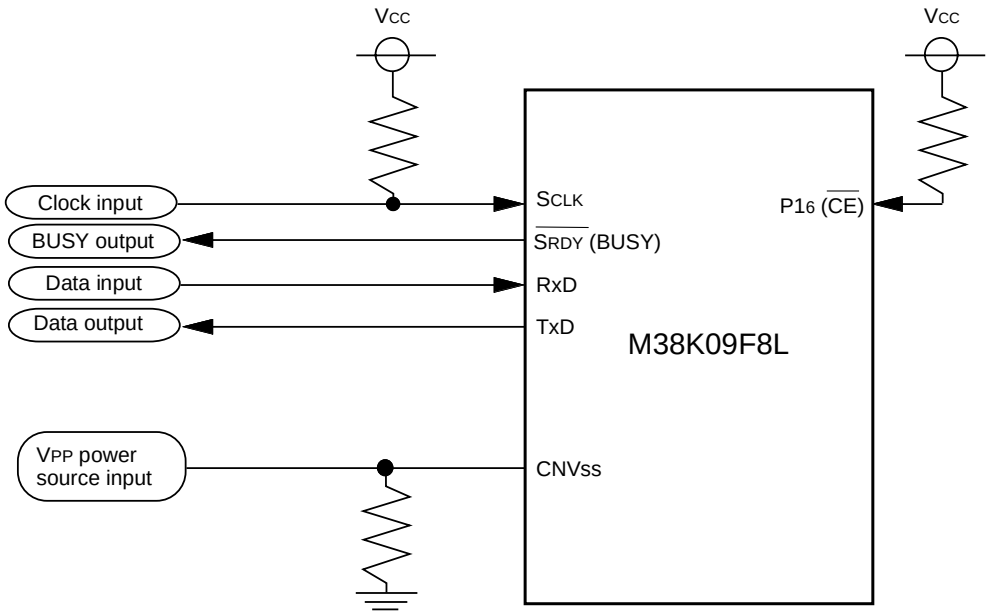


Note: When one of SR5 to SR4 is set to “1” , none of the program, erase all blocks, and block erase commands is accepted. Execute the clear status register command (5016) before executing these commands.

Fig. 135 Full status check flowchart and remedial procedure for errors

Example Circuit Application for Standard Serial I/O Mode

Figure 136 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.



Notes 1: Control pins and external circuitry will vary according to a programmer. For more information, see the programmer manual.

2: In this example, the V_{PP} power supply is supplied from an external source (programmer). To use the user's power source, connect to 4.5 V to 5.25 V.

Fig. 136 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(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 ϕ 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 ϕ cycle.

Definition of A/D Conversion Accuracy

The A/D conversion accuracy is defined below (refer to Figure 137).

•Relative accuracy

① Zero transition voltage (V_{OT})
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_{FST})
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_{OT} and V_{FST} of a converted value between V_{OT} and V_{FST}.

④ Differential non-linearity error
This means a deviation from the input potential difference required to change a converted value between V_{OT} and V_{FST} 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_{REF} of actual A/D conversion characteristics.

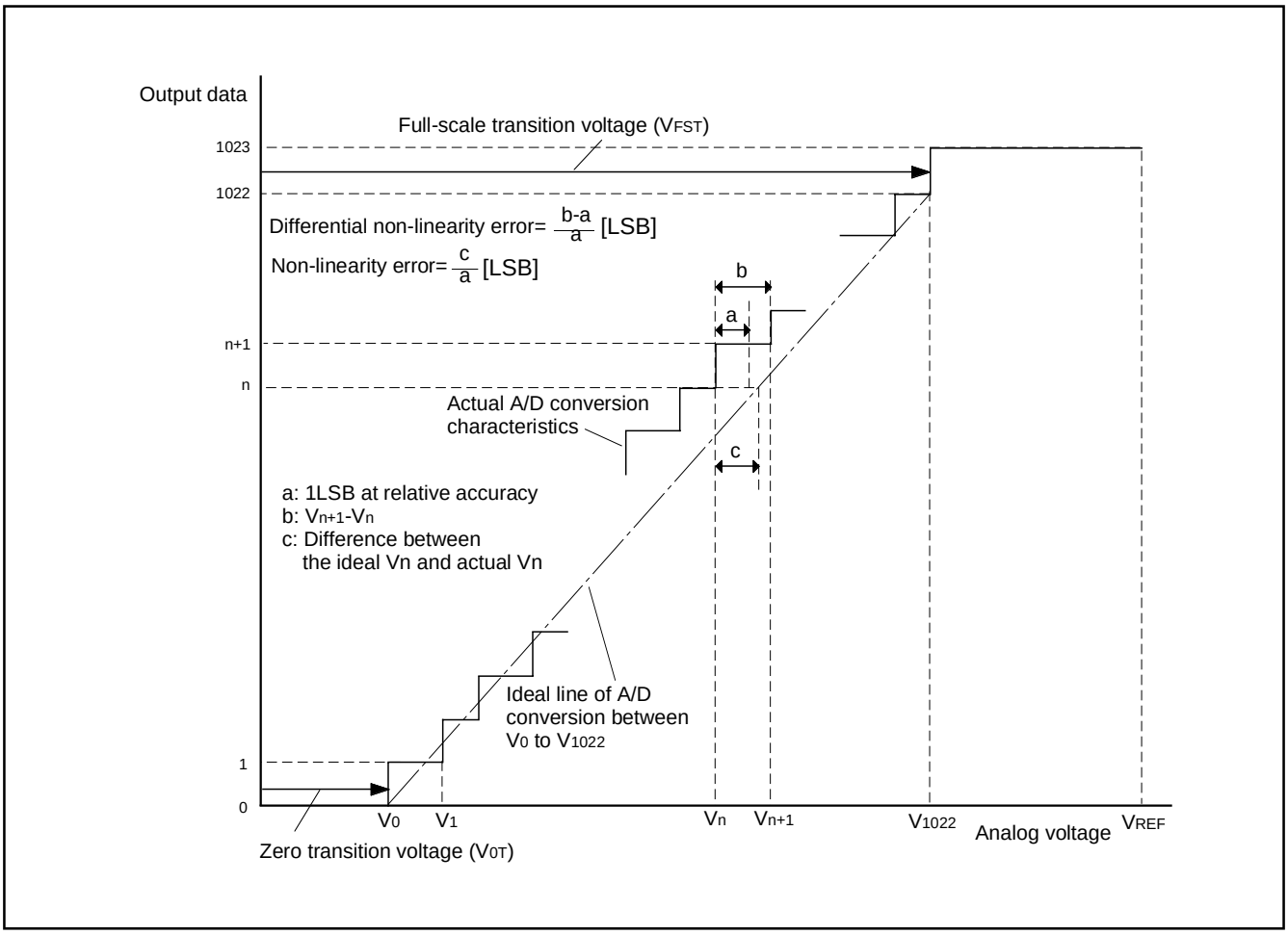


Fig. 137 Definition of A/D conversion accuracy

Vn: Analog input voltage when the output data changes from “n” to “n + 1” (n = 0 to 1022)

- 1 LSB at relative accuracy $\rightarrow \dfrac{V_{FST} - V_{OT}}{1022}$ (V)
- 1 LSB at absolute accuracy $\rightarrow \dfrac{V_{REF}}{1024}$ (V)

NOTES ON USAGE

Power Source Voltage

When the power source voltage value of a microcomputer is less than the value which is indicated as the recommended operating conditions, the microcomputer does not operate normally and may perform unstable operation.

In a system where the power source voltage drops slowly when the power source voltage drops or the power supply is turned off, reset a microcomputer when the power source voltage is less than the recommended operating conditions and design a system not to cause errors to the system by this unstable operation.

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 μF is recommended.

USB Port Pins (D0+, D0-) Treatment

•The USB specification requires a driver-impedance 28 to 44 Ω. In order to meet the USB specification impedance requirements, connect a resistor (27 Ω 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 connecter 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 μF capacitor (electrolytic capacitor) and a 0.1 μF 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.

USB Communication

In applications requiring high-reliability, we recommend providing the system with protective measures such as USB function initialization by software or USB reset by the host to prevent USB communication from being terminated unexpectedly, for example due to external causes such as noise.

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 kΩ 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.

* For the mask ROM confirmation and the mark specifications, refer to the “Renesas Technology Corp.” Homepage (http://www.renesas.com).

ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings

Table 15 Absolute maximum ratings

Symbol	Parameter		Conditions	Ratings	Unit
V_{CC}	Power source voltage		All voltages are based on V_{SS}. Output transistors are cut off.	–0.3 to 6.5	V
AVCC	Analog power source voltage	VCCE, VREF, PVCC, DVCC, USBVREF		–0.3 to VCC + 0.3	V
V_I	Input voltage	P00–P07, P10–P17, P20–P27, P30–P37, P40–P43, P50–P57, P60–P63		–0.3 to VCC + 0.3	V
V_I	Input voltage	$\overline{\text{RESET}}$, XIN, CNVSS2		–0.3 to VCC + 0.3	V
V_I	Input voltage	CNVSS / Mask ROM version		–0.3 to VCC + 0.3	V
		CNVSS / Flash memory version		–0.3 to 6.5	V
V_I	Input voltage	D0+, D0-		–0.5 to 3.8	V
V_O	Output voltage	P00–P07, P10–P17, P20–P27, P30–P37, P40–P43, P50–P57, P60–P63, XOUT		–0.3 to VCC + 0.3	V
V_O	Output voltage	D0+, D0-, TrON		–0.5 to 3.8	V
P_d	Power dissipation	(**Note**)	Ta = 25°C	500	mW
T_{opr}	Operating temperature		MCU operating	–20 to 85	°C
			In flash memory mode (For flash memory version)	25±5	°C
T_{stg}	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 (Vcc = 3.00 to 5.25 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter			Limits			Unit
				Min.	Typ.	Max.	
V_{CC}	Power source voltage	V_{CC}	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
AV_{CC}	Analog power source voltage	PV_{CC}, DV_{CC}			V_{CC}		V
AV_{CC}	Analog power source voltage	$V_{CC}E$			V_{CC}		V
V_{REF}	Analog reference voltage	V_{REF}		2.0		V_{CC}	V
V_{REF}	Analog reference voltage	$USBV_{REF}$	Vcc = 3.6 to 4.0 V	3.0		3.6	V
			Vcc = 3.0 to 3.6 V	3.0		V_{CC}	V
V_{SS}	Power source voltage	V_{SS}			0		V
AV_{SS}	Analog power source voltage	PV_{SS}			0		V
V_{IH}	“H” input voltage		$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$	0.8V_{CC}		V_{CC}	V V
V_{IH}	“H” input voltage		$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$	0.8$V_{CC}E$		$V_{CC}E$	V
V_{IH}	“H” input voltage		$\overline{RESET}$, X_{IN}, CNV_{SS}, CNV_{SS2}	0.8V_{CC}		V_{CC}	V
V_{IH}	“H” input voltage		D0+, D0-	2.0		3.6	V
V_{IL}	“L” input voltage		$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$	0		0.2V_{CC}	V
V_{IL}	“L” input voltage		$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$	0		0.2$V_{CC}E$	V
V_{IL}	“L” input voltage		$\overline{RESET}$, X_{IN}, CNV_{SS}, CNV_{SS2}	0		0.2V_{CC}	V
V_{IL}	“L” input voltage		D0+, D0-	0		0.8	V

Table 17 Recommended operating conditions (Vcc = 3.00 to 5.25 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
$\Sigma I_{OH(peak)}$	"H" total peak output current (**Note 1**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–80	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current (**Note 1**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			–80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current (**Note 1**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current (**Note 1**)	$P6_0$–$P6_3$			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current (**Note 1**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current (**Note 1**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–40	mA
$\Sigma I_{OH(avg)}$	"H" total average output current (**Note 1**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			–40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current (**Note 1**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current (**Note 1**)	$P6_0$–$P6_3$			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current (**Note 1**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			40	mA
$I_{OH(peak)}$	"H" peak output current (**Note 2**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–10	mA
$I_{OH(peak)}$	"H" peak output current (**Note 2**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			–10	mA
$I_{OL(peak)}$	"L" peak output current (**Note 2**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$			10	mA
$I_{OL(peak)}$	"L" peak output current (**Note 2**)	$P6_0$–$P6_3$			20	mA
$I_{OL(peak)}$	"L" peak output current (**Note 2**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			10	mA
$I_{OH(avg)}$	"H" average output current (**Note 3**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–5	mA
$I_{OH(avg)}$	"H" average output current (**Note 3**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			–5	mA
$I_{OL(avg)}$	"L" average output current (**Note 3**)	$P0_0$–$P0_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$			5	mA
$I_{OL(avg)}$	"L" average output current (**Note 3**)	$P6_0$–$P6_3$			10	mA
$I_{OL(avg)}$	"L" average output current (**Note 3**)	$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$			5	mA
$f(X_{IN})$	Main clock input oscillation frequency (**Note 4**)	Vcc = 4.00 to 5.25 V	6		12	MHz
		Vcc = 3.00 to 4.00 V	6		6	MHz
$f(X_{IN})$ or f(SYN)	System clock frequency	Vcc = 4.00 to 5.25 V	6		12	MHz
		Vcc = 3.00 to 4.00 V	6		6	MHz
$f(\phi)$	ϕ frequency	Vcc = 4.00 to 5.25 V			8	MHz
		Vcc = 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 %. 6 MHz or 12 MHz is usable.

Electrical Characteristics

Table 18 Electrical characteristics (1) (Vcc = 3.00 to 5.25 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V_{OH}	"H" output voltage P0$_0$–P0$_7$, P2$_0$–P2$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_3$	I_{OH} = –10 mA (Vcc = 4.00 to 5.25 V)	V_{CC}–2.0			V
		I_{OH} = –1 mA	V_{CC}–1.0			V
V_{OH}	"H" output voltage P1$_0$–P1$_7$, P3$_0$–P3$_7$, P4$_0$–P4$_3$	I_{OH} = –10 mA (V_{CC}E = 4.00 to 5.25 V)	V_{CC}E–2.0			V
		I_{OH} = –1 mA	V_{CC}E–1.0			V
V_{OH}	"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
V_{OL}	"L" output voltage P0$_0$–P0$_7$, P2$_0$–P2$_7$, P5$_0$–P5$_7$	I_{OL} = 10 mA (Vcc = 4.00 to 5.25 V)			2.0	V
		I_{OL} = 1 mA			1.0	V
V_{OL}	"L" output voltage P6$_0$–P6$_3$	I_{OL} = 20 mA (Vcc = 4.00 to 5.25 V)			2.0	V
		I_{OL} = 1 mA			1.0	V
V_{OL}	"L" output voltage P1$_0$–P1$_7$, P3$_0$–P3$_7$, P4$_0$–P4$_3$	I_{OL} = 10 mA (V_{CC}E = 4.00 to 5.25 V)			2.0	V
		I_{OL} = 1 mA (V_{CC}E = 3.00 to 5.25 V)			1.0	V
V_{OL}	"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
V_{T+}–V_{T-}	Hysteresis CNTR$_0$, INT$_0$, INT$_1$			0.6		V
V_{T+}–V_{T-}	Hysteresis P1$_0$/DQ$_0$–P1$_7$/DQ$_7$, P3$_0$–P3$_2$, P3$_3$/ExINT, P3$_4$/ExCS, P3$_5$/ExWR, P3$_6$/ExRD, P3$_7$/ExA0, P4$_0$/ExDREQ/RxD, P4$_1$/ExDACK/TxD, P4$_2$/ExTC/SCLK, P4$_3$/ExA1/$\overline{\text{SRDY}}$			0.6		V
V_{T+}–V_T	Hysteresis D0+, D0-			0.25		V
V_{T+}–V_{T-}	Hysteresis $\overline{\text{RESET}}$			0.5		V
I_{IH}	"H" input current P0$_0$–P0$_7$, P2$_0$–P2$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_3$	V_I = V_{CC} (Pull-ups "off")			5.0	µA
I_{IH}	"H" input current P1$_0$–P1$_7$, P3$_0$–P3$_7$, P4$_0$–P4$_3$	V_I = V_{CC}E			5.0	µA
I_{IH}	"H" input current $\overline{\text{RESET}}$, CNV$_{SS}$	V_I = V_{CC}			5.0	µA
I_{IH}	"H" input current X_{IN}	V_I = V_{CC}		4.0		µA
I_{IL}	"L" input current P0$_0$–P0$_7$, P2$_0$–P2$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_3$	V_I = V_{SS} (Pull-ups "off")			–5.0	µA
I_{IL}	"L" input current P1$_0$–P1$_7$, P3$_0$–P3$_7$, P4$_0$–P4$_3$	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current $\overline{\text{RESET}}$, CNV$_{SS}$, CNV$_{SS2}$	V_I = V_{SS}			–5.0	µA
I_{IL}	"L" input current X_{IN}	V_I = V_{SS}		–4.0		µA
I_{IL}	"L" input current P0$_0$–P0$_7$, P5$_0$, P5$_2$ (Pull-ups "on")	V_I = V_{SS} (Vcc = 4.00 to 5.25 V)	–20.0	–60.0	–120.0	µA
		V_I = V_{SS}	–10.0			µA
V_{RAM}	RAM hold voltage	When clock is stopped	2.00		5.25	V

