REJ09B0338-0200

RENESAS

8

38K2 Group

User's Manual

RENESAS 8-BIT SINGLE-CHIP MICROCOMPUTER

740 FAMILY / 38000 SERIES

User's Manual

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Technology Corp. without notice. Please review the latest information published by Renesas Technology Corp. through various means, including the Renesas Technology Corp. website (http://www.renesas.com).

Rev. 2.00
Revision date: Oct 15, 2006

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

BEFORE USING THIS MANUAL

This user's manual consists of the following three chapters. Refer to the chapter appropriate to your conditions, such as hardware design or software development. Chapter 3 also includes necessary information for systems development. You must refer to that chapter.

1. Organization

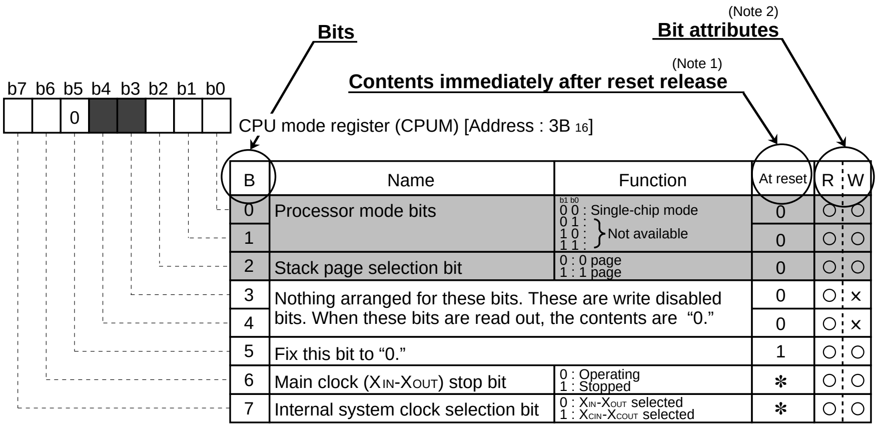
● **CHAPTER 1 HARDWARE**
This chapter describes features of the microcomputer and operation of each peripheral function.

● **CHAPTER 2 APPLICATION**
This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of relevant registers.

● **CHAPTER 3 APPENDIX**
This chapter includes necessary information for systems development using the microcomputer, such as the electrical characteristics, the list of registers.

2. Structure of register

The figure of each register structure describes its functions, contents at reset, and attributes as follows :


(Note 2)
Bit attributes
Bits
(Note 1)
Contents immediately after reset release
b7 b6 b5 b4 b3 b2 b1 b0
0
CPU mode register (CPUM) [Address : 3B 16]

B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : 1 0 : } Not available 1 1 :	0	○	○
1			0	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Nothing arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0."		0	○	×
4			0	○	×
5	Fix this bit to "0."		1	○	○
6	Main clock (X_{IN}-X_{OUT}) stop bit	0 : Operating 1 : Stopped	✻	○	○
7	Internal system clock selection bit	0 : X_{IN}-X_{OUT} selected 1 : X_{CIN}-X_{COUT} selected	✻	○	○

■ : Bit in which nothing is arranged

▨ : Bit that is not used for control of the corresponding function

Note 1:. Contents immediately after reset release
0....... "0" at reset release
1....... "1" at reset release
?....... Undefined at reset release
✻.......Contents determined by option at reset release

Note 2: Bit attributes......... The attributes of control register bits are classified into 3 bytes : read-only, write-only and read and write. In the figure, these attributes are represented as follows :

R.......Read	W......Write
○...... Read enabled	○ ..... Write enabled
×.......Read disabled	×...... Write disabled
	✻......."0" write

3. Supplementation

For details of software, refer to the "740 FAMILY SOFTWARE MANUAL."
For details of development support tools, refer to the "Renesas Technology" Homepage (http://www.renesas.com).

Table of contents

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of figures

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

List of tables

CHAPTER 1 HARDWARE

CHAPTER 2 APPLICATION

CHAPTER 3 APPENDIX

THIS PAGE IS BLANK FOR REASONS OF LAYOUT.

CHAPTER 1

HARDWARE

DESCRIPTION

The 38K2 group is the 8-bit microcomputer based on the 740 family core technology.

The 38K2 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.

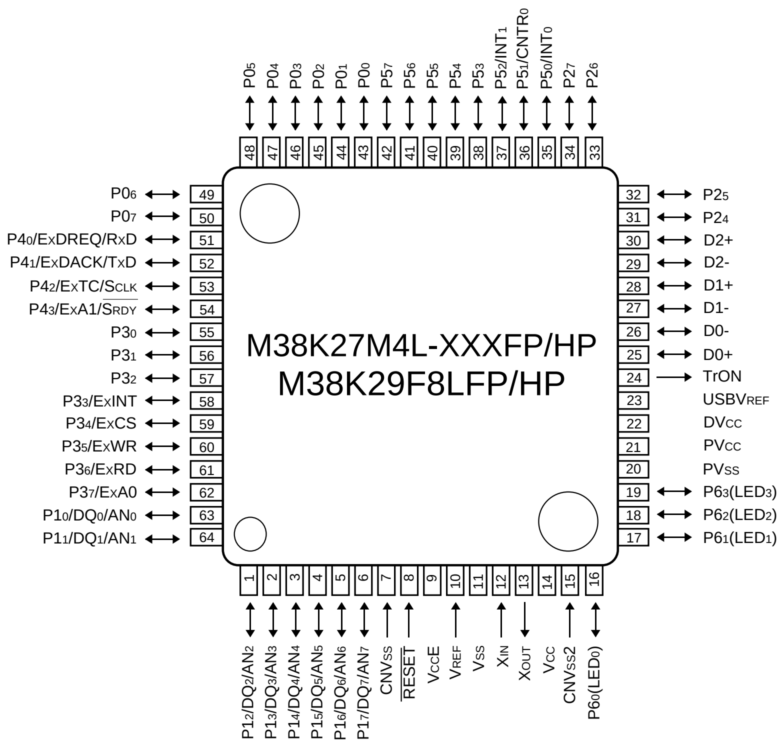
The various microcomputers in the 38K2 group include variations of internal memory size and packaging. For details, refer to the section on part numbering.

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 44
- Software pull-up resistors
- Interrupts .... 16 sources, 16 vectors
- USB function (Full-Speed USB2.0 specification) ...... 4 endpoints
- USB HUB function (Full-Speed USB2.0 specification) .... 2 down ports
- External bus interface .... 8-bit × 1 channel
- Timers .... 8-bit × 3
- Watchdog timer .... 16-bit × 1
- Serial Interface
  Serial I/O .... 8-bit × 1 (UART or Clock-synchronized)
- 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 38K2 group

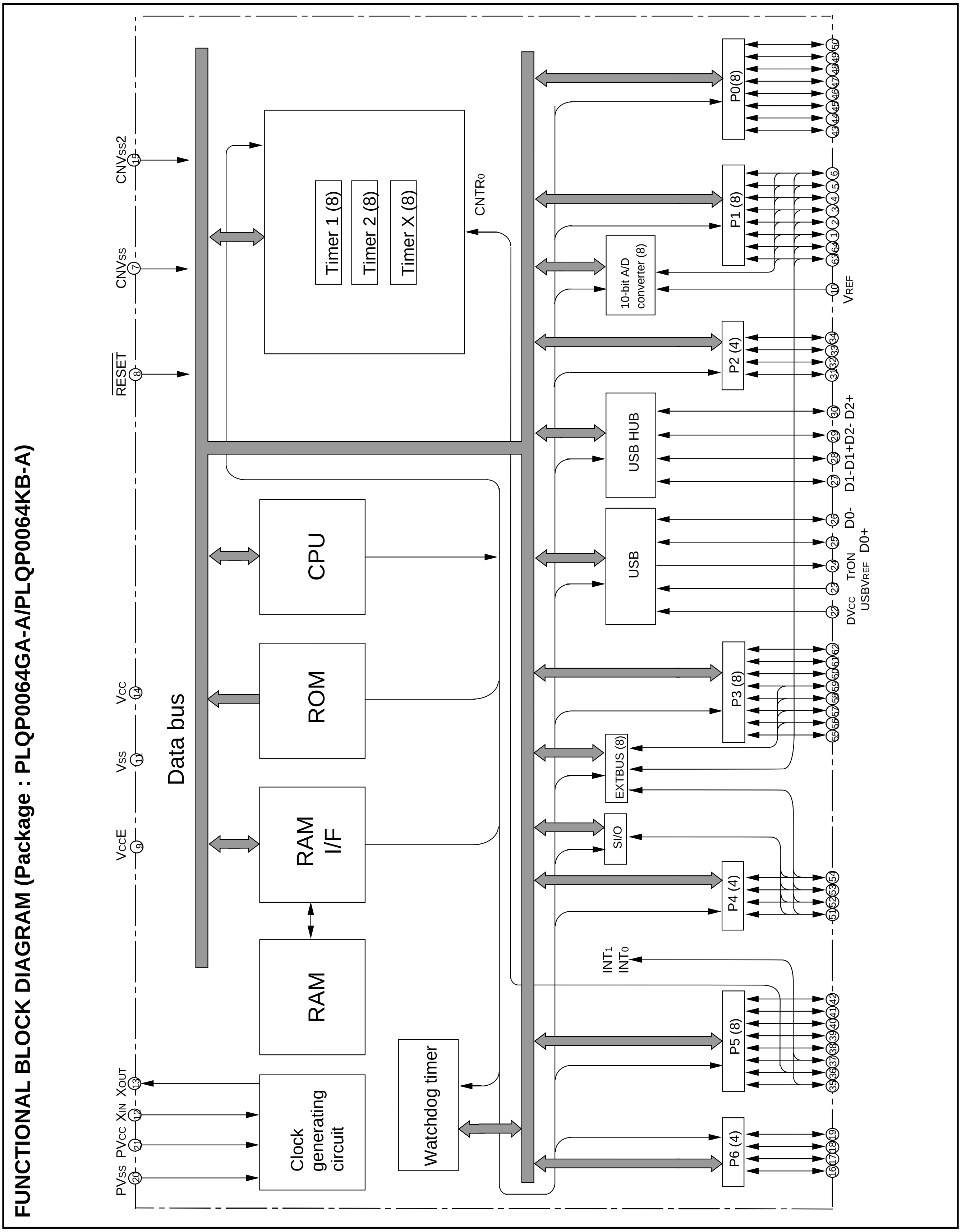


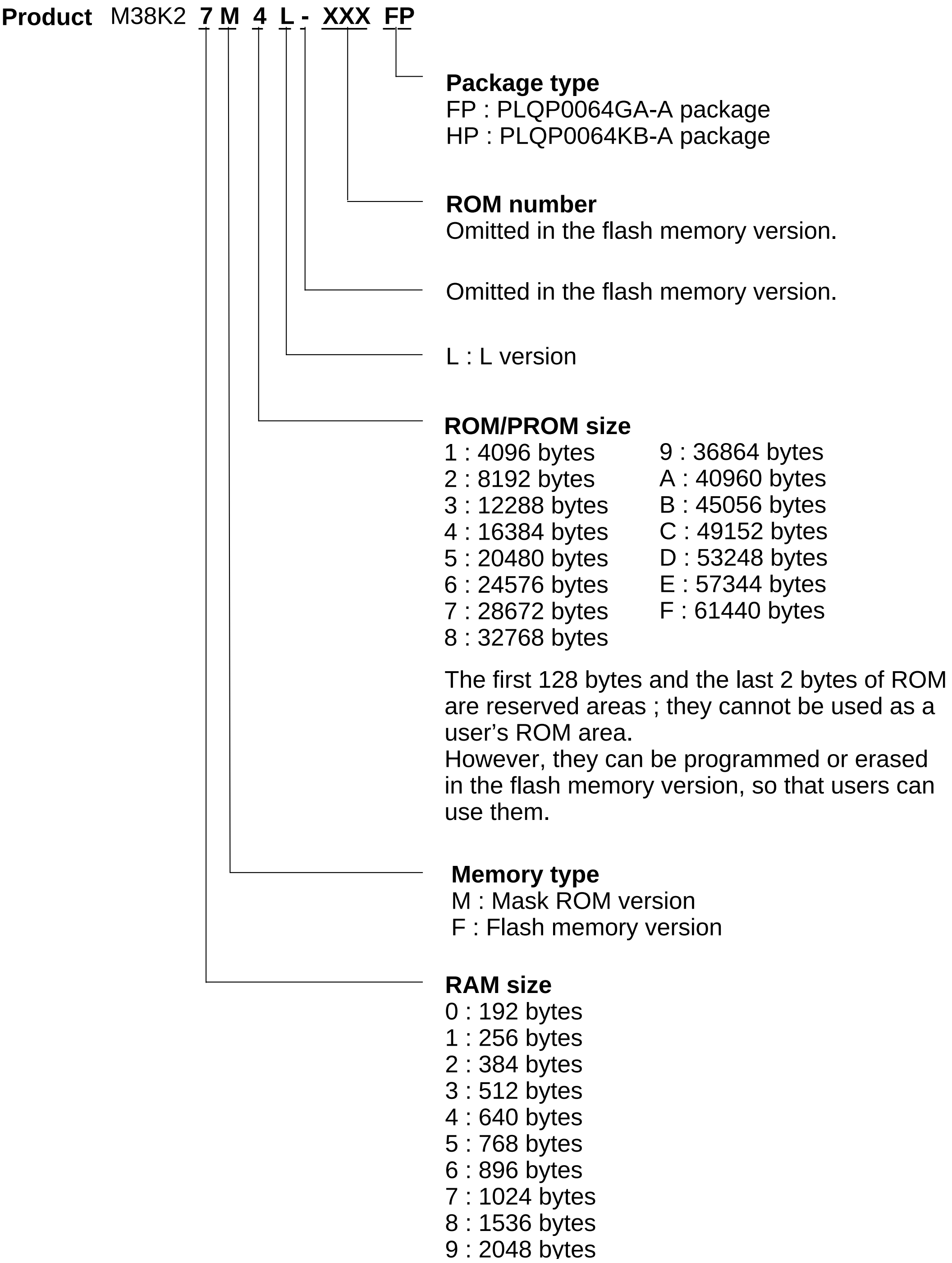
Fig. 2 Functional block diagram

PIN DESCRIPTION

Table 1. Pin description

Pin	Name	Function	Function except a port function
VCC, VSS	Power source	• Apply voltage of 3.0 V – 5.25 V (L version) to VCC, and 0 V to VSS.	
VCCE	Analog power source	• Power source pin for ports P1, P3, P4 and analog circuit. Connect this pin to VCC.	
CNVSS	CNVSS	• This pin controls the operation mode of the chip. Connect this pin to VSS. In the flash memory mode, this pin becoems VPP power source input pin.	
CNVSS2	CNVSS2	• This pin controls the operation mode of the chip. Connect this pin to VSS.	
VREF	Analog reference voltage input	• Reference voltage input pin for A/D converter.	
DVCC PVCC, PVSS	Analog power source	• Power source pin for analog circuit. • Connect the DVCC and PVCC pins to VCC, and the PVSS pin to VSS.	
$\overline{\text{RESET}}$	Reset input	• Reset input pin for active “L”	
XIN	Clock input	• Input and output pins for the main clock generating circuit. • Connect a ceramic resonator or a quartz-crystal oscillator between the XIN and XOUT pins to set the oscillation frequency. •If an external clock is used, connect the clock source to the XIN pin and leave the XOUT pin open.	
XOUT	Clock output		
USBVREF	USB reference power source	• Power source pin for USB port circuit. In Vcc = 4.00 to 5.25 V use the built-in USB reference voltage circuit. In Vcc = 3.60 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 VCC 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	
D1+, D1-, D2+, D2-	USB down-stream I/O	• USB downstream I/O port • USB input level • USB output level output structure	
P00–P07	I/O port P0	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure • Pull-up control is enabled.	• Key input pins (key-on wake up interrupt)
P10/DQ0/AN0– P17/DQ7/AN7	I/O port P1	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	• A/D converter input pins • External bus interface function pins
P24–P27	I/O port P2	• 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	
P30–P32	I/O port P3	• 8-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	
P33/ExINT P34/ExCS P35/ExWR P36/ExRD P37/ExA0			• External bus interface function pins
P40/ExDREQ/RxD P41/ExDACK/TxD P42/ExTC/SCLK P43/ExA1/SRDY	I/O port P4	• 4-bit I/O port • I/O direction register allows each pin to be individually programmed as either input or output. • CMOS compatible input level • CMOS 3-state output structure	• Serial I/O function pins • External bus interface function pins
P50/INT0	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
P51/CNTR0			• Timer X funciton pin
P52/INT1			• Interrupt input pin
P53–P57			
P60–P63	I/O port P6	• 4-bit I/O port; • I/O direction register allows each pin to be individually programmed as either input or output.; • CMOS compatible input level• CMOS 3-state output structure; • Output large current for LED drive is enabled.	

PART NUMBERING



Product M38K2 7 M 4 L - XXX FP

Package type
FP : PLQP0064GA-A package
HP : PLQP0064KB-A package

ROM number
Omitted in the flash memory version.

Omitted in the flash memory version.

L : L version

ROM/PROM size

1 : 4096 bytes
2 : 8192 bytes
3 : 12288 bytes
4 : 16384 bytes
5 : 20480 bytes
6 : 24576 bytes
7 : 28672 bytes
8 : 32768 bytes
9 : 36864 bytes
A : 40960 bytes
B : 45056 bytes
C : 49152 bytes
D : 53248 bytes
E : 57344 bytes
F : 61440 bytes

The first 128 bytes and the last 2 bytes of ROM are reserved areas ; they cannot be used as a user’s ROM area.
However, they can be programmed or erased in the flash memory version, so that users can use them.

Memory type
M : Mask ROM version
F : Flash memory version

RAM size
0 : 192 bytes
1 : 256 bytes
2 : 384 bytes
3 : 512 bytes
4 : 640 bytes
5 : 768 bytes
6 : 896 bytes
7 : 1024 bytes
8 : 1536 bytes
9 : 2048 bytes

Fig. 3 Part numbering

GROUP EXPANSION

Mitsubishi plans to expand the 38K2 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

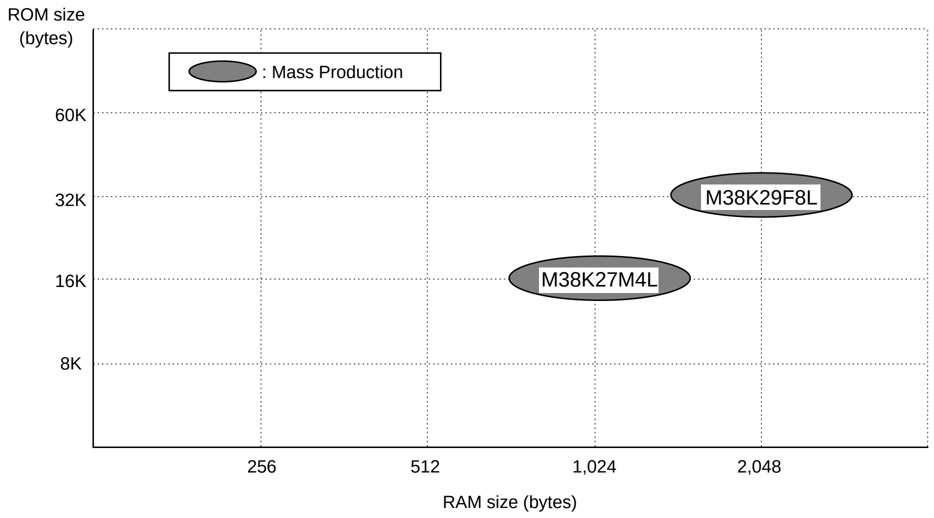


Fig. 4 Memory expansion plan

Currently products are listed below.

Table 2. List of 38K2 group products (L version) **As of October 2006**

Product	ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38K27M4L-XXXFP	16384 (16254)	1024	PLQP0064GA-A	Mask ROM version
M38K27M4L-XXXHP			PLQP0064KB-A	
M38K29F8LFP	32768 (32638)	2048	PLQP0064GA-A	Flash memory version
M38K29F8LHP			PLQP0064KB-A	
M38K29RFS	—	2048	100D0M	

FUNCTIONAL DESCRIPTION
CENTRAL PROCESSING UNIT (CPU)

The 38K2 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 PCH and PCL. It is used to indicate the address of the next instruction to be executed.

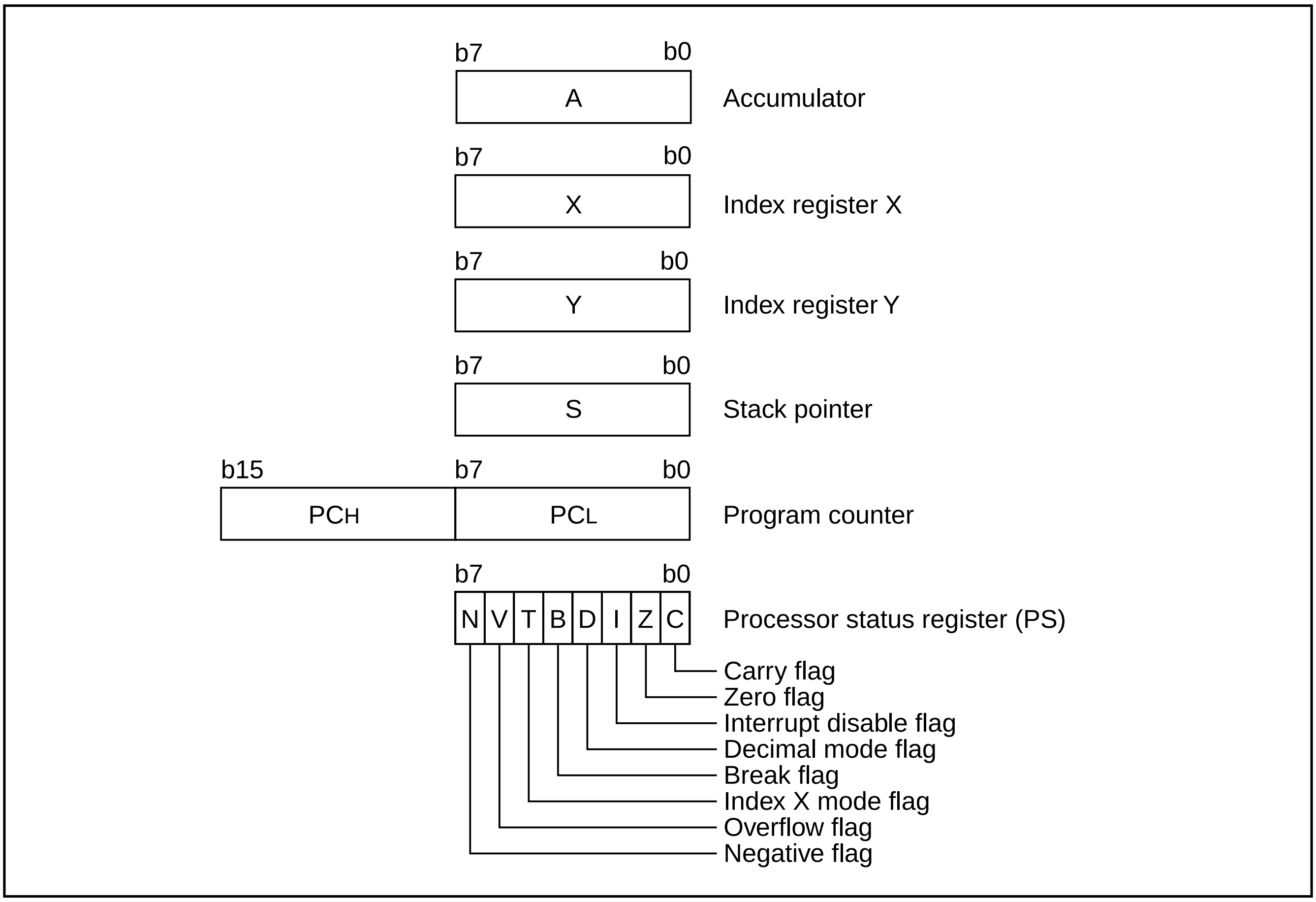


Fig. 5 740 Family CPU register structure

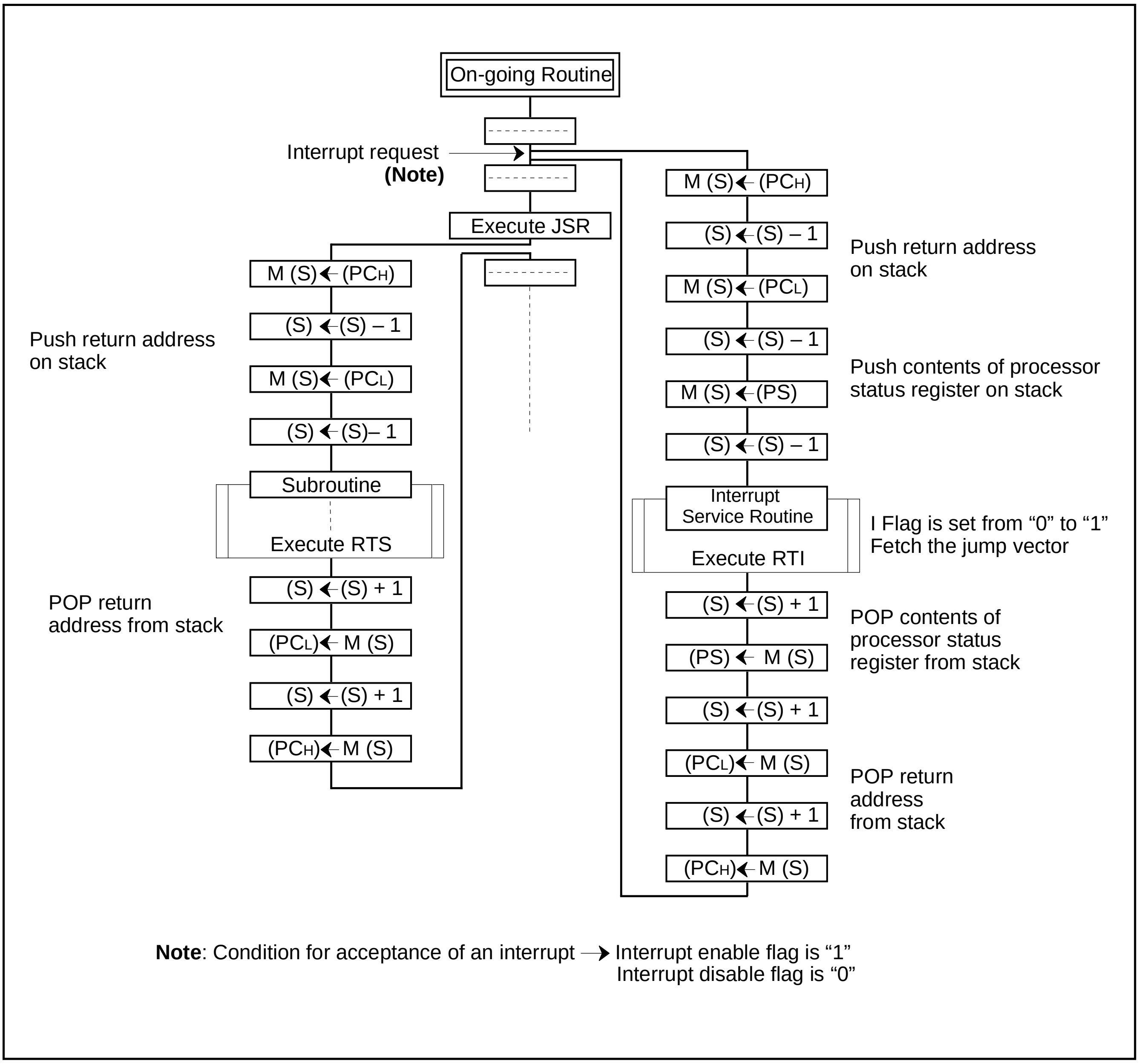

On-going Routine
Interrupt request
(Note)
Execute JSR
M (S) ← (PCH)
(S) ← (S) – 1
M (S) ← (PCL)
(S) ← (S)– 1
Subroutine
Execute RTS
Push return address on stack
POP return address from stack
(S) ← (S) + 1
(PCL) ← M (S)
(S) ← (S) + 1
(PCH) ← M (S)
M (S) ← (PCH)
(S) ← (S) – 1
M (S) ← (PCL)
(S) ← (S) – 1
M (S) ← (PS)
(S) ← (S) – 1
Push return address on stack
Push contents of processor status register on stack
Interrupt Service Routine
Execute RTI
I Flag is set from “0” to “1”
Fetch the jump vector
(S) ← (S) + 1
(PS) ← M (S)
(S) ← (S) + 1
(PCL) ← M (S)
(S) ← (S) + 1
(PCH) ← M (S)
POP contents of processor status register from stack
POP return address from stack
Note: Condition for acceptance of an interrupt → Interrupt enable flag is “1”
Interrupt disable flag is “0”

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 and SBC instructions can execute decimal arithmetic.

•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)] $003B_{16}$

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 $003B_{16}$.

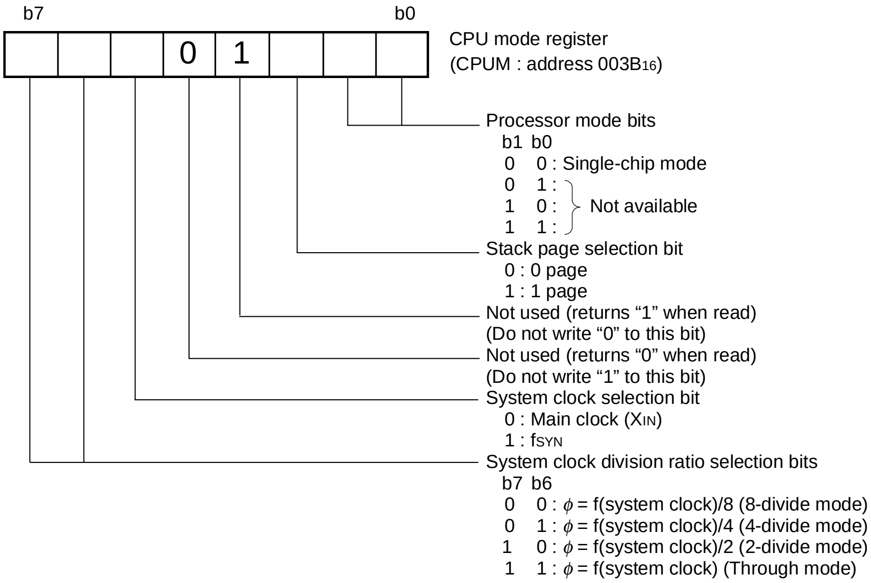

b7
b0
0
1
CPU mode register
(CPUM : address 003B16)
Processor mode bits
b1 b0
0 0 : Single-chip mode
0 1 :
1 0 : Not available
1 1 :
Stack page selection bit
0 : 0 page
1 : 1 page
Not used (returns "1" when read)
(Do not write "0" to this bit)
Not used (returns "0" when read)
(Do not write "1" to this bit)
System clock selection bit
0 : Main clock (XIN)
1 : fSYN
System clock division ratio selection bits
b7 b6
0 0 : φ = f(system clock)/8 (8-divide mode)
0 1 : φ = f(system clock)/4 (4-divide mode)
1 0 : φ = f(system clock)/2 (2-divide mode)
1 1 : φ = f(system clock) (Through mode)

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$_{16}$

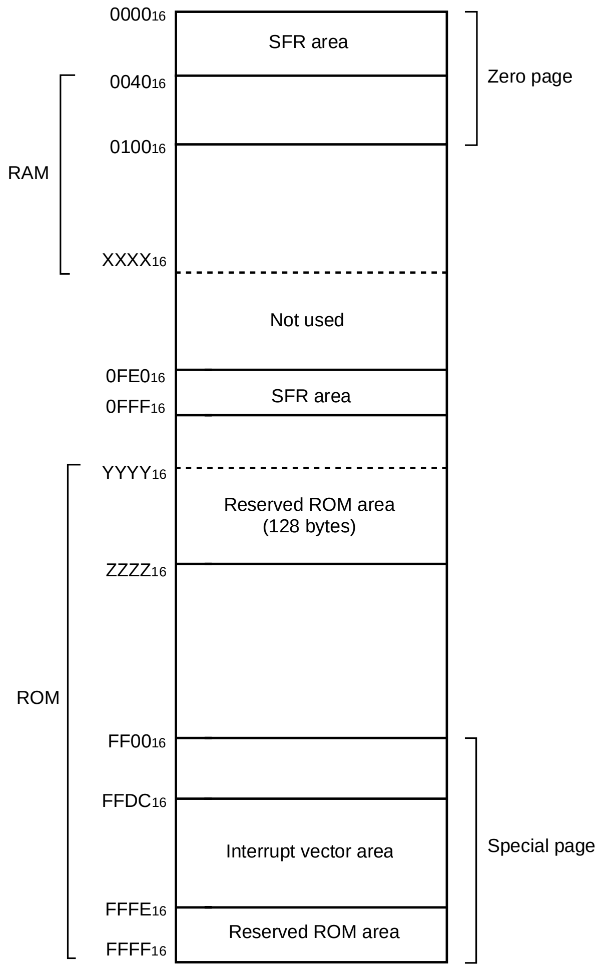


Fig. 8 Memory map diagram

0000_{16}	Port P0 (P0)	0020_{16}	Prescaler 12 (PRE12)
0001_{16}	Port P0 direction register (P0D)	0021_{16}	Timer 1 (T1)
0002_{16}	Port P1 (P1)	0022_{16}	Timer 2 (T2)
0003_{16}	Port P1 direction register (P1D)	0023_{16}	Timer X mode register (TM)
0004_{16}	Port P2 (P2)	0024_{16}	Prescaler X (PREX)
0005_{16}	Port P2 direction register (P2D)	0025_{16}	Timer X (TX)
0006_{16}	Port P3 (P3)	0026_{16}	Transmit/Receive buffer register (TB/RB)
0007_{16}	Port P3 direction register (P3D)	0027_{16}	Serial I/O status register (SIOSTS)
0008_{16}	Port P4 (P4)	0028_{16}	HUB interrupt source enable register (HUBICON)
0009_{16}	Port P4 direction register (P4D)	0029_{16}	HUB interrupt source register (HUBIREQ)
$000A_{16}$	Port P5 (P5)	$002A_{16}$	HUB down stream port index register (HUBINDEX)
$000B_{16}$	Port P5 direction register (P5D)	$002B_{16}$	HUB port field register 1 (DPXREG1)
$000C_{16}$	Port P6 (P6)	$002C_{16}$	HUB port field register 2 (DPXREG2)
$000D_{16}$	Port P6 direction register (P6D)	$002D_{16}$	HUB port field register 3 (DPXREG3)
$000E_{16}$	Reserved **(Note)**	$002E_{16}$	Reserved **(Note)**
$000F_{16}$	Reserved **(Note)**	$002F_{16}$	Reserved **(Note)**
0010_{16}	USB control register (USBCON)	0030_{16}	EXB interrupt source enable register (EXBICON)
0011_{16}	USB function/Hub enable register (USBAE)	0031_{16}	EXB interrupt source register (EXBIREQ)
0012_{16}	USB function address register (USBA0)	0032_{16}	Reserved **(Note)**
0013_{16}	USB HUB address register (USBA1)	0033_{16}	EXB index register (EXBINDEX)
0014_{16}	Frame number register Low (FNUML)	0034_{16}	Register window 1 (EXBREG1)
0015_{16}	Frame number register High (FNUMH)	0035_{16}	Register window 2 (EXBREG2)
0016_{16}	USB interrupt source enable register (USBICON)	0036_{16}	AD control register (ADCON)
0017_{16}	USB interrupt source register (USBIREQ)	0037_{16}	AD conversion register 1 (AD1)
0018_{16}	Endpoint index register (USBINDEX)	0038_{16}	AD conversion register 2 (AD2)
0019_{16}	Endpoint field register 1 (EPXXREG1)	0039_{16}	Watchdog timer control register (WDTCON)
$001A_{16}$	Endpoint field register 2 (EPXXREG2)	$003A_{16}$	Reserved **(Note)**
$001B_{16}$	Endpoint field register 3 (EPXXREG3)	$003B_{16}$	CPU mode register (CPUM)
$001C_{16}$	Endpoint field register 4 (EPXXREG4)	$003C_{16}$	Interrupt request register 1(IREQ1)
$001D_{16}$	Endpoint field register 5 (EPXXREG5)	$003D_{16}$	Interrupt request register 2(IREQ2)
$001E_{16}$	Endpoint field register 6 (EPXXREG6)	$003E_{16}$	Interrupt control register 1(ICON1)
$001F_{16}$	Endpoint field register 7 (EPXXREG7)	$003F_{16}$	Interrupt control register 2(ICON2)

$0FE0_{16}$	Serial I/O control register (SIOCON)	$0FF0_{16}$	Port P0 pull-up control register (PULL0)
$0FE1_{16}$	UART control register (UARTCON)	$0FF1_{16}$	Reserved **(Note)**
$0FE2_{16}$	Baud rate generator (BRG)	$0FF2_{16}$	Port P5 pull-up control register (PULL5)
$0FE3_{16}$	Reserved **(Note)**	$0FF3_{16}$	Interrupt edge selection register (INTEDGE)
$0FE4_{16}$	Reserved **(Note)**	$0FF4_{16}$	Reserved **(Note)**
$0FE5_{16}$	Reserved **(Note)**	$0FF5_{16}$	Reserved **(Note)**
$0FE6_{16}$	Reserved **(Note)**	$0FF6_{16}$	Reserved **(Note)**
$0FE7_{16}$	Reserved **(Note)**	$0FF7_{16}$	Reserved **(Note)**
$0FE8_{16}$	Reserved **(Note)**	$0FF8_{16}$	PLL control register (PLLCON)
$0FE9_{16}$	Reserved **(Note)**	$0FF9_{16}$	Downstream port control register (DPCTL)
$0FEA_{16}$	Reserved **(Note)**	$0FFA_{16}$	Reserved **(Note)**
$0FEB_{16}$	Reserved **(Note)**	$0FFB_{16}$	MISRG
$0FEC_{16}$	Endpoint field register 8 (EPXXREG8)	$0FFC_{16}$	Reserved **(Note)**
$0FED_{16}$	Endpoint field register 9 (EPXXREG9)	$0FFD_{16}$	Reserved **(Note)**
$0FEE_{16}$	Reserved **(Note)**	$0FFE_{16}$	Flash memory control register (FMCR)
$0FEF_{16}$	Reserved **(Note)**	$0FFF_{16}$	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.
P00–P07	Port P0	Input/output, individual bits	CMOS compatible input level CMOS 3-state output	Key-on wake up	Port P0 pull-up control register	(1)
P10–P17	Port P1		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)
P24–P27	Port P2		CMOS compatible input level CMOS 3-state output	——	——	(3)
P30–P32	Port P3		CMOS compatible input level CMOS 3-state output (Power source is VccE)	——	——	(4)
P33/ExINT				External bus interface funciton output	EXB control register	(5)
P34/ExCS P35/ExWR P36/ExRD P37/ExA0				External bus interface funciton input	EXB control register	(6)
P40/RxD/ExDREQ	Port P4			Serial I/O input External bus interface funciton output	Serial I/O control register EXB control register	(7)
P41/TxD/ExDACK				Serial I/O output External bus interface funciton input	Serial I/O control register EXB control register	(8)
P42/SCLK/ExTC				Serial I/O I/O External bus interface funciton input	Serial I/O control register EXB control register	(9)
P43/$\overline{\text{SRDY}}$/ExA1				Serial I/O output External bus interface funciton input	Serial I/O control register EXB control register	(10)
P50/INT0 P52/INT1	Port P5		CMOS compatible input level CMOS 3-state output	External interrupt input	Port P5 pull-up control register Interrupt edge selection register	(11)
P51/CNTR0				Timer X function I/O	Timer X mode register	(12)
P53–P57				——	——	(13)
P60–P63	Port P6			——	——	(14)

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

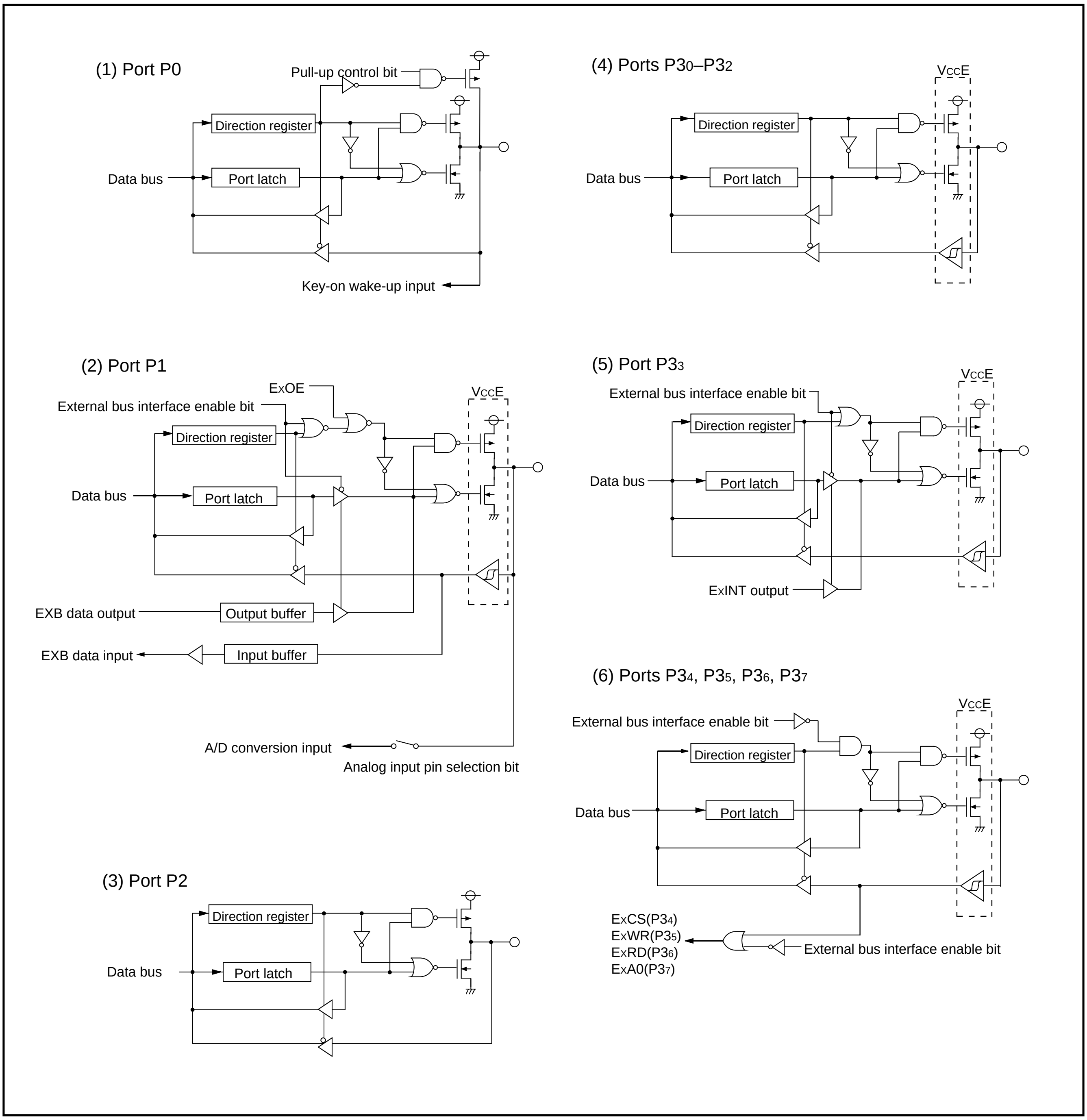


Fig. 10 Port block diagram (1)

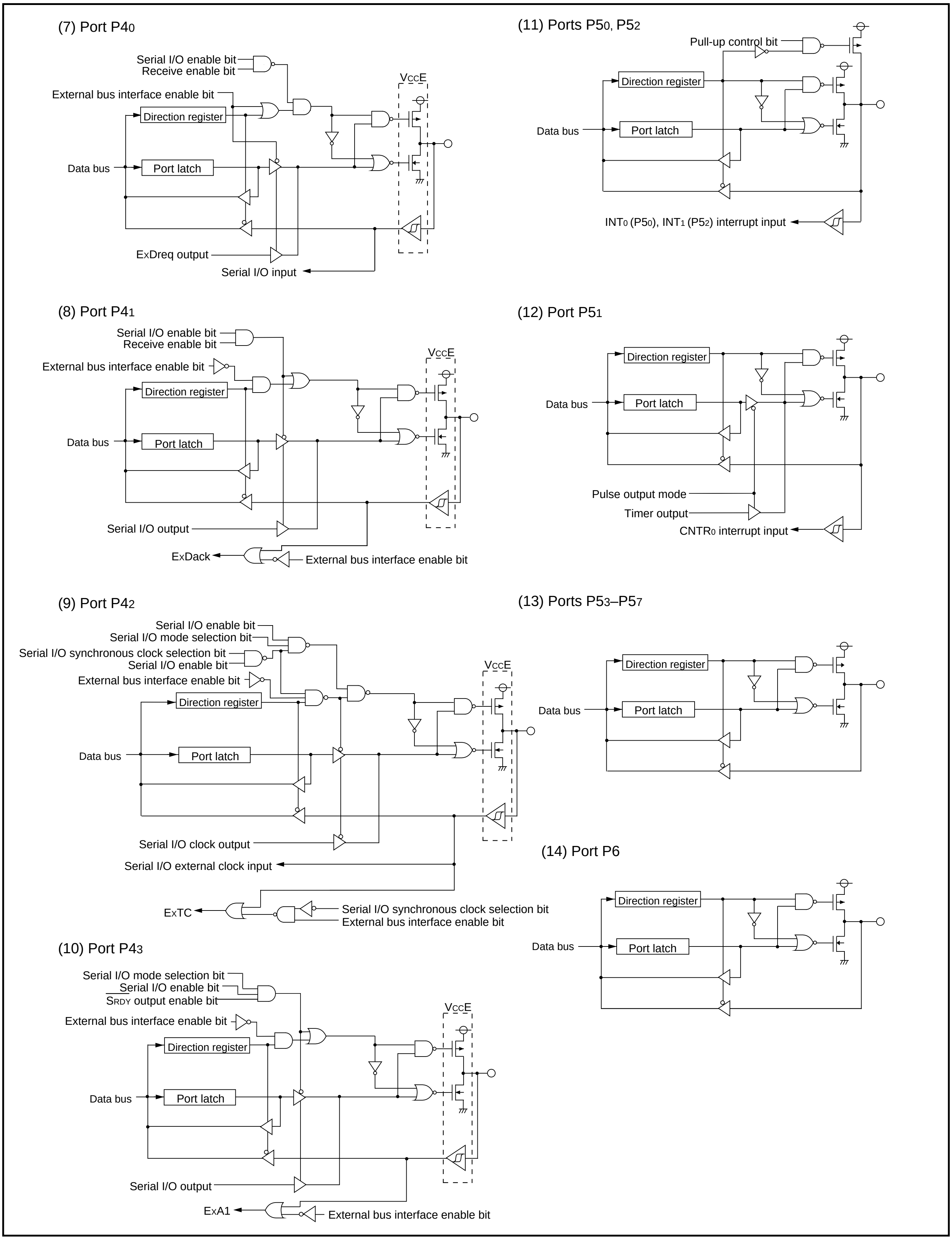


Fig. 11 Port block diagram (2)

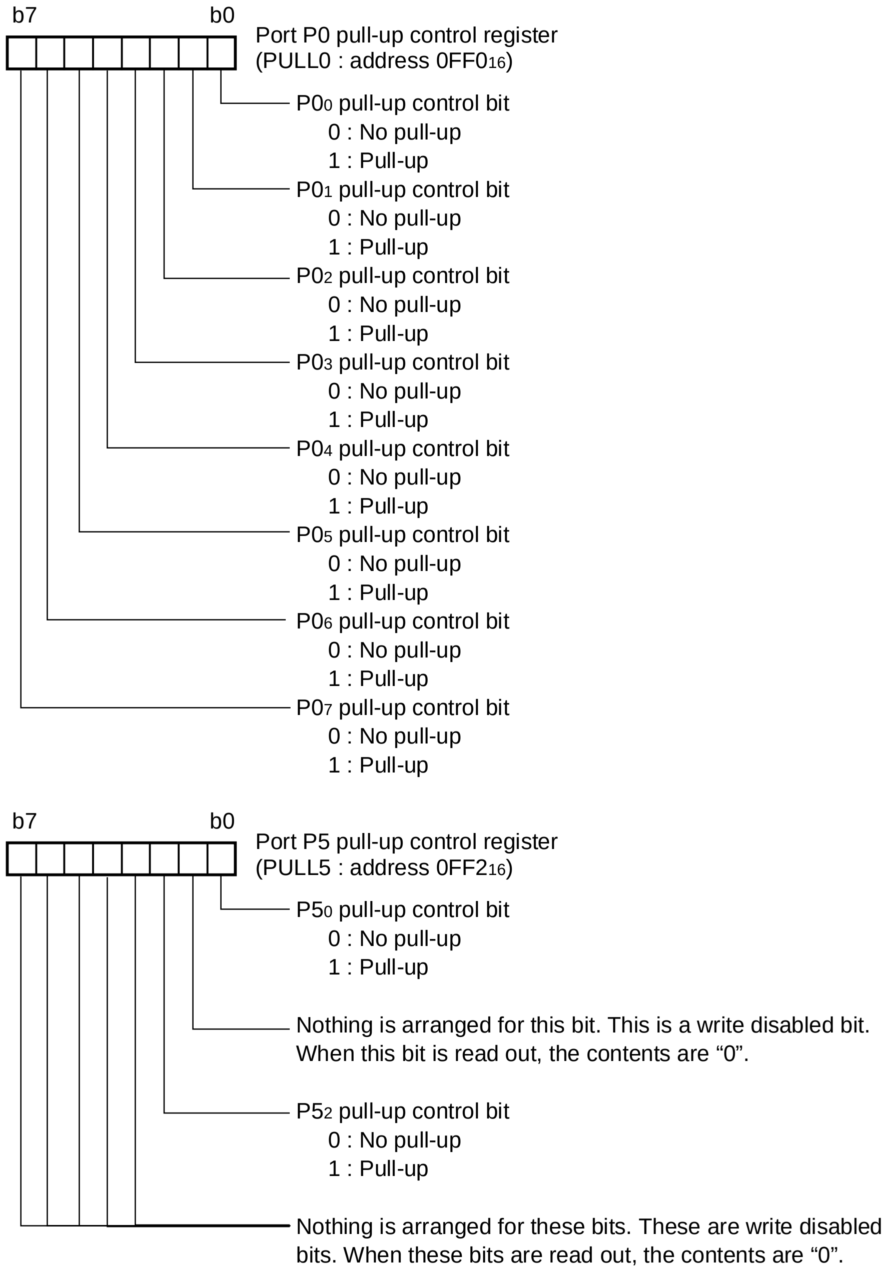

b7
b0
Port P0 pull-up control register
(PULL0 : address 0FF0$_{16}$)
P0$_0$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_1$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_2$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_3$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_4$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_5$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_6$ pull-up control bit
0 : No pull-up
1 : Pull-up
P0$_7$ pull-up control bit
0 : No pull-up
1 : Pull-up
b7
b0
Port P5 pull-up control register
(PULL5 : address 0FF2$_{16}$)
P5$_0$ pull-up control bit
0 : No pull-up
1 : Pull-up
Nothing is arranged for this bit. This is a write disabled bit.
When this bit is read out, the contents are “0”.
P5$_2$ pull-up control bit
0 : No pull-up
1 : Pull-up
Nothing is arranged for these bits. These are write disabled
bits. When these bits are read out, the contents are “0”.