Table 19 Electrical characteristics (2) (Vcc = 3.00 to 5.25 V, Vss = 0 V, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions			Limits			Unit
					Min.	Typ.	Max.	
I_{CC}	Power source current (Output transistor is isolated.)	Normal mode (**Note 1**)	Vcc = 4.00 to 5.25 V	f(XIN) = system clock = 12 MHz, ϕ = 6 MHz, USB reference voltage circuit enabled		21.0	60	mA
				f(XIN) = 12 MHz, System clock = ϕ = 8 MHz, USB reference voltage circuit enabled		22.5	60	mA
				f(XIN) = 6 MHz, System clock = ϕ = 8 MHz, USB reference voltage circuit enabled		22.0	60	mA
				f(XIN) = system clock = ϕ = 6 MHz, USB reference voltage circuit enabled		21.0	60	mA
			Vcc = 3.00 to 4.00 V	f(XIN) = system clock = ϕ = 6 MHz, USB reference voltage circuit disabled			35	mA
			Vcc = 3.00 to 3.60 V	f(XIN) = system clock = ϕ = 6 MHz, USB reference voltage circuit disabled		9.0	30	mA
		Wait mode (**Note 2**)	Vcc = 4.00 to 5.25 V	f(XIN) = 12 MHz, System clock = ϕ = 8 MHz, USB reference voltage circuit enabled		6.0		mA
			Vcc = 3.00 to 4.00 V	f(XIN) = system clock = ϕ = 6 MHz, USB reference voltage circuit disabled		2.0		mA
		Stop mode (**Note 3**)	Vcc = 4.00 to 5.25 V	USB reference voltage circuit enabled Low current mode		125.0	250	µA
			Vcc = 3.00 to 5.25 V	USB reference voltage circuit disabled Ta = 25 °C		0.1		µA
				USB reference voltage circuit disabled Ta = 85 °C			10	µA

<Test conditions>

Notes 1: Operating in single-chip mode
Clock input from XIN pin (XOUT oscillator stopped)
fUSB = 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 XIN pin (XOUT oscillator stopped)
fUSB = 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, Ta = –20 to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				10	Bits
—	Linearity error	Ta = 25 °C			±3	LSB
—	Differential nonlinear error	Ta = 25 °C			±1.5	LSB
V_{OT}	Zero transition voltage	V_{CC} = V_{REF} = 5.12 V	0	15	35	mV
V_{FST}	Full scale transition voltage	V_{CC} = V_{REF} = 5.12 V	5105	5125	5150	mV
t_{CONV}	Conversion time				122	tc(X_{IN}) or tc(f_{SYN})
R_{LADDER}	Ladder resistor			35		kΩ
I_{VREF}	Reference power source input current	A/D converter operating; V_{REF} = 5.0 V	50	150	200	µA
		A/D converter not operating; V_{REF} = 5.0 V			5	
$I_{I(AD)}$	A/D port input current				5.0	µA

Timing Requirements

Table 21 Timing requirements (1) (V_{CC} = 4.00 to 5.25 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{C(XIN)}$	Main clock input cycle time	83			ns
$t_{WH(XIN)}$	Main clock input “H” pulse width	35			ns
$t_{WL(XIN)}$	Main clock input “L” pulse width	35			ns
$t_{C(CNTR)}$	$CNTR_0$ input cycle time	200			ns
$t_{WH(CNTR)}$	$CNTR_0$ input “H” pulse width	80			ns
$t_{WL(CNTR)}$	$CNTR_0$ input “L” pulse width	80			ns
$t_{WH(INT)}$	INT_0, INT_1 input “H” pulse width	80			ns
$t_{WL(INT)}$	INT_0, INT_1 input “L” pulse width	80			ns
$t_{C(SCLK)}$	Serial I/O clock input cycle time (**Note**)	800			ns
$t_{WH(SCLK)}$	Serial I/O clock input “H” pulse width (**Note**)	370			ns
$t_{WL(SCLK)}$	Serial I/O clock input “L” pulse width (**Note**)	370			ns
$t_{su(RxD–SCLK)}$	Serial I/O input set up time	220			ns
$t_{h(SCLK–RxD)}$	Serial I/O input hold time	100			ns

Note: These limits are the rating values in the clock synchronous mode, bit 6 of address $0FE0_{16}$ = “1”. In the UART mode, bit 6 of address $0FE0_{16}$ = “0”; the rating values are set to one fourth.

Table 22 Timing requirements (2) (V_{CC} = 3.00 to 4.00 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(\overline{RESET})}$	Reset input “L” pulse width	2			µs
$t_{C(XIN)}$	Main clock input cycle time	166			ns
$t_{WH(XIN)}$	Main clock input “H” pulse width	70			ns
$t_{WL(XIN)}$	Main clock input “L” pulse width	70			ns
$t_{C(CNTR)}$	$CNTR_0$ input cycle time	500			ns
$t_{WH(CNTR)}$	$CNTR_0$ input “H” pulse width	230			ns
$t_{WL(CNTR)}$	$CNTR_0$ input “L” pulse width	230			ns
$t_{WH(INT)}$	INT_0, INT_1 input “H” pulse width	230			ns
$t_{WL(INT)}$	INT_0, INT_1 input “L” pulse width	230			ns
$t_{C(SCLK)}$	Serial I/O clock input cycle time (**Note**)	2000			ns
$t_{WH(SCLK)}$	Serial I/O clock input “H” pulse width (**Note**)	950			ns
$t_{WL(SCLK)}$	Serial I/O clock input “L” pulse width (**Note**)	950			ns
$t_{su(RxD–SCLK)}$	Serial I/O input set up time	400			ns
$t_{h(SCLK–RxD)}$	Serial I/O input hold time	200			ns

Note: These limits are the rating values in the clock synchronous mode, bit 6 of address $0FE0_{16}$ = “1”. In the UART mode, bit 6 of address $0FE0_{16}$ = “0”; the rating values are set to one fourth.

Table 23 Timing requirements of external bus interface (EXB) (1)

(V_{CC} = 4.00 to 5.25 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
t_{su}(S-R)	ExCS setup time for read		0			ns
t_{su}(S-W)	ExCS setup time for write		0			ns
t_h(R-S)	ExCS hold time for read		0			ns
t_h(W-S)	ExCS hold time for write		0			ns
t_{su}(A-R)	ExA0, ExA1 setup time for read		10			ns
t_{su}(A-W)	ExA0, ExA1 setup time for write		10			ns
t_h(R-A)	ExA0, ExA1 hold time for read		0			ns
t_h(W-A)	ExA0, ExA1 hold time for write		0			ns
t_{su}(ACK-R)	ExDACK setup time for read		10			ns
t_{su}(ACK-W)	ExDACK setup time for write		10			ns
t_h(R-ACK)	ExDACK hold time for read		0			ns
t_h(W-ACK)	ExDACK hold time for write		0			ns
t_{WH}(R)	Read “H” pulse width		80			ns
t_{WL}(R)	Read “L” pulse width		80			ns
t_{WH}(W)	Write “H” pulse width		80			ns
t_{WL}(W)	Write “L” pulse width		80			ns
t_{WH}(ACK)	ExDACK “H” pulse width		120			ns
t_{WL}(ACK)	ExDACK “L” pulse width		120			ns
t_{su}(D-W)	Data input setup time before write		40			ns
t_h(W-D)	Data input hold time after write		0			ns
t_{su}(D-ACK)	Data input setup time before ExDACK		60			ns
t_h(ACK-W)	Data input hold time after ExDACK		5			ns
$t_C(\phi)$	CPU clock cycle time		125			ns
tw(cycle)	Burst mode access cycle time	USB function not operating	$t_C(\phi)$•3+10			ns
		USB function operating	$t_C(\phi)$•5+10			ns

Table 24 Timing requirements of external bus interface (EXB) (2)