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

INTERRUPTS

Interrupts occur by sixteen sources: four external, eleven 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
USB HUB	11	FFE9$_{16}$	FFE8$_{16}$	At detection of USB HUB downport’s state switch
Serial I/O reception	12	FFE7$_{16}$	FFE6$_{16}$	At completion of serial I/O data reception
Serial I/O transmission	13	FFE5$_{16}$	FFE4$_{16}$	At completion of serial I/O data transmission
CNTR$_0$	14	FFE3$_{16}$	FFE2$_{16}$	At detection of either rising or falling edge of CNTR$_0$ input
Key-on wake up	15	FFE1$_{16}$	FFE0$_{16}$	At falling of conjunction of input level for port P0 (at input mode)
A/D conversion	16	FFDF$_{16}$	FFDE$_{16}$	At completion of A/D conversion
BRK instruction	17	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.

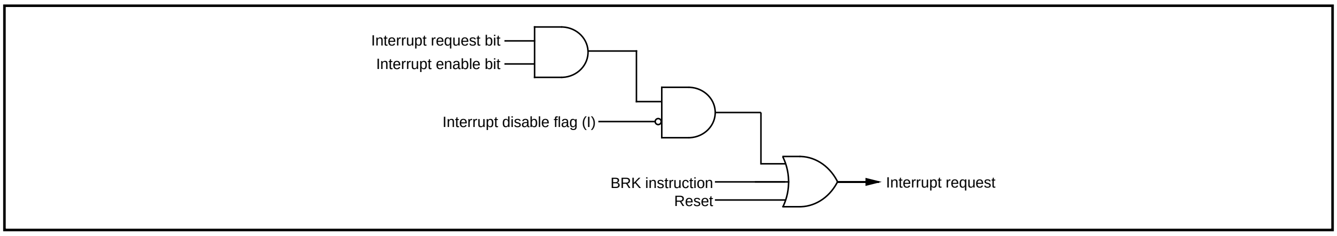
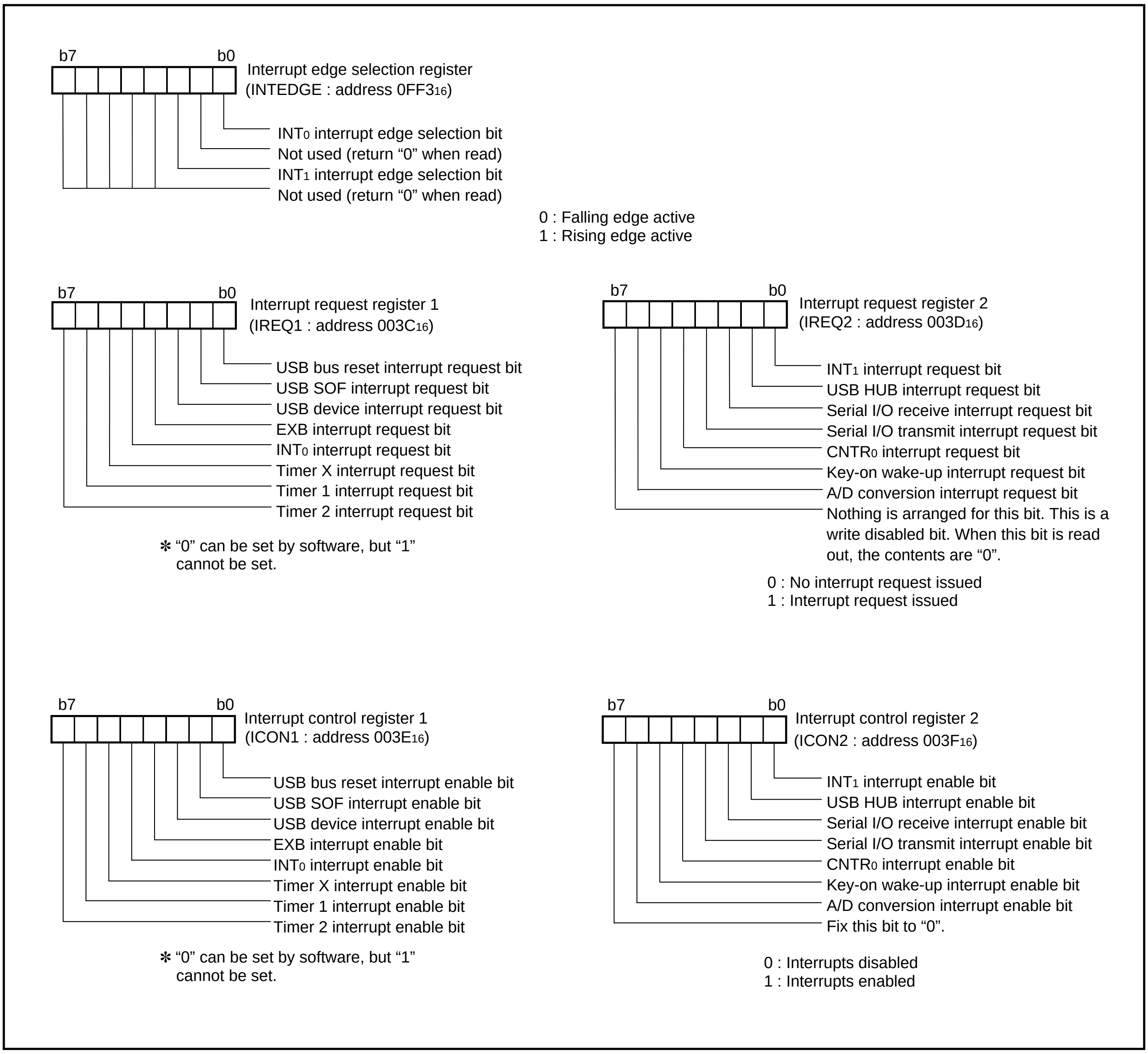


Fig. 13 Interrupt control



b7 b0 Interrupt edge selection register (INTEDGE : address 0FF3$_{16}$)

- INT$_0$ interrupt edge selection bit
- Not used (return “0” when read)
- INT$_1$ interrupt edge selection bit
- Not used (return “0” when read)

0 : Falling edge active
1 : Rising edge active

b7 b0 Interrupt request register 1 (IREQ1 : address 003C$_{16}$)

- USB bus reset interrupt request bit
- USB SOF interrupt request bit
- USB device interrupt request bit
- EXB interrupt request bit
- INT$_0$ interrupt request bit
- Timer X interrupt request bit
- Timer 1 interrupt request bit
- Timer 2 interrupt request bit

✻ “0” can be set by software, but “1” cannot be set.

b7 b0 Interrupt request register 2 (IREQ2 : address 003D$_{16}$)

- INT$_1$ interrupt request bit
- USB HUB interrupt request bit
- Serial I/O receive interrupt request bit
- Serial I/O transmit interrupt request bit
- CNTR$_0$ interrupt request bit
- Key-on wake-up interrupt request bit
- A/D conversion interrupt request bit
- Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.

0 : No interrupt request issued
1 : Interrupt request issued

b7 b0 Interrupt control register 1 (ICON1 : address 003E$_{16}$)

- USB bus reset interrupt enable bit
- USB SOF interrupt enable bit
- USB device interrupt enable bit
- EXB interrupt enable bit
- INT$_0$ interrupt enable bit
- Timer X interrupt enable bit
- Timer 1 interrupt enable bit
- Timer 2 interrupt enable bit

✻ “0” can be set by software, but “1” cannot be set.

b7 b0 Interrupt control register 2 (ICON2 : address 003F$_{16}$)

- INT$_1$ interrupt enable bit
- USB HUB interrupt enable bit
- Serial I/O receive interrupt enable bit
- Serial I/O transmit interrupt enable bit
- CNTR$_0$ interrupt enable bit
- Key-on wake-up interrupt enable bit
- A/D conversion interrupt enable bit
- Fix this bit to “0”.

0 : Interrupts disabled
1 : Interrupts enabled

Fig. 14 Structure of interrupt-related registers

Key Input Interrupt (Key-on Wake Up)

A Key-on wake up interrupt request is generated by applying a falling edge to any pin of port P0 that have been set to input mode. In other words, it is generated when AND of input level goes from "1" to "0". An example of using a key input interrupt is shown in Figure 15, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports P0₀–P0₃.

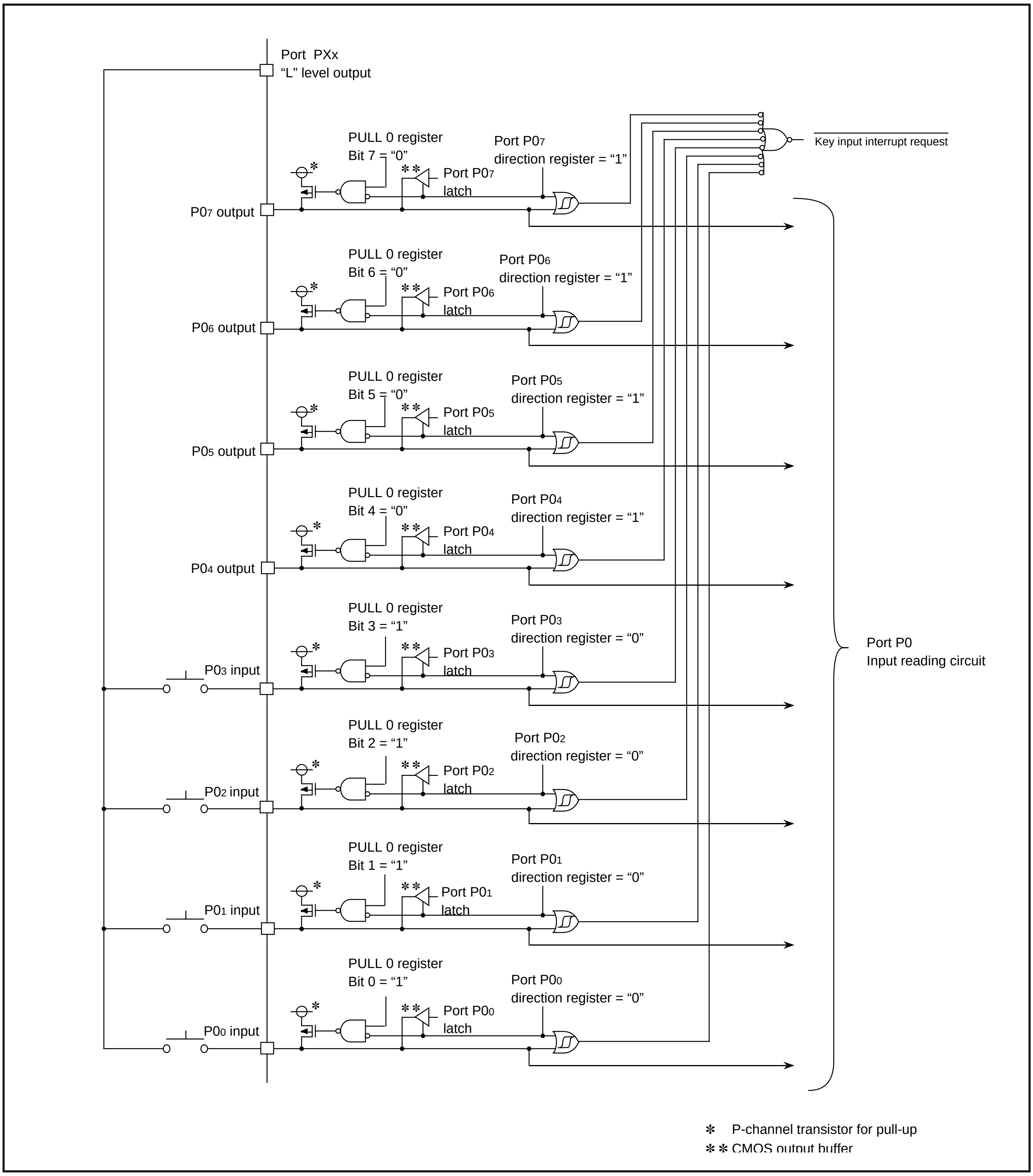


Fig. 15 Connection example when using key input interrupt and port P0 block diagram

TIMERS

The 38K2 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”.

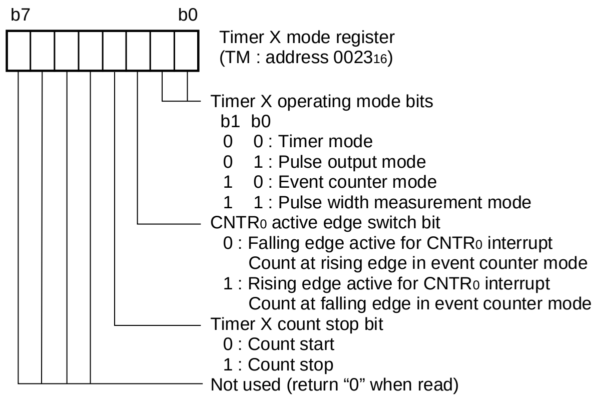

b7
b0
Timer X mode register
(TM : address 0023_{16})
Timer X operating mode bits
b1 b0
0 0 : Timer mode
0 1 : Pulse output mode
1 0 : Event counter mode
1 1 : Pulse width measurement mode
CNTR$_0$ active edge switch bit
0 : Falling edge active for CNTR$_0$ interrupt
Count at rising edge in event counter mode
1 : Rising edge active for CNTR$_0$ interrupt
Count at falling edge in event counter mode
Timer X count stop bit
0 : Count start
1 : Count stop
Not used (return “0” when read)

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.

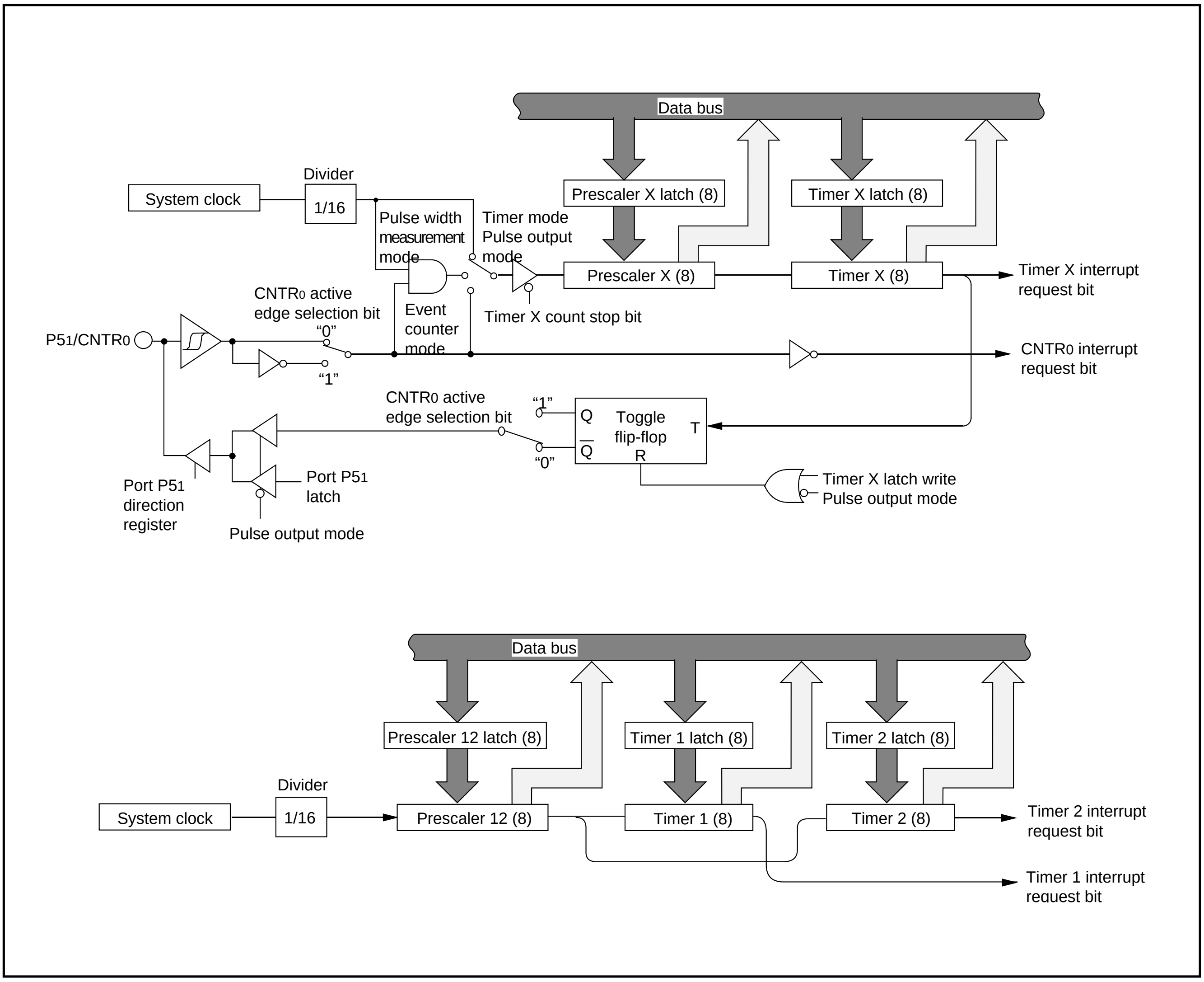


Fig. 17 Timer block diagram

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.

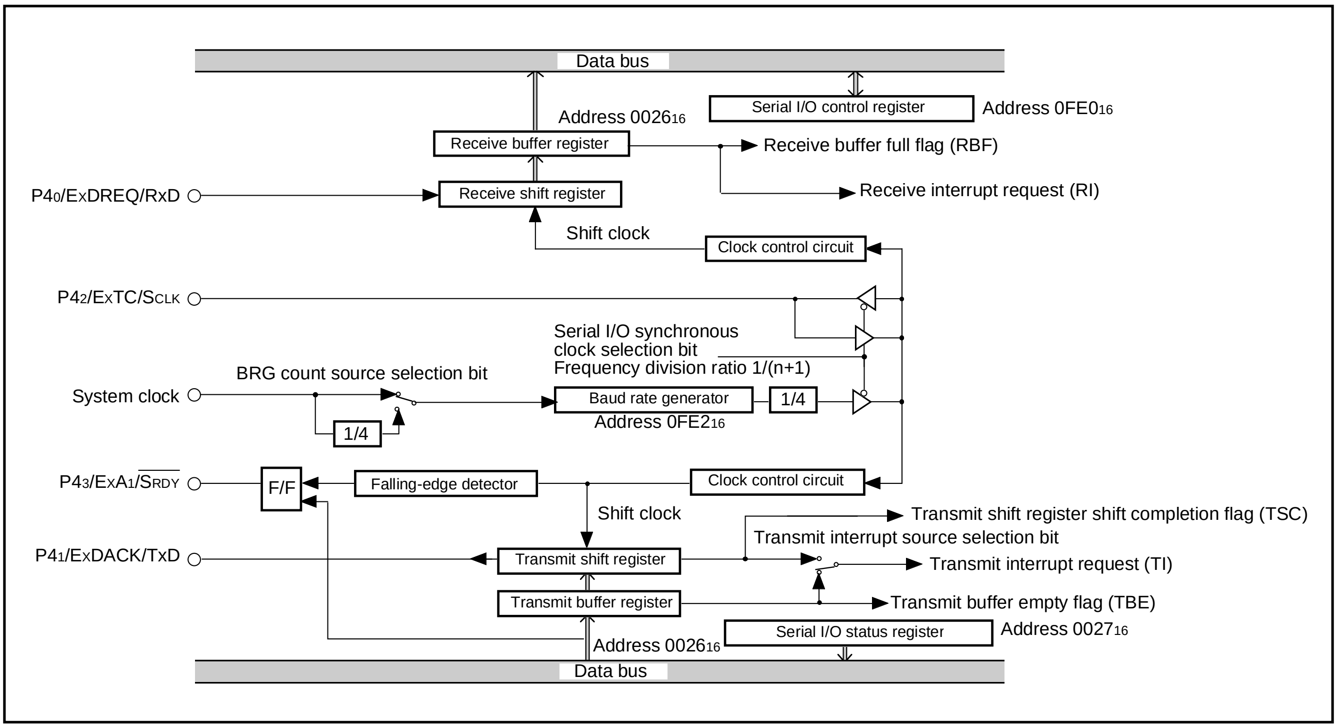
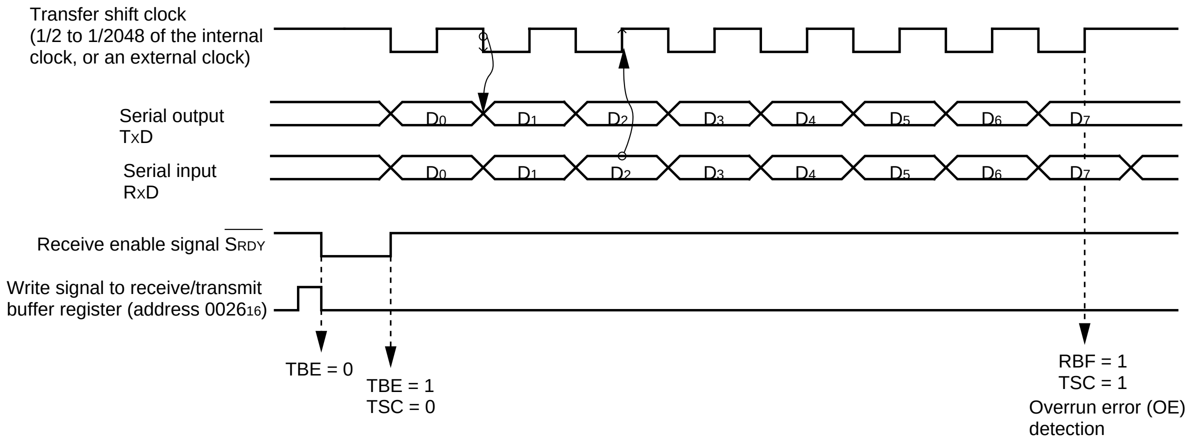

Data bus
Serial I/O control register
Address 0FE0$_{16}$
Address 0026$_{16}$
Receive buffer register
Receive buffer full flag (RBF)
P4$_0$/ExDREQ/RxD
Receive shift register
Receive interrupt request (RI)
Shift clock
Clock control circuit
P4$_2$/ExTC/S$_{CLK}$
Serial I/O synchronous clock selection bit
Frequency division ratio 1/(n+1)
BRG count source selection bit
System clock
Baud rate generator
1/4
Address 0FE2$_{16}$
1/4
P4$_3$/ExA$_1$/$\overline{S_{RDY}}$
F/F
Falling-edge detector
Clock control circuit
Shift clock
Transmit shift register shift completion flag (TSC)
Transmit interrupt source selection bit
P4$_1$/ExDACK/TxD
Transmit shift register
Transmit interrupt request (TI)
Transmit buffer register
Transmit buffer empty flag (TBE)
Serial I/O status register
Address 0027$_{16}$
Address 0026$_{16}$
Data bus

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


Transfer shift clock (1/2 to 1/2048 of the internal clock, or an external clock)
Serial output TxD
D$_0$ D$_1$ D$_2$ D$_3$ D$_4$ D$_5$ D$_6$ D$_7$
Serial input RxD
D$_0$ D$_1$ D$_2$ D$_3$ D$_4$ D$_5$ D$_6$ D$_7$
Receive enable signal $\overline{S_{RDY}}$
Write signal to receive/transmit buffer register (address 0026$_{16}$)
TBE = 0
TBE = 1
TSC = 0
RBF = 1
TSC = 1
Overrun error (OE) detection

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.

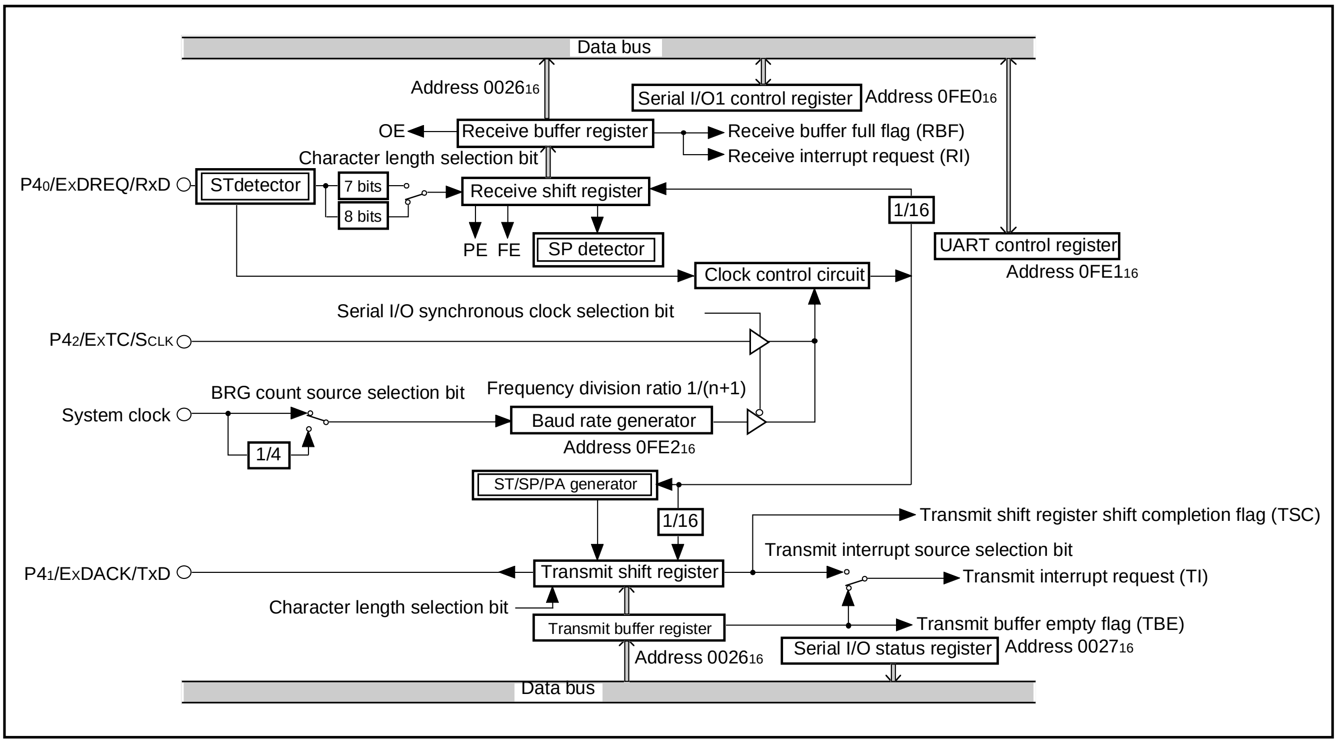
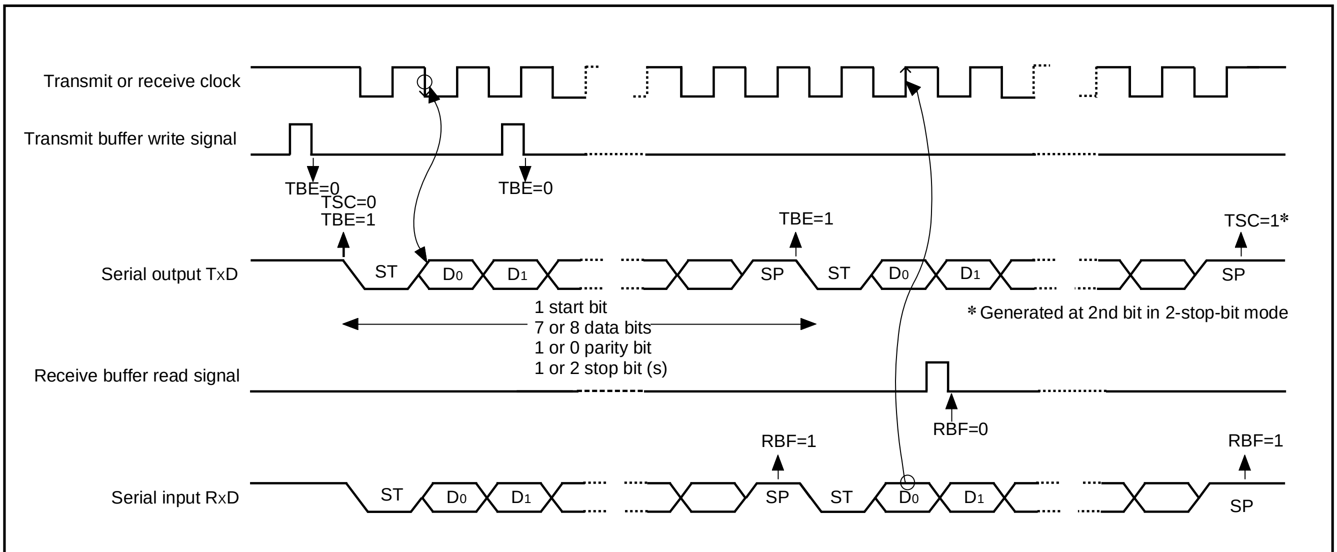

Data bus
Address 002616
Serial I/O1 control register
Address 0FE016
OE
Receive buffer register
Receive buffer full flag (RBF)
Receive interrupt request (RI)
Character length selection bit
P40/ExDREQ/RxD
STdetector
7 bits
8 bits
Receive shift register
1/16
UART control register
Address 0FE116
PE FE
SP detector
Clock control circuit
Serial I/O synchronous clock selection bit
P42/ExTC/SCLK
BRG count source selection bit
Frequency division ratio 1/(n+1)
System clock
Baud rate generator
1/4
Address 0FE216
ST/SP/PA generator
1/16
Transmit shift register shift completion flag (TSC)
Transmit interrupt source selection bit
P41/ExDACK/TxD
Transmit shift register
Transmit interrupt request (TI)
Character length selection bit
Transmit buffer register
Transmit buffer empty flag (TBE)
Serial I/O status register
Address 002716
Address 002616
Data bus

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).

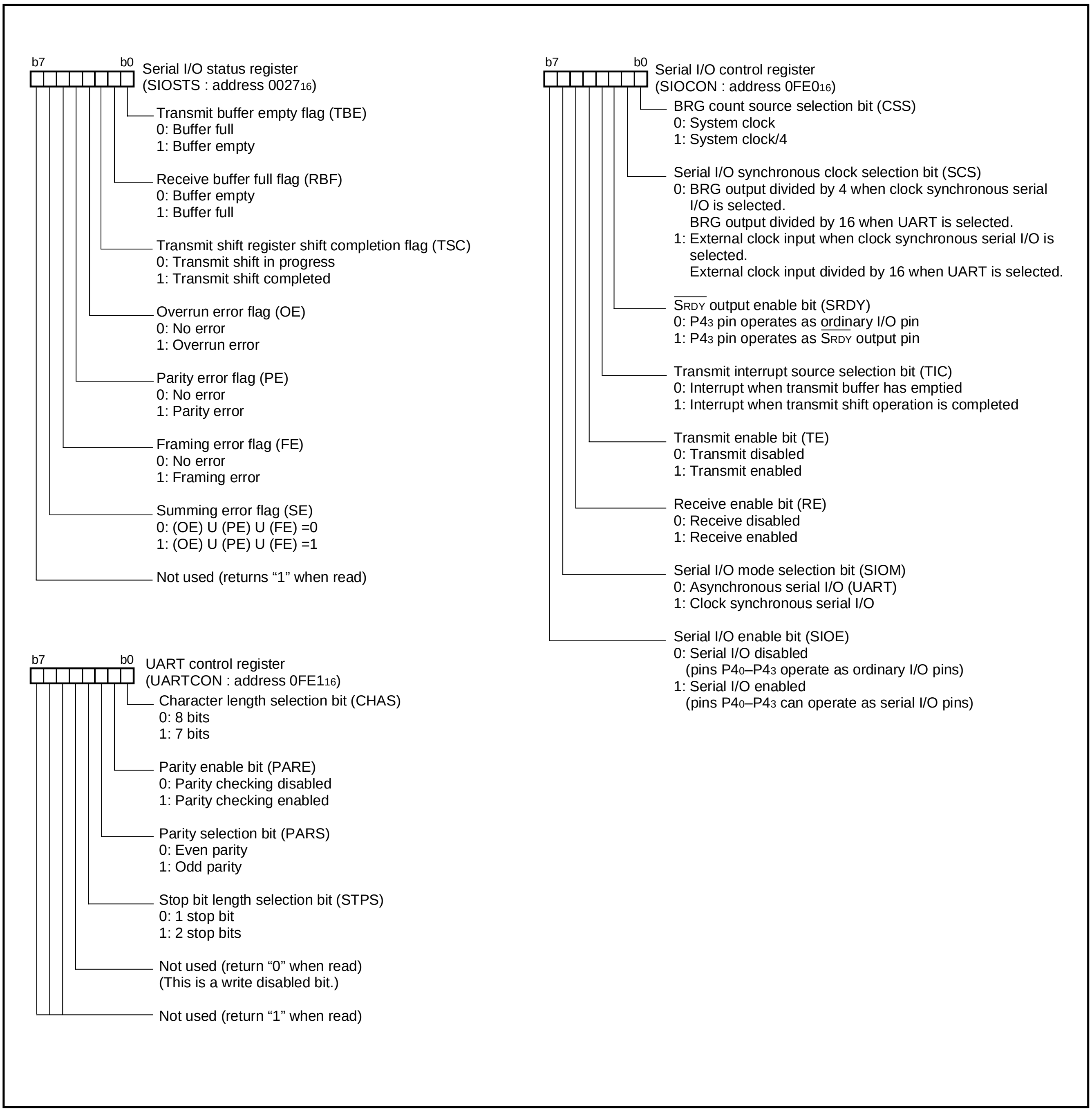

b7 b0
Serial I/O status register
(SIOSTS : address 0027_{16})
Transmit buffer empty flag (TBE)
0: Buffer full
1: Buffer empty
Receive buffer full flag (RBF)
0: Buffer empty
1: Buffer full
Transmit shift register shift completion flag (TSC)
0: Transmit shift in progress
1: Transmit shift completed
Overrun error flag (OE)
0: No error
1: Overrun error
Parity error flag (PE)
0: No error
1: Parity error
Framing error flag (FE)
0: No error
1: Framing error
Summing error flag (SE)
0: (OE) U (PE) U (FE) =0
1: (OE) U (PE) U (FE) =1
Not used (returns “1” when read)
b7 b0
UART control register
(UARTCON : address $0FE1_{16}$)
Character length selection bit (CHAS)
0: 8 bits
1: 7 bits
Parity enable bit (PARE)
0: Parity checking disabled
1: Parity checking enabled
Parity selection bit (PARS)
0: Even parity
1: Odd parity
Stop bit length selection bit (STPS)
0: 1 stop bit
1: 2 stop bits
Not used (return “0” when read)
(This is a write disabled bit.)
Not used (return “1” when read)
b7 b0
Serial I/O control register
(SIOCON : address $0FE0_{16}$)
BRG count source selection bit (CSS)
0: System clock
1: System clock/4
Serial I/O synchronous clock selection bit (SCS)
0: BRG output divided by 4 when clock synchronous serial I/O is selected.
BRG output divided by 16 when UART is selected.
1: External clock input when clock synchronous serial I/O is selected.
External clock input divided by 16 when UART is selected.
$\overline{\text{S}_{\text{RDY}}}$ output enable bit (SRDY)
0: P4_3 pin operates as ordinary I/O pin
1: P4_3 pin operates as $\overline{\text{S}_{\text{RDY}}}$ output pin
Transmit interrupt source selection bit (TIC)
0: Interrupt when transmit buffer has emptied
1: Interrupt when transmit shift operation is completed
Transmit enable bit (TE)
0: Transmit disabled
1: Transmit enabled
Receive enable bit (RE)
0: Receive disabled
1: Receive enabled
Serial I/O mode selection bit (SIOM)
0: Asynchronous serial I/O (UART)
1: Clock synchronous serial I/O
Serial I/O enable bit (SIOE)
0: Serial I/O disabled
(pins P4_0–P4_3 operate as ordinary I/O pins)
1: Serial I/O enabled
(pins P4_0–P4_3 can operate as serial I/O pins)

Fig. 22 Structure of serial I/O control registers

USB FUNCTION

38K2 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 two USB addresses and 6 endpoints, enabling separate control of the HUB functions and peripheral functions.

The USB address for HUB functions is equipped with two endpoints. Each endpoint is fixed to a specified transfer type: Endpoint 0 is fixed to Control Transfer and Endpoint 1 is fixed to Interrupt Transfer.

The USB address for peripheral functions is equipped with 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/24 factors).

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

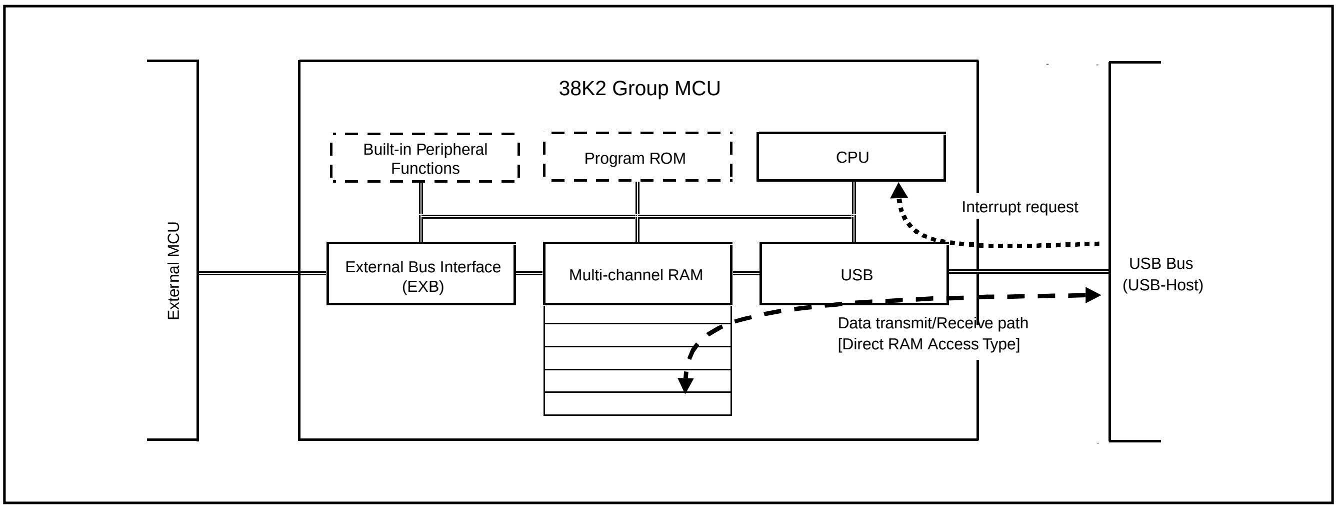

38K2 Group MCU
Built-in Peripheral Functions
Program ROM
CPU
Interrupt request
External MCU
External Bus Interface (EXB)
Multi-channel RAM
USB
USB Bus (USB-Host)
Data transmit/Receive path [Direct RAM Access Type]

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)

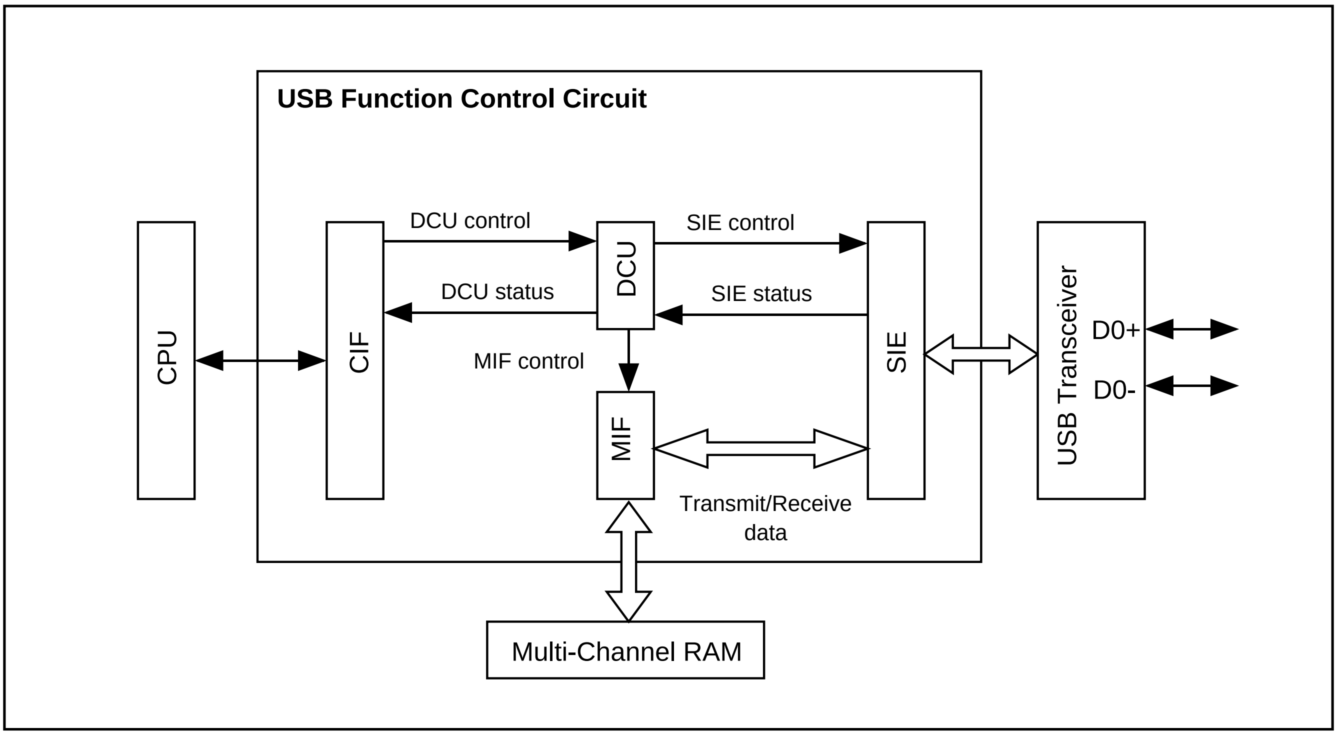


Fig. 24 USB Function Control Circuit (USBFCC) block diagram

(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)

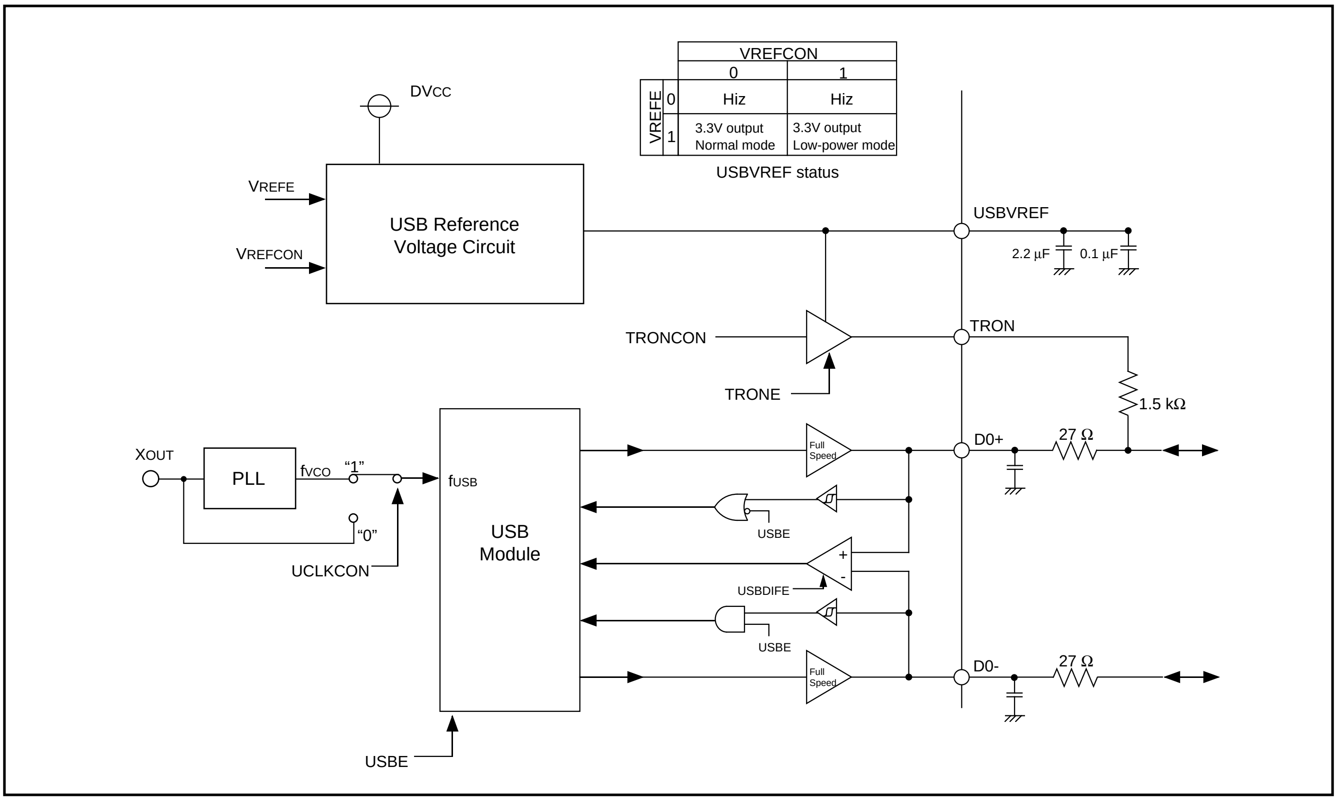
The CIF performs the following functions:

- 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 0010₁₆).

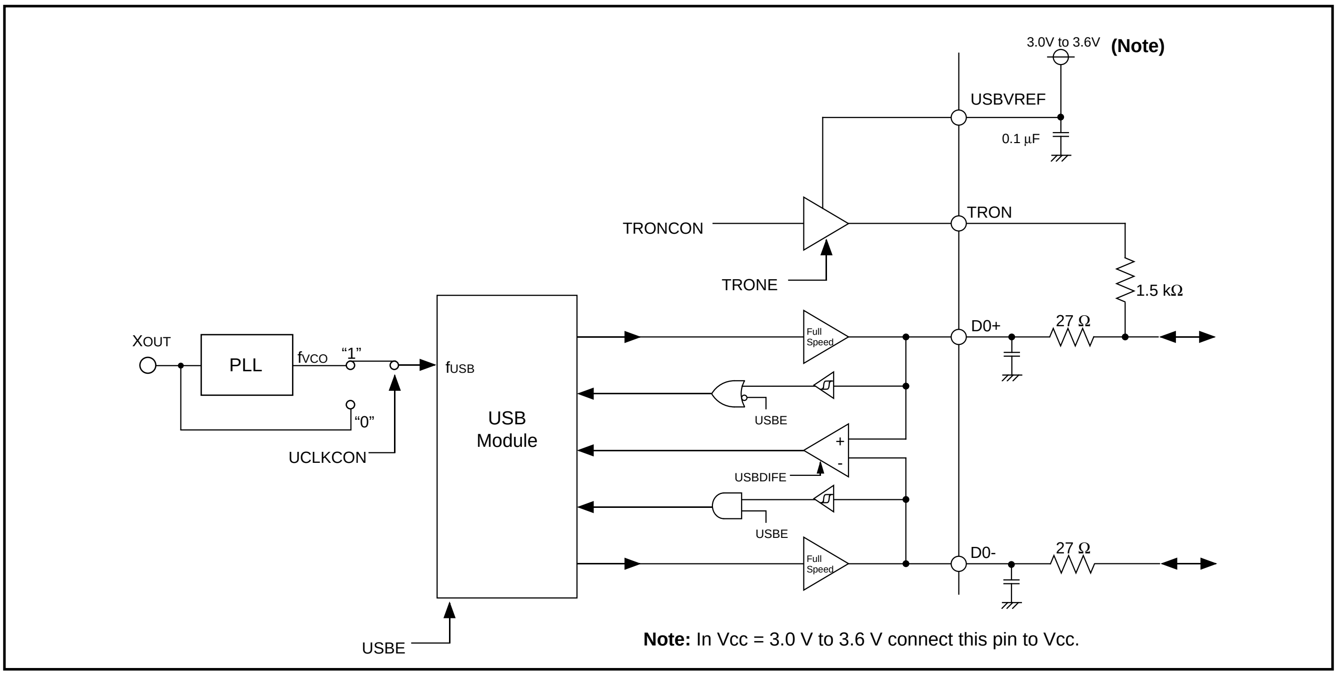
Figure 25 and Figure 26 show the USB port external circuit block diagram.



VREFE	VREFCON 0	VREFCON 1
0	Hiz	Hiz
1	3.3V output Normal mode	3.3V output Low-power mode

USBVREF status

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



Note: In Vcc = 3.0 V to 3.6 V connect this pin to Vcc.

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
•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

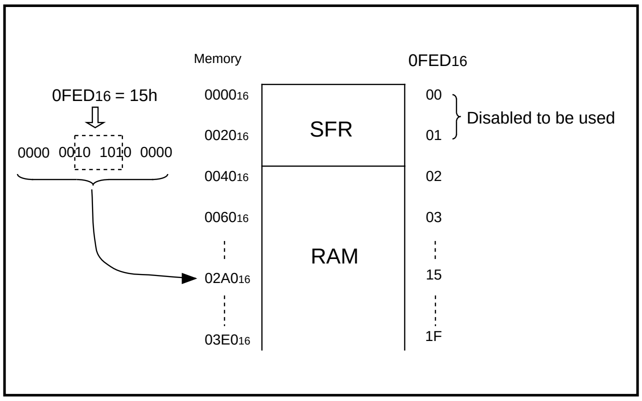


Fig. 27 Example setting of buffer area beginning address

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.

(4) Endpoint 10

Same as Endpoint 00.

(5) Endpoint 11

Endpoint 11 has only one interrupt source for accessing the buffer.
B0RDY11 (Buffer 0 Ready Interrupt): Offset address = 00h

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.

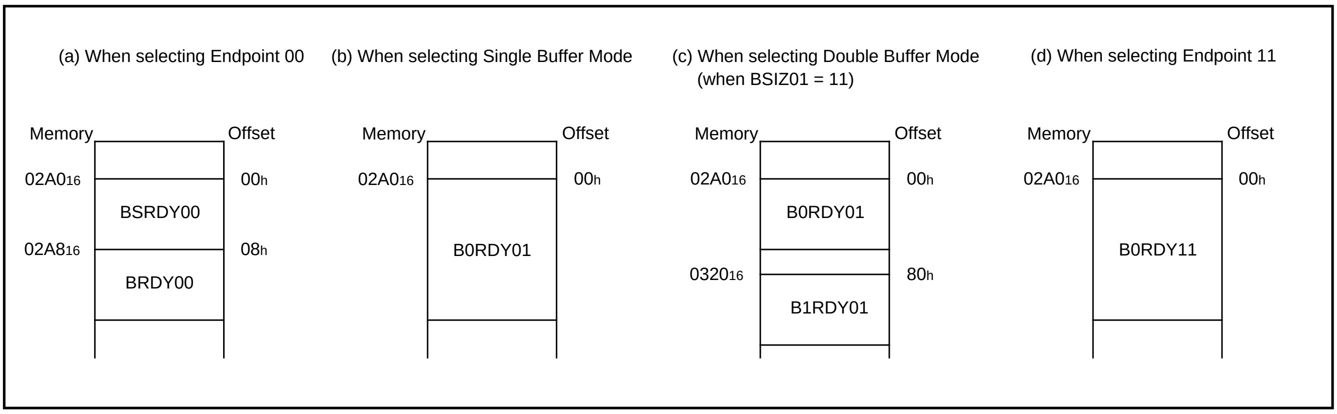

(a) When selecting Endpoint 00
(b) When selecting Single Buffer Mode
(c) When selecting Double Buffer Mode (when BSIZ01 = 11)
(d) When selecting Endpoint 11
Memory
Offset
02A016
00h
BSRDY00
02A816
08h
BRDY00
B0RDY01
032016
80h
B1RDY01
B0RDY11

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

USB Interrupt Function

USB Interrupt Control Circuit (USBINTCON) has 3 requests and 22 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 $003C_{16}$)	USB interrupt bit (USBIREQ: Address 0017_{16})	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 f_{USB} = 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 f_{USB} = 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
	EP10	At Endpoint 10 data transfer complete: •Buffer ready (read/write enabled state) •Control transfer completed •Status stage transition •SETUP buffer ready (read enabled state) •Control transfer error
	EP11	At Endpoin 11 data transfer complete: •Buffer 0 ready (write enabled state)
	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 f_{USB} = 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.

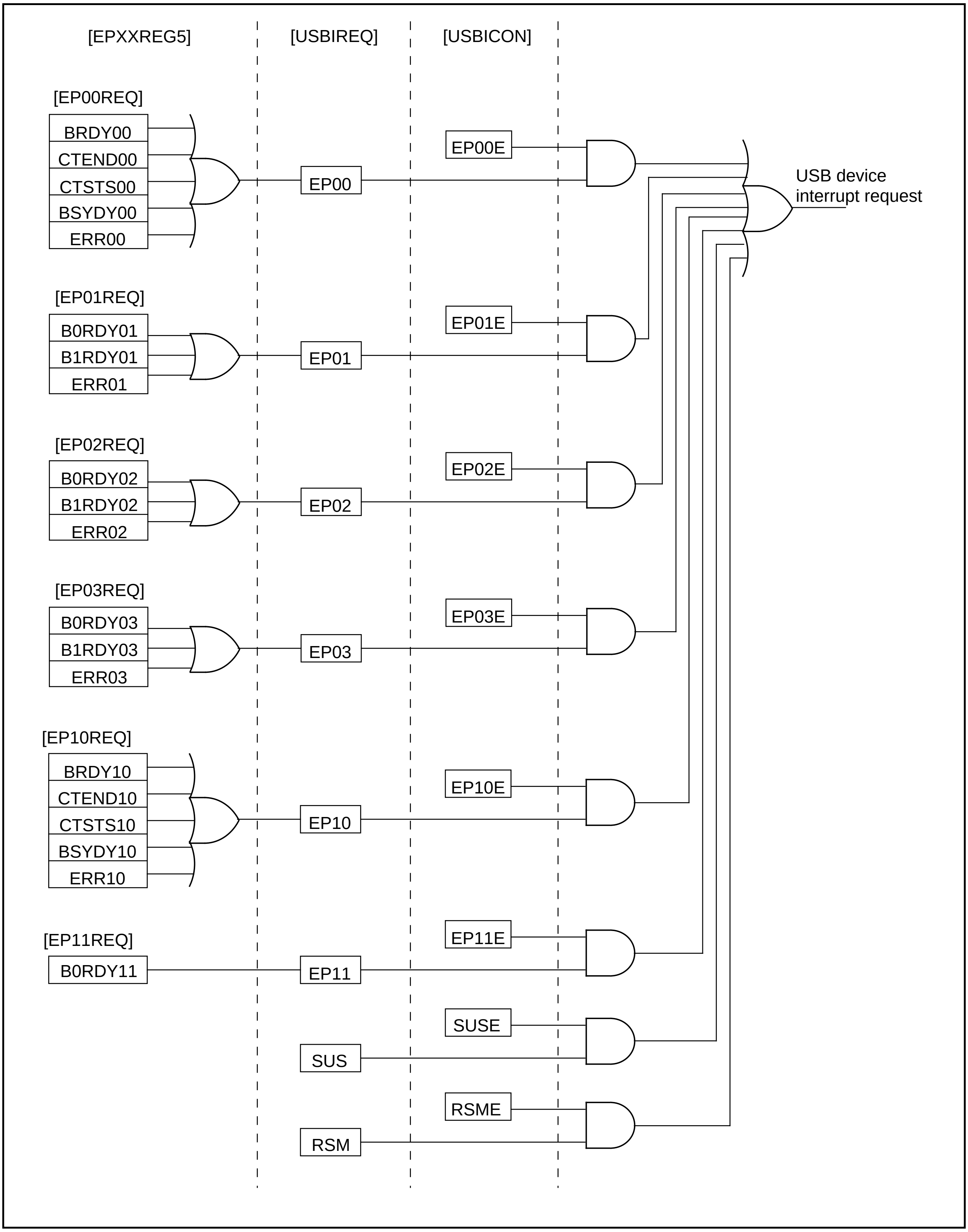


Fig. 29 USB device interrupt control

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
0010_{16}	USB control register	USBCON	USBE	UCLKCON	USBDIFE	VREFE	VREFCON	TRONE	TRONCON	WKUP
0011_{16}	USB Function/Hub enable register	USBAE	☒	☒	☒	☒	☒	☒	AD1E	AD0E
0012_{16}	USB function address register	USBA0	☒	USBADD0[6:0]						
0013_{16}	USB HUB address register	USBA1	☒	USBADD1[6:0]						
0014_{16}	Frame number register Low	FNUML	FNUM[7:0]							
0015_{16}	Frame number register High	FNUMH	☒	☒	☒	☒	☒	FNUM[10:8]		
0016_{16}	USB interrupt source enable register	USBICON	RSME	SUSE	EP11E	EP10E	EP03E	EP02E	EP01E	EP00E
0017_{16}	USB interrupt source register	USBIREQ	RSM	SUS	EP11	EP10	EP03	EP02	EP01	EP00
0018_{16}	Endpoint index register	USBINDEX	☒	☒	☒	☒	☒	ADIDX	EPIDX[1:0]	
0019_{16}	Endpoint field register 1	EPXXREG1								
$001A_{16}$	Endpoint field register 2	EPXXREG2								
$001B_{16}$	Endpoint field register 3	EPXXREG3								
$001C_{16}$	Endpoint field register 4	EPXXREG4								
$001D_{16}$	Endpoint field register 5	EPXXREG5								
$001E_{16}$	Endpoint field register 6	EPXXREG6								
$001F_{16}$	Endpoint field register 7	EPXXREG7								
$0FEC_{16}$	Endpoint field register 8	EPXXREG8								
$0FED_{16}$	Endpoint field register 9	EPXXREG9								

(1) Endpoint 00

0019_{16}	EP00 stage register	EP00STG	☒	☒	☒	☒	☒	☒	☒	SETUP00
$001A_{16}$	EP00 control register 1	EP00CON1	☒	☒	☒	☒	☒	☒	PID00[1:0]	
$001B_{16}$	EP00 control register 2	EP00CON2	☒	☒	☒	☒	☒	☒	☒	BVAL00
$001C_{16}$	EP00 control register 3	EP00CON3	☒	☒	☒	☒	☒	☒	☒	CTENDE00
$001D_{16}$	EP00 interrupt source register	EP00REQ	☒	☒	☒	ERR00	BSRDY00	CTSTS00	CTEND00	BRDY00
$001E_{16}$	EP00 byte number register	EP00BYT	☒	☒	☒	☒	BBYT00[3:0]			
$001F_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$0FEC_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$0FED_{16}$	EP00 buffer area set register	EP00BUF	☒	☒	☒	BADD00[4:0]				

(2) Endpoint 01

0019_{16}	EP01 set register	EP01CFG	TYP01[1:0]		DIR01	ITMD01	SQCL01	DBLB01	BSIZ01[1:0]	
$001A_{16}$	EP01 control register 1	EP01CON1	☒	☒	☒	☒	☒	☒	PID01[1:0]	
$001B_{16}$	EP01 control register 2	EP01CON2	☒	☒	☒	☒	☒	☒	☒	B0VAL01
$001C_{16}$	EP01 control register 3	EP01CON3	☒	☒	☒	☒	☒	☒	☒	B1VAL01
$001D_{16}$	EP01 interrupt source register	EP01REQ	☒	☒	☒	☒	☒	ERR01	B1RDY01	B0RDY01
$001E_{16}$	EP01 byte number register 0	EP01BYT0	☒	B0BYT01[6:0]						
$001F_{16}$	EP01 byte number register 1	EP01BYT1	☒	B1BYT01[6:0]						
$0FEC_{16}$	EP01 MAX. packet size register	EP01MAX	☒	MXPS01[6:0]						
$0FED_{16}$	EP01 buffer area set register	EP01BUF	☒	☒	☒	BADD01[4:0]				

(3) Endpoint 02

0019_{16}	EP02 set register	EP02CFG	TYP02[1:0]		DIR02	ITMD02	SQCL02	DBLB02	BSIZ02[1:0]	
$001A_{16}$	EP02 control register 1	EP02CON1	☒	☒	☒	☒	☒	☒	PID02[1:0]	
$001B_{16}$	EP02 control register 2	EP02CON2	☒	☒	☒	☒	☒	☒	☒	B0VAL02
$001C_{16}$	EP02 control register 3	EP02CON3	☒	☒	☒	☒	☒	☒	☒	B1VAL02
$001D_{16}$	EP02 interrupt source register	EP02REQ	☒	☒	☒	☒	☒	ERR02	B1RDY02	B0RDY02
$001E_{16}$	EP02 byte number register 0	EP02BYT0	☒	B0BYT02[6:0]						
$001F_{16}$	EP02 byte number register 1	EP02BYT1	☒	B1BYT02[6:0]						
$0FEC_{16}$	EP02 MAX. packet size register	EP02MAX	☒	MXPS02[6:0]						
$0FED_{16}$	EP02 buffer area set register	EP02BUF	☒	☒	☒	BADD02[4:0]				

(4) Endpoint 03

0019_{16}	EP03 set register	EP03CFG	TYP03[1:0]		DIR03	ITMD03	SQCL03	DBLB03	BSIZ03[1:0]	
$001A_{16}$	EP03 control register 1	EP03CON1	☒	☒	☒	☒	☒	☒	PID03[1:0]	
$001B_{16}$	EP03 control register 2	EP03CON2	☒	☒	☒	☒	☒	☒	☒	B0VAL03
$001C_{16}$	EP03 control register 3	EP03CON3	☒	☒	☒	☒	☒	☒	☒	B1VAL03
$001D_{16}$	EP03 interrupt source register	EP03REQ	☒	☒	☒	☒	☒	ERR03	B1RDY03	B0RDY03
$001E_{16}$	EP03 byte number register 0	EP03BYT0	☒	B0BYT03[6:0]						
$001F_{16}$	EP03 byte number register 1	EP03BYT1	☒	B1BYT03[6:0]						
$0FEC_{16}$	EP03 MAX. packet size register	EP03MAX	☒	MXPS03[6:0]						
$0FED_{16}$	EP03 buffer area set register	EP03BUF	☒	☒	☒	BADD03[4:0]				

(5) Endpoint 10

0019_{16}	EP10 set register	EP10STG	☒	☒	☒	☒	☒	☒	☒	SETUP10
$001A_{16}$	EP10 control register 1	EP10CON1	☒	☒	☒	☒	☒	☒	PID10[1:0]	
$001B_{16}$	EP10 control register 2	EP10CON2	☒	☒	☒	☒	☒	☒	☒	BVAL10
$001C_{16}$	EP10 control register 3	EP10CON3	☒	☒	☒	☒	☒	☒	☒	CTENDE10
$001D_{16}$	EP10 interrupt source register	EP10REQ	☒	☒	☒	ERR10	BSRDY10	CTSTS10	CTEND10	BRDY10
$001E_{16}$	EP10 byte number register	EP10BYT	☒	☒	☒	☒	BBYT10[3:0]			
$001F_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$0FEC_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$0FED_{16}$	EP10 buffer area set register	EP10BUF	☒	☒	☒	BADD10[4:0]				

(6) Endpoint 11

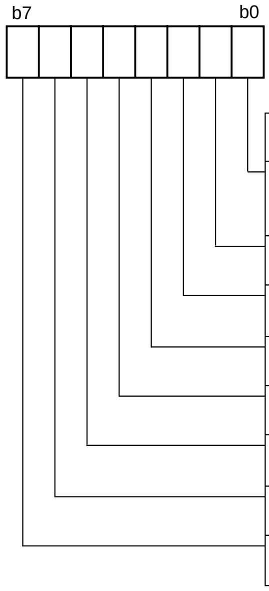
0019_{16}	EP11 set register	EP11CFG	TYP11	☒	DIR11	☒	SQCL11	☒	☒	☒
$001A_{16}$	EP11 control register 1	EP11CON1	☒	☒	☒	☒	☒	☒	PID11[1:0]	
$001B_{16}$	EP11 control register 2	EP11CON2	☒	☒	☒	☒	☒	☒	☒	B0VAL11
$001C_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$001D_{16}$	EP11 interrupt source register	EP11REQ	☒	☒	☒	☒	☒	☒	☒	B0RDY11
$001E_{16}$	EP11 byte number register	EP11BYT0	☒	☒	☒	☒	☒	☒	☒	B0BYT11
$001F_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$0FEC_{16}$			☒	☒	☒	☒	☒	☒	☒	☒
$0FED_{16}$	EP11 buffer area set register	EP11BUF	☒	☒	☒	BADD11[4:0]				

☒ : Not used

Fig. 30 USB related registers

USB Related Registers

The USB related registers are shown below.

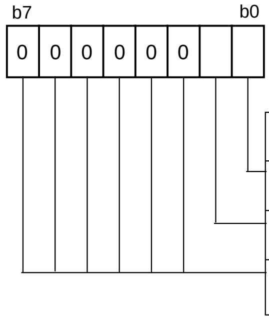


USB control register (USBCON) [address 0010$_{16}$]

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

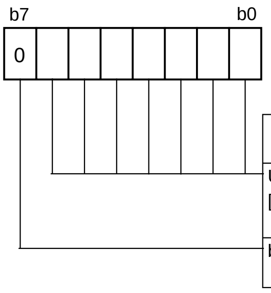


USB function/HUB enable register (USBAE) [address 0011$_{16}$]

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
AD1E	USB HUB enable bit	0: USB HUB address register invalidated 1: USB HUB address register validated	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 32 Structure of USB function/HUB enable register

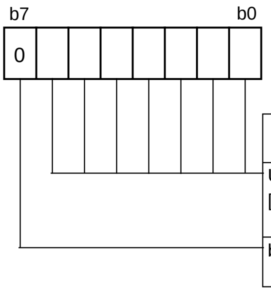


USB function address register (USBA0) [address 0012$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

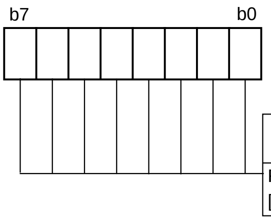


USB HUB address register (USBA1) [address 0013$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
USBADD1 [6:0]	USB HUB address bit	In AD1E = “0”, this value changes after writing. In AD1E = “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

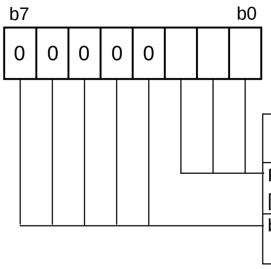
Fig. 34 Structure of USB HUB address register



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

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

Fig. 35 Structure of Frame number register Low

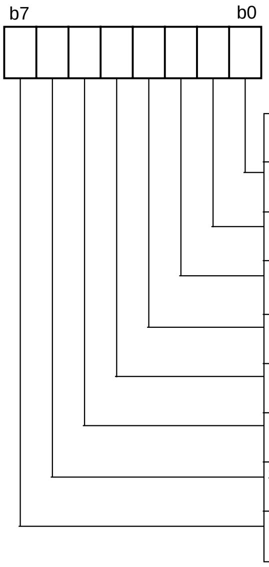


Frame number register High (FNUMH) [address 0015$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 36 Structure of Frame number register High



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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
EP10E	USB HUB/Endpoint 0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
EP11E	USB HUB/Endpoint 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	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

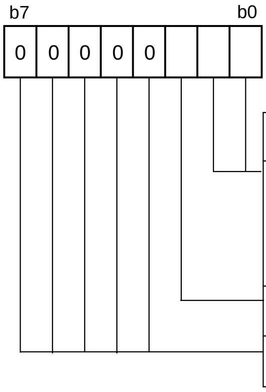
Fig. 37 Structure of USB interrupt source enable register

b7 b0

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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
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	X
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	X
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	X
EP10	USB HUB/Endpoint 0 interrupt bit	This bit is set to “1” when any one of EP10 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP10 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	X
EP11	USB HUB/Endpoint 1 interrupt bit	This bit is set to “1” when any one of EP11 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP11 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	X
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	X

Fig.38 Structure of USB interrupt source register



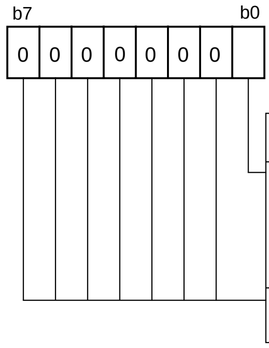
Endpoint index register (USBINDEX) [address 0018_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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
ADIDX	Address index bit	0 : USB function 1 : USB HUB	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 39 Structure of Endpoint index register

(1) Endpoint 00

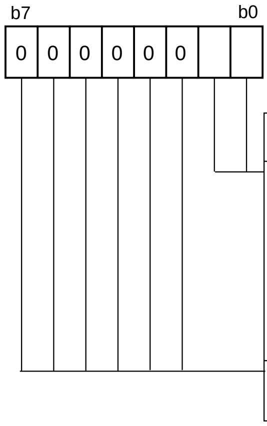


EP00 stage register (EP00STG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 40 Structure of EP00 stage register

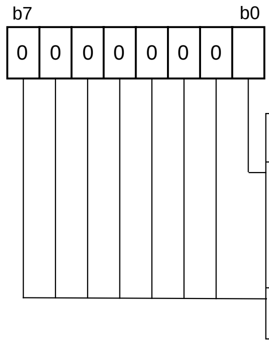


EP00 control register 1 (EP00CON1) [address $001A_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 41 Structure of EP00 control register 1

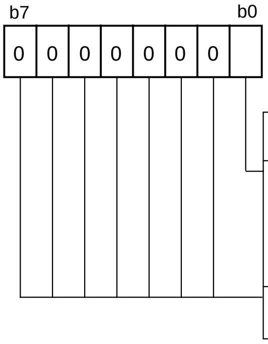


EP00 control register 2 (EP00CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 42 Structure of EP00 control register 2

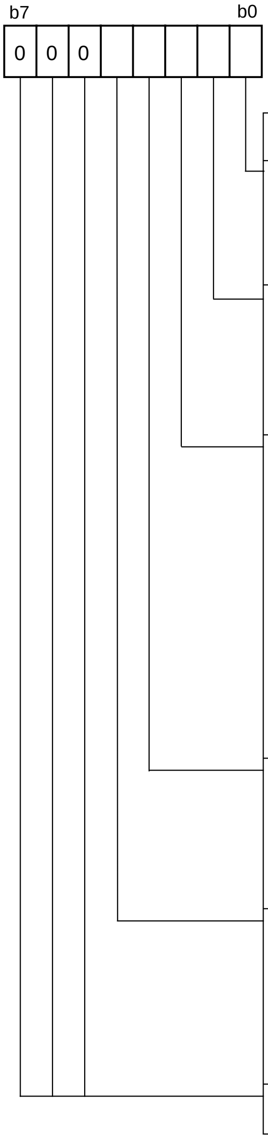


EP00 control register 3 (EP00CON3) [address $001C_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 = "01_2") At reception of SETUP token: This bit is cleared to "0" by the hardware.	0	–	○	○
b7:b1	Not used	Write "0" when writing. "0" is read when reading.	–	–	○	○

–: State remaining

Fig. 43 Structure of EP00 control register 3

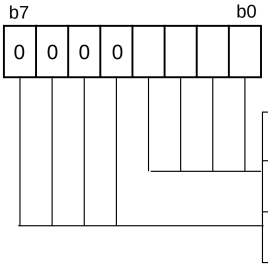


EP00 interrupt source register (EP00REQ) [address $001D_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	○	○
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	○	○
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	○	○
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	○	○
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	○	○
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	○	○

–: State remaining

Fig. 44 Structure of EP00 interrupt source register

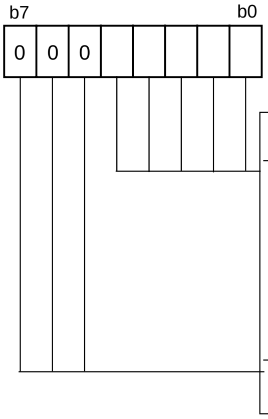


EP00 byte number register (EP00BYT) [address 001E$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 45 Structure of EP00 byte number register



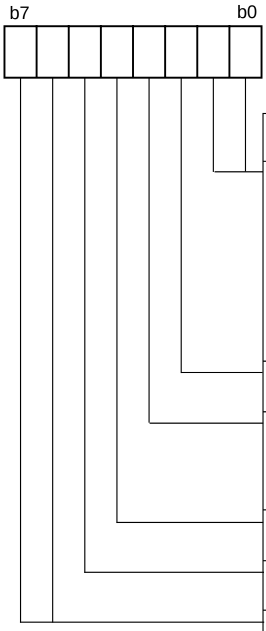
EP00 buffer area set register (EP00BUF) [address 0FED$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 : 0040$_{16}$ 0 0 0 1 1 : 0060$_{16}$ .... 1 1 1 1 0 : 03C0$_{16}$ 1 1 1 1 1 : 03E0$_{16}$	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 46 Structure of EP00 buffer area set register

(2) Endpoint 01

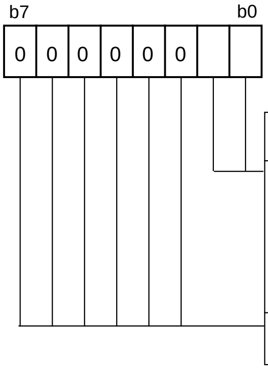


EP01 set register (EP01CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 47 Structure of EP01 set register

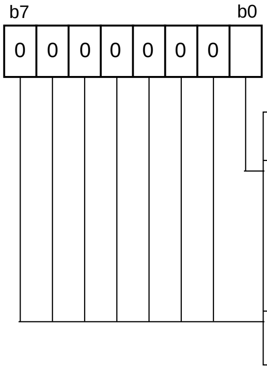


EP01 control register 1 (EP01CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 48 Structure of EP01 control register 1

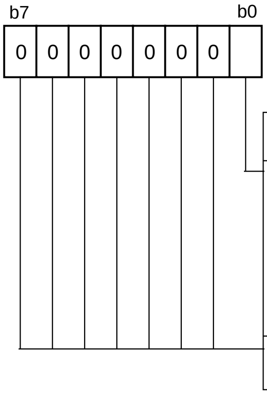


EP01 control register 2 (EP01CON2) [address 001B$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 49 Structure of EP01 control register 2

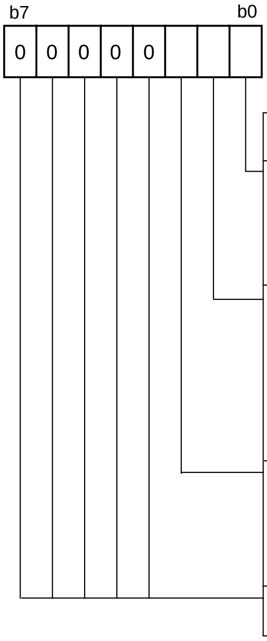


EP01 control register 3 (EP01CON3) [address 001C$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

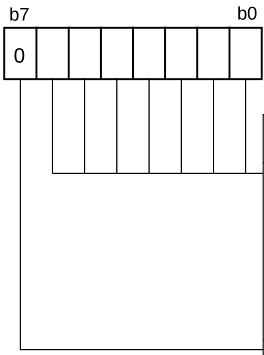
Fig.50 Structure of EP01 control register 3



EP01 interrupt source register (EP01REQ) [address 001D$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 51 Structure of EP01 interrupt source register

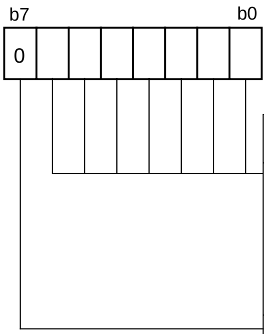


EP01 byte number register 0 (EP01BYT0) [address $001E_{16}$]

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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 52 Structure of EP01 byte number register 0

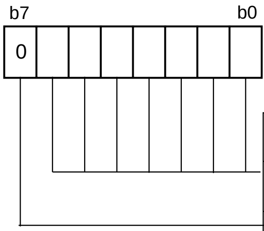


EP01 byte number register 1 (EP01BYT1) [address $001F_{16}$]

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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 53 Structure of EP01 byte number register 1

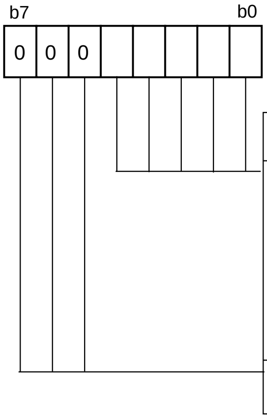


EP01 MAX. packet size register (EP01MAX) [address $0FEC_{16}$]

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. 54 Structure of EP01 MAX. packet size register



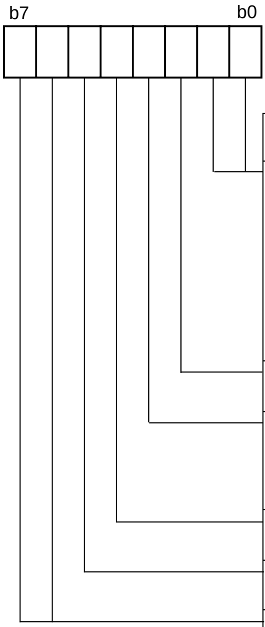
EP01 buffer area set register (EP01BUF) [address 0FED$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 : 0040$_{16}$ 0 0 0 1 1 : 0060$_{16}$ .... 1 1 1 1 0 : 03C0$_{16}$ 1 1 1 1 1 : 03E0$_{16}$	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 55 Structure of EP01 buffer area set register

(3) Endpoint 02

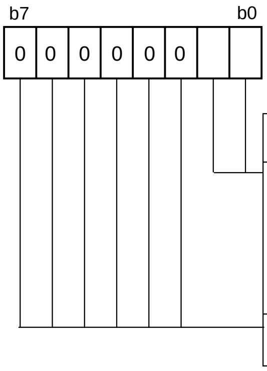


EP02 set register (EP02CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 56 Structure of EP02 set register

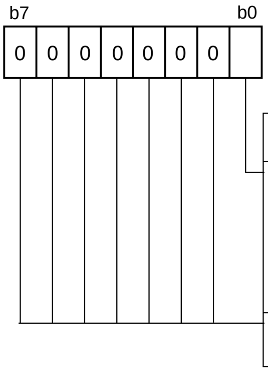


EP02 control register 1 (EP02CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 57 Structure of EP02 control register 1

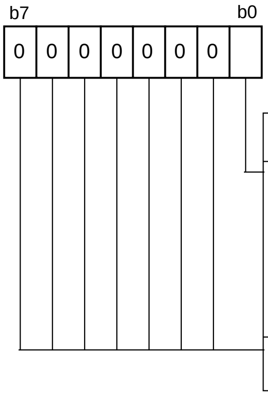


EP02 control register 2 (EP02CON2) [address 001B16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 58 Structure of EP02 control register 2

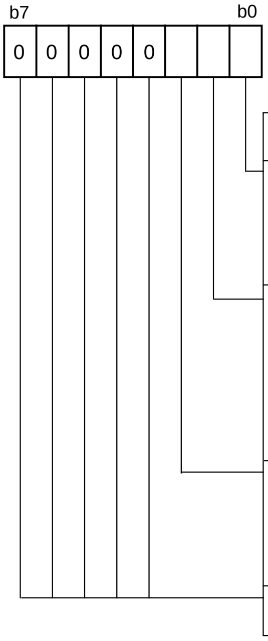


EP02 control register 3 (EP02CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

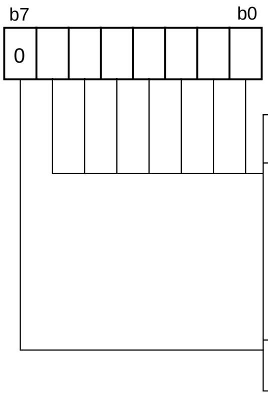
Fig. 59 Structure of EP02 control register 3



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 60 Structure of EP02 interrupt source register

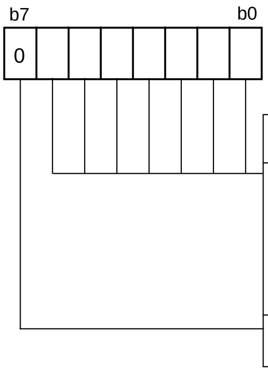


EP02 byte number register 0 (EP02BYT0) [address 001E$_{16}$]

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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 61 Structure of EP02 byte number register 0

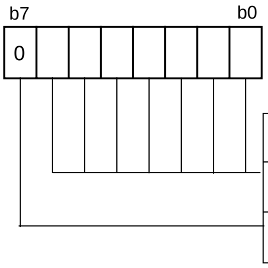


EP02 byte number register 1 (EP02BYT1) [address 001F$_{16}$]

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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 62 Structure of EP02 byte number register 1

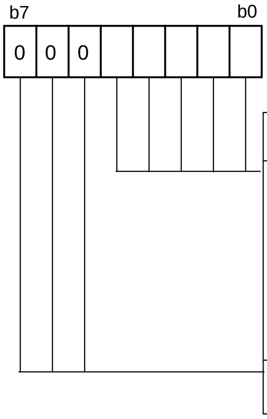


EP02 MAX. packet size register (EP02MAX) [address 0FEC$_{16}$]

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. 63 Structure of EP02 MAX. packet size register



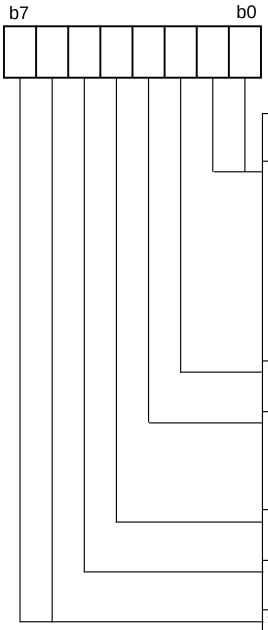
EP02 buffer area set register (EP02BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 64 Structure of EP02 buffer area set register

(4) Endpoint 03

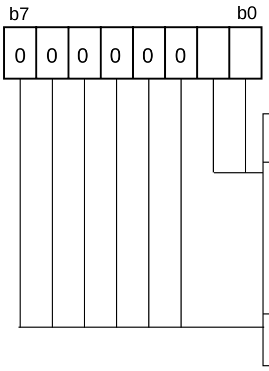


EP03 set register (EP03CFG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 65 Structure of EP03 set register

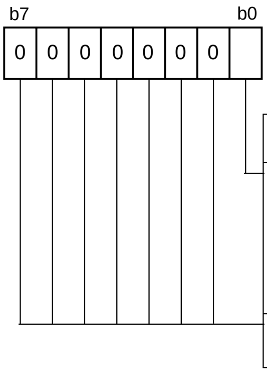


EP03 control register 1 (EP03CON1) [address $001A_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 66 Structure of EP03 control register 1

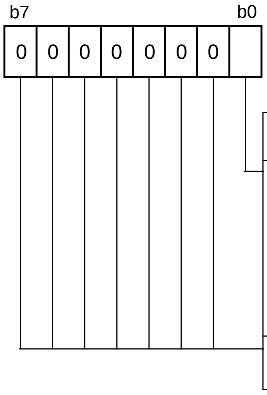


EP03 control register 2 (EP03CON2) [address 001B16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 67 Structure of EP03 control register 2