(V_{CC} = 3.00 to 4.00 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
t_{su}(S-R)	ExCS setup time for read		0			ns
t_{su}(S-W)	ExCS setup time for write		0			ns
t_h(R-S)	ExCS hold time for read		0			ns
t_h(W-S)	ExCS hold time for write		0			ns
t_{su}(A-R)	ExA0, ExA1 setup time for read		30			ns
t_{su}(A-W)	ExA0, ExA1 setup time for write		30			ns
t_h(R-A)	ExA0, ExA1 hold time for read		0			ns
t_h(W-A)	ExA0, ExA1 hold time for write		0			ns
t_{su}(ACK-R)	ExDACK setup time for read		30			ns
t_{su}(ACK-W)	ExDACK setup time for write		30			ns
t_h(R-ACK)	ExDACK hold time for read		0			ns
t_h(W-ACK)	ExDACK hold time for write		0			ns
t_{WH}(R)	Read “H” pulse width		120			ns
t_{WL}(R)	Read “L” pulse width		120			ns
t_{WH}(W)	Write “H” pulse width		120			ns
t_{WL}(W)	Write “L” pulse width		120			ns
t_{WH}(ACK)	ExDACK “H” pulse width		160			ns
t_{WL}(ACK)	ExDACK “L” pulse width		160			ns
t_{su}(D-W)	Data input setup time before write		60			ns
t_h(W-D)	Data input hold time after write		0			ns
t_{su}(D-ACK)	Data input setup time before ExDACK		80			ns
t_h(ACK-W)	Data input hold time after ExDACK		10			ns
$t_C(\phi)$	CPU clock cycle time		166			ns
tw(cycle)	Burst mode access cycle time	USB function not operating	$t_C(\phi)$•3+30			ns
		USB function operating	$t_C(\phi)$•5+30			ns

Switching Characteristics

Table 25 Switching characteristics (1) (V_{CC} = 4.00 to 5.25 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{WH(SCLK)}$	Serial I/O clock output “H” pulse width	$t_{C(SCLK)}$/2–30			ns
$t_{WL(SCLK)}$	Serial I/O clock output “L” pulse width	$t_{C(SCLK)}$/2–30			ns
$t_{d(SCLK–TxD)}$	Serial I/O output delay time			140	ns
$t_{v(SCLK–TxD)}$	Serial I/O output valid time	–30			ns
$t_{r(SCLK)}$	Serial I/O clock output rising time			30	ns
$t_{f(SCLK)}$	Serial I/O clock output falling time			30	ns
$t_{r(CMOS)}$	CMOS output rising time (**Note**)			30	ns
$t_{f(CMOS)}$	CMOS output falling time (**Note**)			30	ns

Notes: Pins X_{OUT}, D0+, D0- are excluded.

Table 26 Switching characteristics (2) (V_{CC} = 3.00 to 4.00 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{WH(SCLK)}$	Serial I/O clock output “H” pulse width	$t_{C(SCLK)}$/2–50			ns
$t_{WL(SCLK)}$	Serial I/O clock output “L” pulse width	$t_{C(SCLK)}$/2–50			ns
$t_{d(SCLK–TxD)}$	Serial I/O output delay time			350	ns
$t_{v(SCLK–TxD)}$	Serial I/O output valid time	–30			ns
$t_{r(SCLK)}$	Serial I/O clock output rising time			50	ns
$t_{f(SCLK)}$	Serial I/O clock output falling time			50	ns
$t_{r(CMOS)}$	CMOS output rising time (**Note**)			50	ns
$t_{f(CMOS)}$	CMOS output falling time (**Note**)			50	ns

Notes: Pins X_{OUT}, D0+, D0- are excluded.

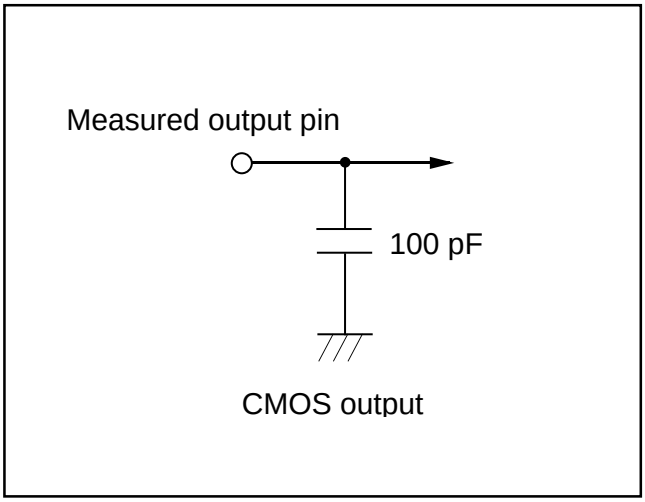


Fig. 138 Output switching characteristics measurement circuit

Table 27 Switching characteristics of external bus interface (EXB) (1)

(V_{CC} = 4.00 to 5.25 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
t_a(R-D)	Data output enable time after read				60	ns
t_v(R-D)	Data output disable time after read		0			ns
t_a(ACK-D)	Data output enable time after ExDACK				80	ns
t_v(ACK-D)	Data output disable time after ExDACK		0			ns
t_d(R-Mdis)	In cycle mode Mch_req disable output delay time after read				$t_{c(\phi)}$+10	ns
t_d(W-Mdis)	In cycle mode Mch_req disable output delay time after write				$t_{c(\phi)}$+10	ns
t_d(R-Men)	In cycle mode Mch_req enable output delay time after read	USB function not operating			$t_{c(\phi)}$•3+10	ns
		USB function operating			$t_{c(\phi)}$•5+10	ns
t_d(W-Men)	In cycle mode Mch_req enable output delay time after write	USB function not operating			$t_{c(\phi)}$•3+10	ns
		USB function operating			$t_{c(\phi)}$•5+10	ns

Table 28 Switching characteristics of external bus interface (EXB) (2)

(V_{CC} = 3.00 to 4.00 V, V_{SS} = 0 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
t_a(R-D)	Data output enable time after read				80	ns
t_v(R-D)	Data output disable time after read		0			ns
t_a(ACK-D)	Data output enable time after ExDACK				120	ns
t_v(ACK-D)	Data output disable time after ExDACK		0			ns
t_d(R-Mdis)	In cycle mode Mch_req disable output delay time after read				$t_{c(\phi)}$+30	ns
t_d(W-Mdis)	In cycle mode Mch_req disable output delay time after write				$t_{c(\phi)}$+30	ns
t_d(R-Men)	In cycle mode Mch_req enable output delay time after read	USB function not operating			$t_{c(\phi)}$•3+30	ns
		USB function operating			$t_{c(\phi)}$•5+30	ns
t_d(W-Men)	In cycle mode Mch_req enable output delay time after write	USB function not operating			$t_{c(\phi)}$•3+30	ns
		USB function operating			$t_{c(\phi)}$•5+30	ns

Table 29 Switching characteristics (USB ports) (V_{CC} = 3.00 to 5.25 V, V_{SS} = 0 V, Ta = –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 rising 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