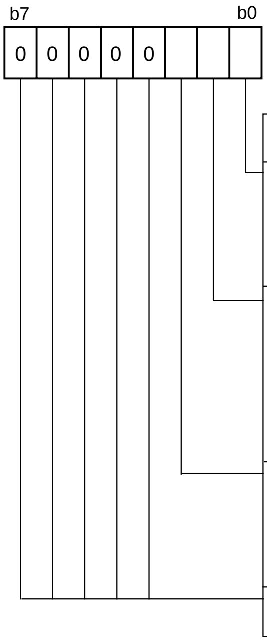


EP03 control register 3 (EP03CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

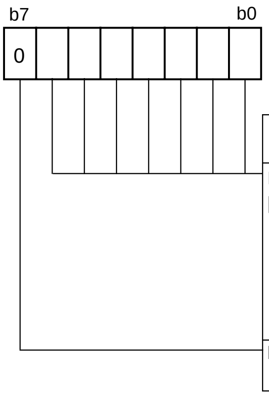
Fig. 68 Structure of EP03 control register 3



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 69 Structure of EP03 interrupt source register

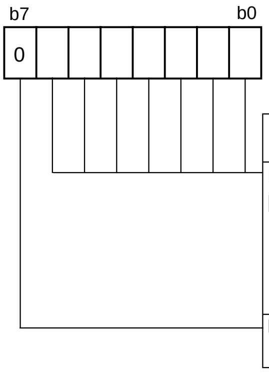


EP03 byte number register 0 (EP03BYT0) [address $001E_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 70 Structure of EP03 byte number register 0

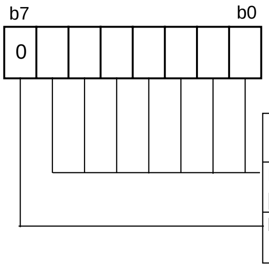


EP03 byte number register 1 (EP03BYT1) [address $001F_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 71 Structure of EP03 byte number register 1

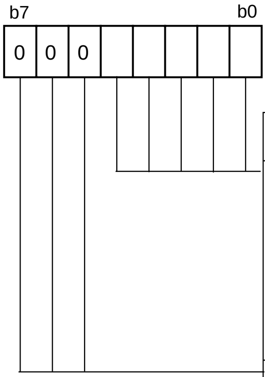


EP03 MAX. packet size register (EP03MAX) [address $0FEC_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 72 Structure of EP03 MAX. packet size register



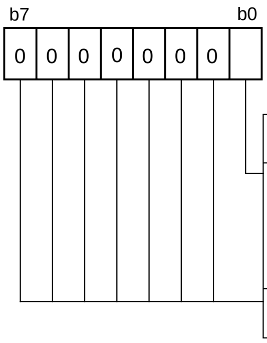
EP03 buffer area set register (EP03BUF) [address 0FED$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 : 0040$_{16}$ 0 0 0 1 1 : 0060$_{16}$ .... 1 1 1 1 0 : 03C0$_{16}$ 1 1 1 1 1 : 03E0$_{16}$	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 73 Structure of EP03 buffer area set register

(5) Endpoint 10

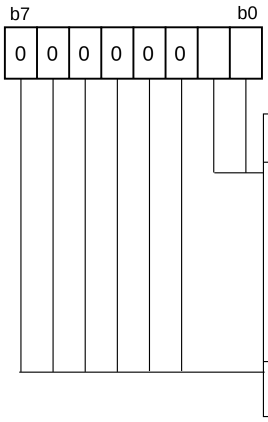


EP10 stage register (EP10STG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
SETUP10	SETUP packet detection bit	This bit is set to “1” at reception of SETUP packet. Writing “0” clears this bit if the next SETUP token does not occur. Writing “1” causes no state change of the status flags. This bit change is not for an interrupt source.	1	1	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 74 Structure of EP10 stage register

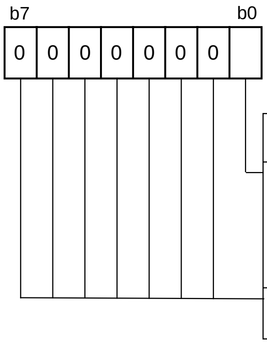


EP10 control register 1 (EP10CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PID10 [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. 75 Structure of EP10 control register 1

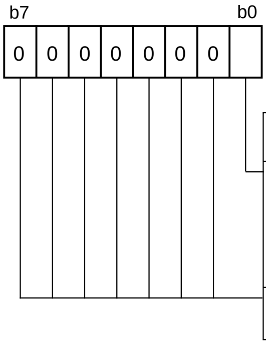


EP10 control register 2 (EP10CON2) [address 001B$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BVAL10	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.) (Valid in PID10 = “01$_2$”) 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. 76 Structure of EP10 control register 2

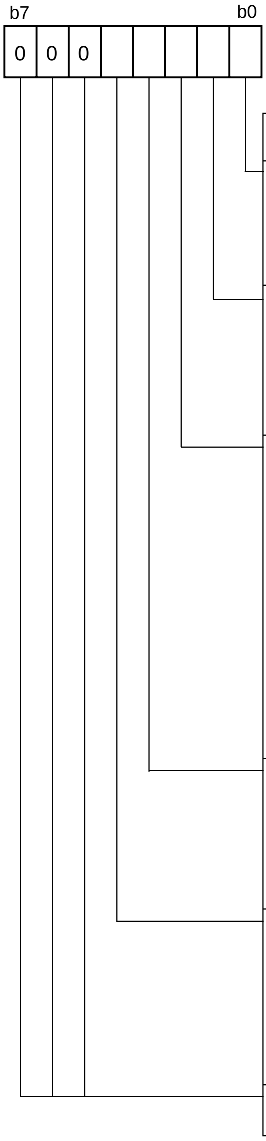


EP10 control register 3 (EP10CON3) [address $001C_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
CTENDE10	Control transfer completion enable bit	0 : NAK transmission in the status stage 1 : Control transfer completion enabled (SIE transmits NULL/ACK.) (Valid in PID10 = “01_2”) 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. 77 Structure of EP10 control register 3

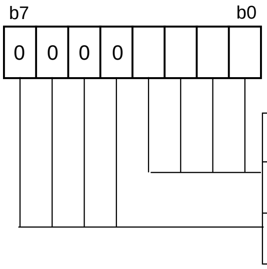


EP10 interrupt source register (EP10REQ) [address $001D_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BRDY10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. “0” can be set by software, but “1” cannot be set.	0	0	O	O
CTEND10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. “0” can be set by software, but “1” cannot be set.	0	0	O	O
CTSTS10	USB HUB/Endpoint 10 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 CTENDE10 = “0” (control transfer completion disabled) on USB HUB/Endpoint 10. “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
BSRDY10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. “0” can be set by software, but “1” cannot be set.	0	0	O	O
ERR10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. 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. 78 Structure of EP10 interrupt source register

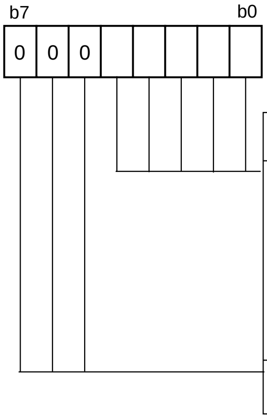


EP10 byte number register (EP10BYT) [address 001E$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BBYT10 [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. 79 Structure of EP10 byte number register



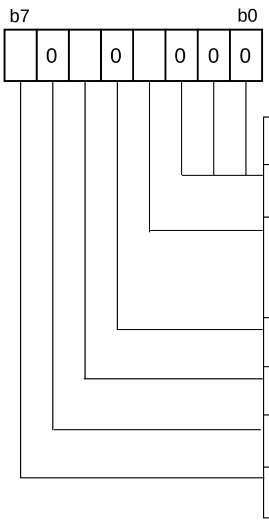
EP10 buffer area set register (EP10BUF) [address 0FED$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BADD10 [4:0]	EP10 beginning address set bit	Set the beginning address of EP10's buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040$_{16}$ 0 0 0 1 1 : 0060$_{16}$ .... 1 1 1 1 0 : 03C0$_{16}$ 1 1 1 1 1 : 03E0$_{16}$	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 80 Structure of EP10 buffer area set register

(6) Endpoint 11

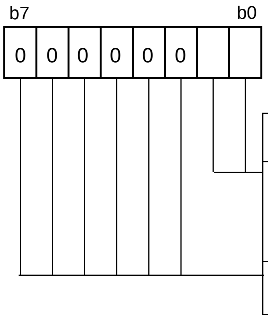


EP11 set register (EP11CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
b2:b0	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
SQCL11	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	–		
b4	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
DIR11	Transfer direction bit	0 : IN transfer disabled 1 : IN (Data is transmitted to the host.)	0	–	O	O
b6	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
TYP11	Transfer type bite	0 : Transfer disabled 1 : Interrupt transfer	0	–	O	O

–: State remaining

Fig. 81 Structure of EP11 set register

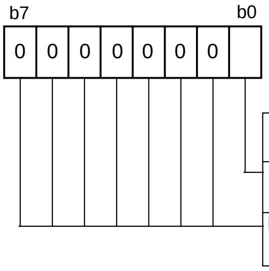


EP11 control register 1 (EP11CON1) [address 001A$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PID11 [1:0]	Response PID bit	b1 b0 0 0 : NAK 0 1 : Automatic response (NAK, DATA0, DATA1) 1 X : STALL	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 82 Structure of EP11 control register 1

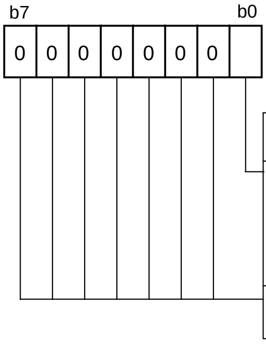


EP11 control register 2 (EP11CON2) [address 001B$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
B0VAL11	Buffer 0 status bit	This bit set to “1” shows the transmitting data is in a set state (SIE is possible to read).	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 83 Structure of EP11 control register 2

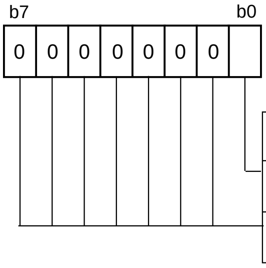


EP11 interrupt source register (EP11REQ) [address 001D$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0RDY11	USB HUB/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 is ready state (enabled to be read/written) on USB HUB/Endpoint 1. “0” can be set by software, but “1” cannot be set.	0	0	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 84 Structure of EP11 interrupt source register

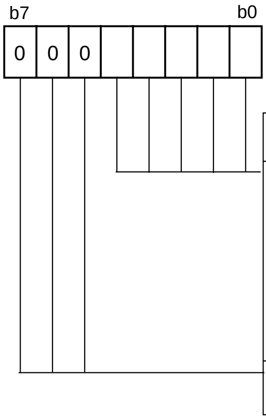


EP11 byte number register (EP11BYT0) [address 001E$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
B0BYT11	Transmit byte number bit	IN : Set the transmitting byte number.	0	—	O	O
b7:b1	Not used	Write 0 when writing. 0 is read when reading.	—	—	O	O

—: State remaining

Fig. 85 Structure of EP11 byte number register



EP11 buffer area set register (EP11BUF) [address 0FED$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
BADD11 [4:0]	EP11 beginning address set bit	Set the beginning address of EP11’s buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040$_{16}$ 0 0 0 1 1 : 0060$_{16}$ .... 1 1 1 1 0 : 03C0$_{16}$ 1 1 1 1 1 : 03E0$_{16}$	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 86 Structure of EP11 buffer area set register

HUB FUNCTION

The 38K2 Group has a HUB Function Control Circuit (HUBFCC) that offers easy implementation of USB-hub functions (signal repeat and bus state detection). This circuit is in compliance with USB Specification Version 2.0 Full-Speed/Low-Speed Transfer Modes (12 Mbps/1.5 Mbps, equivalent to Version 1.1).

The HUBFCC operates with two external down-ports and one internal down-port, which is utilized by the USB addresses of the built-in peripherals, enabling management of a total of three down-ports independently.

A dedicated circuit automatically performs the bus state change detection and error detection needed for the sequence management of the hub repeater circuit, data repeat function, and down-port status management. This dedicated control circuit ensures the user easy development of a program or timing design.

Each down-port register can be controlled by USB commands using USB addresses for HUB functions or detecting changes in the bus state of down-ports. The HUBFCC is also equipped with a remote wakeup signal transfer function for use during global resume as other special signals management. The HUBFCC generates an interrupt to the CPU when detecting a down-port state change (1 vector, 10 sources).

The flexibility of the indispensable yet wide-ranging HUBFCC structure and an external interrupt function and I/O ports implemented in the standard features of this MCU enable the power supply management essential for USB-HUB functions and also allow users to easily and effortlessly configure their optimum system.

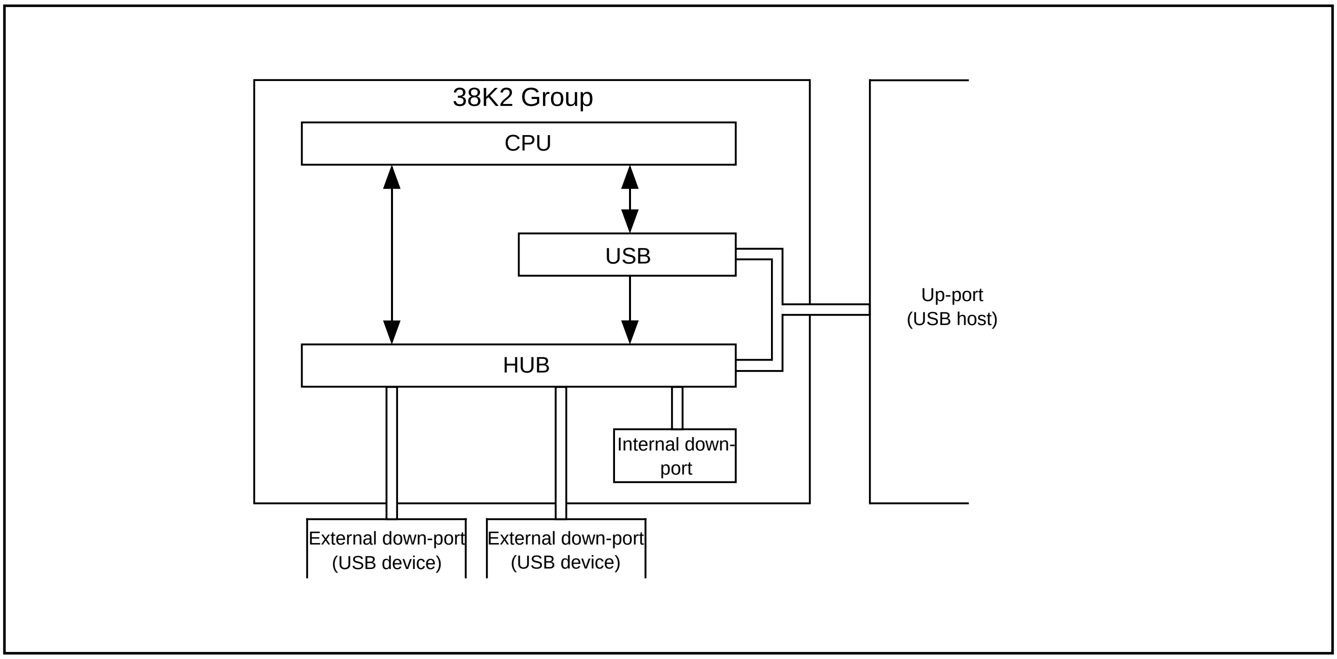


Fig. 87 HUB functions

HUB Function Control Circuit Block Diagram

The HUB function control circuit, as show in the diagram below, consists of the following blocks.

(1) HUB repeater block
(2) Down-port control block
(3) CPU interface block (CIF)

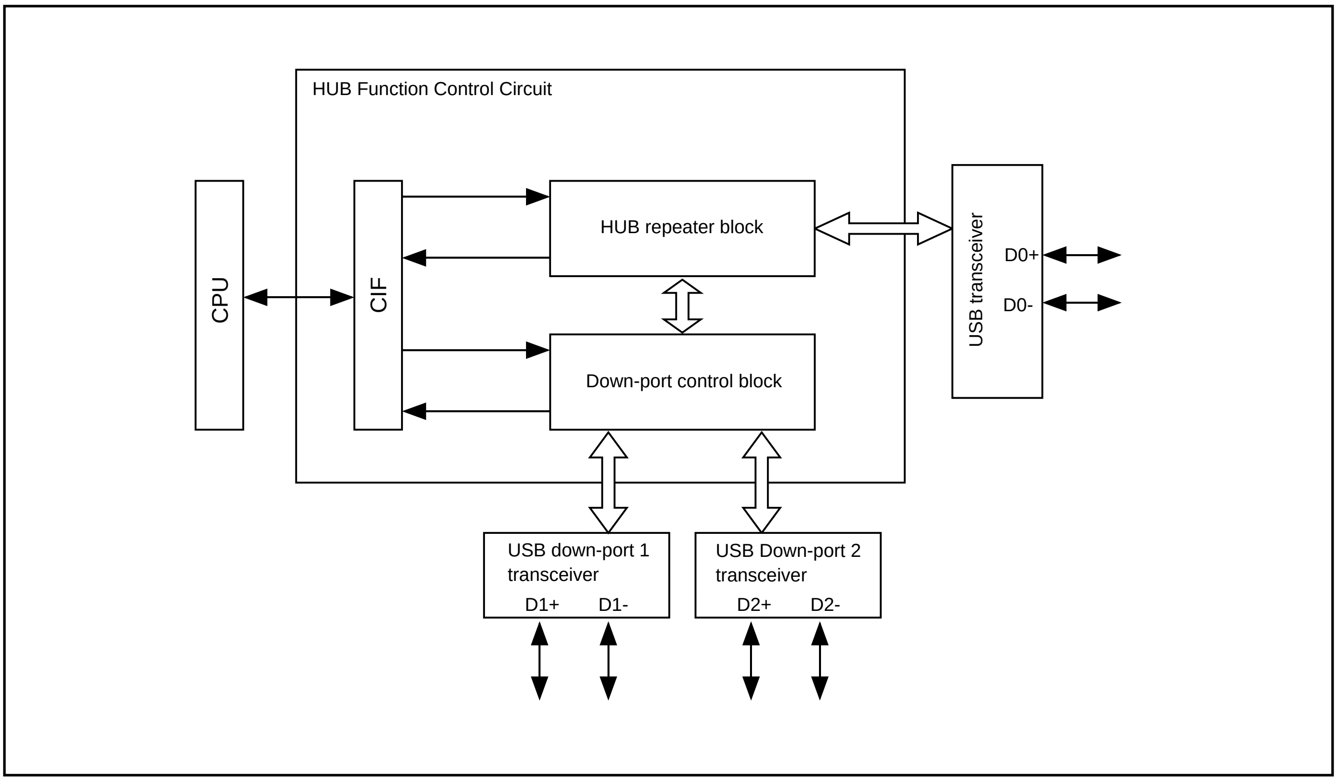

HUB Function Control Circuit
CPU
CIF
HUB repeater block
Down-port control block
USB transceiver
D0+
D0-
USB down-port 1 transceiver
D1+
D1-
USB Down-port 2 transceiver
D2+
D2-

Fig. 88 HUB function control circuit block diagram

(1) HUB repeater block

The HUB repeater block, consisting of the circuits listed below, processes the HUB repeater function sequence. The HUB repeater is ready for operation after enabling the USB module (USBE = “1”).

- •Repeater circuit (detects SOP/EOP signal)
- •Frame-time circuit (synchronizes to SOF signal and manages frames in 1 ms)
- •Receiver circuit (manages up-port states)
- •Transmitter circuit (controls up-port outputs)

(2) Down-port control block

The down-port control block, consisting of the circuits listed below, performs down-port controls under supervision of the HUB repeater state operation.

- •Down-port sequencer circuit
- •Down-port state change detect circuit

(3) CPU interface block (CIF)

The CPU interface block performs the following processes.

- •Control of repeater/down-port states through registers.
- •Generates interrupt signal
- •Controls internal bus interface

USB Down-port Peripheral Circuit Setting

The USB down-port peripheral circuits can be set with the downstream port control register (address 0FF9$_{16}$). Figures 89 and 90 show the circuit block diagrams.

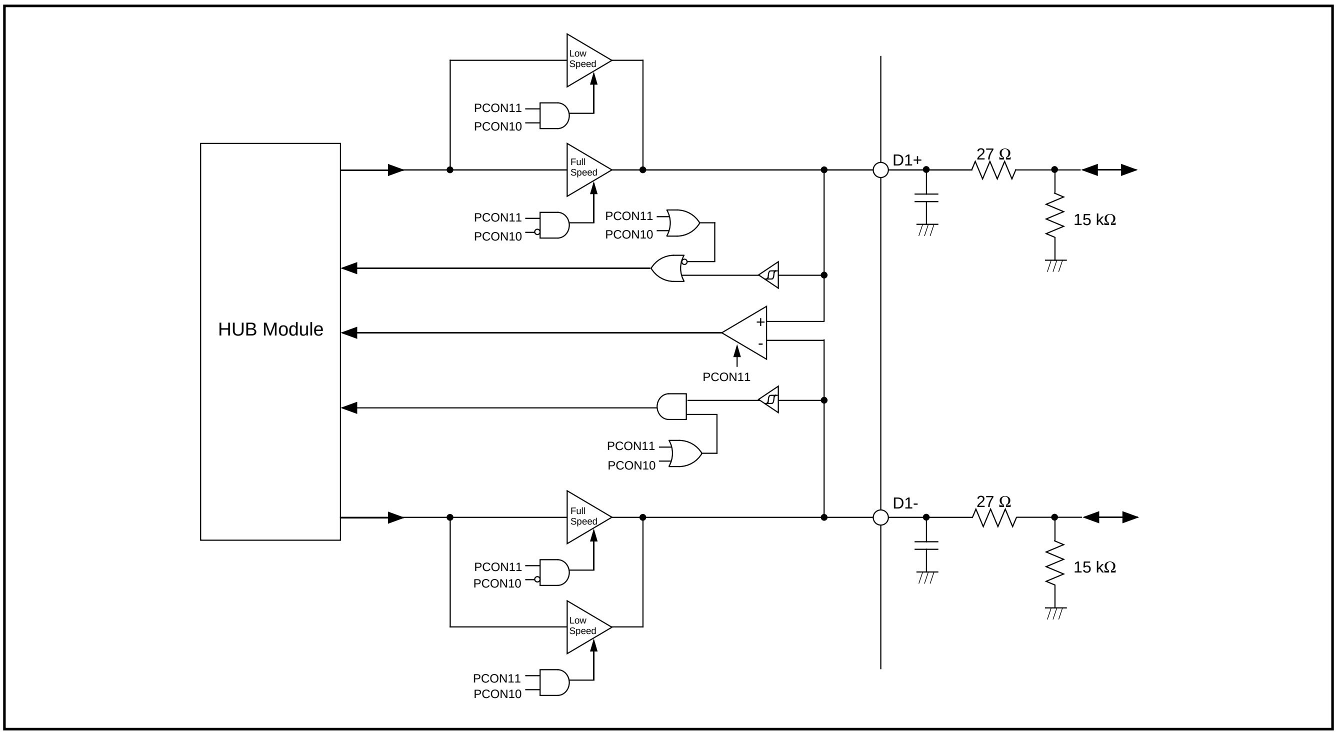


Fig. 89 Block diagram of USB down-port peripheral circuits (D1+, D1-)

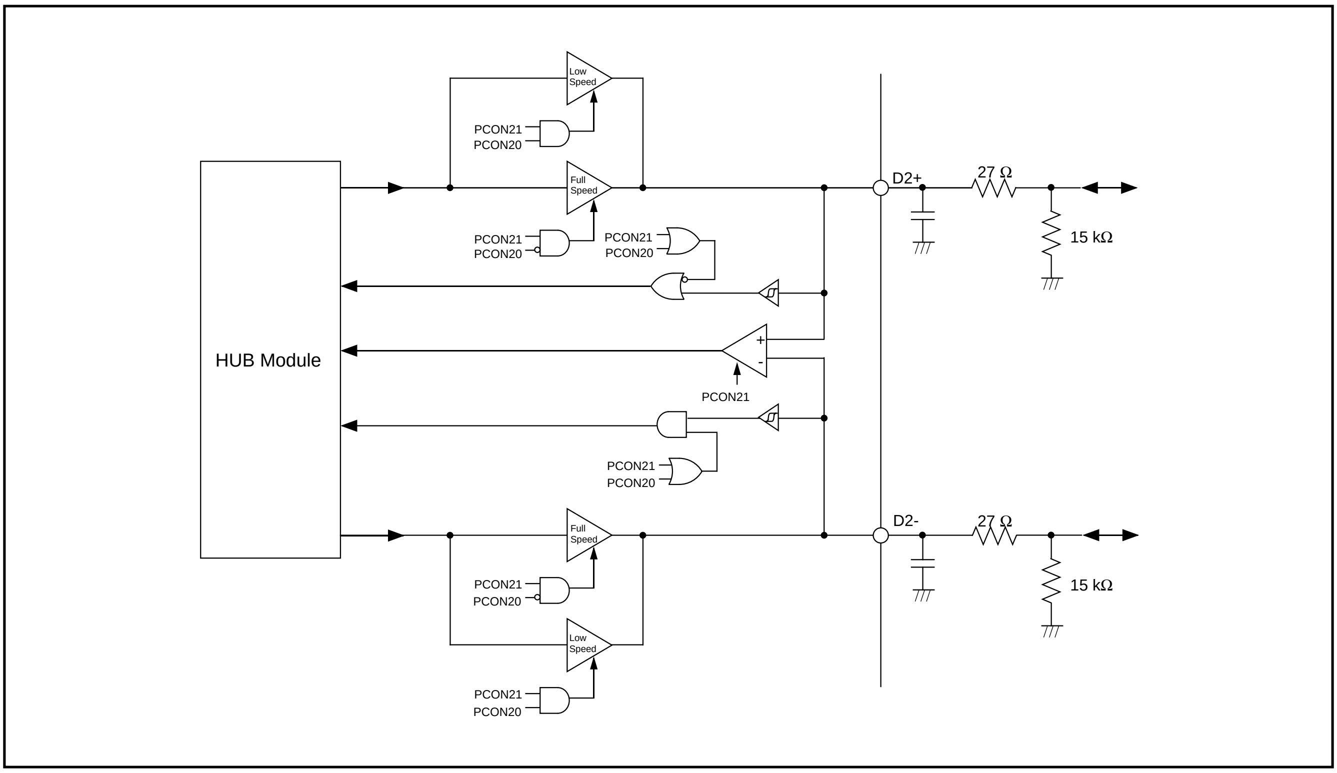


Fig. 90 Block diagram of USB down-port peripheral circuits (D2+, D2-)

HUB Interrupt Function

The HUB function control circuit has one interrupt request consisting of 10 interrupt sources each of which can be determined through the interrupt source register. Table 8 shows the HUB interrupt sources.

Table 8 HUB interrupt sources

Interrupt request bit (IREQ2: Address $003D_{16}$)	HUB interrupt bit (HUBIREQ: Address 0029_{16})	Interrupt source
USB HUB	DP1	At HUB down-port 1 state change detected: •Disconnected state detected •Connected state detected •Port error state detected •Resume signal detected •Bus state change detected
	DP2	At HUB down-port 2 state change detected: •Disconnected state detected •Connected state detected •Port error state detected •Resume signal detected •Bus state change detected

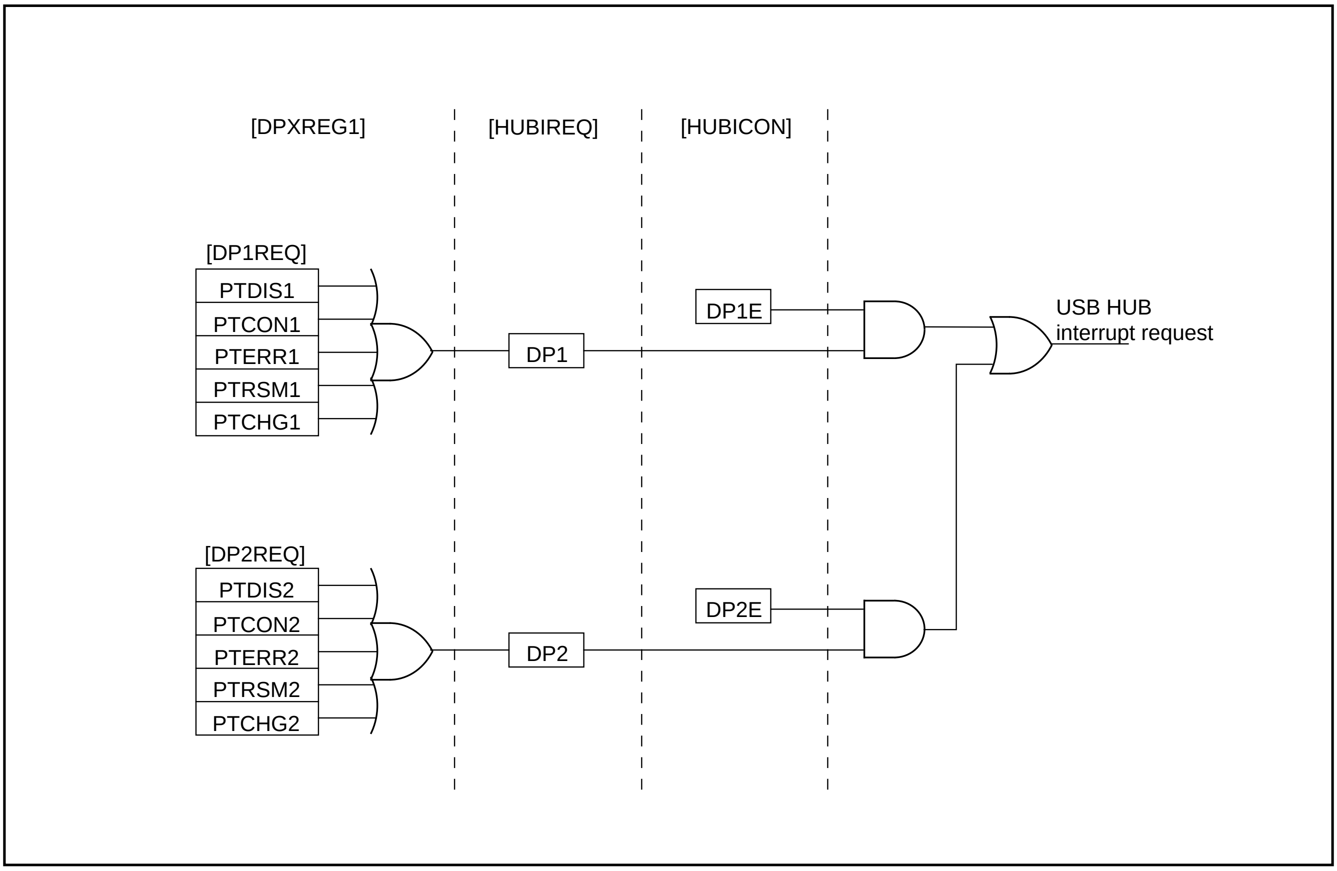


Fig. 91 USB HUB interrupt control

HUB Register List

The HUB register list is shown below.

Address	Register Name	SYMBOL	USB SFR							
			bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
0028_{16}	HUB interrupt source enable register	HUBICON	HRWUE	☒	☒	☒	☒	☒	DP2E	DP1E
0029_{16}	HUB interrupt source register	HUBIREQ	HRWU	☒	☒	☒	☒	☒	DP2	DP1
$002A_{16}$	HUB downstream port index register	HUBINDEX	☒	☒	☒	☒	☒	☒	☒	DPIDX
$002B_{16}$	HUB port field register 1	DPXREG1								
$002C_{16}$	HUB port field register 2	DPXREG2								
$002D_{16}$	HUB port field register 3	DPXREG3								

(1) HUB port 1

Address	Register Name	SYMBOL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$002B_{16}$	DP1 interrupt source register	DP1REQ	☒	☒	☒	PTCHG1	PTRSM1	PTERR1	PTCON1	PTDIS1
$002C_{16}$	DP1 control register	DP1CON	DSLSPD1	DSRMOD1	DSRSMO1	DSRSTO1	DSDETE1	DSSUSP1	DSPTEN1	DSCONN1
$002D_{16}$	DP1 status register	DP1STS	☒	☒	☒	☒	☒	☒	D1PLUS	D1MINUS

(2) HUB port 2

Address	Register Name	SYMBOL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$002B_{16}$	DP2 interrupt source register	DP2REQ	☒	☒	☒	PTCHG2	PTRSM2	PTERR2	PTCON2	PTDIS2
$002C_{16}$	DP2 control register	DP2CON	DSLSPD2	DSRMOD2	DSRSMO2	DSRSTO2	DSDETE2	DSSUSP2	DSPTEN2	DSCONN2
$002D_{16}$	DP2 status register	DP2STS	☒	☒	☒	☒	☒	☒	D2PLUS	D2MINUS

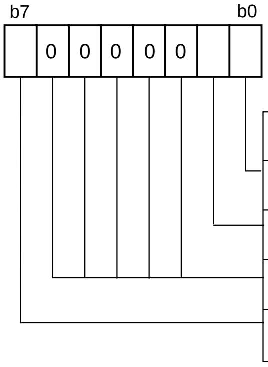
Address	Register Name	SYMBOL	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
$0FF9_{16}$	Downstream port control register	DPCTL	☒	☒	☒	☒	PCON2[1:0]		PCON1[1:0]	

☒ : Not used

Fig. 92 HUB related registers

HUB Related Registers

The HUB related registers are shown below.

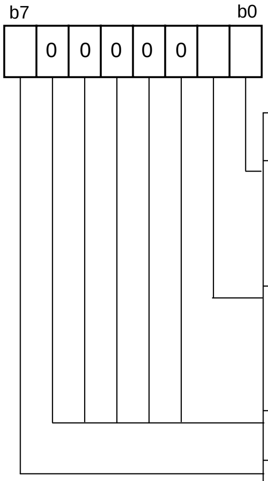


HUB interrupt source enable register (HUBICON) [address 0028$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DP1E	HUB downstream port 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	–	O	O
DP2E	HUB downstream port 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	–	O	O
b6:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
HRWUE	HUB upstream port remote-wakeup output enable bit	0 : Disabled 1 : Enabled	0	–	O	O

–: State remaining

Fig. 93 Structure of HUB interrupt source enable register

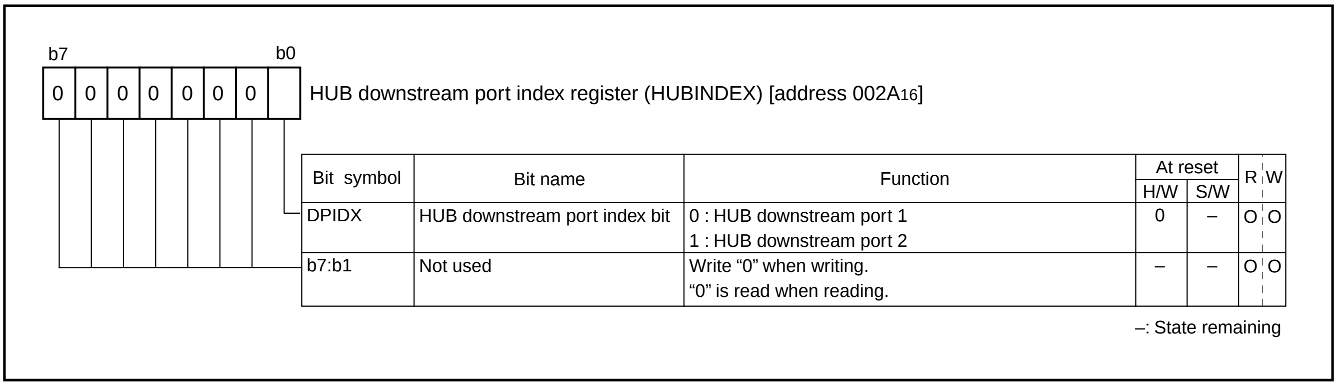


HUB interrupt source register (HUBIREQ) [address 0029$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DP1	HUB downstream port 1 interrupt bit	This bit is set to “1” when any one of DP1 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing DP1 interrupt source register to “00$_{16}$”. Writing to this bit causes no state change.	0	–	O	×
DP2	HUB downstream port 1 interrupt bit	This bit is set to “1” when any one of DP2 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing DP2 interrupt source register to “00$_{16}$”. Writing to this bit causes no state change.	0	–	O	×
b6:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
HRWU	HUB upstream port remote -wakeup output enable bit	0 : Remote-wakeup being not output 1 : Remote-wakeup being output This bit change is not for a interrupt source. When detecting 2.5 µs or more of K-signal on a downstream port in Hub-suspended state, K-signal is output on from a upstream port and this bit is simultaneously set to “1”. “0” can be set by software, but “1” cannot be set.	0	–	O	O

–: State remaining

Fig. 94 Structure of HUB interrupt source register



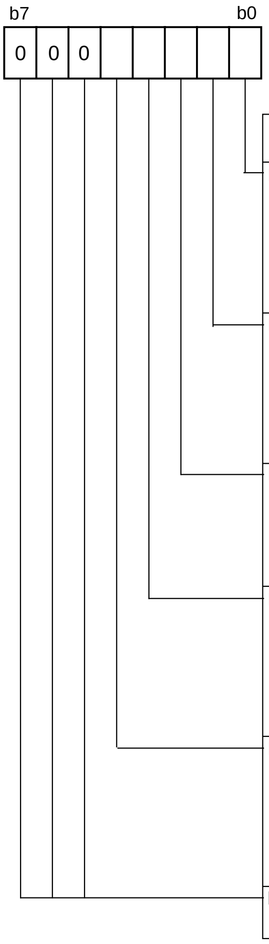
HUB downstream port index register (HUBINDEX) [address $002A_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DPIDX	HUB downstream port index bit	0 : HUB downstream port 1 1 : HUB downstream port 2	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 95 Structure of HUB downstream port index register

(1) Downstream port 1

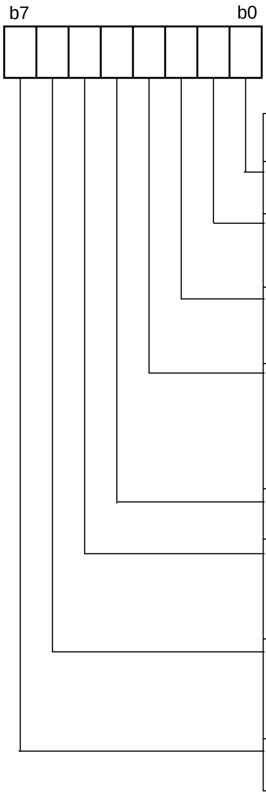


DP1 interrupt source register (DP1REQ) [address $002B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PTDIS1	Downstream port 1 disconnect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-disconnect state (2.5 µs or more of SE0) on a downstream port 1 in DSCONN1 = “1”. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTCON1	Downstream port 1 connect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-connect state (2.5 µs or more of J- or K- state) on a downstream port 1 in DSCONN1 = “0”. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTERR1	Downstream port 1 port error interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when an error occurs on a downstream port 1. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTRSM1	Downstream port 1 resume interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a resume signal on a downstream port 1 in the condition of HUB suspended or port suspended state. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTCHG1	Downstream port 1 bus-change detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-change of a downstream port 1 in the condition of HUB suspended state. It is also “1” in the internal clock halted. “0” can be set by software, but “1” cannot be set.	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 96 Structure of DP1 interrupt source register

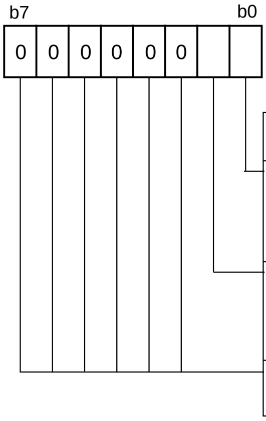


DP1 control register (DP1CON) [address 002C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DSCONN1	Downstream port 1 connect bit	0 : Disconnect ; PTCON1 interrupt enabled 1 : Connect ; PTDIS1 interrupt enabled	0	–	O	O
DSPTEN1	Downstream port 1 enable bit	0 : Downstream port 1 disabled 1 : Downstream port 1 enabled ; This bit is cleared when an interrupt of PTDIS1 or PTERR1 is generated.	0	–	O	O
DSSUSP1	Downstream port 1 suspend bit	0 : No port suspended 1 : Port suspended; This bit is cleared when an interrupt of PTDIS1 or PTRSM1 is generated.	0	–	O	O
DSDETE1	Downstream port 1 connect-state detection enable bit	0 : Connect/disconnect-state detection disabled ; PTCON1 and PTDIS1 interrupts disabled 1 : Connect/disconnect-state detection enabled ; This bit is cleared when an interrupt of PTCON1, PTDIS1 or PTERR1 is generated.	0	–	O	O
DSRSTO1	Downstream port 1 SE0 signal transmit bit	0 : Being not output 1 : SE0 signal being output	0	–	O	O
DSRSMO1	Downstream port 1 resume signal transmit bit	0 : Being not output 1 : K-signal being output ; When writing “0”, a low-speed EOP is output and then a transition to being not output occurs.	0	–	O	O
DSRMOD1	Downstream port 1 bus-state read mode control bit	0 : Mode where a downstream port 1 bus-state is read, using RD signal 1 : Mode where a downstream port 1 bus-state is read, using EOF2 signal (internal signal)	0	–	O	O
DSLSPD1	Downstream port 1 USB transfer	0 : Full-speed mode (12MHz) 1 : Low-speed mode (1.5 MHz)	0	–	O	O

–: State remaining

Fig. 97 Structure of DP1 control register



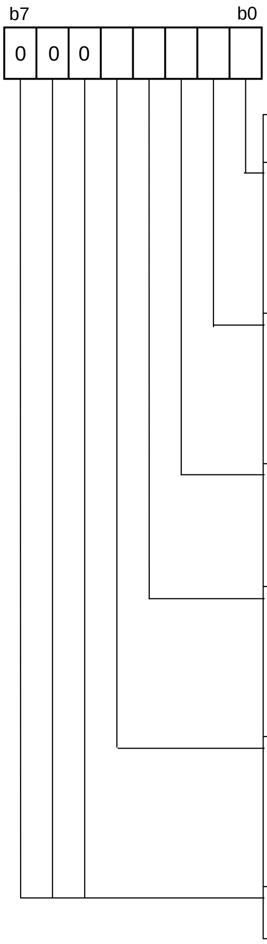
DP1 status register (DP1STS) [address 002D16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
D1MINUS	D1- signal bit	In DSRMOD1 = “0”, a downstream port 1 bus-state is read, using RD signal. In DSRMOD1 = “1”, a downstream port 1 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	O	X
D1PLUS	D1+ signal bit	In DSRMOD1 = “0”, a downstream port 1 bus-state is read, using RD signal. In DSRMOD1 = “1”, a downstream port 1 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	O	X
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 98 Structure of DP1 status register

(2) Downstream port 2

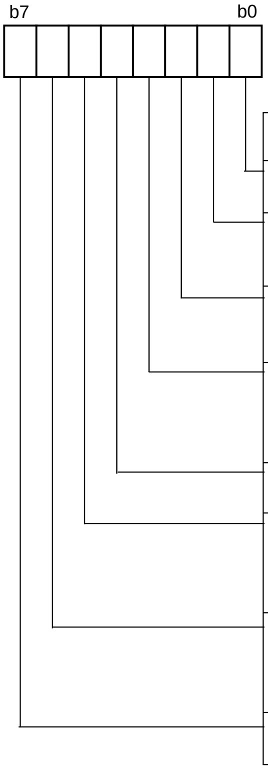


DP2 interrupt source register (DP2REQ) [address $002B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PTDIS2	Downstream port 2 disconnect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-disconnect state (2.5 µs or more of SE0) on a downstream port 2 in DSCONN2 = “1”. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTCON2	Downstream port 2 connect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-connect state (2.5 µs or more of J- or K- state) on a downstream port 2 in DSCONN2 = “0”. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTERR2	Downstream port 2 port error interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when an error occurs on a downstream port 2. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTRSM2	Downstream port 2 resume interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a resume signal on a downstream port 2 in the condition of HUB suspended or port suspended state. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTCHG2	Downstream port 2 bus-change detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-change of a downstream port 2 in the condition of HUB suspended state. It is also “1” in the internal clock halted. “0” can be set by software, but “1” cannot be set.	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 99 Structure of DP2 interrupt source register

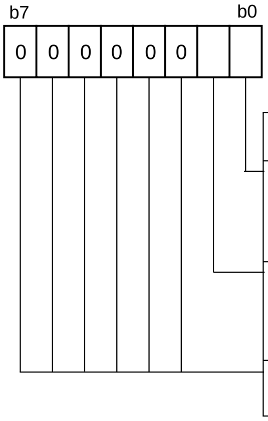


DP2 control register (DP2CON) [address 002C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DSCONN2	Downstream port 2 connect bit	0 : Disconnect ; PTCON2 interrupt enabled 1 : Connect ; PTDIS2 interrupt enabled	0	–	○	○
DSPTEN2	Downstream port 2 enable bit	0 : Downstream port 2 disabled 1 : Downstream port 2 enabled ; This bit is cleared when an interrupt of PTDIS2 or PTERR2 is generated.	0	–	○	○
DSSUSP2	Downstream port 2 suspend bit	0 : No port suspended 1 : Port suspended; This bit is cleared when an interrupt of PTDIS2 or PTRSM2 is generated.	0	–	○	○
DSDETE2	Downstream port 2 connect-state detection enable bit	0 : Connect-state detection disabled ; PTCON2 and PTDIS2 interrupts disabled 1 : Connect-state detection enabled ; This bit is cleared when an interrupt of PTCON2, PTDIS2 or PTERR2 is generated.	0	–	○	○
DSRSTO2	Downstream port 2 SE0 signal transmit bit	0 : Being not output 1 : SE0 signal being output	0	–	○	○
DSRSMO2	Downstream port 2 resume signal transmit bit	0 : Being not output 1 : K-signal being output ; When writing "0", a low-speed EOP is output and then a transition to being not output occurs.	0	–	○	○
DSRMOD2	Downstream port 2 bus-state read mode control bit	0 : Mode where a downstream port 2 bus-state is read, using RD signal 1 : Mode where a downstream port 2 bus-state is read, using EOF2 signal (internal signal)	0	–	○	○
DSLSPD2	Downstream port 2 USB transfer speed select bit	0 : Full-speed mode (12MHz) 1 : Low-speed mode (1.5 MHz)	0	–	○	○

–: State remaining

Fig. 100 Structure of DP2 control register

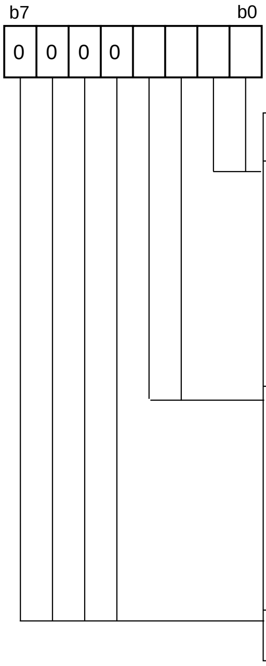


DP2 status register (DP2STS) [address 002D16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
D2MINUS	D2- signal bit	In DSRMOD2 = "0", a downstream port 2 bus-state is read, using RD signal. In DSRMOD2 = "1", a downstream port 2 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	○	×
D2PLUS	D2+ signal bit	In DSRMOD2 = "0", a downstream port 2 bus-state is read, using RD signal. In DSRMOD2 = "1", a downstream port 2 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	○	×
b7:b2	Not used	Write "0" when writing. "0" is read when reading.	–	–	○	○

–: State remaining

Fig. 101 Structure of DP2 status register



Downstream port control register (DPCTL) [address 0FF9$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PCON1 [1:0]	Downstream port 1 function select bit	b1b0 0 0 : USB port (D1-, D1+) OFF, USB difference amplifier OFF 0 1 : USB exclusive input port (D1-, D1+), USB difference amplifier OFF 1 0 : Full-speed port (D1-, D1+), USB difference amplifier ON 1 1 : Low-speed port (D1-, D1+), USB difference amplifier ON	0	–	O	O
PCON2 [1:0]	Downstream port 2 function select bit	b3b2 0 0 : USB port (D2-, D2+) OFF, USB difference amplifier OFF 0 1 : USB exclusive input port (D2-, D2+), USB difference amplifier OFF 1 0 : Full-speed port (D2-, D2+), USB difference amplifier ON 1 1 : Low-speed port (D2-, D2+), USB difference amplifier ON	0	–	O	O
b7:b4	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 102 Structure of Downstream port control register

EXTERNAL BUS INTERFACE (EXB)

The external bus interface (EXB) controls the data transfer between the external MCU and the 38K2 group's CPU or its memory (multichannel RAM). The external bus interface is shown below.

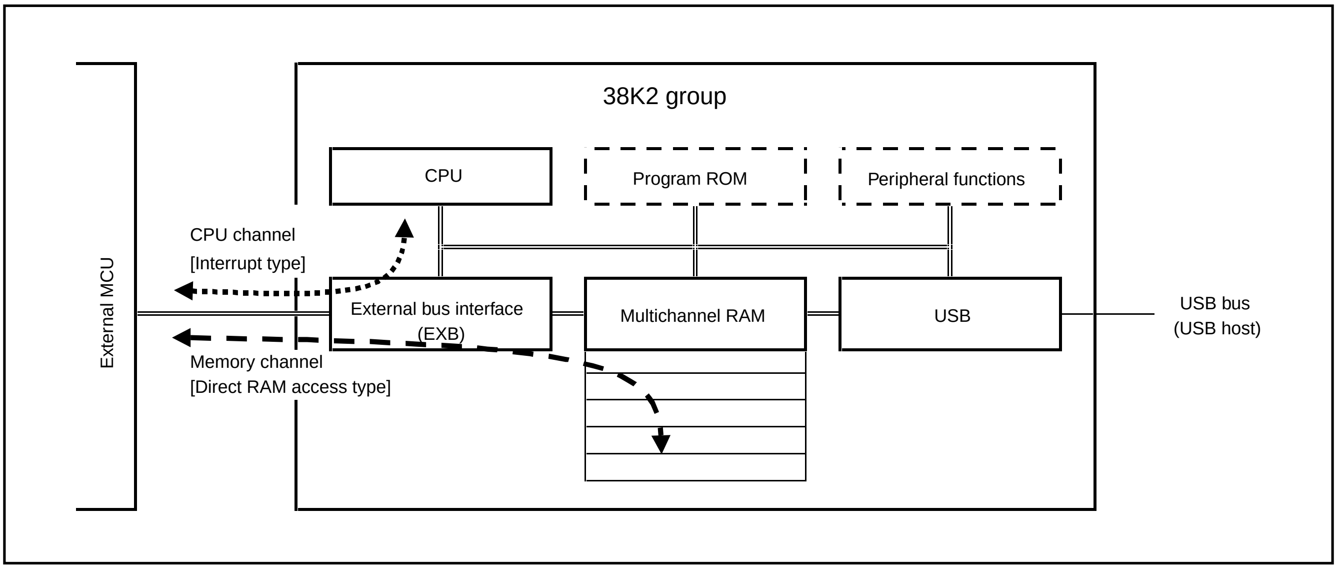


Fig. 103 External bus interface

●CPU channel

It is a data transfer course by the interrupt processing between the external MCU and the 38K2 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 38K2 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.

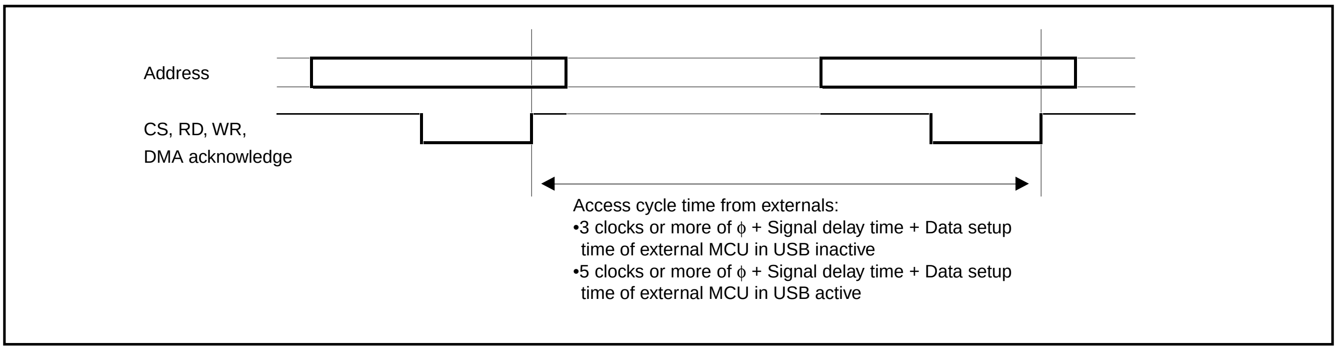

Address
CS, RD, WR,
DMA acknowledge
Access cycle time from externals:
•3 clocks or more of ϕ + Signal delay time + Data setup
time of external MCU in USB inactive
•5 clocks or more of ϕ + Signal delay time + Data setup
time of external MCU in USB active

Fig. 104 Data transfer timing of memory channel

EXB Pin Assignment

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

The 38K2 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 38K2 group can be set up per one by the program.

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

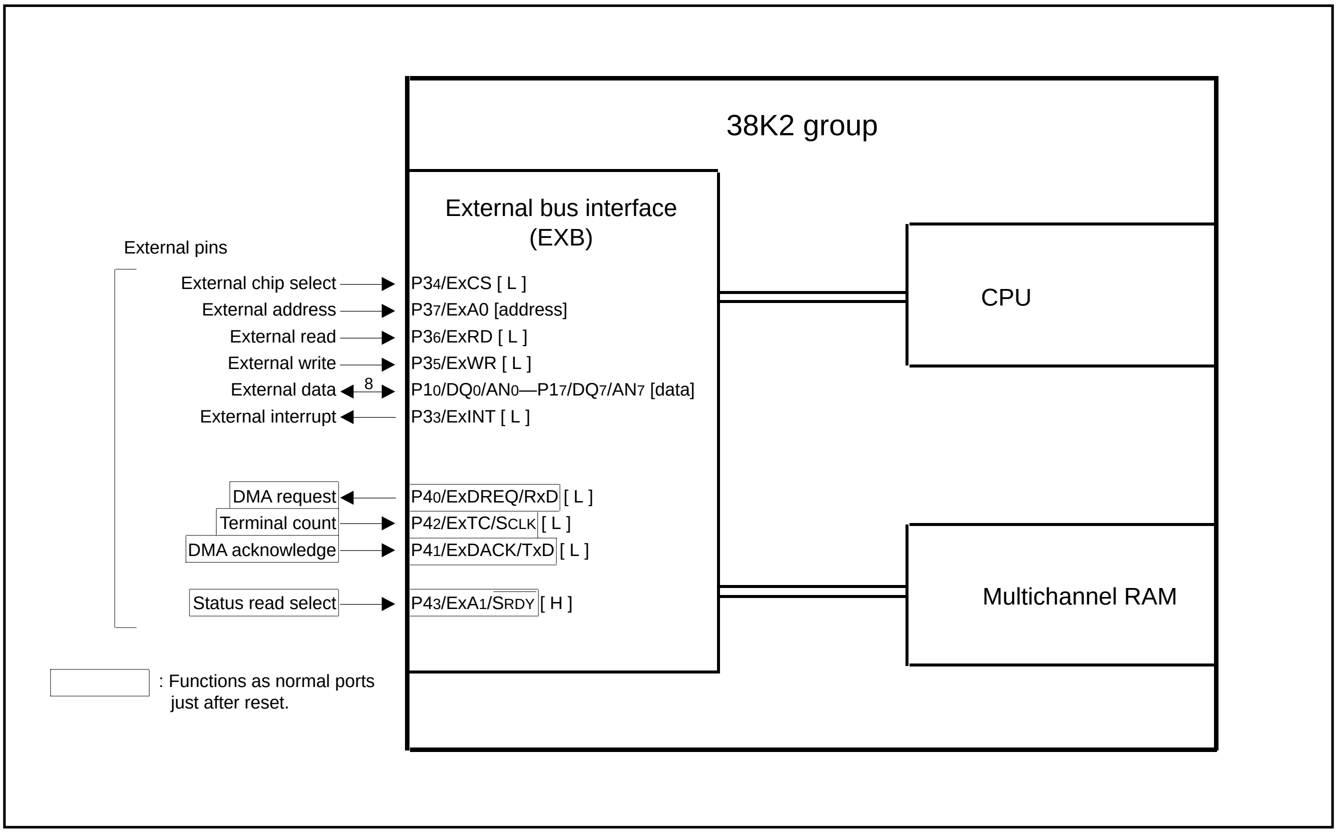


Fig. 105 External bus interface (EXB) pin assignment

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

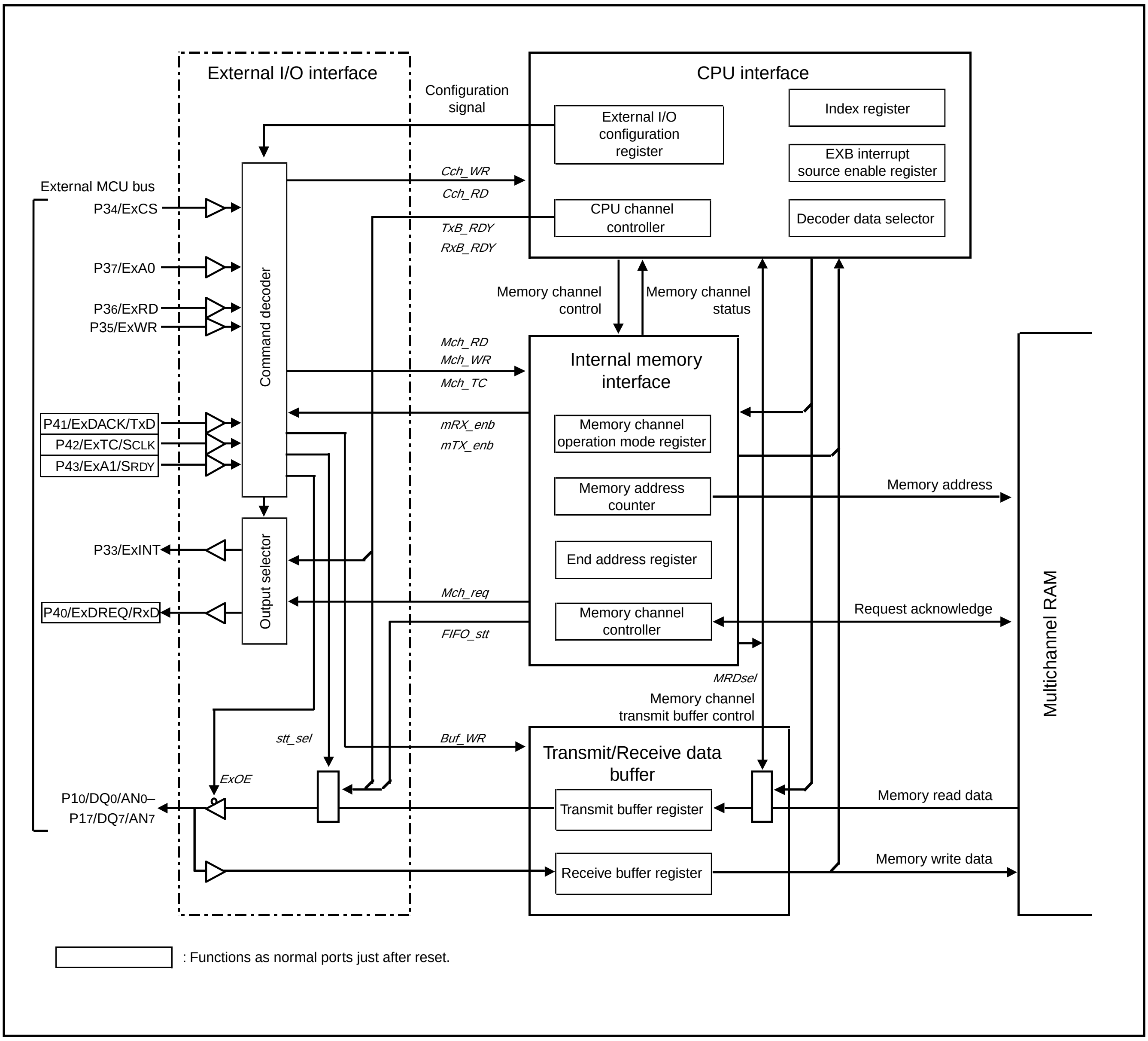


Fig. 106 Block diagram of external bus interface (EXB)

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

•Status read select	P4$_3$/ExA1/$\overline{\text{SRDY}}$

●CPU channel: Communication with 38K2 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 38K2 group CPU can confirm its access. The 38K2 group CPU judges the interrupt source and it starts a data transmission/reception with an external MCU bus.

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

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

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

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

●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 38K2 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 124 and 127.)

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 125 and 128.)

●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 38K2 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 126.)

EXB Register List

The EXB register list is shown below.

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

☒ : Not used
0 : "0" fixed

Fig. 107 EXB related registers (1)

•EXB interrupt source enable register

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

•EXB interrupt source register

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

•EXB index register/Register windows 1, 2

The accessible register is switched by treating addresses 0034_{16} and 0035_{16} as a register window depending on the value of EXB index register at address 0033_{16}.

Index	low high	Register Name	SYMBOL	EXB SFR							
				bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00_{16}	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]	
01_{16}	low	Transmit/Receive buffer register	RXBUF/TXBUF	At CPU read : RXBUF[7:0] At CPU write : TXBUF[7:0]							
	high		—	☒	☒	☒	☒	☒	☒	☒	☒
02_{16}	low	Memory channel operation mode register	MCHMOD	☒	☒	☒	☒	☒	BURST	MC_DIR[1:0]	
	high		—	☒	☒	☒	☒	☒	☒	☒	☒
03_{16}	low	Memory address counter	MEMADL	IM_A[7:0]							
	high		MEMADH	0	0	0	0	0	IM_A[10:8]		
04_{16}	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. 108 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

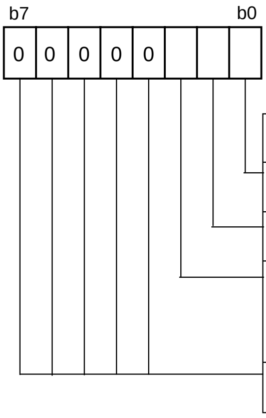
This is a counter to set the beginning address which FIFO accesses. This register is increased by access from the external MCU bus.

•End address register

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

EXB Related Registers

The EXB related registers are shown below.



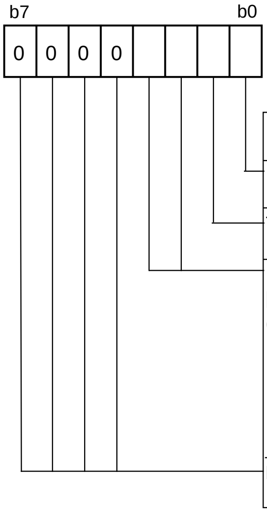
EXB interrupt source enable register (EXBICON) [address 003016] (**Note**)

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 109 Structure of EXB interrupt source enable register



EXB interrupt source register (EXBIREQ) [address 003116] (**Note 1**)

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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”.

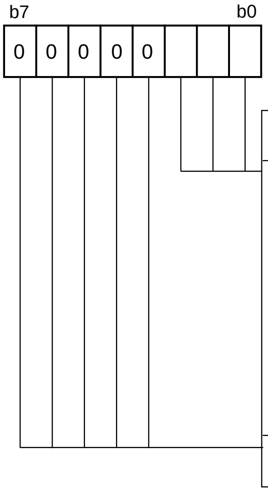
2: The memory channel status bits indicate the status of memory channel. In MC_ENB = “0” these bits are always “00$_2$”. When the memory channel operation ends, these bits are set to “11$_2$” and the memory channel operation end interrupt is generated.

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. 110 Structure of EXB interrupt source register

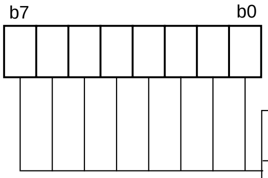


EXB index register (EXBINDEX) [address 0033_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

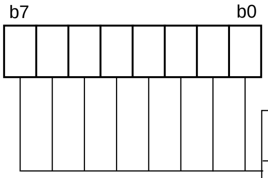
Fig. 111 Structure of EXB index register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

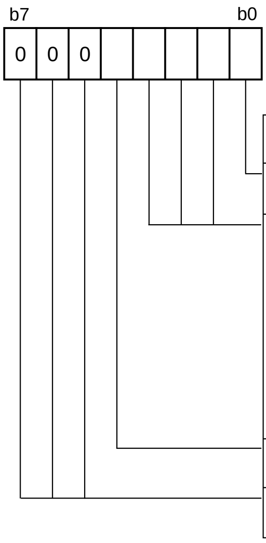
Fig. 112 Structure of Register window 1



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 113 Structure of Register window 2

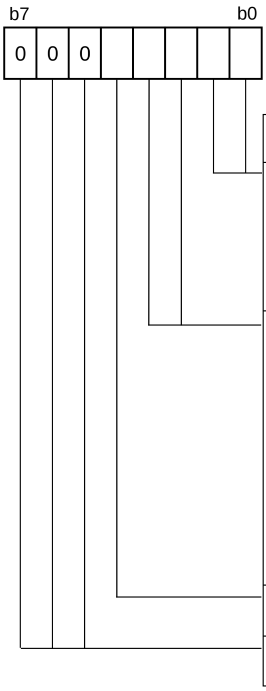


Index = 00_{16} : External I/O configuration register (EXBCFGL) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 114 Index00[low]; Structure of External I/O configuration register

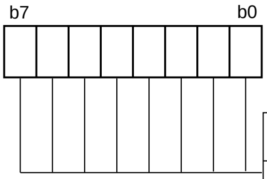


Index = 00_{16} : External I/O configuration register (EXBCFGH) [address 0035_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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/SCLK 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. 115 Index00[high]; Structure of External I/O configuration register

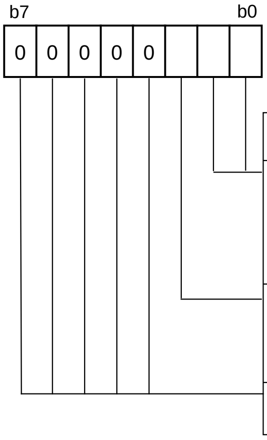


Index =01_{16} : Transmit/Receive buffer register (RXBUF/TXBUF) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 116 Index01[low]; Structure of Transmit/Receive buffer register

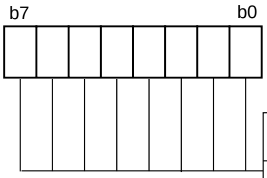


Index =02_{16} : Memory channel operation mode register (MCHMOD) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

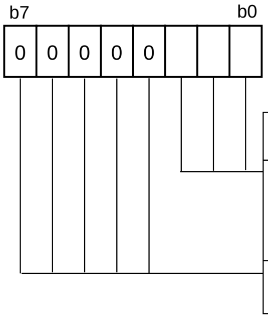
Fig. 117 Index02[low]; Structure of Memory channel operation mode register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 118 Index03[low]; Structure of Memory address counter

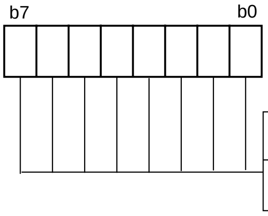


Index = 03_{16} : Memory address counter (MEMADH) [address 0035_{16}]

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. 119 Index03[high]; Structure of Memory address counter

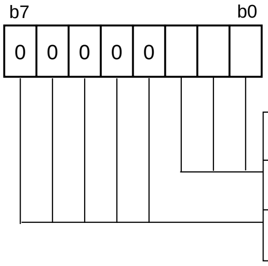


Index = 04_{16} : End address register (ENDADL) [address 0034_{16}]

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. 120 Index04[low]; Structure of End address register



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

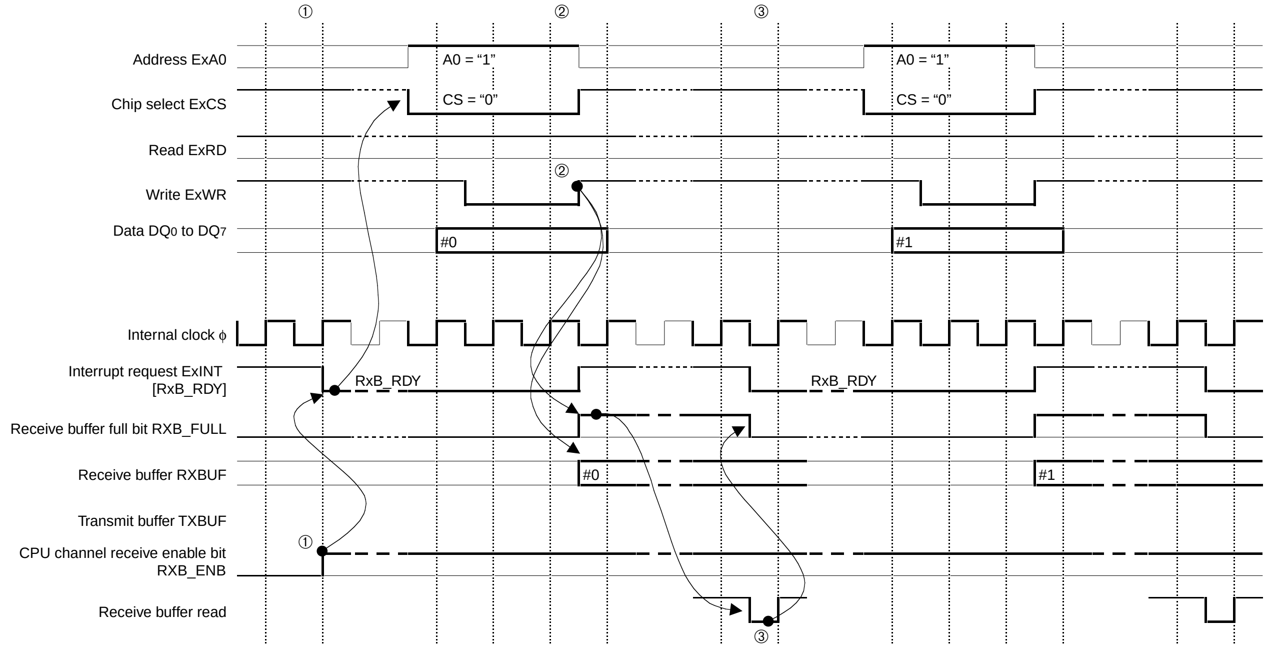
Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
END_A [10:8]	–	Register to set the high-order address of memory channel operation end.	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 121 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] ($P3_3$/ExINT pin control) = 001_2 (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 $P3_3$/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.

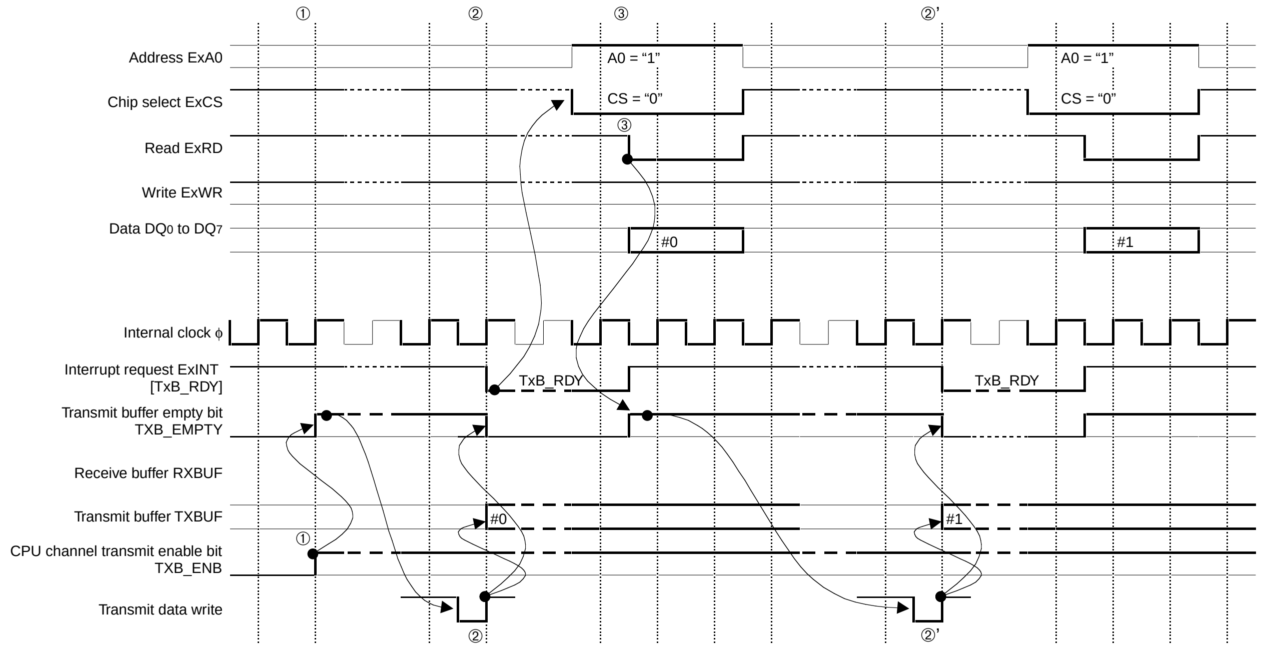
② When a write operation is performed from an external MCU bus in the condition of ExCS = “L” and WxA0 = “H”, it will result in as follows:
- The data is written into the receive buffer (RXBUF)
- Negation of the receive buffer ready signal (RxB_RDY) to an external is made
- The RXB_FULL interrupt is 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. 122 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.

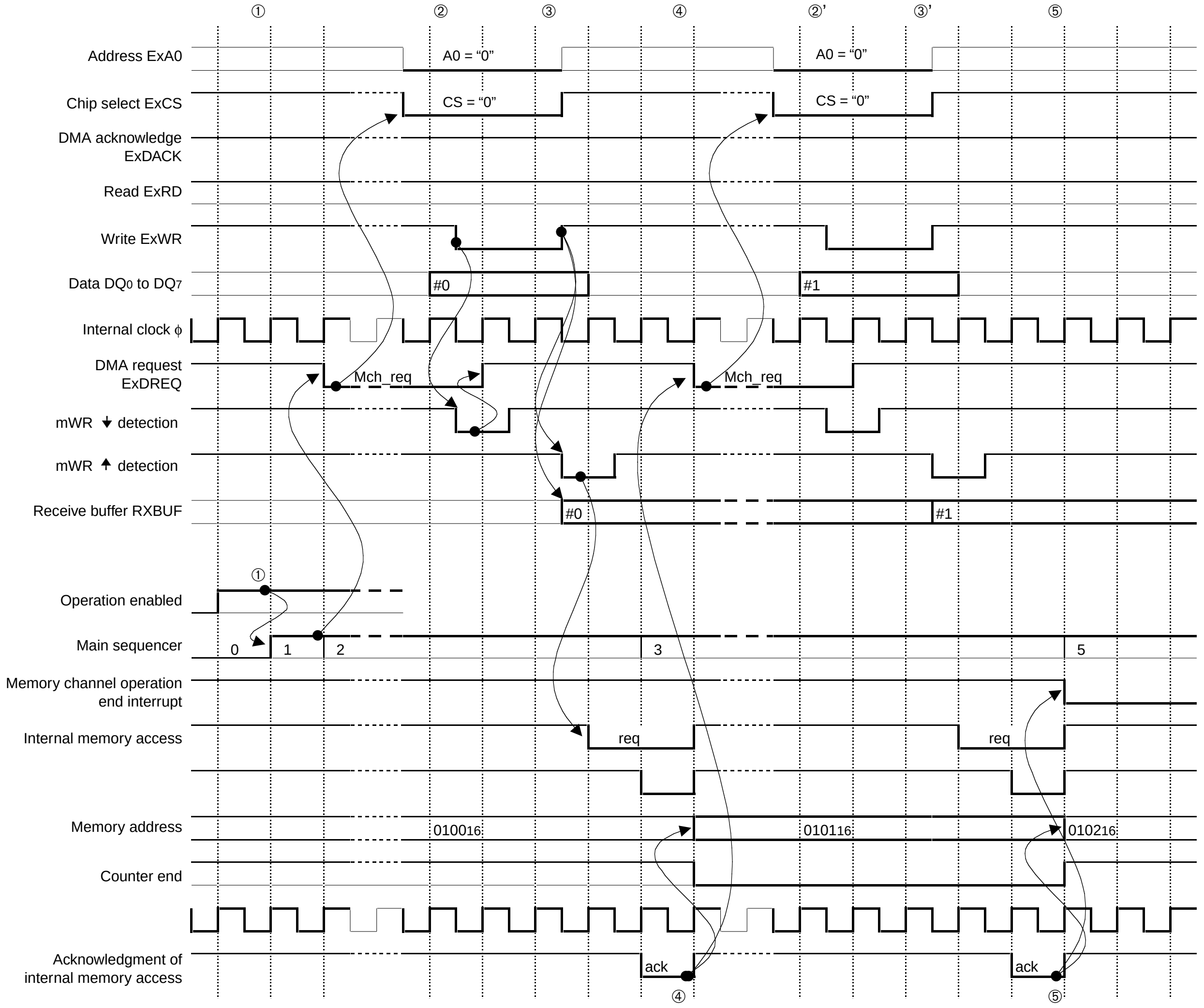
② When the CPU writes the data into the transmit buffer (TXBUF) with an interrupt processing program, the transmit buffer empty bit (TXB_EMPTY) is cleared to "0" and assertion of the transmit buffer ready signal (TxB_RDY) to an external is made.

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. 123 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.

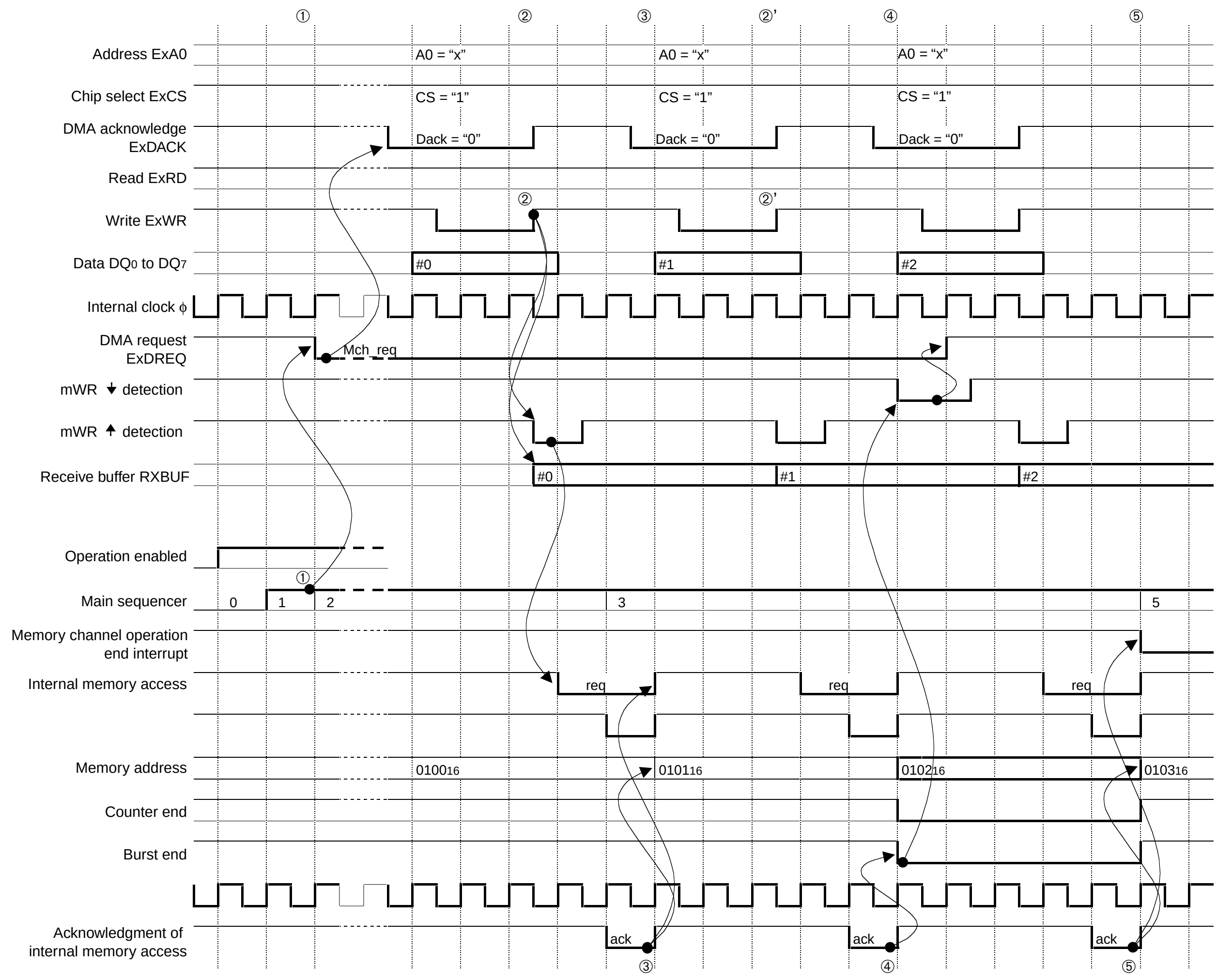
④ The memory address counter is increased simultaneously at write completion and assertion of the next memory channel request is made.

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. 124 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) = 01_2 (Receive 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 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.

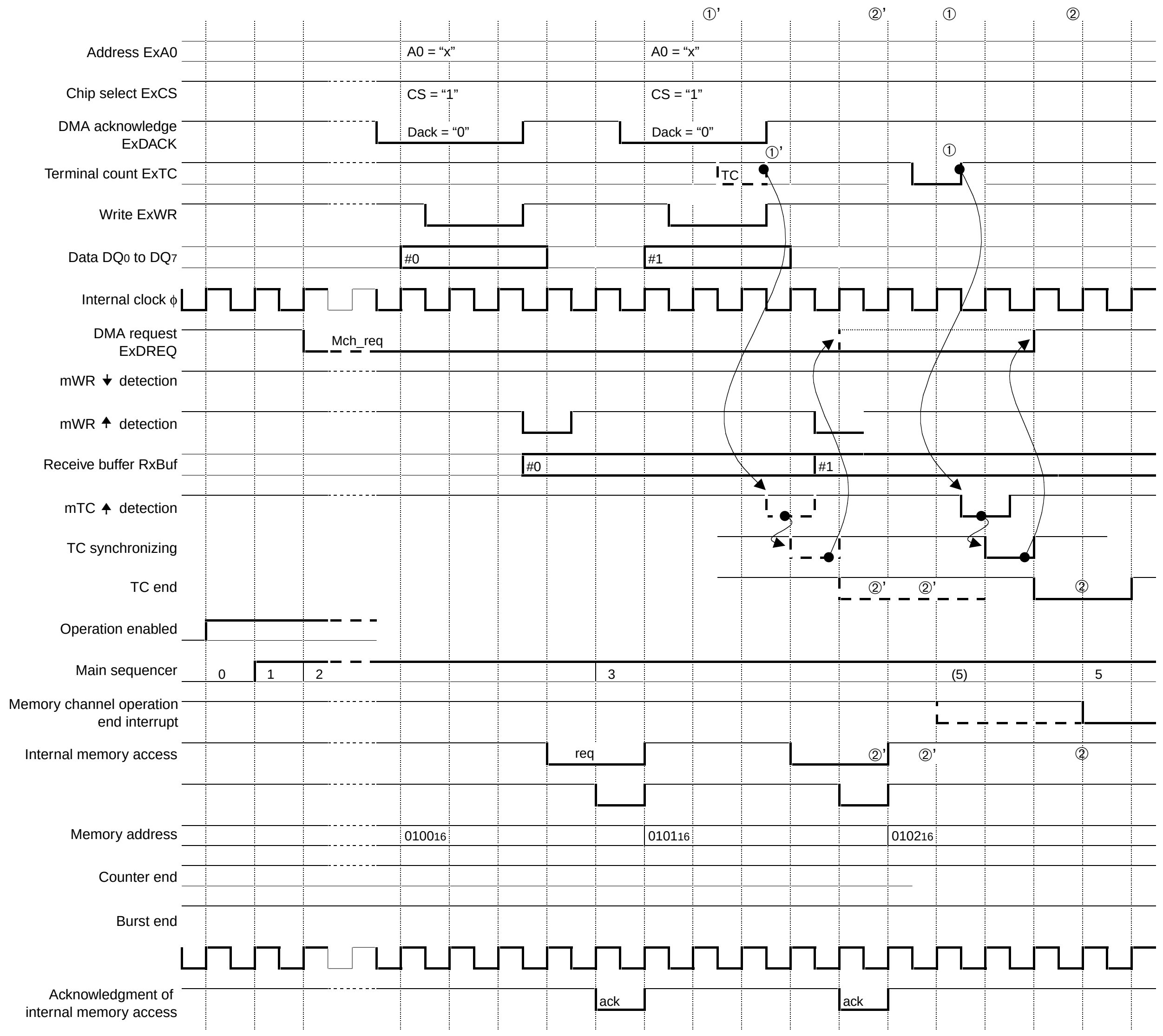
④ When the memory address counter reaches the end address, the detection circuit of external write signal (ExWR) operation is enabled and negation of the memory channel request which synchronized with the following ϕ is made.

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. 125 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)

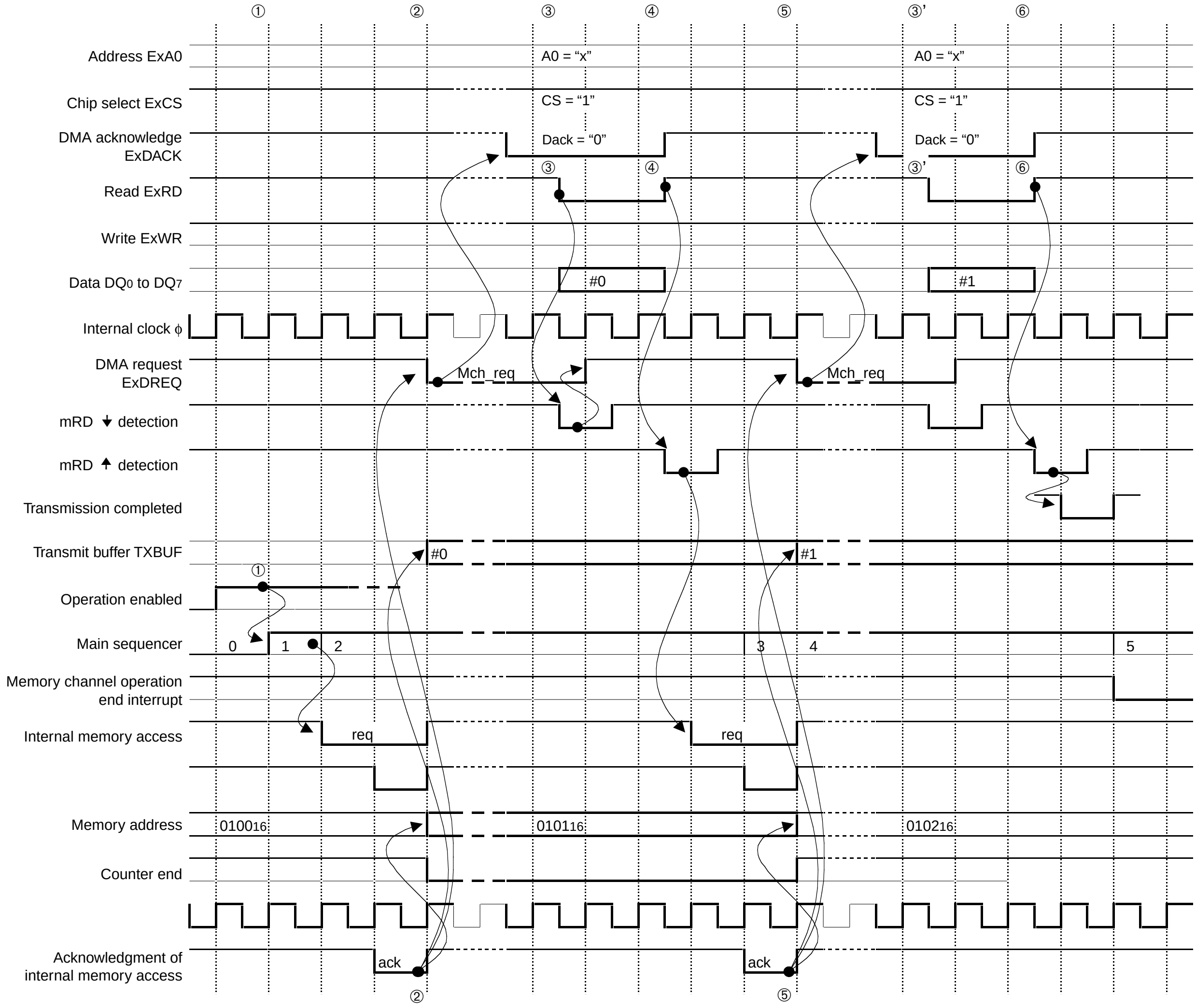
① When a rise of TC is detected, negation of the memory channel request which synchronized with a rise of ϕ is made.

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

Fig. 126 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.