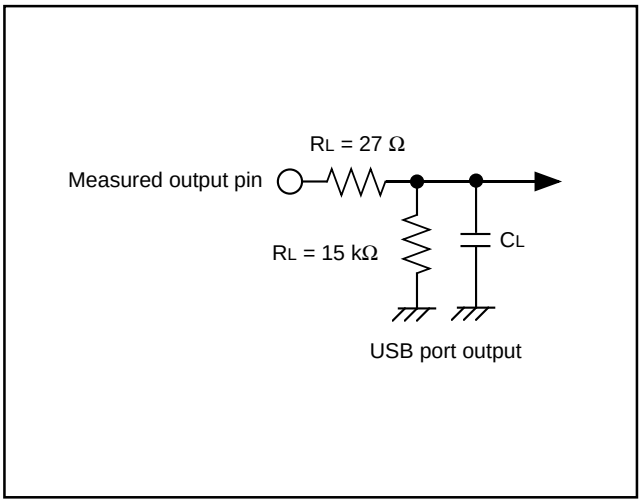


Fig. 139 USB output switching characteristics measurement circuit (1) for D0-

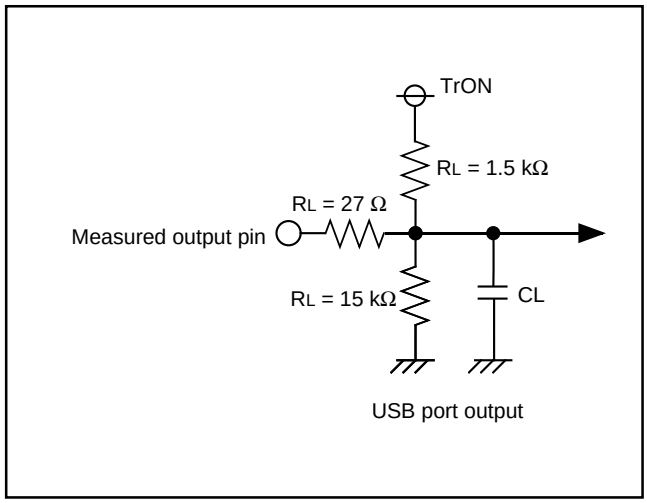


Fig. 140 USB output switching characteristics measurement circuit (2) for D0+

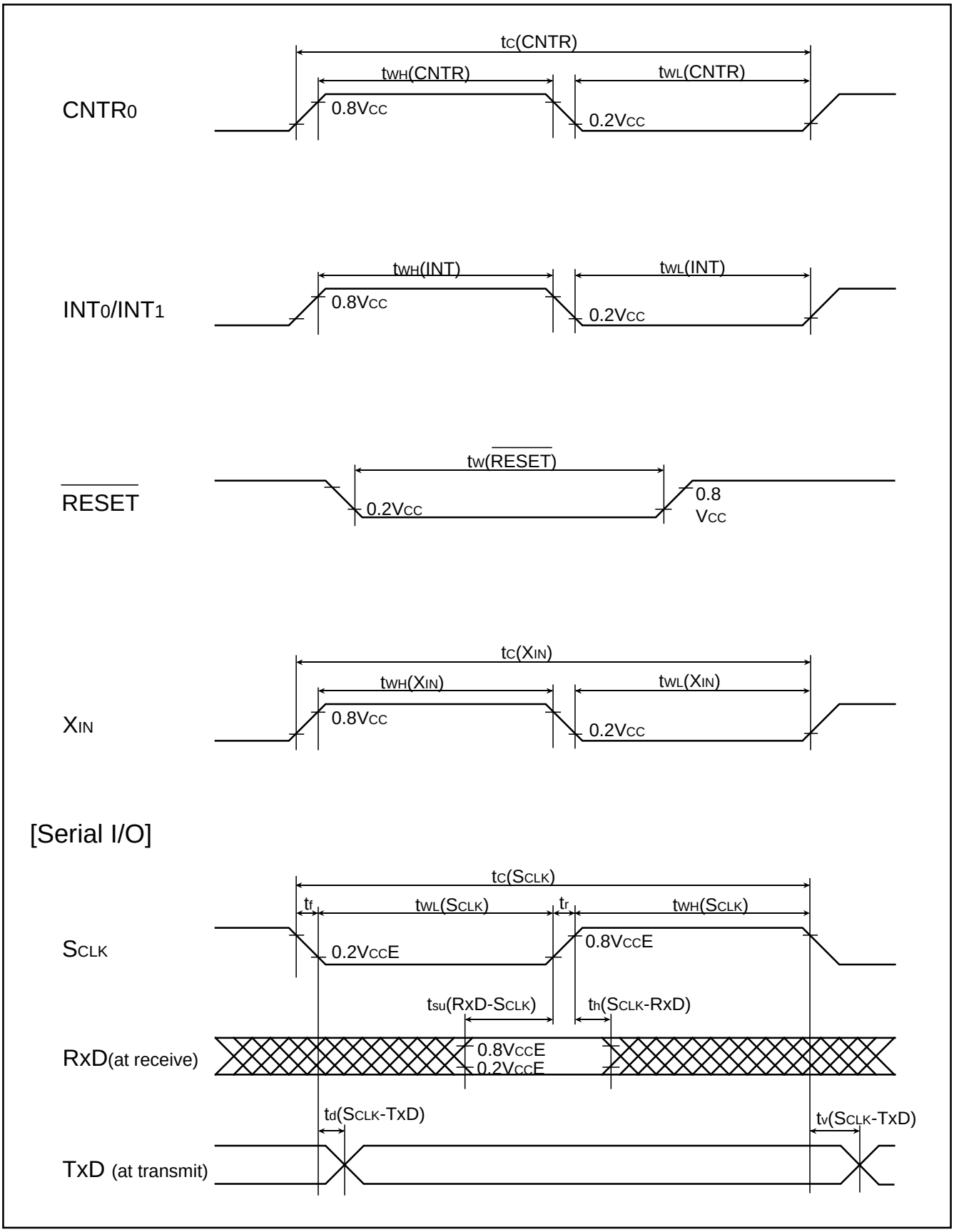
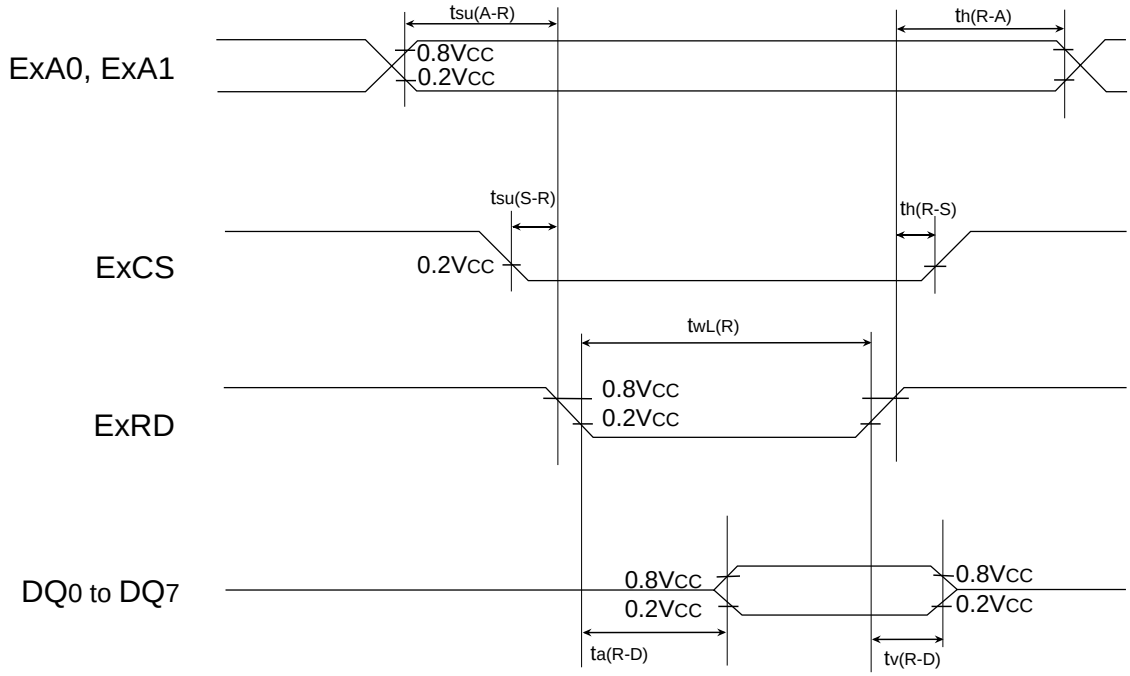


Fig. 141 Timing chart (1)

● Timing chart

[EXB <CPU channel mode>]

< Read >



< Write >

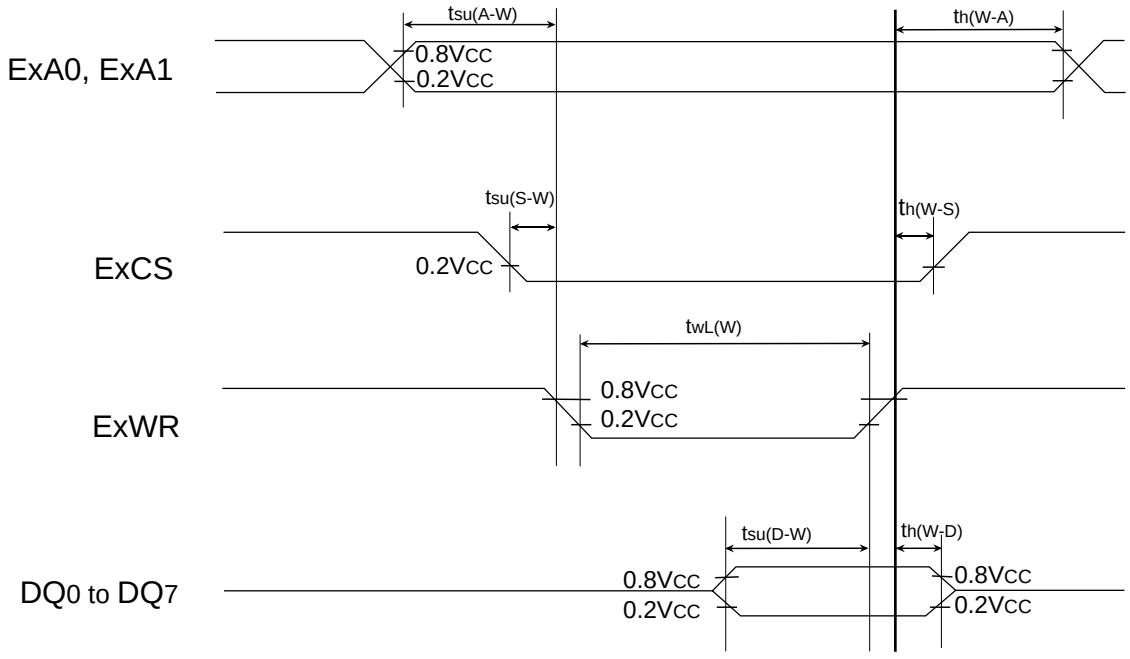
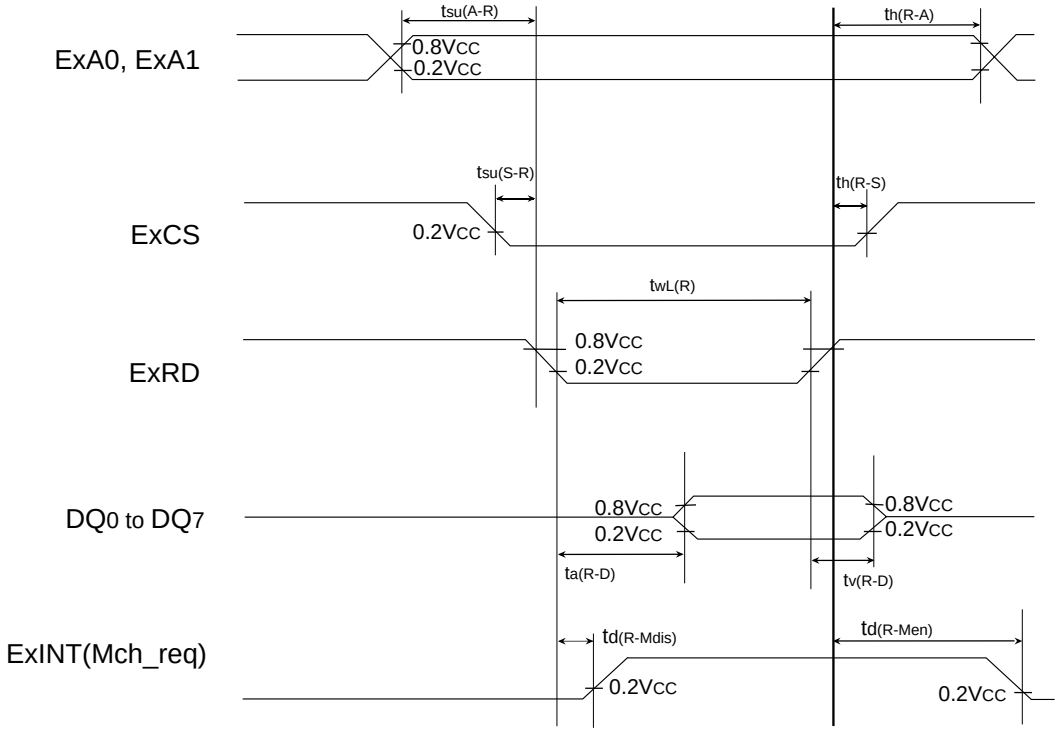


Fig. 142 Timing chart (2)

● Timing chart

[EXB <Memory channel mode, Normal port function>]

< Read >



< Write >

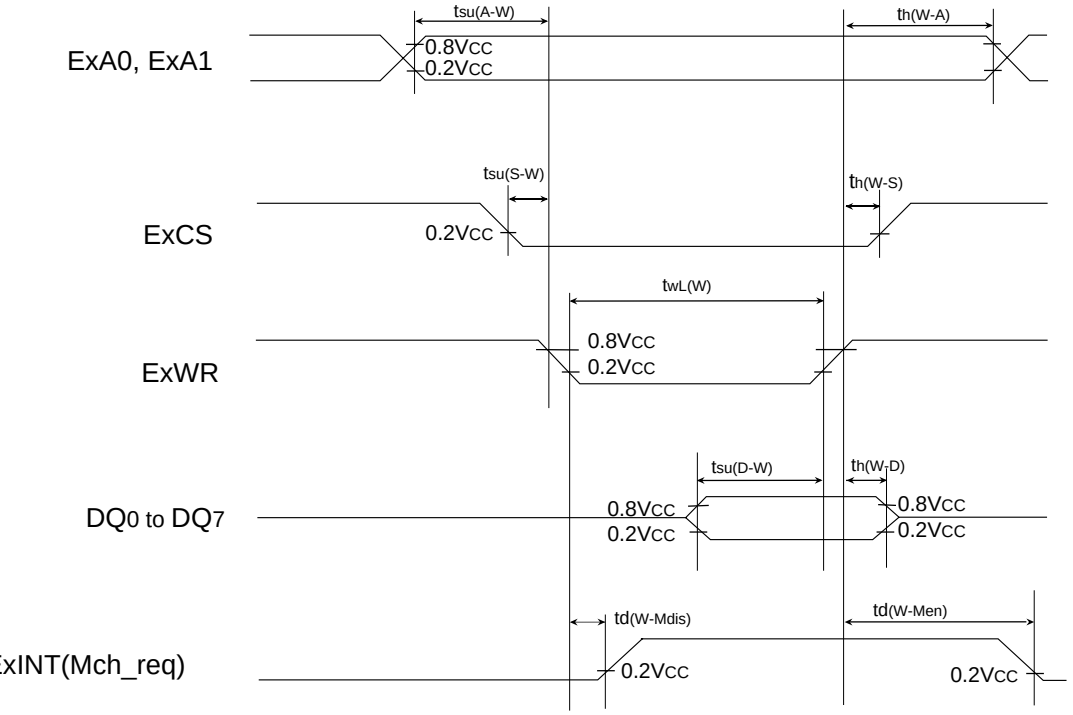
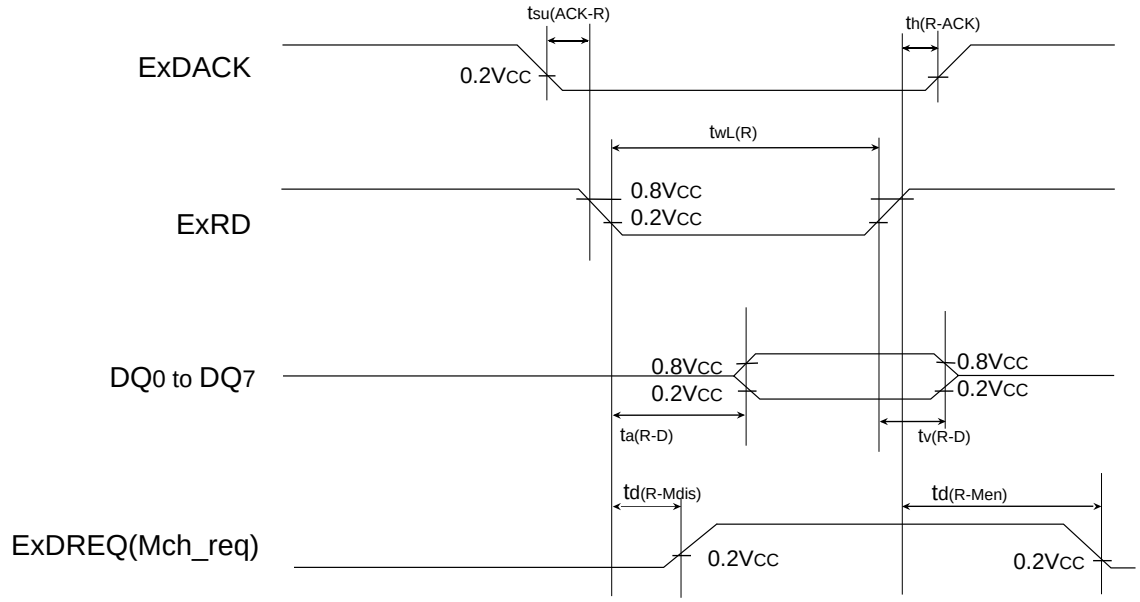


Fig. 143 Timing chart (3)

● Timing chart

[EXB <Memory channel mode, DMA interface pin function,
Read and write signals used together mode>]