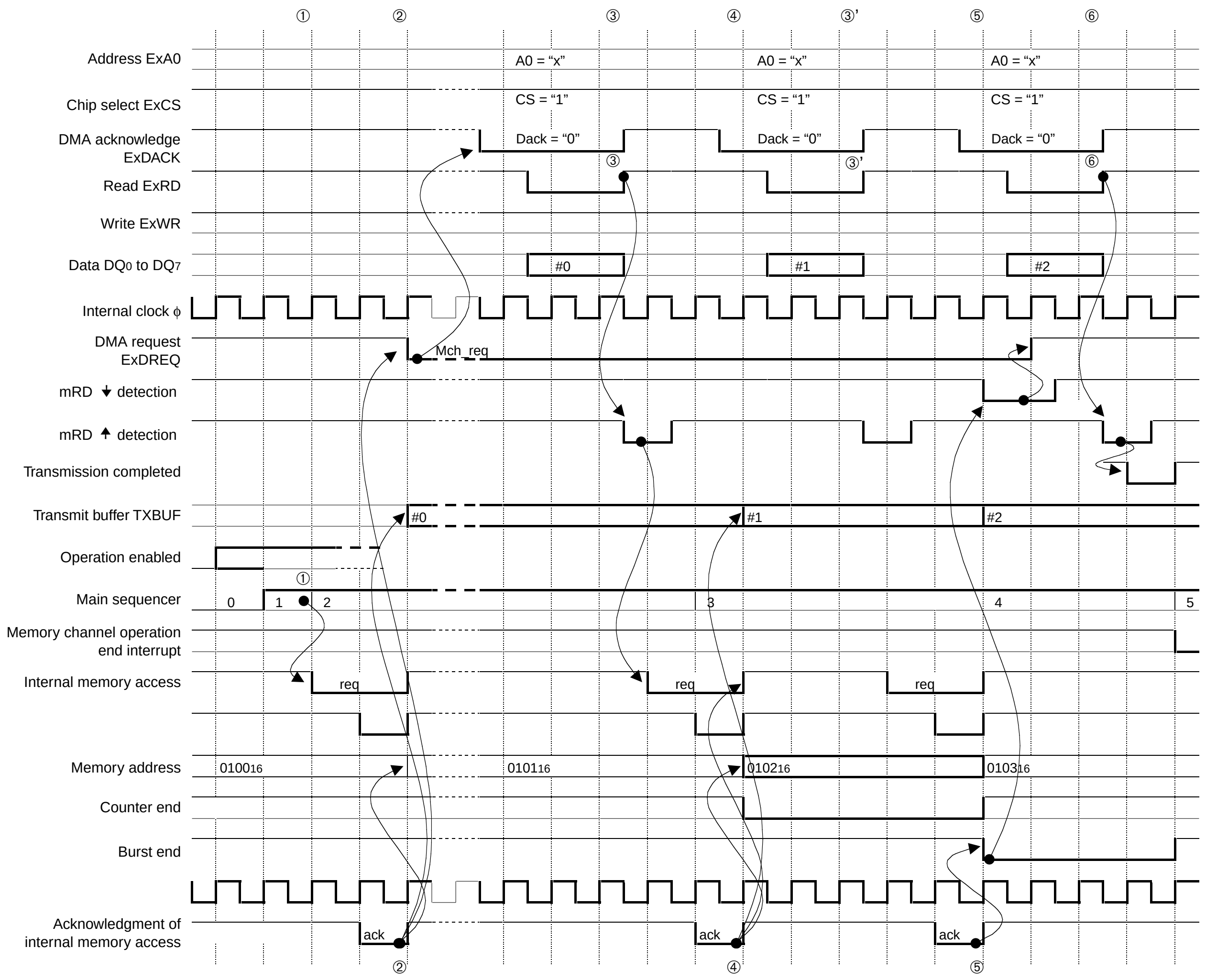
⑤ A data is read out from the internal memory within two clocks at a minimum and this data is stored in the transmit buffer (TXBUF). The memory address counter is simultaneously increased and assertion of the memory channel request is made.
When the read operation from the end address has been completed, 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. 127 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. 128 Memory channel tranmitting operation (2)

MULTICHANNEL RAM

The 38K2 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.

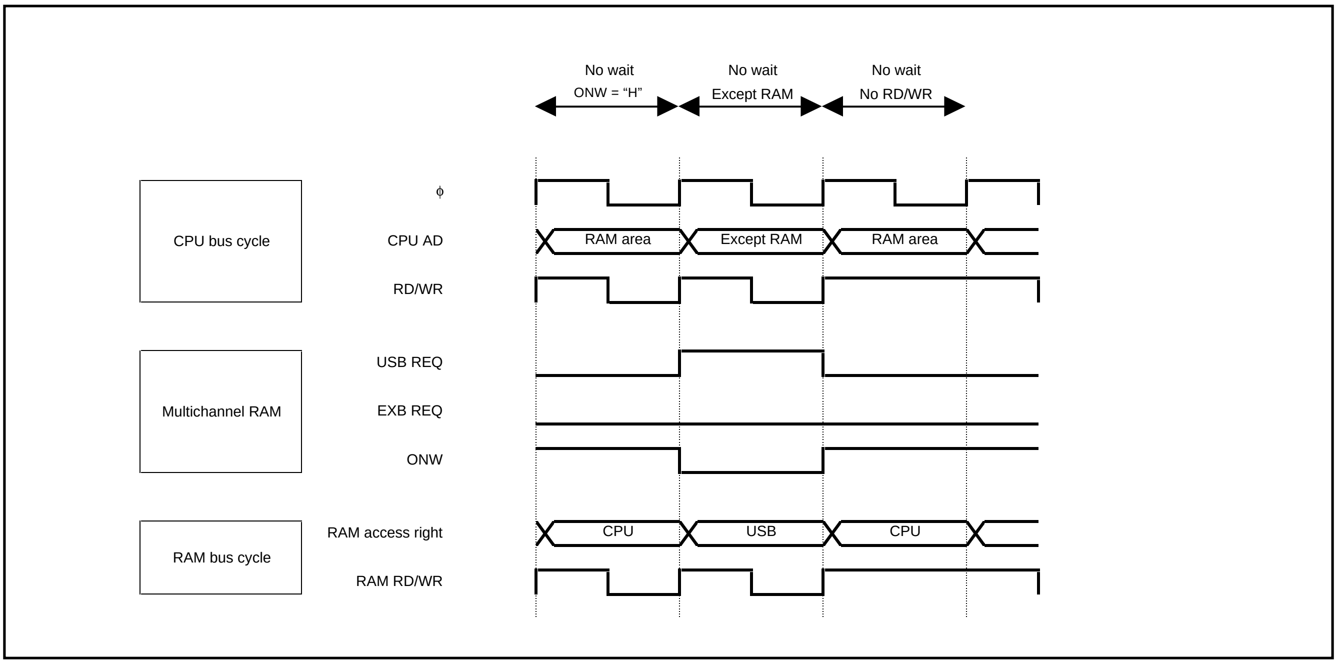


Fig. 129 Multichannel RAM timing diagram (no wait)

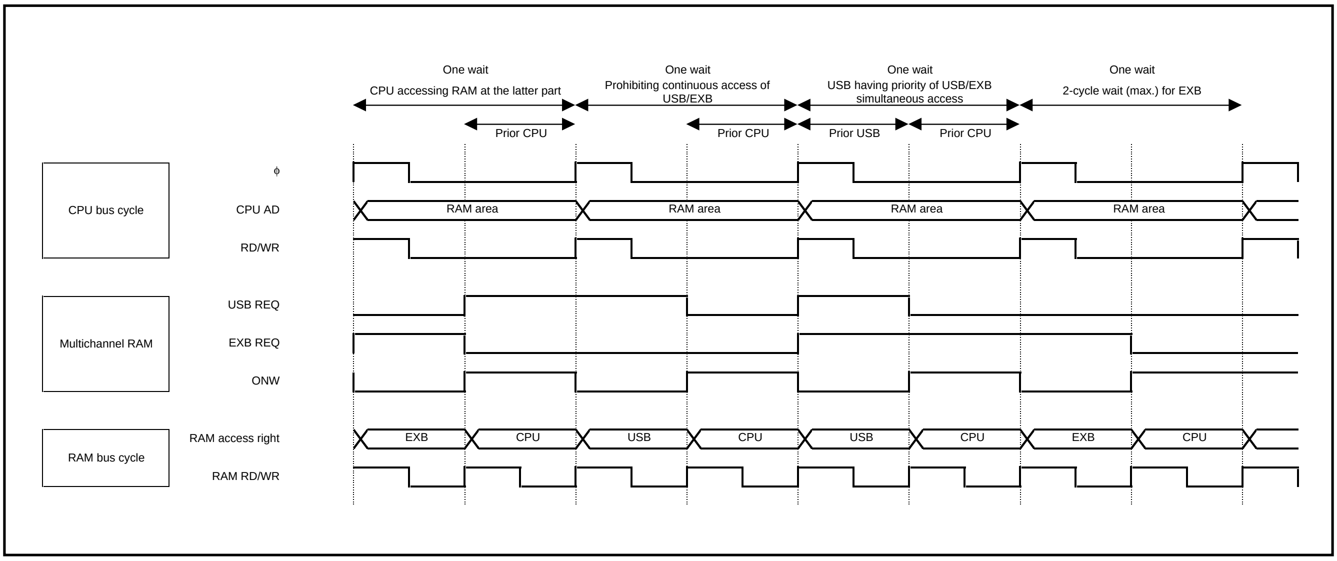


Fig. 130 Multichannel RAM timing diagram (one wait)

Multichannel RAM Operation Example

The multichannel RAM operation example is shown below.
This example shows the case that an external MCU uses the 38K2 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.

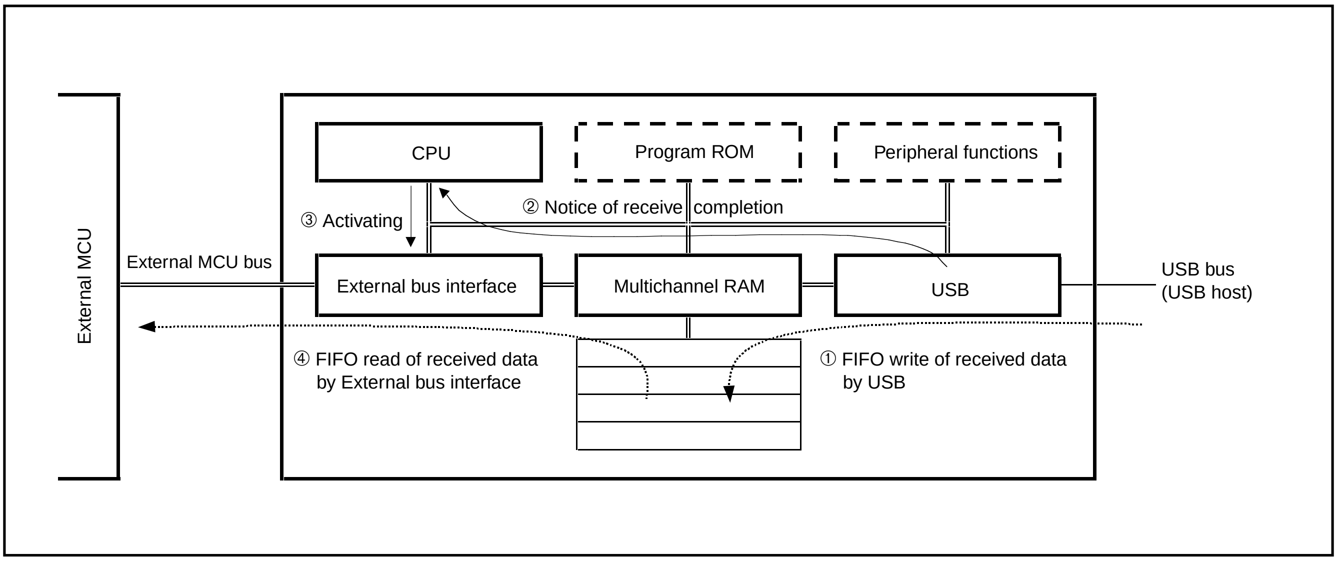


Fig. 131 Multichannel RAM operation example

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 133).

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

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

- 8-bit reading

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

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.

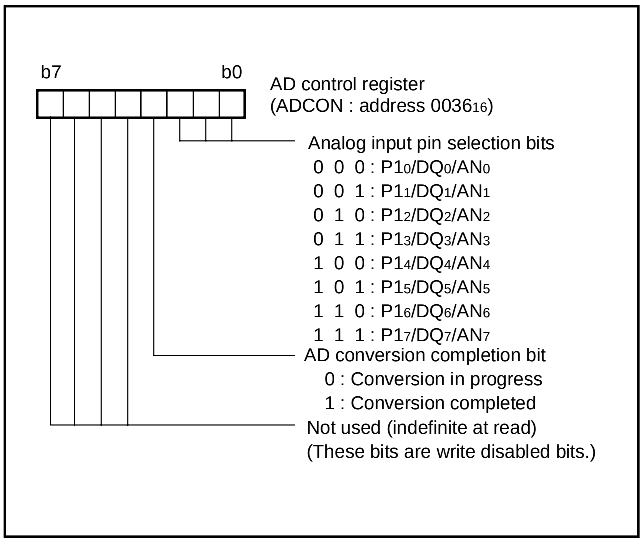


Fig. 132 Structure of AD control register

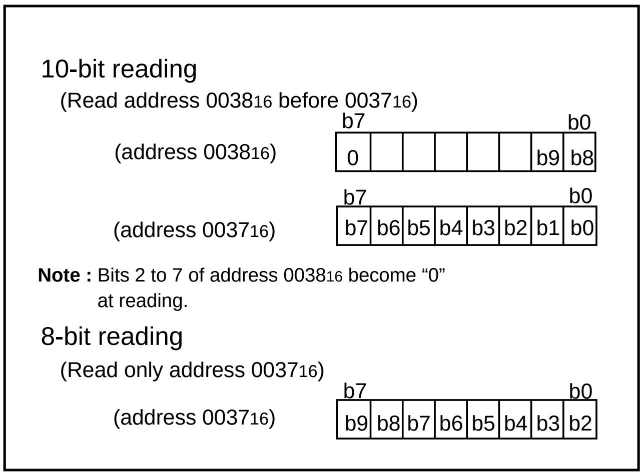


Fig. 133 10-bit/8-bit reading

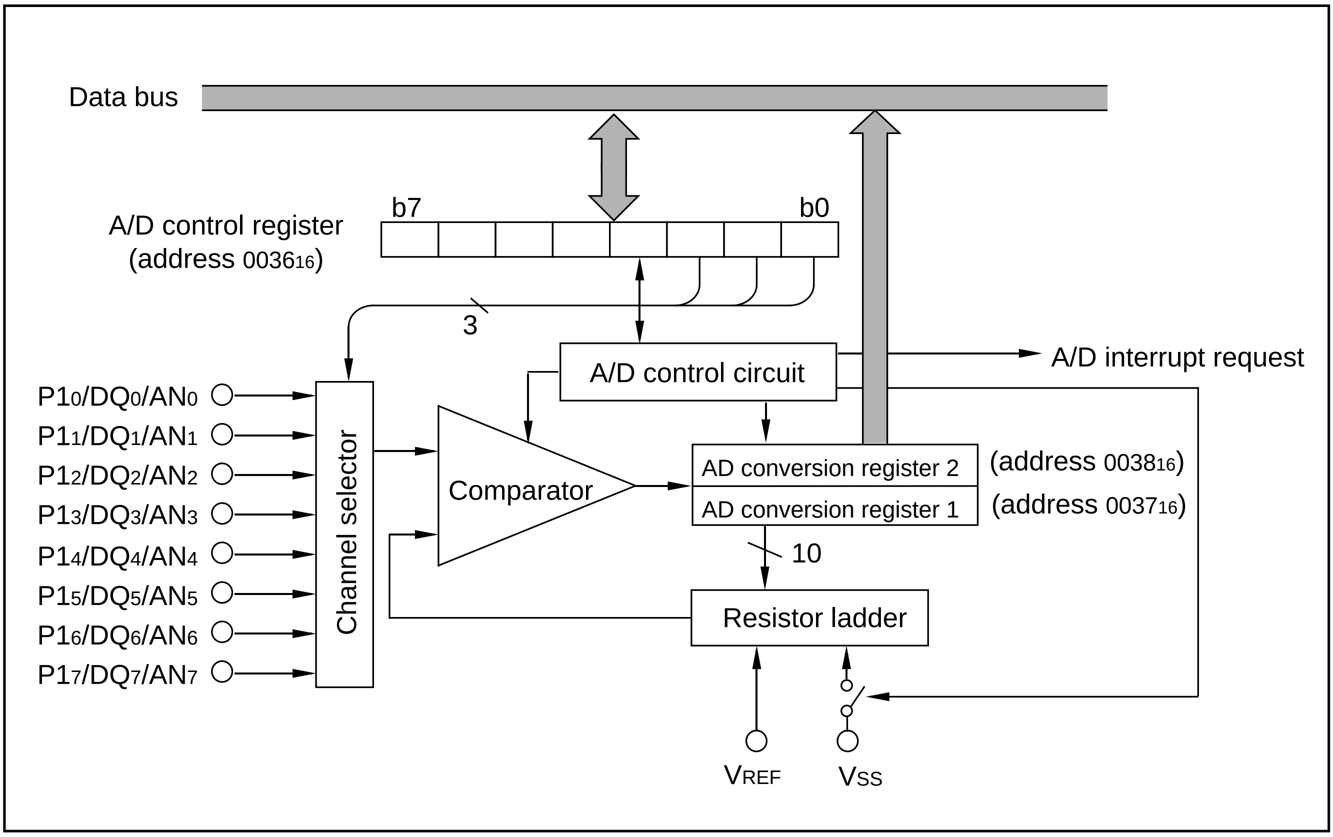


Fig. 134 A/D converter block diagram

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.

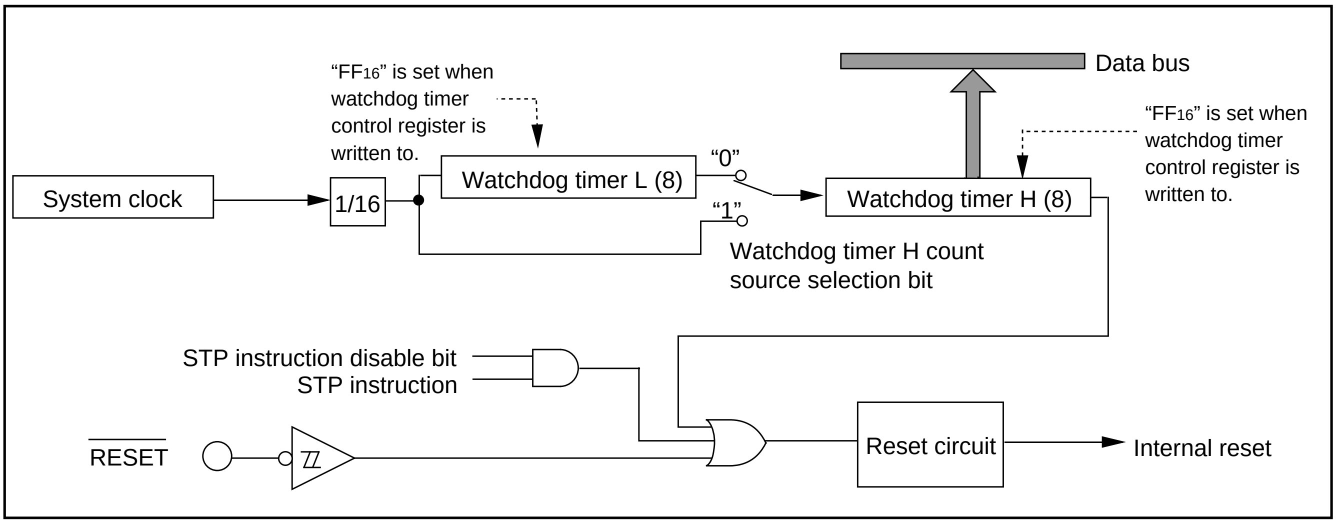

"FF16" is set when watchdog timer control register is written to.
Data bus
"FF16" is set when watchdog timer control register is written to.
System clock
1/16
Watchdog timer L (8)
"0"
"1"
Watchdog timer H (8)
Watchdog timer H count source selection bit
STP instruction disable bit
STP instruction
RESET
Reset circuit
Internal reset

Fig. 135 Block diagram of Watchdog timer

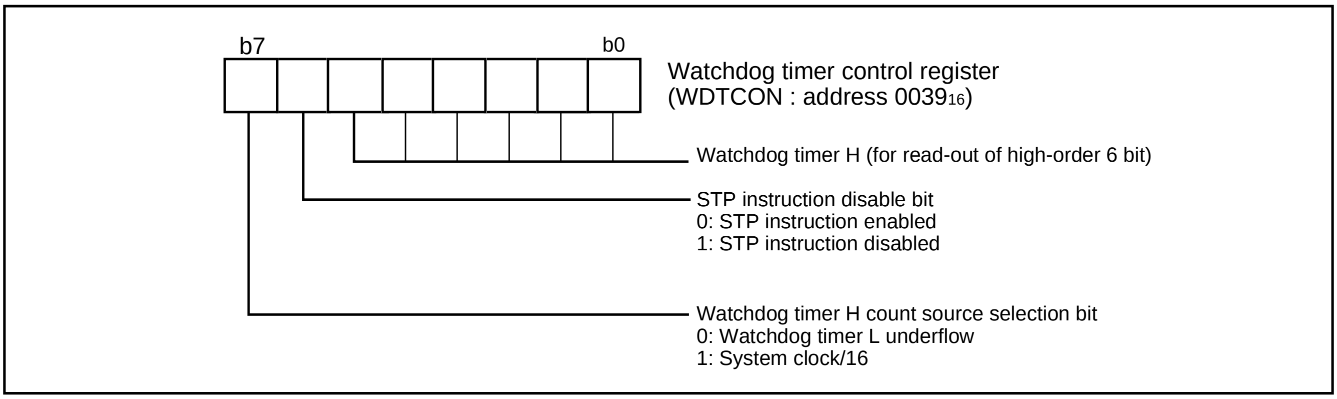

b7
b0
Watchdog timer control register
(WDTCON : address 003916)
Watchdog timer H (for read-out of high-order 6 bit)
STP instruction disable bit
0: STP instruction enabled
1: STP instruction disabled
Watchdog timer H count source selection bit
0: Watchdog timer L underflow
1: System clock/16

Fig. 136 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 $FFFD_{16}$ (high-order byte) and address $FFFC_{16}$ (low-order byte). Make sure that the reset input voltage is under 0.6 V for V_{CC} of 3.0 V (L version).

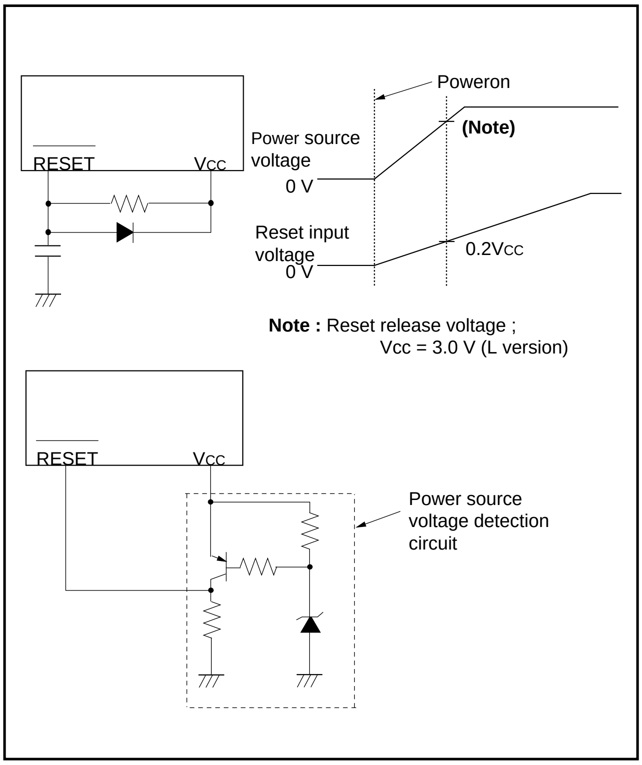
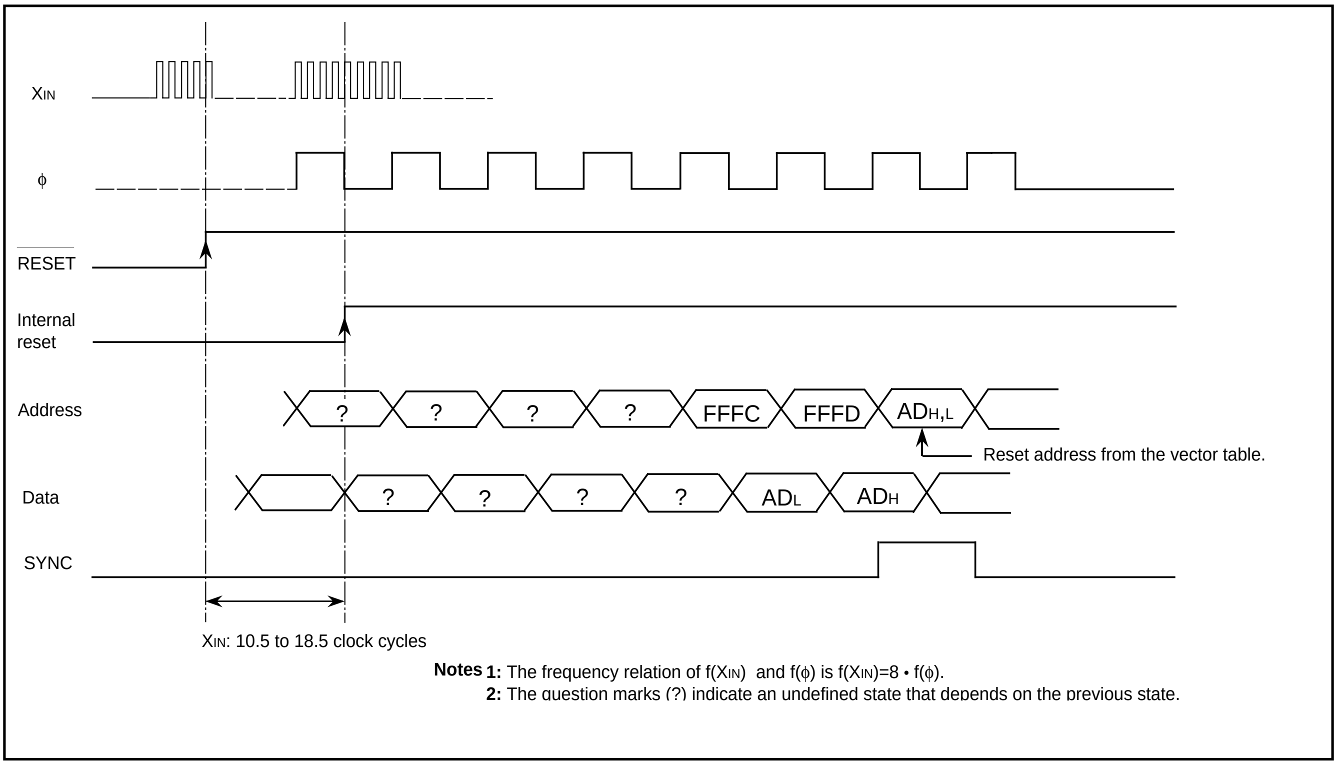


Fig. 137 Example of reset circuit


XIN
φ
RESET
Internal reset
Address
?
?
?
?
FFFC
FFFD
ADH,L
Reset address from the vector table.
Data
?
?
?
?
ADL
ADH
SYNC
XIN: 10.5 to 18.5 clock cycles

Notes 1: The frequency relation of $f(X_{IN})$ and $f(\phi)$ is $f(X_{IN})=8 \cdot f(\phi)$.
2: The question marks (?) indicate an undefined state that depends on the previous state.

Fig. 138 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 139 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).

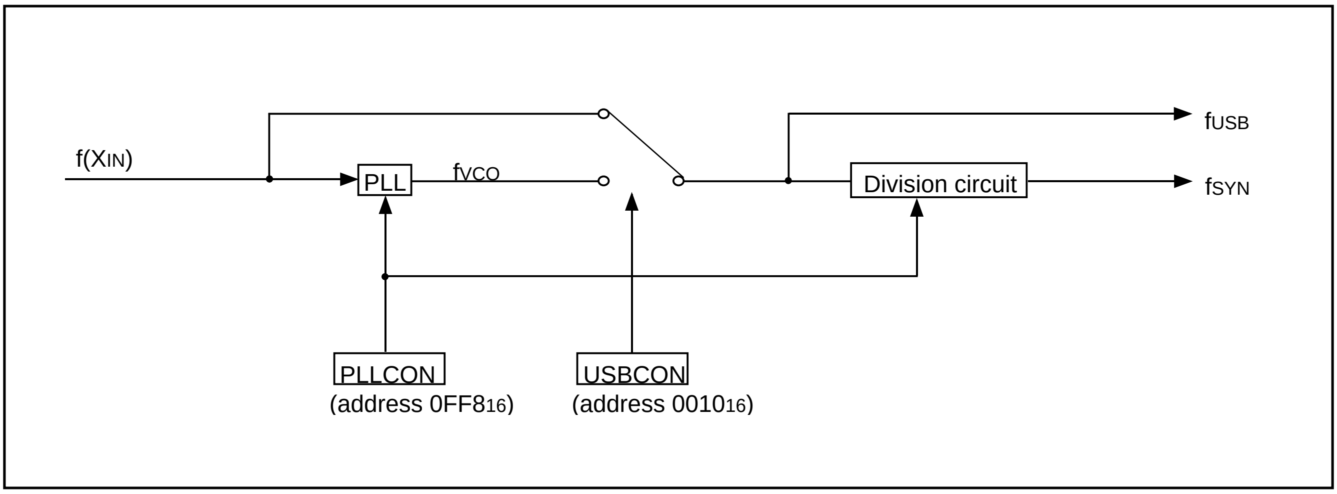


Fig. 139 Block diagram of PLL circuit

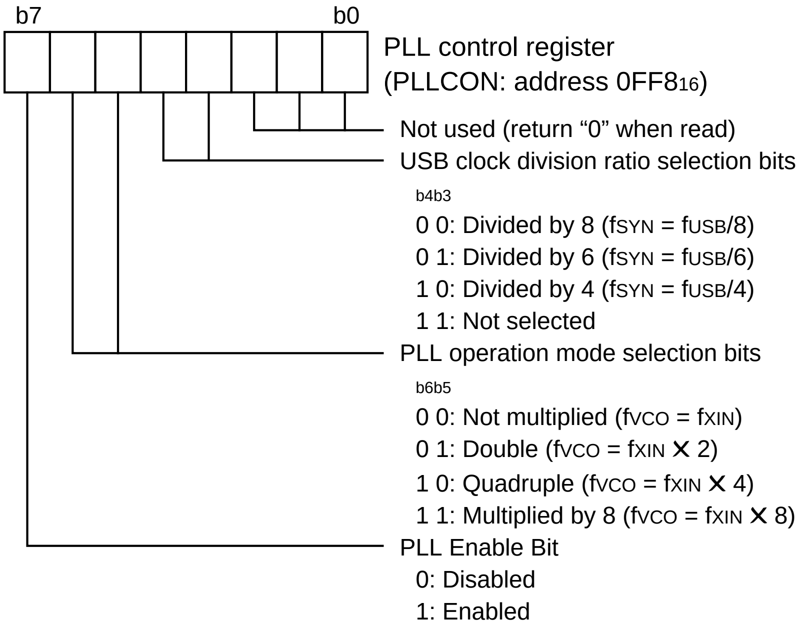

b7
b0
PLL control register
(PLLCON: address 0FF8$_{16}$)
Not used (return "0" when read)
USB clock division ratio selection bits
b4b3
0 0: Divided by 8 (f_{SYN} = f_{USB}/8)
0 1: Divided by 6 (f_{SYN} = f_{USB}/6)
1 0: Divided by 4 (f_{SYN} = f_{USB}/4)
1 1: Not selected
PLL operation mode selection bits
b6b5
0 0: Not multiplied (f_{VCO} = f_{XIN})
0 1: Double (f_{VCO} = f_{XIN} × 2)
1 0: Quadruple (f_{VCO} = f_{XIN} × 4)
1 1: Multiplied by 8 (f_{VCO} = f_{XIN} × 8)
PLL Enable Bit
0: Disabled
1: Enabled

Fig. 140 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

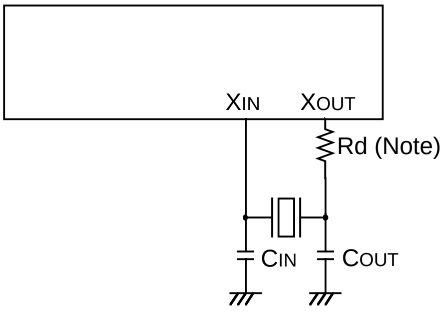
If the WIT instruction is executed, the internal clock ϕ stops at an "H" level, but the oscillator does not stop. The internal clock ϕ restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

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.

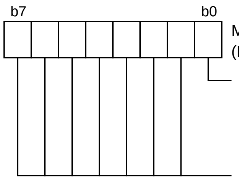
■Note

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.



Note : 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. 141 Ceramic resonator or quartz-crystal oscilltor circuit



MISRG
(MISRG: address 0FFB$_{16}$)

Oscillation stabilizing time set after STP instruction released bit
0: Automatically set “01$_{16}$” to Timer 1, “FF$_{16}$” to Prescaler 12
1: Automatically set nothing
Not used (indefinite at read)

Fig. 143 Structure of MISRG

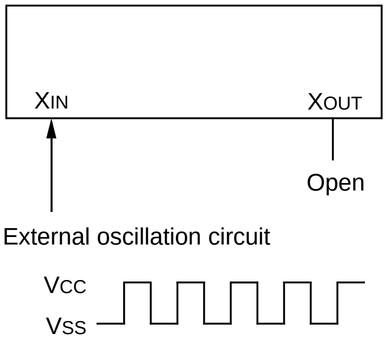


Fig. 142 External clock input circuit

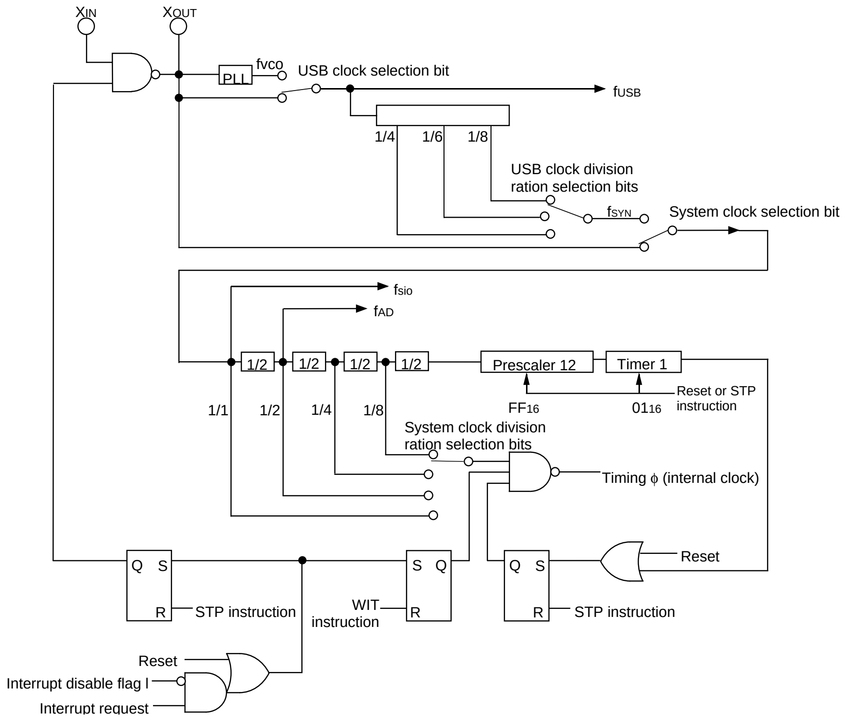
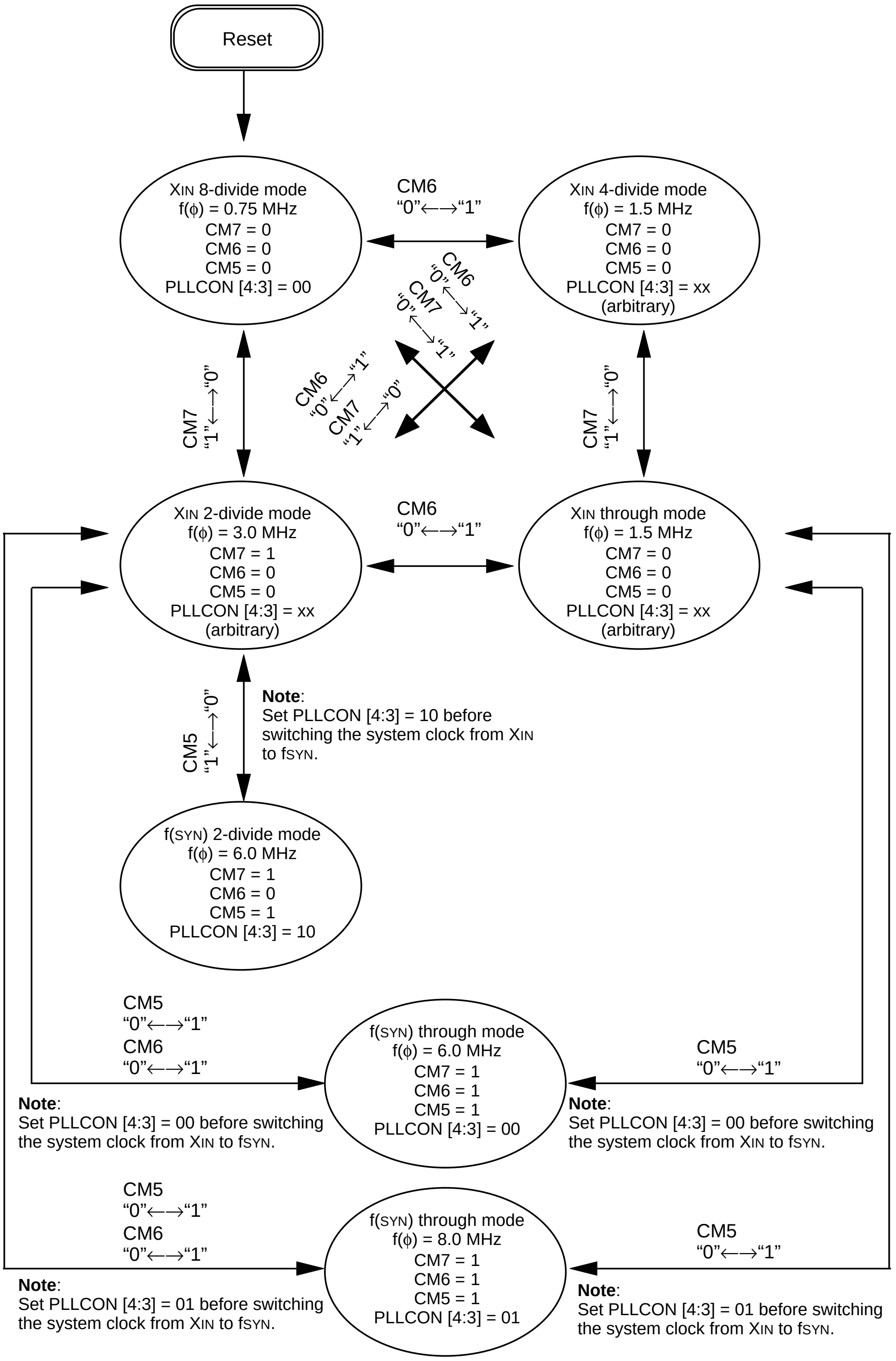


Fig. 144 System clock generating circuit block diagram (single-chip mode)



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. 145 State transitions of clock

FLASH MEMORY MODE

The 38K2 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 9 lists the summary of the 38K2 group's flash memory version.

This flash memory version has some blocks on the flash memory as shown in Figure 146 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 9 Summary of 38K2 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 38K2 Group (flash memory version).

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

(1) CPU Rewrite Mode

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

In CPU rewrite mode, only the User ROM area shown in Figure 146 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.)

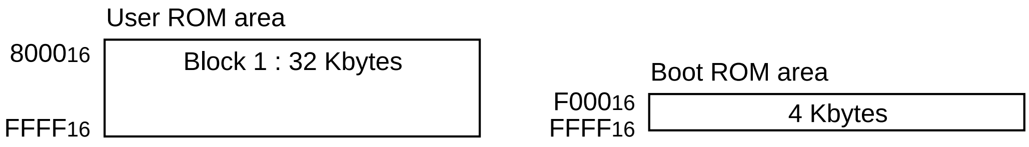
See Figure 146 for details about the Boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNV$_{SS}$ 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 P1$_6$ ($\overline{CE}$) pin high, the CNV$_{SS}$ 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.



Notes 1: The Boot ROM area can be rewritten in only parallel I/O mode. (Access to any other areas is inhibited.)

2: To specify a block, use the maximum address in the block.

Fig. 146 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 CNV$_{SS}$ pin and setting “1” to the CPU Rewrite Mode Select Bit (bit 1 of address 0FFE$_{16}$). 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 147 shows the flash memory control register.

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

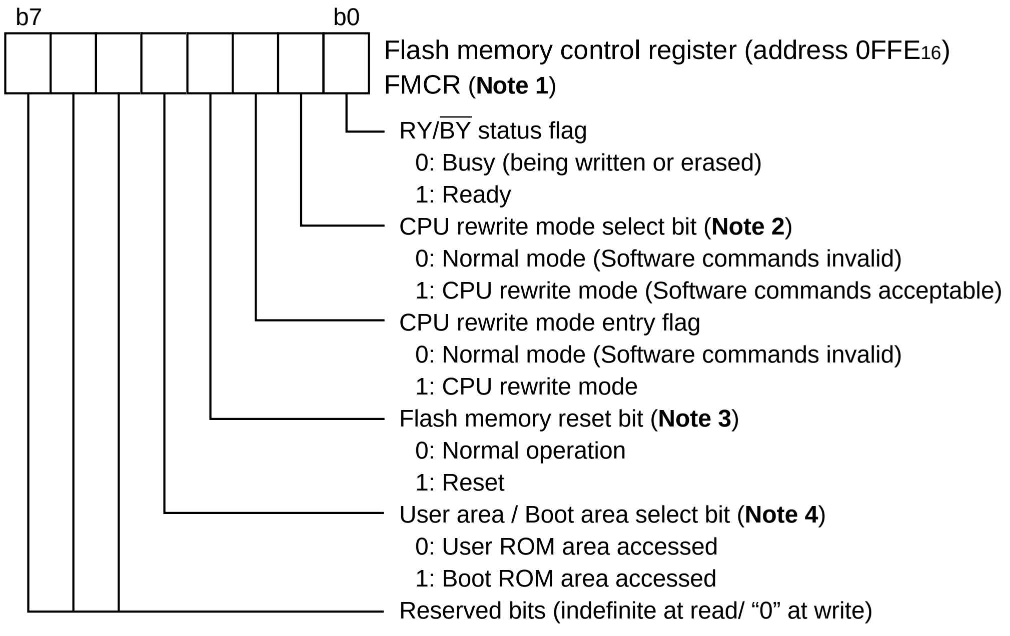
Bit 1 is the CPU Rewrite Mode Select Bit. When this bit is set to “1”, the MCU enters CPU rewrite mode. Software commands are accepted once the mode is entered. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in a memory other than internal flash memory for write to bit 1. To set this bit to “1”, it is necessary to write “0” and then write “1” in succession. The bit can be set to “0” by only writing “0”.

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

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

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

Figure 148 shows a flowchart for setting/releasing CPU rewrite mode.


b7
b0
Flash memory control register (address 0FFE16)
FMCR (Note 1)
RY/BY status flag
0: Busy (being written or erased)
1: Ready
CPU rewrite mode select bit (Note 2)
0: Normal mode (Software commands invalid)
1: CPU rewrite mode (Software commands acceptable)
CPU rewrite mode entry flag
0: Normal mode (Software commands invalid)
1: CPU rewrite mode
Flash memory reset bit (Note 3)
0: Normal operation
1: Reset
User area / Boot area select bit (Note 4)
0: User ROM area accessed
1: Boot ROM area accessed
Reserved bits (indefinite at read/ “0” at write)

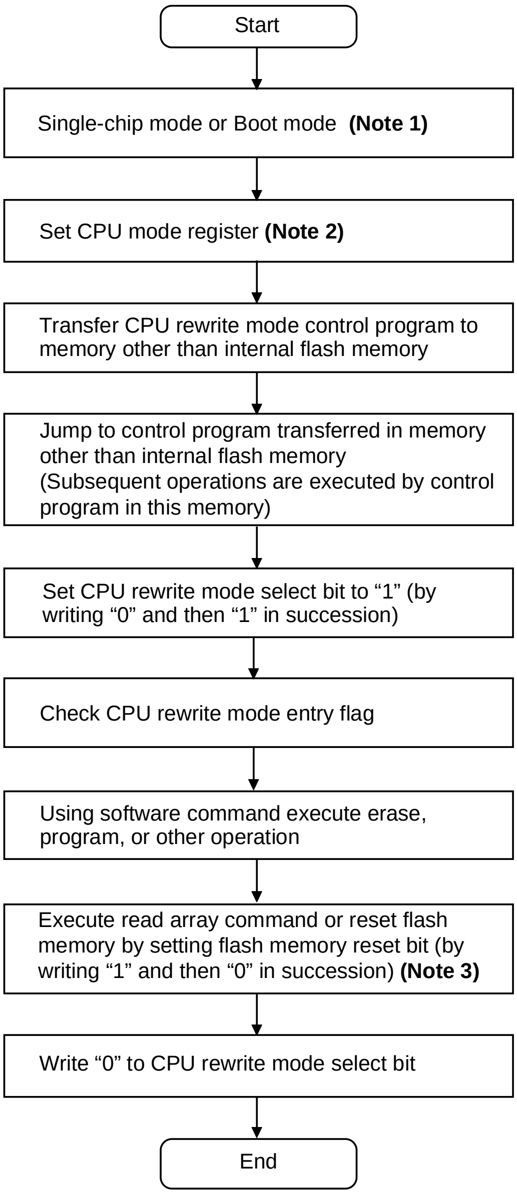
Notes **1**: The contents of flash memory control register are “XXX00001” just after reset release.