< Read >



< Write >

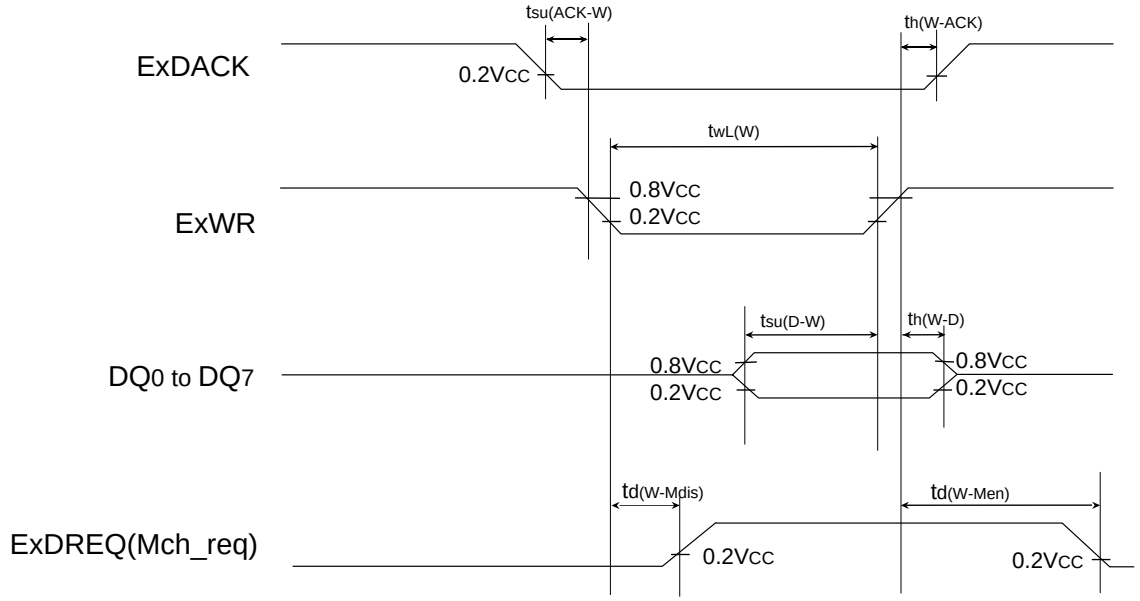
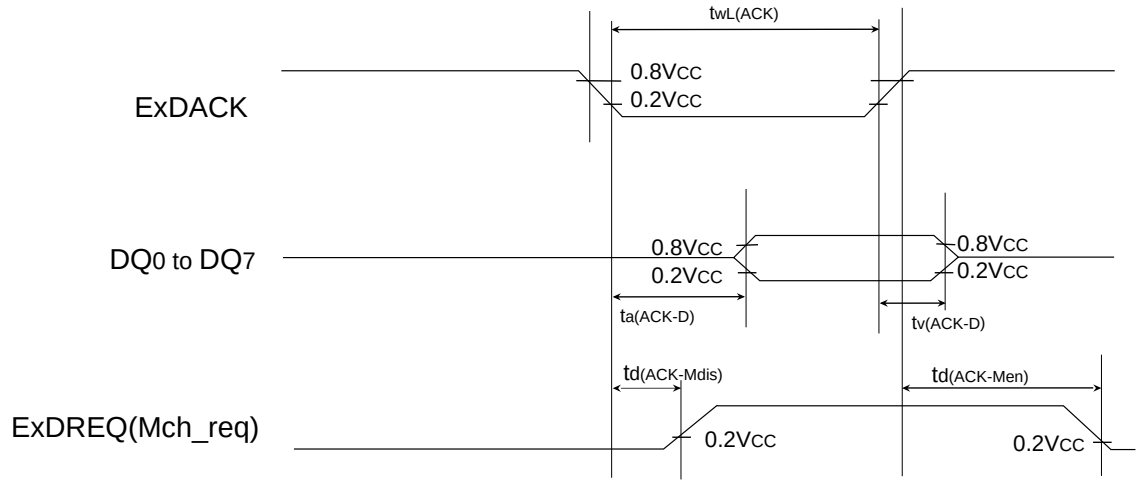


Fig. 144 Timing chart (4)

● Timing chart

[EXB <Memory channel mode, DMA interface pin function,
Read and write signals not required mode>]

< Read >



< Write >

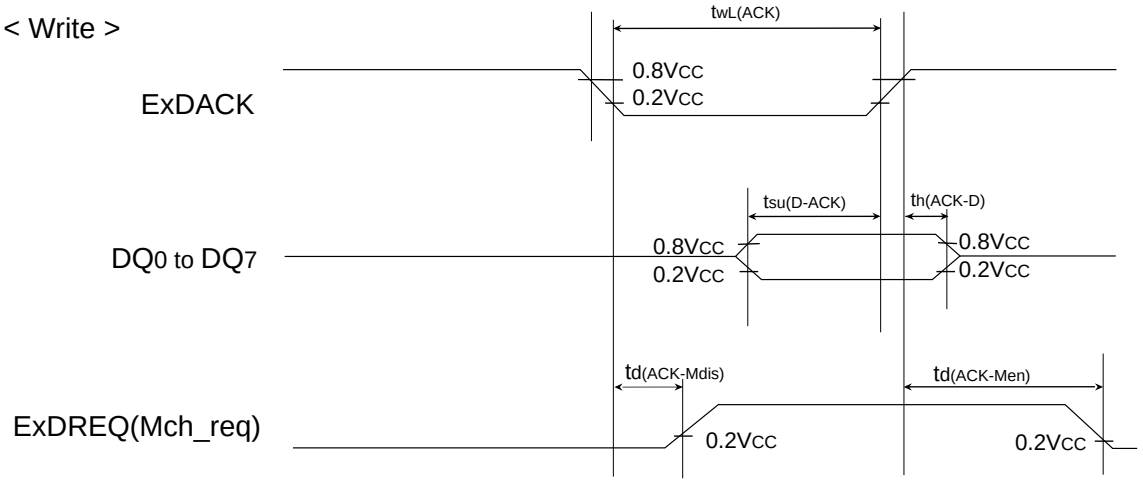
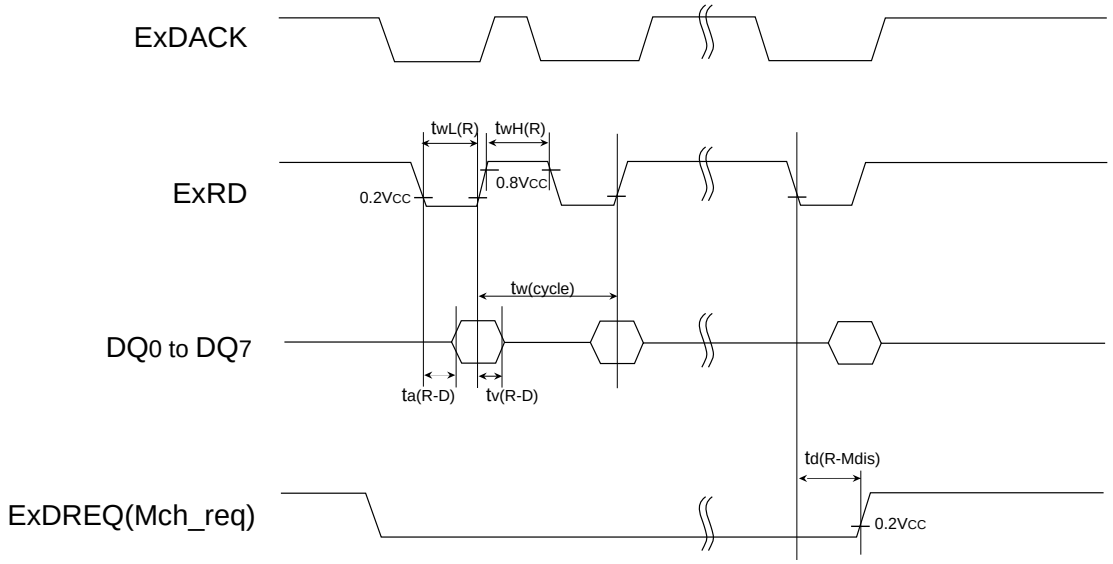


Fig. 145 Timing chart (5)

● Timing chart

[EXB <Memory channel mode, Burst transfer>]

< Read >



< Write >

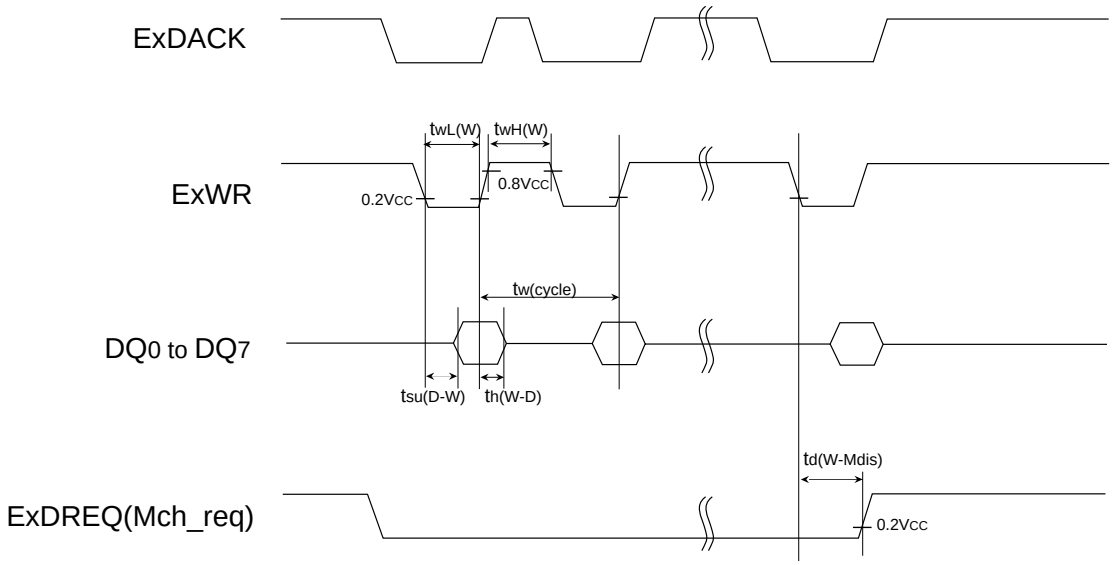
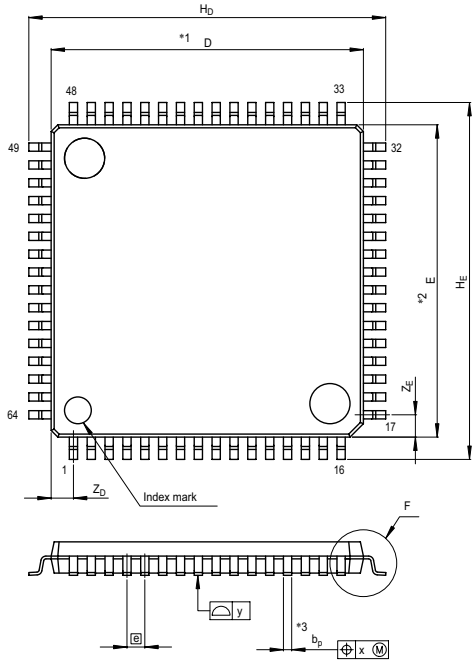


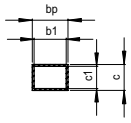
Fig. 146 Timing chart (6)

PACKAGE OUTLINE

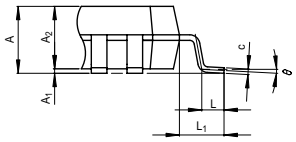
PLQP0064GA-A

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP64-14x14-0.80	PLQP0064GA-A	64P6U-A	0.7g





Terminal cross section



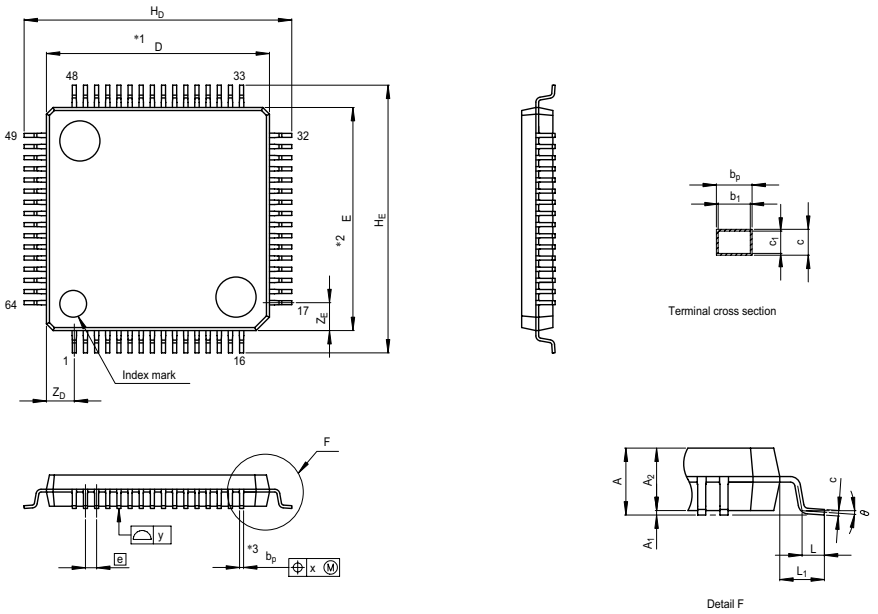
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.3	0.5	0.7
L_1	—	1.0	—

PLQP0064KB-A

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP64-10x10-0.50	PLQP0064KB-A	64P6Q-A / FP-64K / FP-64KV	0.3g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

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.

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(ϕ) 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.
2.1	10/09/02	P. 1	FEATURES: USB specification ver.1.1→ Full-Speed USB2.0 specification Power source voltage: Standard and L version are indicated respectively. Power dissipation(at 3.3V): 45mW→ 30mW 2. of Notes is deleted. Fig. 1: Product name of L version is added.
		P. 3	Table 1: Vcc and USBREF are revised.
		P. 5	Fig. 4: L version is added. Table 3 is added. Subsequent table numbers are changed.
		P. 16	Table 7: Key-on wake up is revised.
		P.25	4th line: USB specification ver.1.1→ USB2.0 specification, Ver1.1→ USB1.1 specification
		P. 27	Fig. 25: Tytle: Vcc condition is added. Capacitors are added. Fig. 26 is added. Subsequent figure numbers are changed.
		P. 31	Fig. 30: EP00 transmit/receive byte number register→EP00 byte number register
		P. 38	Fig. 44: EP00 transmit/receive byte number register→EP00 byte number register
		P. 75	Power source voltage and reset input voltage of Standard and L version are indicated. Fig. 107:Reset release voltage of Standard and L version are indicated.
		P. 81	Table 9: Vcc of Standard and L version are indicated.
		P. 94	Table 12:Vcc of Standard and L version are indicated. I/O of USBVREF: Input → empty, I/O of TrON: Empty → Output
		P. 95	Fig.124: Product name of L version is added. Vcc when connect to Vpp is revised.

Rev.	Date	Description	
		Page	Summary
2.1	10/09/02	P. 111 --	Electrical characteristics of Standard and L version are indicated respectively (Those of L version are indicated from page 119). Table 17 to Tbale 21: Tytle: Vcc=3.00 to 5.25 V →Vcc=4.00 to 5.25 V, Vcc when system clock 6MHz is deleted, 2,4-divide mode when system clock 12MHz is deleted.
		P. 111	Table 7: Vcc when system clock 8MHz→Vcc when system clock ≤ 8MHz USBVREF is deleted.
		P. 112	Table 18:Vcc conditions of f(XIN), f(XIN) or f(SYN), and f(ϕ) are deleted. Data when Vcc=3.00 to 4.00 V is deleted. Notes 4 is revised.
		P. 113	Table 19: Indications of Vcc and VccE in Test conditions are deleted. Data when VI=VSS in IIL pull-up is deleted.
		P. 114	Table 20: Ranges of Vcc in Test conditions are deleted. Data when Vcc= 3.00 to 4.00 V is deleted. Typ. in normal mode are revised.
		P. 116	Timing requirements table when Vcc= 3.00 to 4.00 V is deleted. Table 23 is added.
		P. 117	Switching characteristics table when Vcc= 3.00 to 4.00 V is deleted. Table 25 is added.
		P. 118	Table 26: Tytle: Vcc=3.00 to 5.25 V→Vcc 4.00 to 5.25V
		P. 119 --	Electrical characteristics and switching characteristics of L version are added.
		P. 131 --	Timing chart (2) to (6) are added.
3.0	10/05/06	All pages	Package names “64P6U-A” → “PLQP0064GA-A” revised Package names “64P6Q-A” → “PLQP0064KB-A” revised
		P. 78	CLOCK GENERATING CIRCUIT; “No external resistor is needed .... resistor exists on-chip.” → “No external resistor is needed .... depending on conditions.)
		P. 79	Fig. 111; Pulled up added, NOTE added
		P. 109	NOTES ON USAGE; Power Source Voltage, USB Communication added
		P. 136,137	PACKAGE OUTLINE revised

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

Keep safety first in your circuit designs!
1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials
1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors.
Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

© 2006. Renesas Technology Corp., All rights reserved. Printed in Japan.

Colophon .6.0

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View M38K09F8LHP on WIN SOURCE
- Renesas Electronics America Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management