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. 147 Structure of flash memory control register


Start
Single-chip mode or Boot mode **(Note 1)**
Set CPU mode register **(Note 2)**
Transfer CPU rewrite mode control program to memory other than internal flash memory
Jump to control program transferred in memory other than internal flash memory (Subsequent operations are executed by control program in this memory)
Set CPU rewrite mode select bit to “1” (by writing “0” and then “1” in succession)
Check CPU rewrite mode entry flag
Using software command execute erase, program, or other operation
Execute read array command or reset flash memory by setting flash memory reset bit (by writing “1” and then “0” in succession) **(Note 3)**
Write “0” to CPU rewrite mode select bit
End

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. 148 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 10 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/$\overline{\text{BY}}$ 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/$\overline{\text{BY}}$ 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.

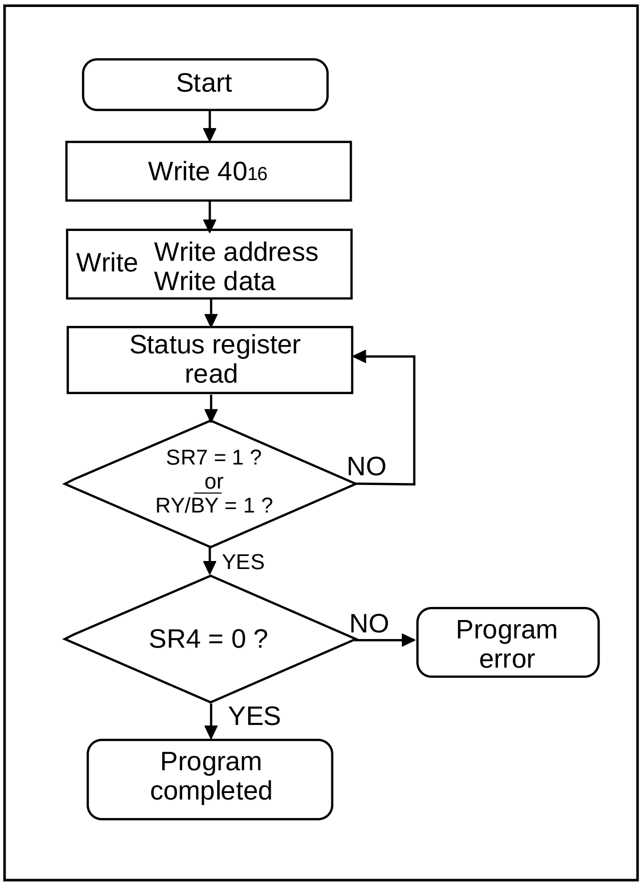


Fig. 149 Program flowchart

Table 10 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/$\overline{\text{BY}}$ Status Flag of flash memory control register. When the erase all blocks operation starts, the read status register mode is entered automatically and the contents of the status register can be read out at the data bus (D_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/$\overline{\text{BY}}$ Status Flag is “0” during erase operation and “1” when the erase operation is completed as is the status register bit 7.

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

●Block Erase Command (20_{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/$\overline{\text{BY}}$ Status Flag of flash memory control register. At the same time the block erase operation starts, the read status register mode is automatically entered, so that the contents of the status register can be read out. The status register bit 7 (SR7) is set to “0” at the same time the block erase operation starts and is returned to “1” upon completion of the block erase operation. In this case, the read status register mode remains active until the read array command (FF_{16}) is written.

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

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

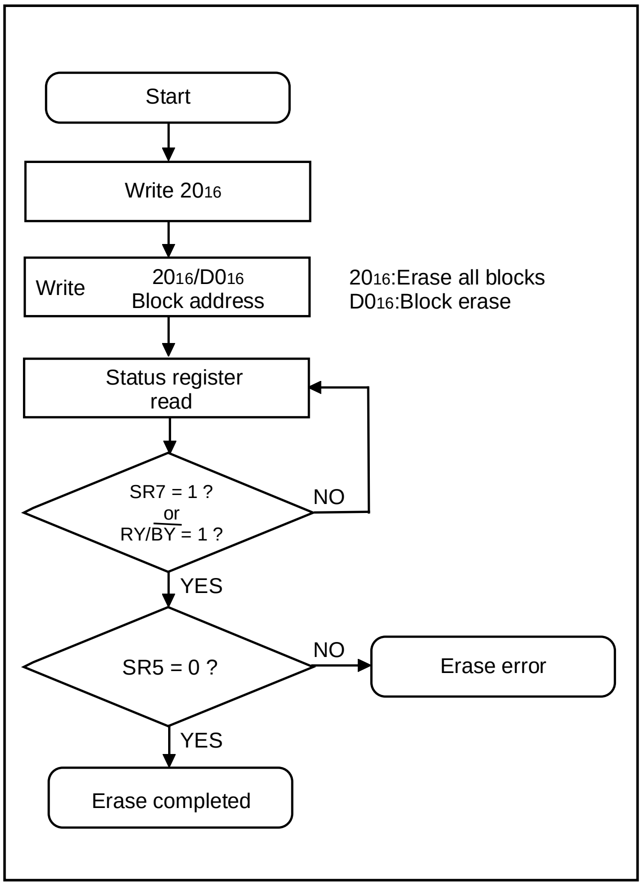


Fig. 150 Erase flowchart

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 11 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.

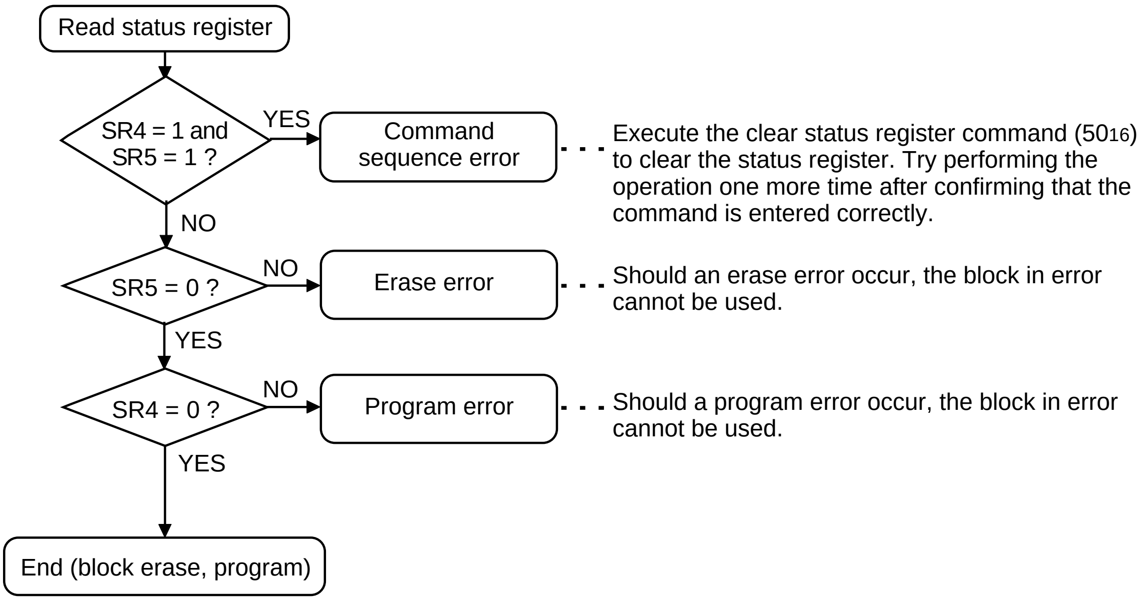
If “1” is written for any of the SR5 and SR4 bits, the program, erase all blocks, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.

Table 11 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 151 shows a full status check flowchart and the action to be taken when each error occurs.



Execute the clear status register command (50_{16}) to clear the status register. Try performing the operation one more time after confirming that the command is entered correctly.

Should an erase error occur, the block in error cannot be used.

Should a program error occur, the block in error cannot be used.

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. 151 Full status check flowchart and remedial procedure for errors

Functions To Inhibit Rewriting Flash Memory Version

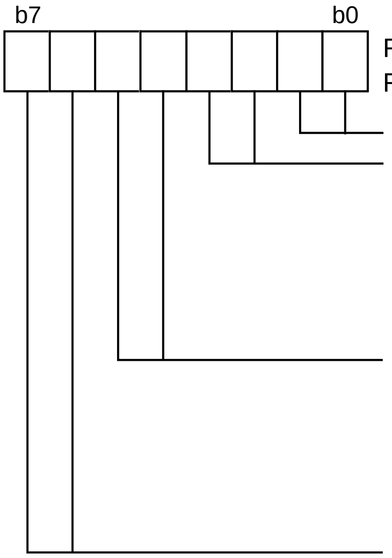
To prevent the contents of internal flash memory from being read out or rewritten easily, this MCU incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

●ROM Code Protect Function

The ROM code protect function is the function to inhibit reading out or modifying the contents of internal flash memory by using the ROM code protect control register (address FFDB$_{16}$) in parallel I/O mode. Figure 152 shows the ROM code protect control register (address FFDB$_{16}$). (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.



ROM code protect control register (address FFDB$_{16}$)
ROMCP

Reserved bits (“1” at read/write)

ROM code protect level 2 set bits (ROMCP2) (**Notes 1, 2**)

b3b2	
0 0:	Protect enabled
0 1:	Protect enabled
1 0:	Protect enabled
1 1:	Protect disabled

ROM code protect reset bits (**Note 3**)

b5b4	
0 0:	Protect removed
0 1:	Protect set bits effective
1 0:	Protect set bits effective
1 1:	Protect set bits effective

ROM code protect level 1 set bits (ROMCP1) (**Note 1**)

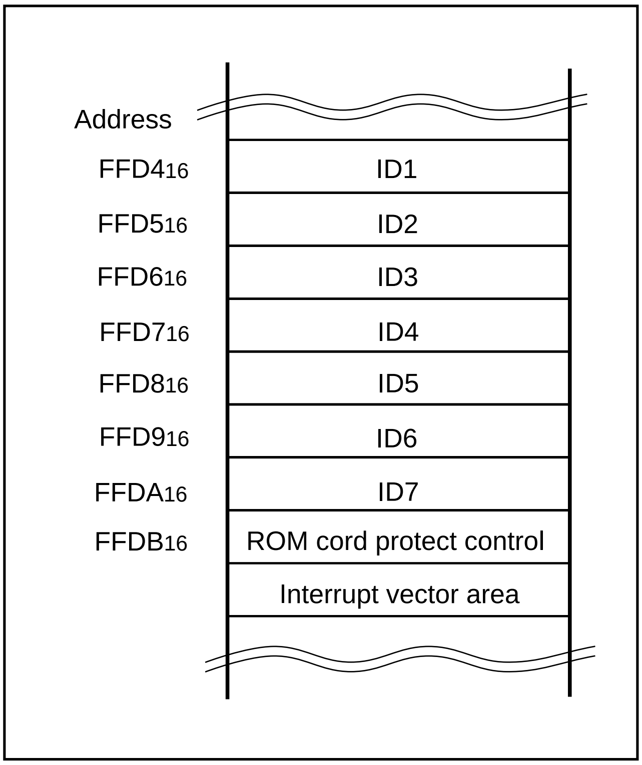
b7b6	
0 0:	Protect enabled
0 1:	Protect enabled
1 0:	Protect enabled
1 1:	Protect disabled

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. 152 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.



Address	
FFD4$_{16}$	ID1
FFD5$_{16}$	ID2
FFD6$_{16}$	ID3
FFD7$_{16}$	ID4
FFD8$_{16}$	ID5
FFD9$_{16}$	ID6
FFDA$_{16}$	ID7
FFDB$_{16}$	ROM cord protect control
	Interrupt vector area

Fig. 153 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 38K2 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 146 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 P1$_6$ ($\overline{\text{CE}}$) pin and “H” to the P4$_2$ (SCLK) pin and “H” to the CNV$_{SS}$ (V$_{PP}$) 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 154 shows the pin connections for the standard serial I/O mode.

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

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

In standard serial I/O mode, only the User ROM area shown in Figure 146 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.

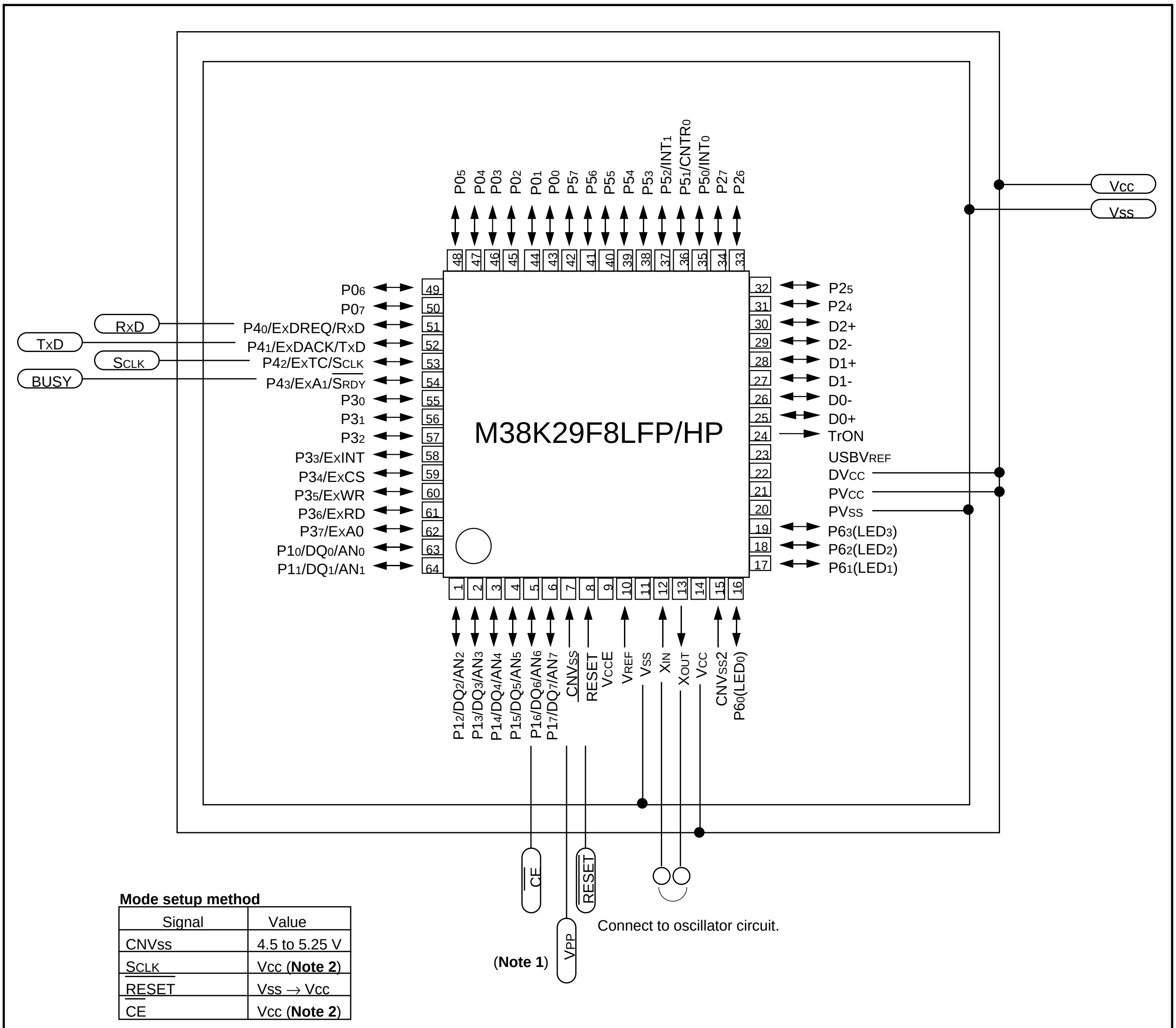
The TxD pin is for CMOS output. Transfer is in 8-bit units with LSB first.

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

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

Table 12 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	I	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.
D1+,D1-	USB downstream input	I/O	Input “L” level when not using.
D2+,D2-	USB downstream 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 P24	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. 154 Pin connection diagram in standard serial I/O mode

Software Commands

Table 13 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 13 Software commands (Standard serial I/O mode)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte	.....	When ID is not verified
1	Page read	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Erase all blocks	$A7_{16}$	$D0_{16}$						Not acceptable
4	Read status register	70_{16}	SRD output	SRD1 output					Acceptable
5	Clear status register	50_{16}							Not acceptable
6	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
7	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
8	Version data output function	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
9	Boot ROM area output function	FC_{16}	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 "FF$_{16}$" 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, data (D$_0$ to D$_7$) for the page (256 bytes) specified with addresses A$_8$ to A$_{23}$ will be output sequentially from the smallest address first synchronized with the fall of the clock.

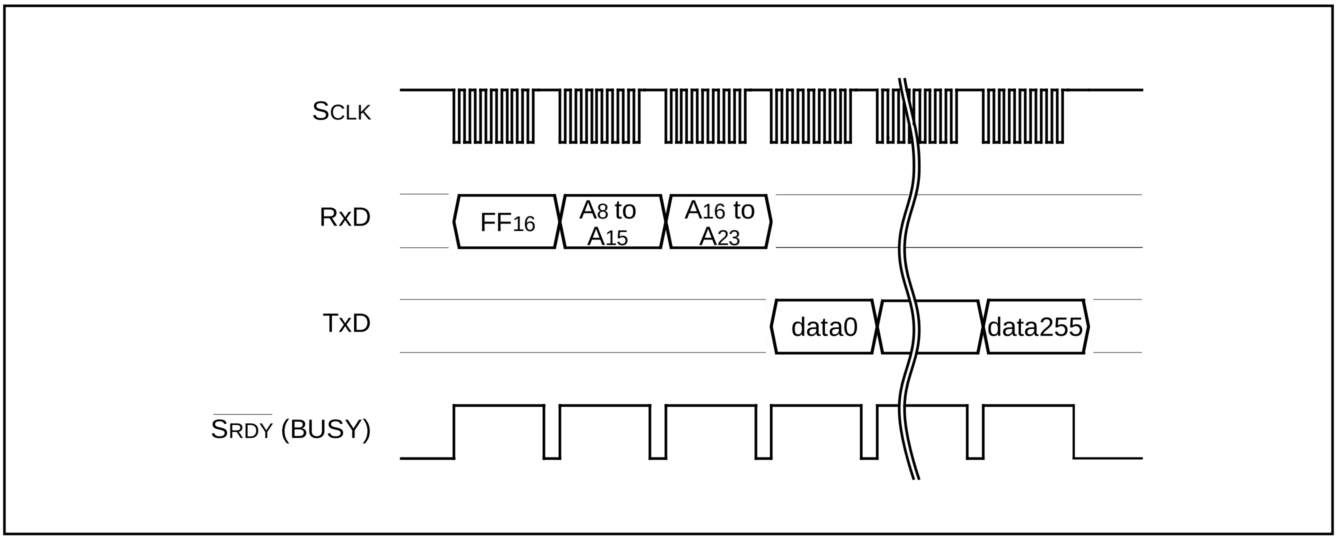


Fig. 155 Timing for page read

●Read Status Register Command

This command reads status information. When the "70$_{16}$" 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.

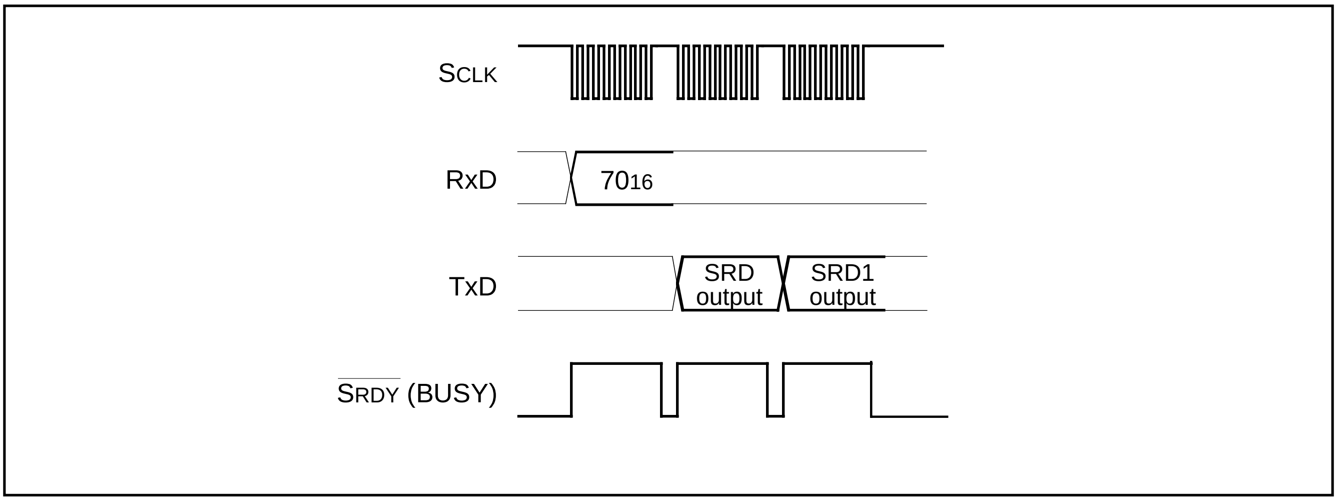


Fig. 156 Timing for reading status register

●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 $\overline{\text{SRDY}}$ (BUSY) signal changes from “H” to “L” level.

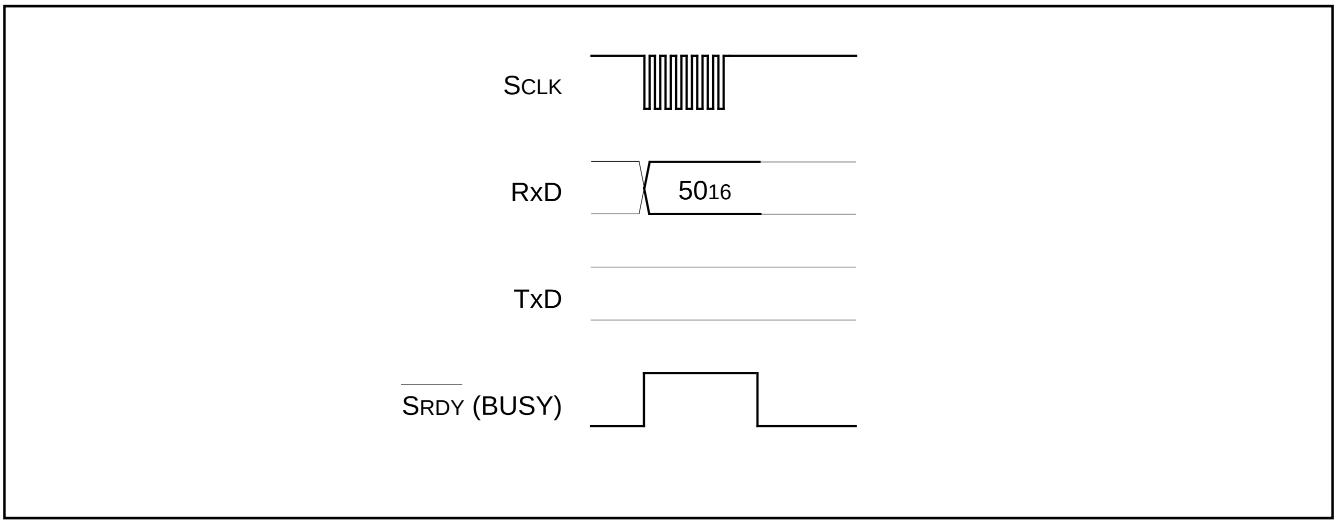


Fig. 157 Timing for clear status register

●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 $\overline{\text{SRDY}}$ (BUSY) signal changes from “H” to “L” level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

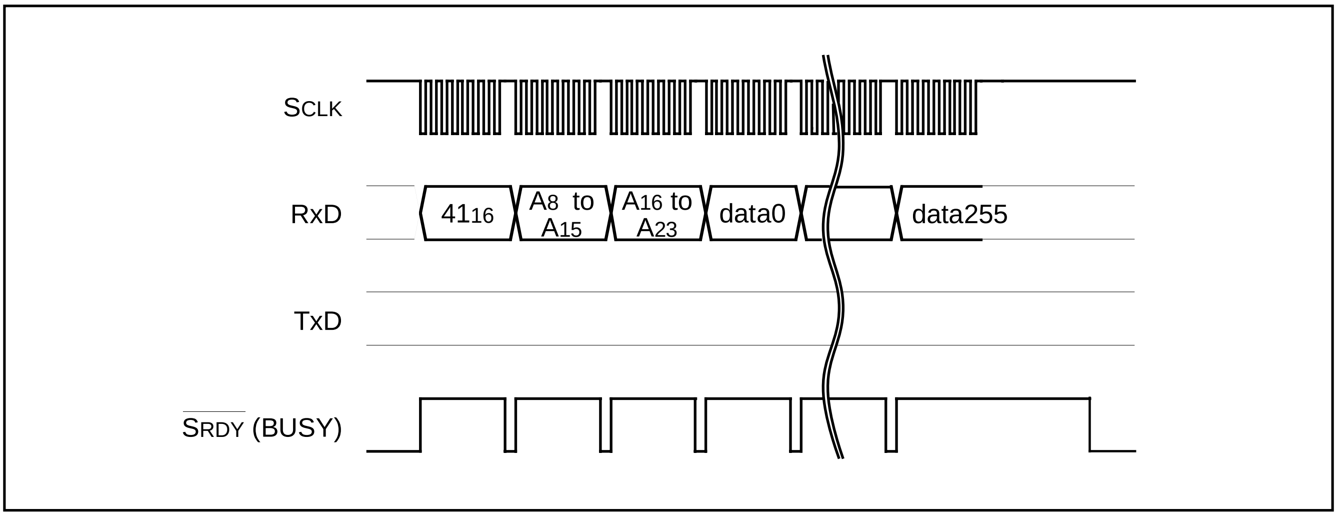


Fig. 158 Timing for page program

●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 “A7₁₆” command code with the 1st byte.

(2) Transfer the verify command code “D0₁₆” 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.

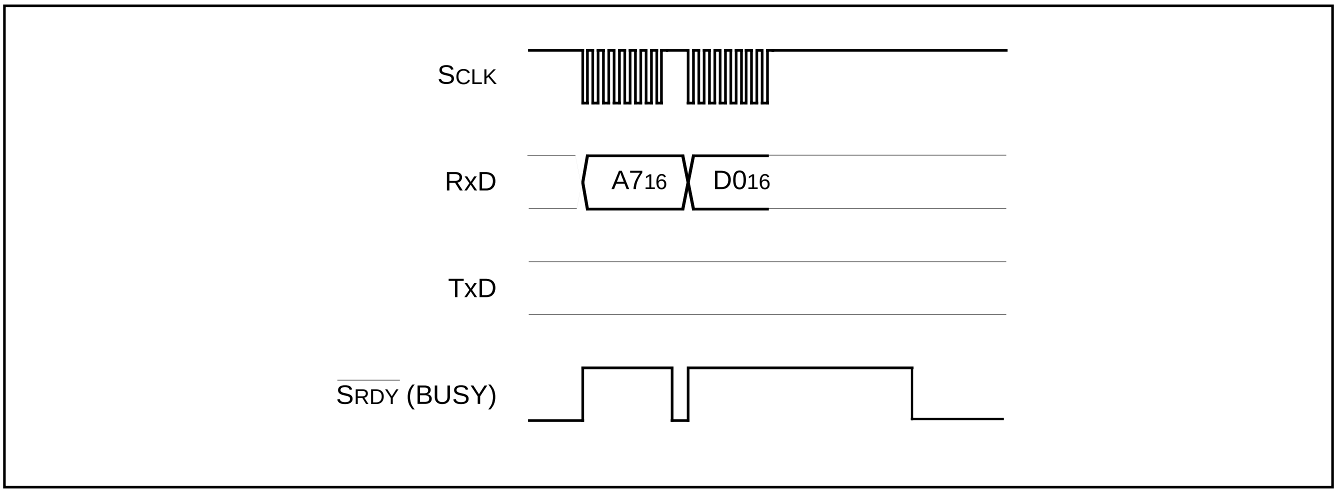


Fig. 159 Timing for erase all blocks

●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.

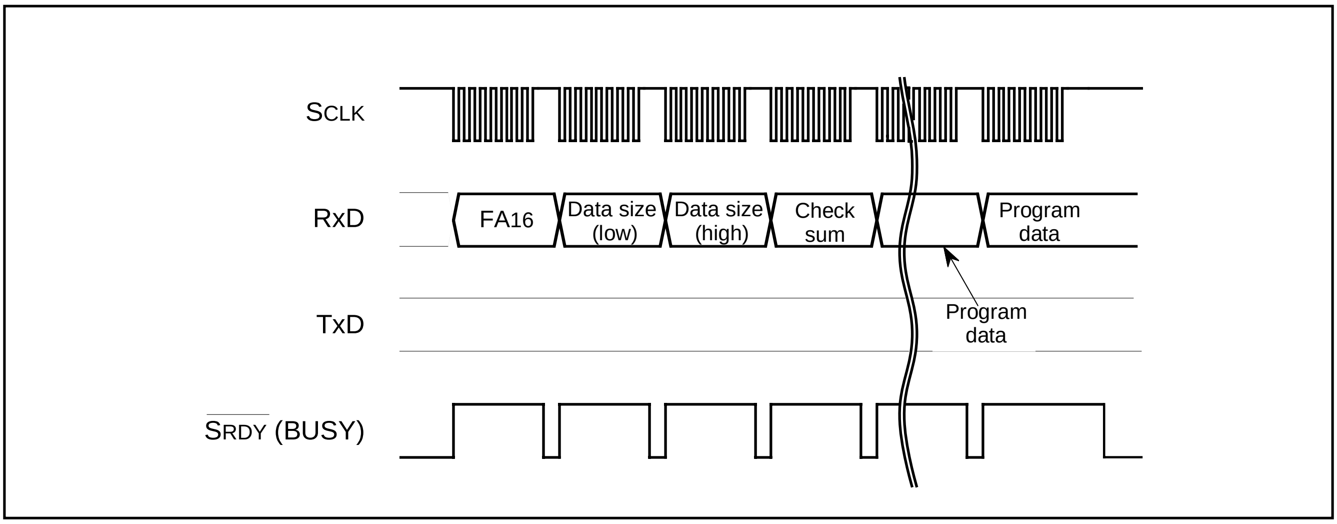


Fig. 160 Timing for download

●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.

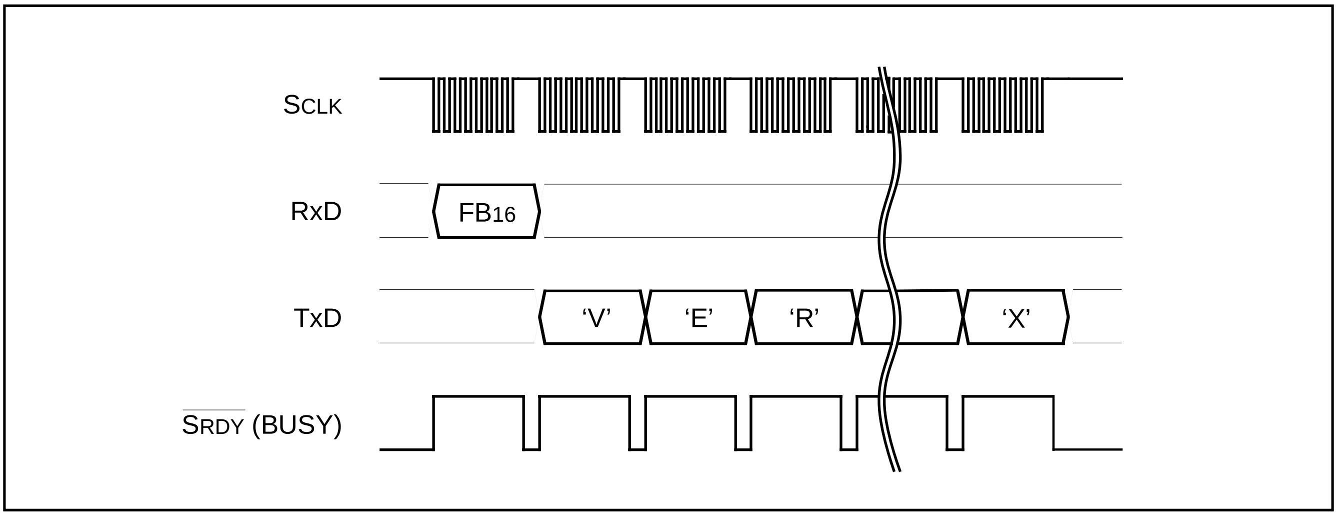


Fig. 161 Timing for version information output

●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 A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0 to D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first synchronized with the fall of the clock.

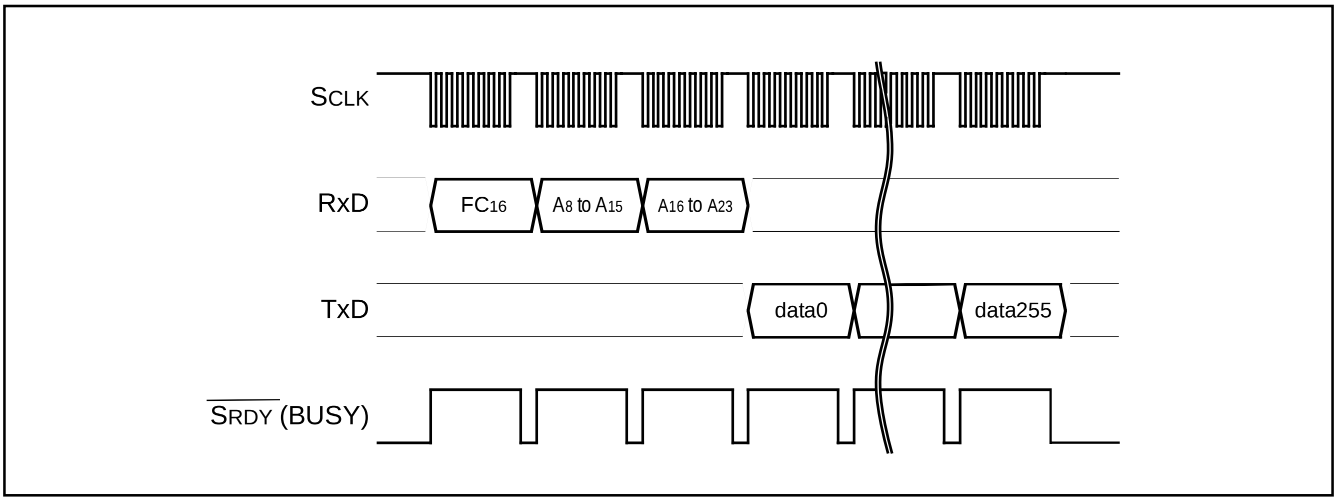


Fig. 162 Timing for Boot ROM area output

●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.

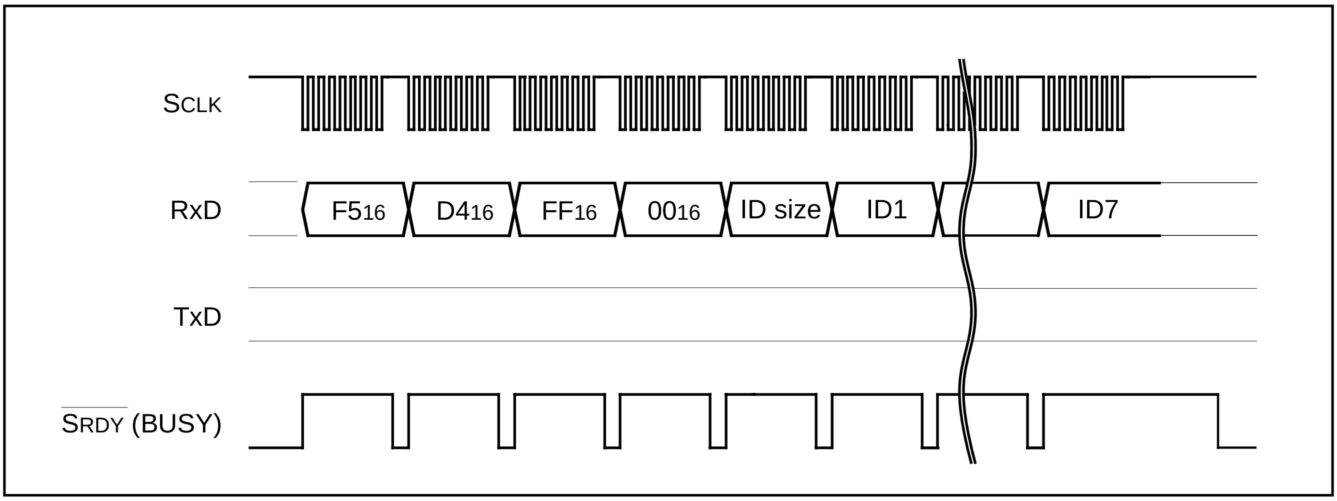
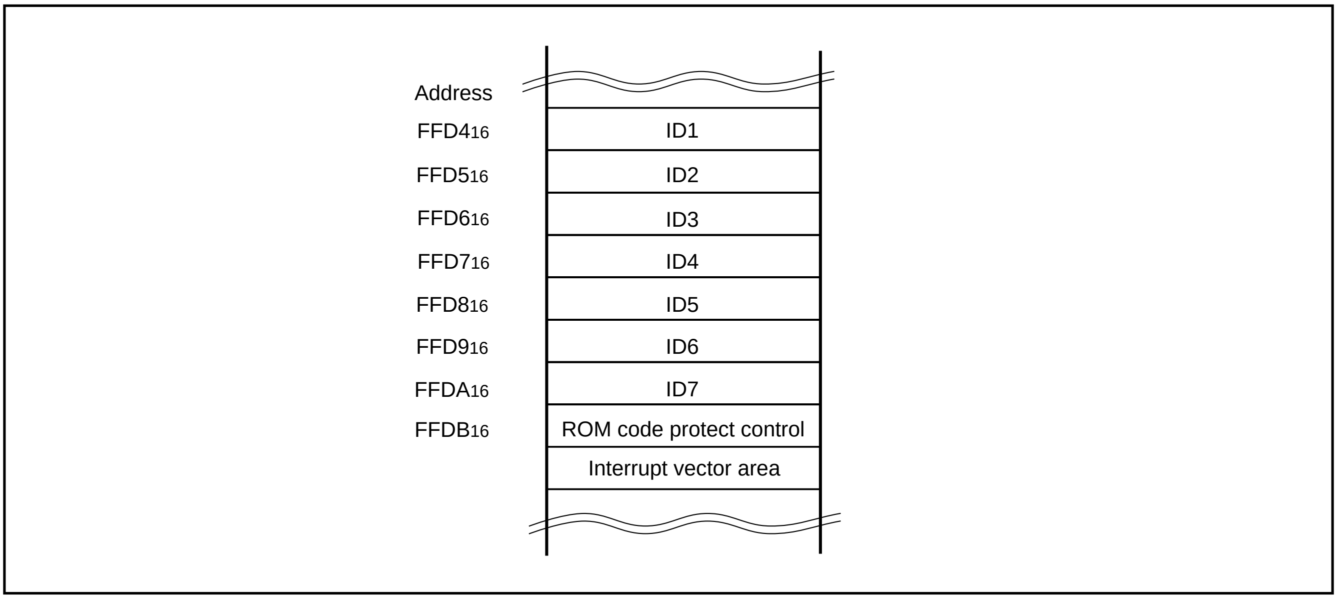


Fig. 163 Timing for ID check

●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.



Address	
FFD4$_{16}$	ID1
FFD5$_{16}$	ID2
FFD6$_{16}$	ID3
FFD7$_{16}$	ID4
FFD8$_{16}$	ID5
FFD9$_{16}$	ID6
FFDA$_{16}$	ID7
FFDB$_{16}$	ROM code protect control
	Interrupt vector area

Fig. 164 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 14 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 14 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 15 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)

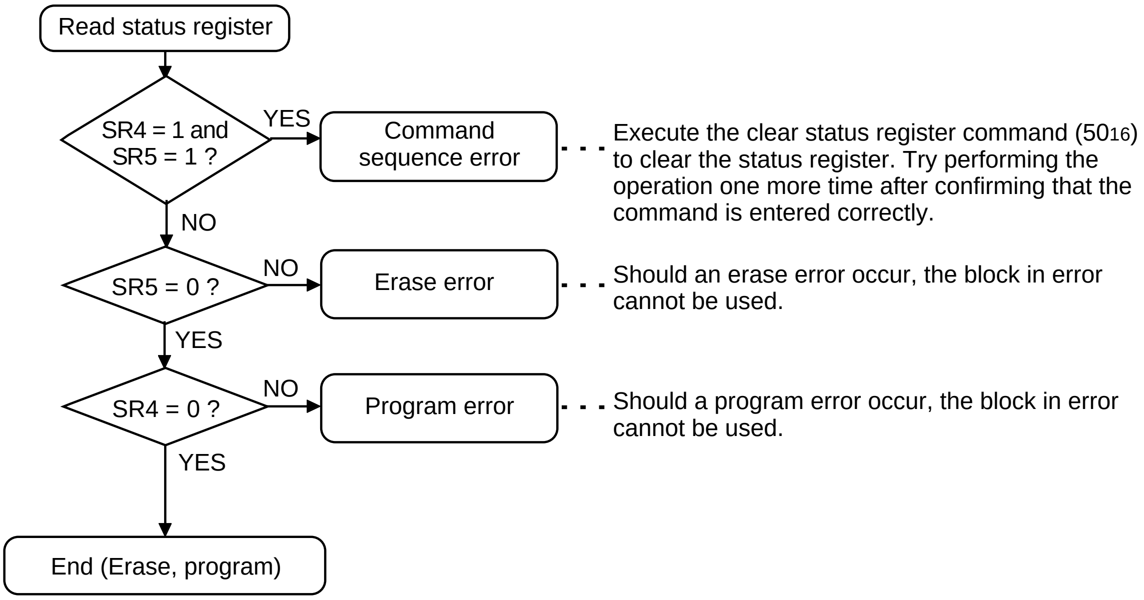
This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the MCU returns to the command wait state.

Table 15 Status register 1 (SRD1)

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

Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 165 shows a flowchart of the full status check and explains how to remedy errors which occur.



Execute the clear status register command (50_{16}) to clear the status register. Try performing the operation one more time after confirming that the command is entered correctly.

Should an erase error occur, the block in error cannot be used.

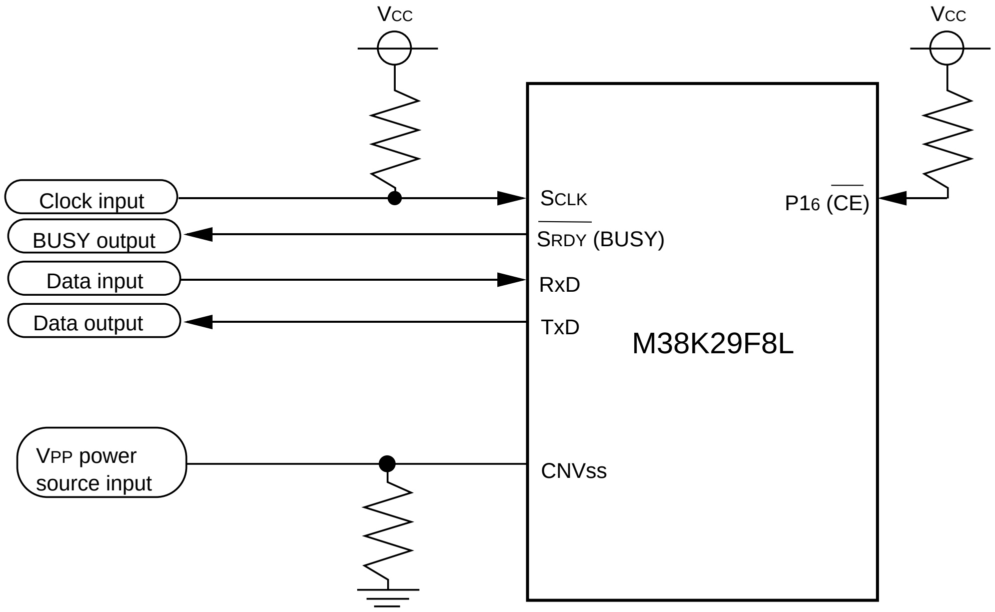
Should a program error occur, the block in error cannot be used.

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 (50_{16}) before executing these commands.

Fig. 165 Full status check flowchart and remedial procedure for errors

Example Circuit Application for Standard Serial I/O Mode

Figure 166 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 VPP 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. 166 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 167).

•Relative accuracy

① Zero transition voltage (VOT)
This means an analog input voltage when the actual A/D conversion output data changes from “0” to “1.”

② Full-scale transition voltage (VFST)
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 VOT and VFST of a converted value between VOT and VFST.

④ Differential non-linearity error
This means a deviation from the input potential difference required to change a converted value between VOT and VFST by 1 LSB of the 1 LSB at the relative accuracy.

•Absolute accuracy

This means a deviation from the ideal characteristics between 0 to VREF of actual A/D conversion characteristics.

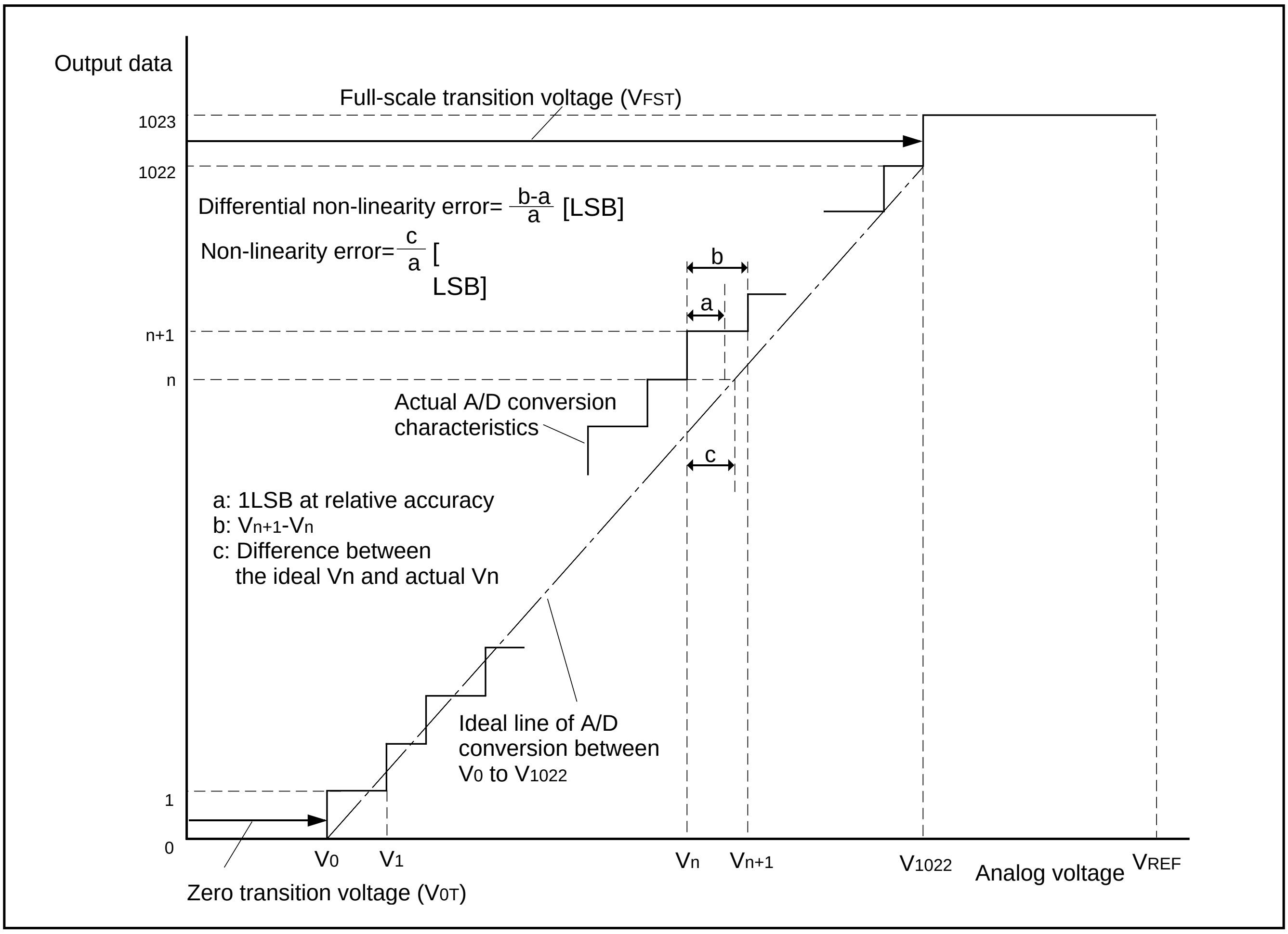

Output data
Full-scale transition voltage (VFST)
1023
1022
Differential non-linearity error= $\frac{b-a}{a}$ [LSB]
Non-linearity error= $\frac{c}{a}$ [LSB]
b
a
n+1
n
Actual A/D conversion characteristics
c
a: 1LSB at relative accuracy
b: V_{n+1}-V_n
c: Difference between the ideal Vn and actual Vn
Ideal line of A/D conversion between V_0 to V_{1022}
1
0
V_0
V_1
V_n
V_{n+1}
V_{1022}
Analog voltage
VREF
Zero transition voltage (VOT)

Fig. 167 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 → $\frac{V_{FST} - V_{OT}}{1022}$ (V)

• 1 LSB at absolute accuracy → $\frac{V_{REF}}{1024}$ (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-, D1+, D1-, D2+, D2-) 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).

FUNCTIONAL DESCRIPTION SUPPLEMENT

A/D Converter

A/D conversion is started by setting AD conversion completion bit to "0." During A/D conversion, internal operations are performed as follows.

1. After the start of A/D conversion, AD conversion register goes to "00_{16}."
2. The highest-order bit of AD conversion register is set to "1," and the comparison voltage Vref is input to the comparator. Then, V_{ref} is compared with analog input voltage V_{IN}.
3. As a result of comparison, when $V_{ref} < V_{IN}$, the highest-order bit of AD conversion register becomes "1." When $V_{ref} > V_{IN}$, the highest-order bit becomes "0."

By repeating the above operations up to the lowest-order bit of the AD conversion register, an analog value converts into a digital value.

A/D conversion completes at 122 clock cycles (15.25 µs at system clock = 8 MHz, Through mode) after it is started, and the result of the conversion is stored into the AD conversion register.

Concurrently with the completion of A/D conversion, A/D conversion interrupt request occurs, so that the AD conversion interrupt request bit is set to "1."

Table 16 Relative formula for a reference voltage V_{REF} of A/D converter and V_{ref}

When n = 0	$V_{ref} = 0$
When n = 1 to 1023	$V_{ref} = \frac{V_{REF}}{1024} \times n$

n: Value of A/D converter (decimal numeral)

Table 17 Change of AD conversion register during A/D conversion

	Change of AD conversion register	Value of comparison voltage (V_{ref})
At start of conversion	0 0 0 0 0 0 0 0 0 0	0
First comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$
Second comparison	✻1 1 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4}$
Third comparison	✻1 ✻2 1 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8}$
	≈	≈
After completion of tenth comparison	A result of A/D conversion ✻1 ✻2 ✻3 ✻4 ✻5 ✻6 ✻7 ✻8 ✻9 ✻10	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \cdots\cdot \pm \frac{V_{REF}}{1024}$

✻1–✻10: A result of the first comparison to the tenth comparison

Figures 168 shows the A/D conversion equivalent circuit, and Figure 169 shows the A/D conversion timing chart.

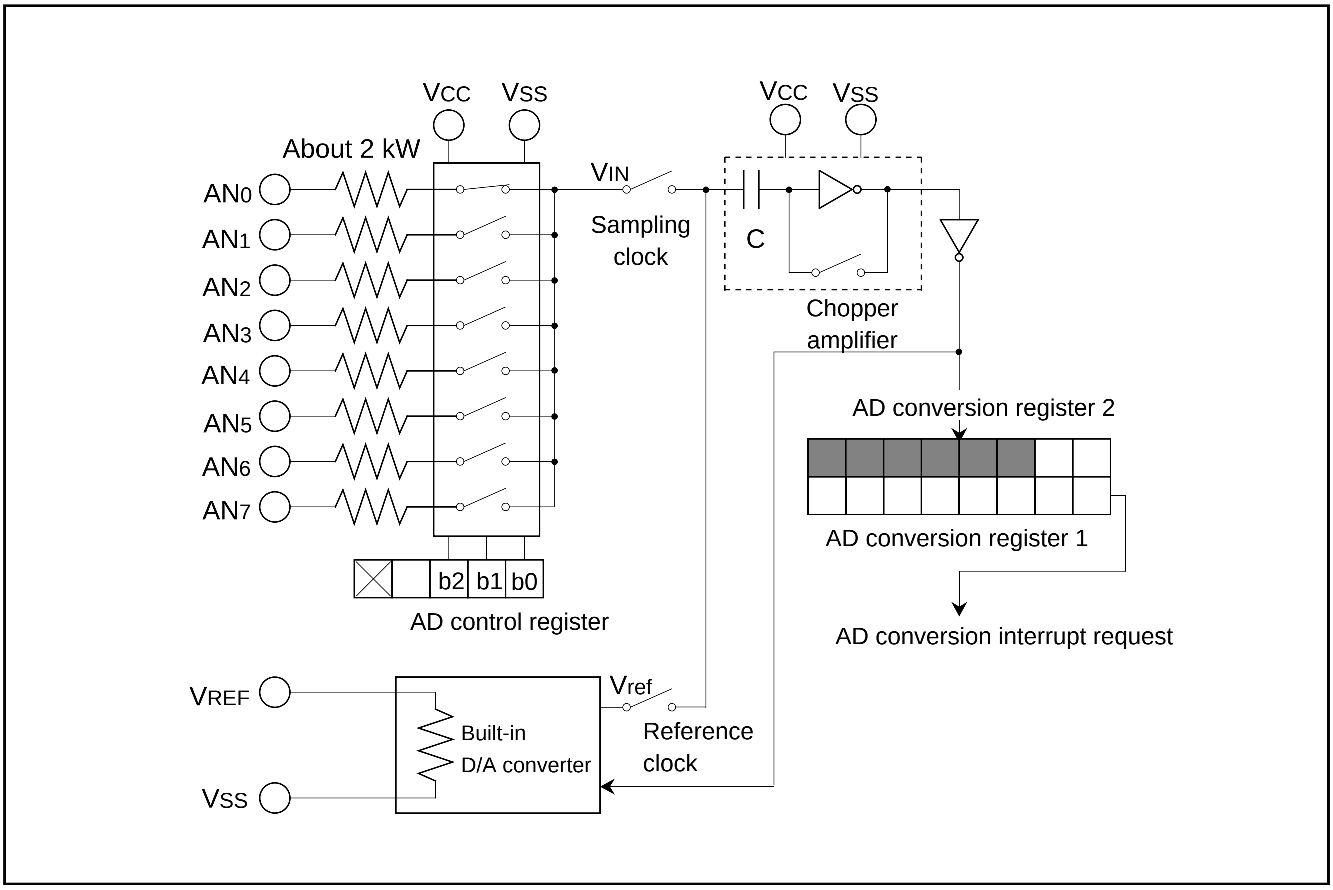


Fig. 168 A/D conversion equivalent circuit

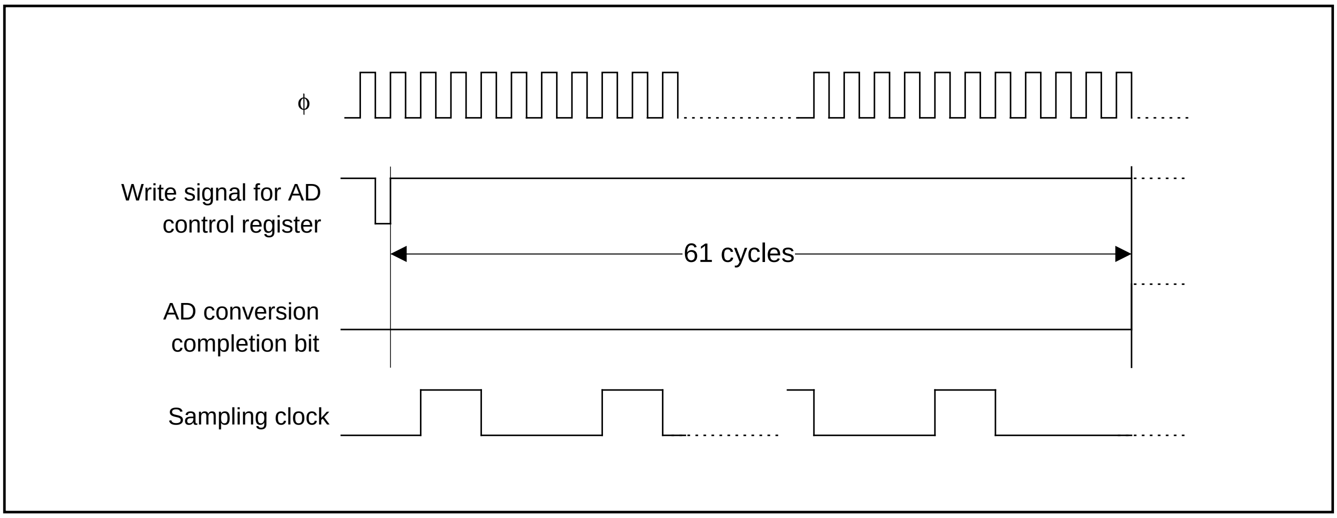

ϕ
Write signal for AD control register
61 cycles
AD conversion completion bit
Sampling clock

Fig. 169 A/D conversion timing chart

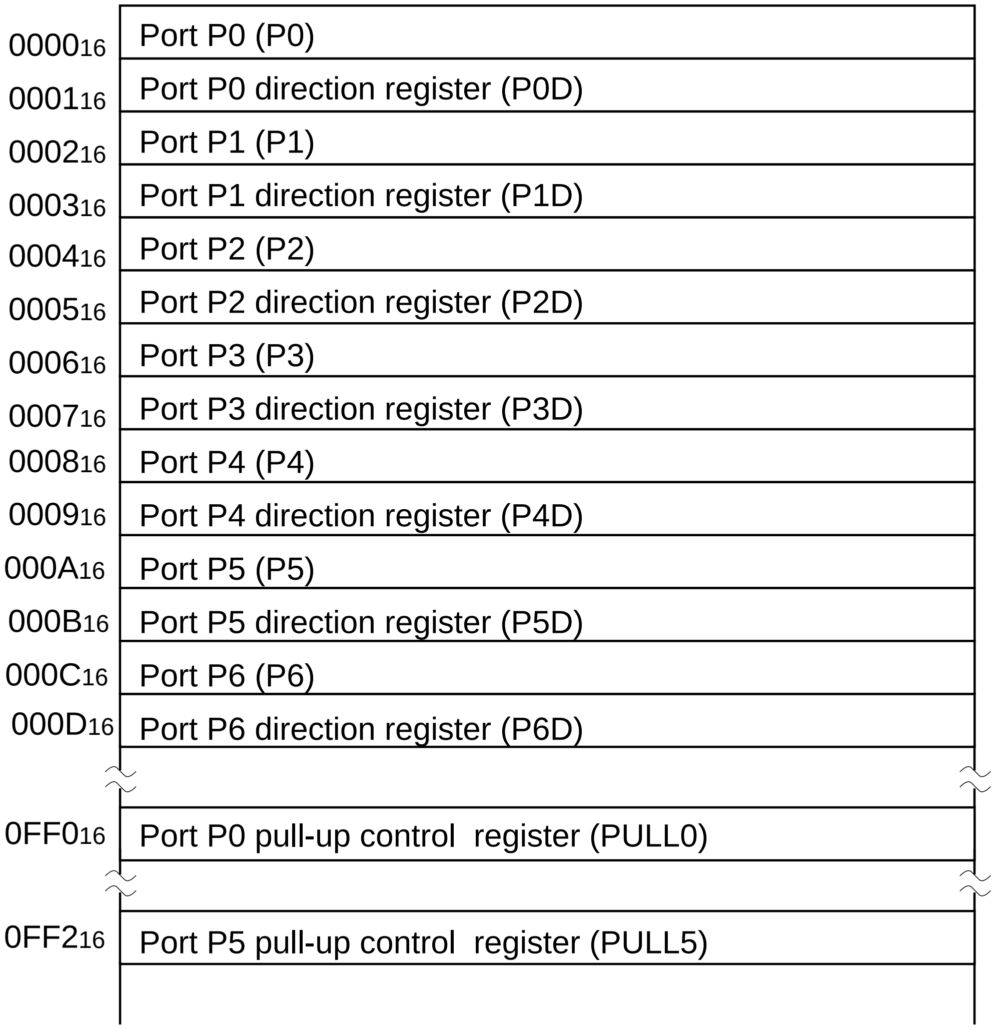
CHAPTER 2

APPLICATION

2.1 I/O port

This paragraph explains the registers setting method and the notes related to the I/O ports.

2.1.1 Memory map



0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	Port P1 (P1)
0003_{16}	Port P1 direction register (P1D)
0004_{16}	Port P2 (P2)
0005_{16}	Port P2 direction register (P2D)
0006_{16}	Port P3 (P3)
0007_{16}	Port P3 direction register (P3D)
0008_{16}	Port P4 (P4)
0009_{16}	Port P4 direction register (P4D)
$000A_{16}$	Port P5 (P5)
$000B_{16}$	Port P5 direction register (P5D)
$000C_{16}$	Port P6 (P6)
$000D_{16}$	Port P6 direction register (P6D)
$0FF0_{16}$	Port P0 pull-up control register (PULL0)
$0FF2_{16}$	Port P5 pull-up control register (PULL5)

Fig. 2.1.1 Memory map of registers related to I/O port

2.1.2 Related registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi) (i = 0, 1, 2, 3, 4, 5, 6) **(Note)**
[Address : 00_{16}, 02_{16}, 04_{16}, 06_{16}, 08_{16}, $0A_{16}$, $0C_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi$_0$	● In output mode Write } Port latch Read } ● In input mode Write : Port latch Read : Value of pins	?	○	○
1	Port Pi$_1$		?	○	○
2	Port Pi$_2$		?	○	○
3	Port Pi$_3$		?	○	○
4	Port Pi$_4$		?	○	○
5	Port Pi$_5$		?	○	○
6	Port Pi$_6$		?	○	○
7	Port Pi$_7$		?	○	○

Note: Since the following ports are not allocated, the corrrsponding bits can not be used.

- • P2$_0$ to P2$_3$
- • P4$_4$ to P4$_7$
- • P6$_4$ to P6$_7$

Fig. 2.1.2 Structure of Port Pi (i = 0 to 6)

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi direction register (PiD) (i = 0, 1, 2, 3, 4, 5, 6) **(Note)**
[Address : 01_{16}, 03_{16}, 05_{16}, 07_{16}, 09_{16}, $0B_{16}$, $0D_{16}$]

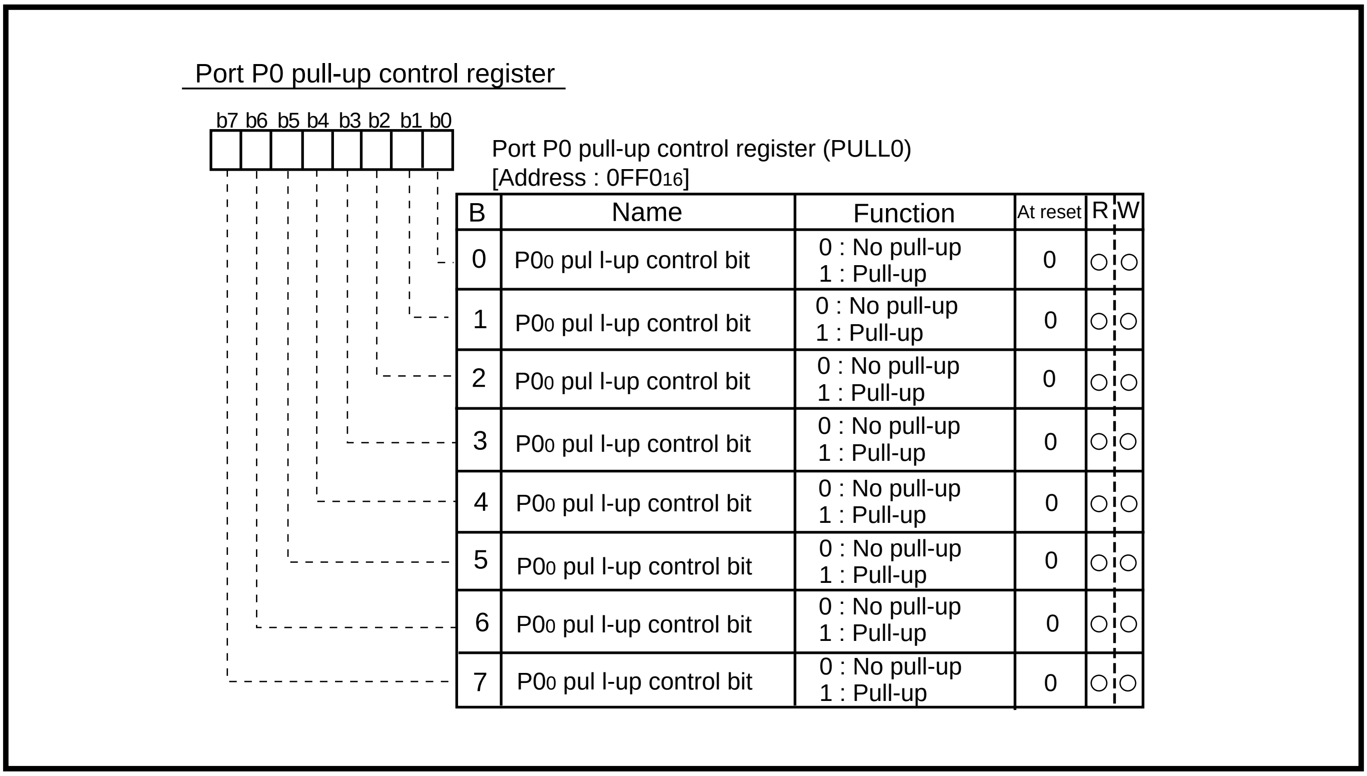
B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi$_0$ input mode 1 : Port Pi$_0$ output mode	0	×	○
1		0 : Port Pi$_1$ input mode 1 : Port Pi$_1$ output mode	0	×	○
2		0 : Port Pi$_2$ input mode 1 : Port Pi$_2$ output mode	0	×	○
3		0 : Port Pi$_3$ input mode 1 : Port Pi$_3$ output mode	0	×	○
4		0 : Port Pi$_4$ input mode 1 : Port Pi$_4$ output mode	0	×	○
5		0 : Port Pi$_5$ input mode 1 : Port Pi$_5$ output mode	0	×	○
6		0 : Port Pi$_6$ input mode 1 : Port Pi$_6$ output mode	0	×	○
7		0 : Port Pi$_7$ input mode 1 : Port Pi$_7$ output mode	0	×	○

Note: Since the following ports are not allocated, the corrrsponding bits can not be used.

- • P2$_0$ to P2$_3$
- • P4$_4$ to P4$_7$
- • P6$_4$ to P6$_7$

Do not set bits of the direction register corresponding to ports P2$_0$–P2$_3$ (bits 0–3 of port P2 direction register (address 05_{16})) to output mode (“1”).
If writing to these bits, write “0”.

Fig. 2.1.3 Structure of Port Pi direction register (i = 0 to 6)

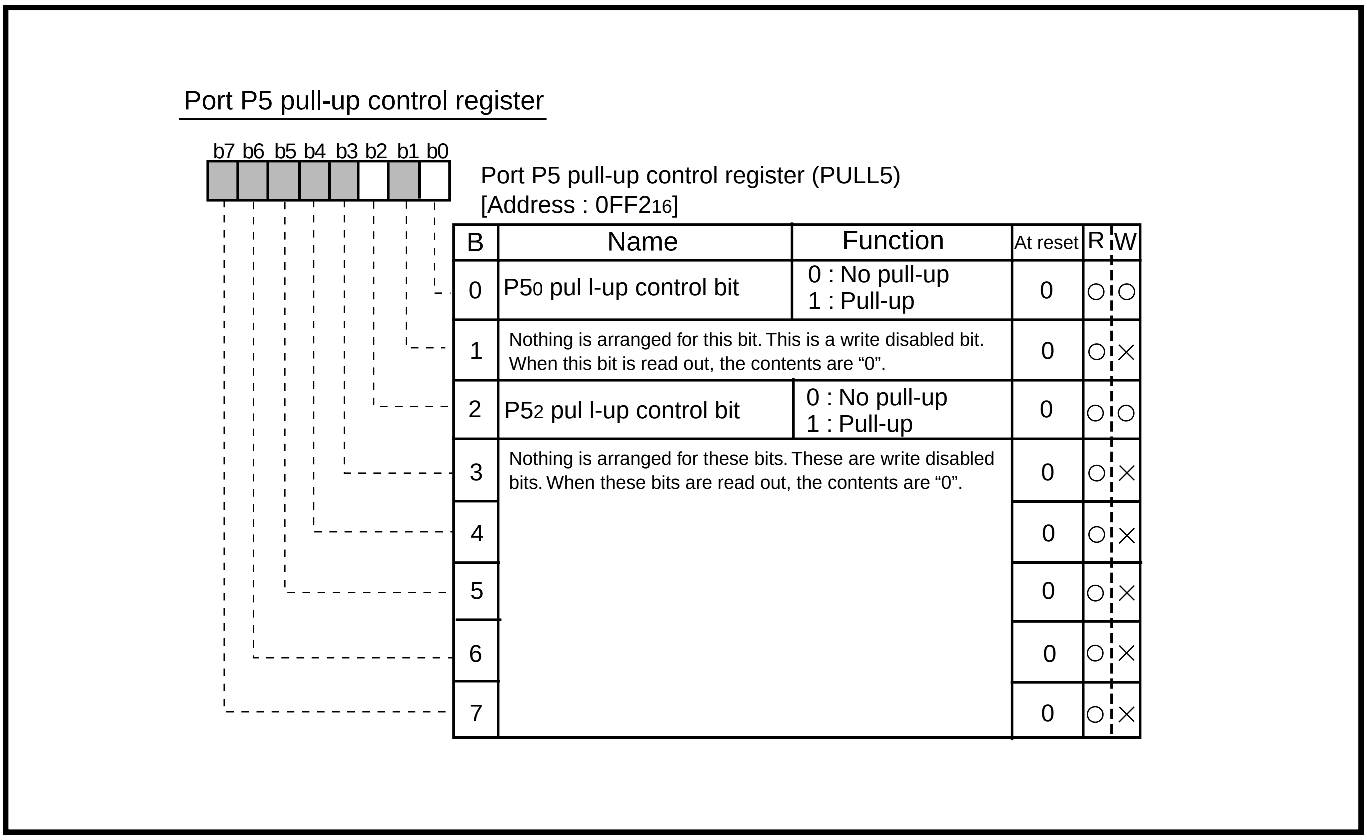


Port P0 pull-up control register

Port P0 pull-up control register (PULL0)
[Address : 0FF0$_{16}$]

B	Name	Function	At reset	R	W
0	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
2	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
3	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
4	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
5	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
6	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
7	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○

Fig. 2.1.4 Structure of Port P0 pull-up control register



Port P5 pull-up control register

Port P5 pull-up control register (PULL5)
[Address : 0FF2$_{16}$]

B	Name	Function	At reset	R	W
0	P5$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
2	P5$_2$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.1.5 Structure of Port P5 pull-up control register

2.1.3 Handling of unused pins

Table 2.1.1 Handling of unused pins

Pins/Ports name	Handling
P0, P1, P2, P3, P4, P5, P6	•Set to the input mode and connect each to Vcc or Vss through a resistor of 1 kΩ to 10 kΩ. •Set to the output mode and open at “L” or “H” level.
V_{REF}	•Connect to Vss (GND).
X_{OUT}	•Open, only when using an external clock.
$USBV_{REF}$	•Connect to V_{CC}
TrON	•Open
D0+, D0-, D1+, D1-, D2+, D2-	•Connect each to Vss through a resistor of 1 kΩ to 10 kΩ.

2.1.4 Notes on input and output pins

(1) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*[1], the value of the unspecified bit may be changed.

● **Reason**

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the port latch.

- As for a bit which is set for an input port :
  The pin state is read in the CPU, and is written to this bit after bit managing.
- As for a bit which is set for an output port :
  The bit value of the port latch is read in the CPU, and is written to this bit after bit managing.

Note the following :

- Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.
- As for a bit of the port latch which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

*[1] bit managing instructions : **SEB**, and **CLB** instructions

2.1.5 Termination of unused pins

(1) Terminate unused pins

① I/O ports :

- Set the I/O ports for the input mode and connect them to Vcc or Vss through each resistor of 1 kΩ to 10 kΩ. With regard to ports which can select the built-in pull-up resistor, the built-in pull-up resistor can be used.
  Set the I/O ports for the output mode and open them at “L” or “H”.
- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

(2) Termination remarks

① I/O ports :
Do not open in the input mode.

● **Reason**

- The power source current may increase depending on the first-stage circuit.
- An effect due to noise may be easily produced as compared with proper termination shown in (1).

② I/O ports :
When setting for the input mode, do not connect to Vcc or Vss directly.

● **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and Vcc (or Vss).

③ I/O ports :
When setting for the input mode, do not connect multiple ports in a lump to Vcc or Vss through a resistor.

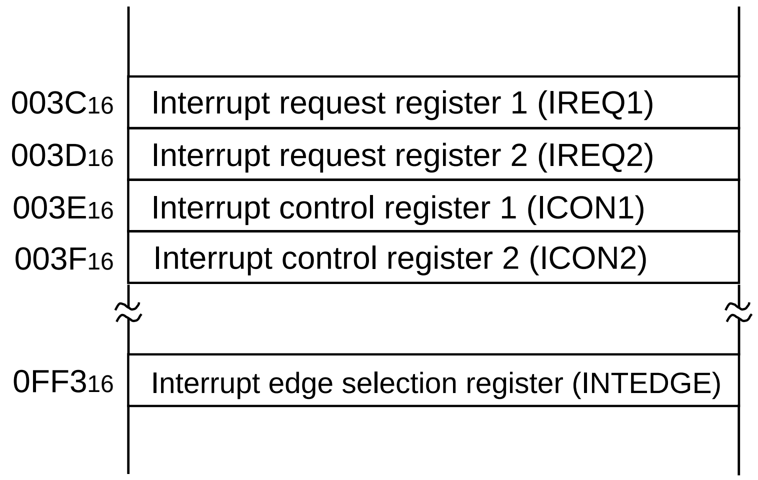
● **Reason**
If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

- At the termination of unused pins, perform wiring at the shortest possible distance (20 mm or less) from microcomputer pins.

2.2 Interrupt

This paragraph explains the registers setting method and the notes related to the interrupt.

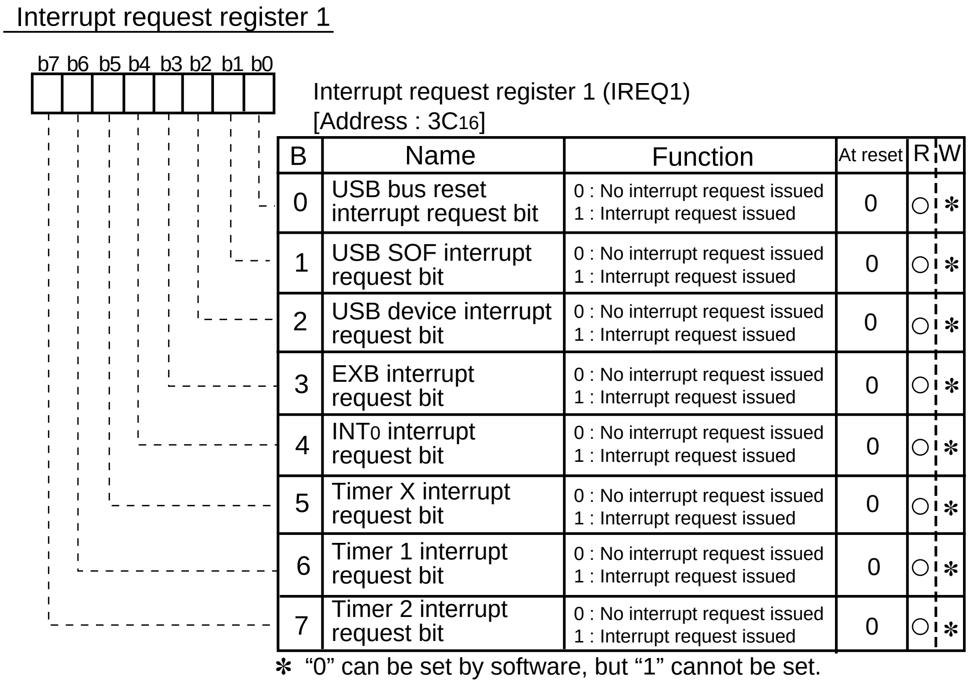
2.2.1 Memory map



$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)
$0FF3_{16}$	Interrupt edge selection register (INTEDGE)

Fig. 2.2.1 Memory map of registers related to interrupt

2.2.2 Related registers



Interrupt request register 1

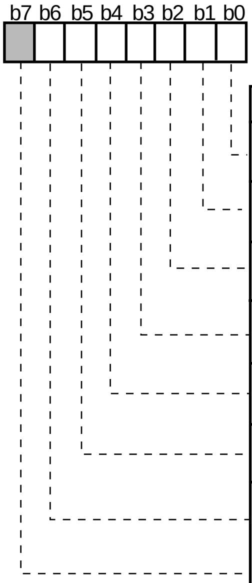
Interrupt request register 1 (IREQ1)
[Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	USB bus reset interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB SOF interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	USB device interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	EXB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻ "0" can be set by software, but "1" cannot be set.

Fig. 2.2.2 Structure of Interrupt request register 1

Interrupt request register 2



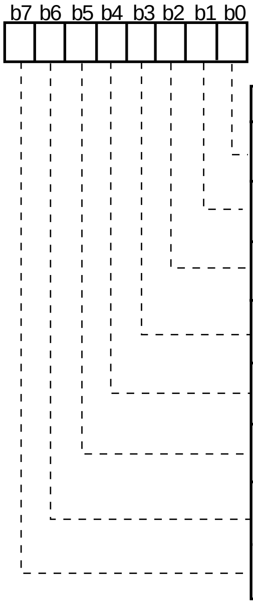
Interrupt request register 2 (IREQ2)
[Address : 3D16]

B	Name	Function	At reset	R	W
0	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB HUB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A/D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

✻ “0” can be set by software, but “1” cannot be set.

Fig. 2.2.3 Structure of Interrupt request register 2

Interrupt control register 1

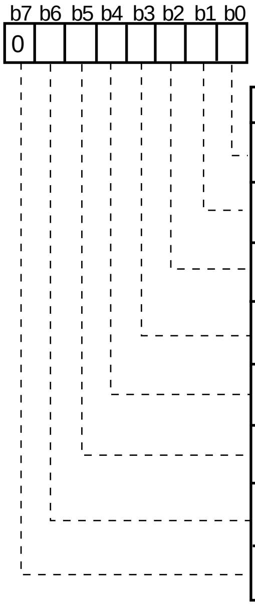


Interrupt control register 1 (ICON1)
[Address : 3E16]

B	Name	Function	At reset	R	W
0	USB bus reset interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB SOF interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	USB device interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	EXB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.2.4 Structure of Interrupt control register 1

Interrupt control register 2

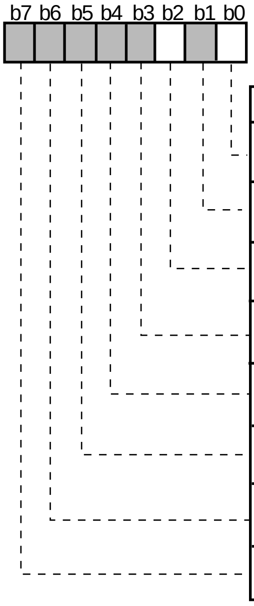


Interrupt control register 2 (ICON2)
[Address : 3F16]

B	Name	Function	At reset	R	W
0	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB HUB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	CNTR0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A/D conversion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.2.5 Structure of Interrupt control register 2

Interrupt edge selection register



Interrupt edge selection register (INTEDGE)
[Address : 0FF316]

B	Name	Function	At reset	R	W
0	INT0 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
2	INT1 interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.2.6 Structure of Interrupt edge selection register

2.2.3 Interrupt source

The 38K2 group permits interrupts of 16 sources. These are vector interrupts with a fixed priority system. Accordingly, when two or more interrupt requests occur during the same sampling, the higher-priority interrupt is accepted first. This priority is determined by hardware, but a variety of priority processing can be performed by software, using an interrupt enable bit and an interrupt disable flag.

For interrupt sources, vector addresses and interrupt priority, refer to Table 2.2.1.

Table 2.2.1 Interrupt sources, vector addresses and priority of 38K2 group

Interrupt Source	Priority	Vector Addresses **(Note 1)**		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset **(Note 2)**	1	$FFFD_{16}$	$FFFC_{16}$	At reset	Non-maskable
USB bus reset	2	$FFFB_{16}$	$FFFA_{16}$	At detection of USB bus reset signal (2.5 μs interval SE0)	Valid when USB is selected
USB SOF	3	$FFF9_{16}$	$FFF8_{16}$	At detection of USB SOF signal	Valid when USB is selected
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	Valid when USB is selected
External bus	5	$FFF5_{16}$	$FFF4_{16}$	At completion of reception or transmission or at completion of DMA transmission	Valid when external bus is selected
INT_0	6	$FFF3_{16}$	$FFF2_{16}$	At detection of either rising or falling edge of INT_0 input	External interrupt (active edge selectable)
Timer X	7	$FFF1_{16}$	$FFF0_{16}$	At timer X underflow	
Timer 1	8	$FFEF_{16}$	$FFEE_{16}$	At timer 1 underflow	STP release timer 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	External interrupt (active edge selectable)
USB HUB	11	$FFE9_{16}$	$FFE8_{16}$	At detection of status change of USB HUB down ports	Valid when USB HUB is selected
Serial I/O reception	12	$FFE7_{16}$	$FFE6_{16}$	At completion of serial I/O data reception	Valid when serial I/O is selected
Serial I/O transmission	13	$FFE5_{16}$	$FFE4_{16}$	At completion of serial I/O data transmission	Valid when serial I/O is selected
$CNTR_0$	14	$FFE3_{16}$	$FFE2_{16}$	At detection of either rising or falling edge of $CNTR_0$ input	External interrupt (active edge selectable)
Key-on wake up	15	$FFE1_{16}$	$FFE0_{16}$	At falling of conjunction of input level for port P0 (at input mode)	External interrupt (active edge selectable)
A/D conversion	16	$FFDF_{16}$	$FFDE_{16}$	At completion of A/D conversion	
BRK instruction	17	$FFDD_{16}$	$FFDC_{16}$	At BRK instruction execution	Non-maskable software interrupt

Notes 1: Vector addresses contain interrupt jump destination addresses.
2: Reset function in the same way as an interrupt with the highest priority.

2.2.4 Interrupt operation

When an interrupt request is accepted, the contents of the following registers just before acceptance of the interrupt requests is automatically pushed onto the stack area in the order of ①, ② and ③.

①High-order contents of program counter (PC_H)
②Low-order contents of program counter (PC_L)
③Contents of processor status register (PS)

After the contents of the above registers are pushed onto the stack area, the accepted interrupt vector address enters the program counter and consequently the interrupt processing routine is executed.
When the RTI instruction is executed at the end of the interrupt processing routine, the contents of the above registers pushed onto the stack area are restored to the respective registers in the order of ③, ② and ①; and the microcomputer resumes the processing executed just before acceptance of the interrupts.
Figure 2.2.7 shows an interrupt operation diagram.

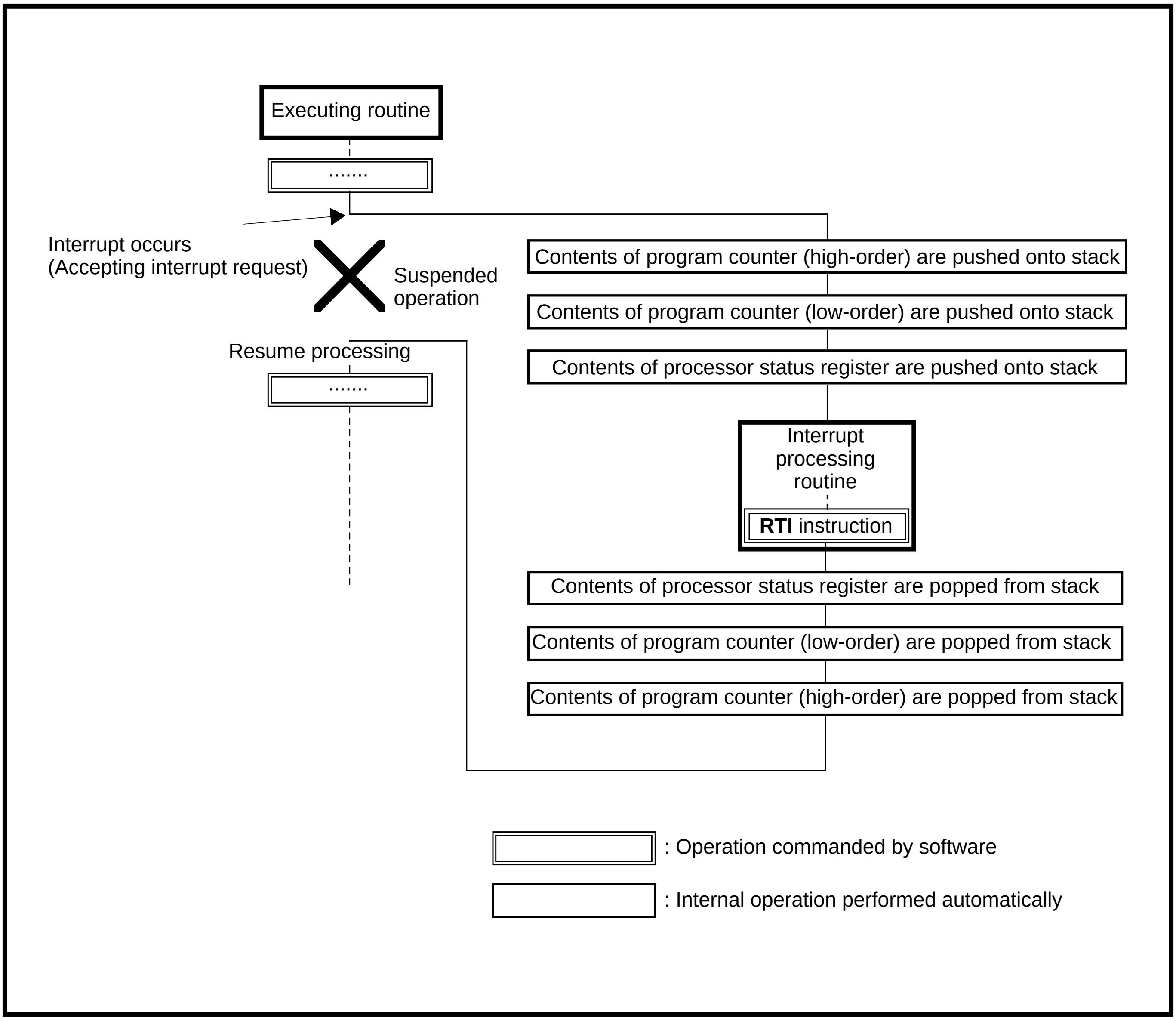

Executing routine
....
Interrupt occurs
(Accepting interrupt request)
Suspended operation
Resume processing
....
Contents of program counter (high-order) are pushed onto stack
Contents of program counter (low-order) are pushed onto stack
Contents of processor status register are pushed onto stack
Interrupt processing routine
RTI instruction
Contents of processor status register are popped from stack
Contents of program counter (low-order) are popped from stack
Contents of program counter (high-order) are popped from stack
: Operation commanded by software
: Internal operation performed automatically

Fig. 2.2.7 Interrupt operation diagram

(1) Processing upon acceptance of interrupt request

Upon acceptance of an interrupt request, the following operations are automatically performed.

①The processing being executed is stopped.

②The contents of the program counter and the processor status register are pushed onto the stack area. Figure 2.2.8 shows the changes of the stack pointer and the program counter upon acceptance of an interrupt request.

③Concurrently with the push operation, the jump destination address (the beginning address of the interrupt processing routine) of the occurring interrupt stored in the vector address is set in the program counter, then the interrupt processing routine is executed.

④After the interrupt processing routine is started, the corresponding interrupt request bit is automatically cleared to “0”. The interrupt disable flag is set to “1” so that multiple interrupts are disabled.

Accordingly, for executing the interrupt processing routine, it is necessary to set the jump destination address in the vector area corresponding to each interrupt.

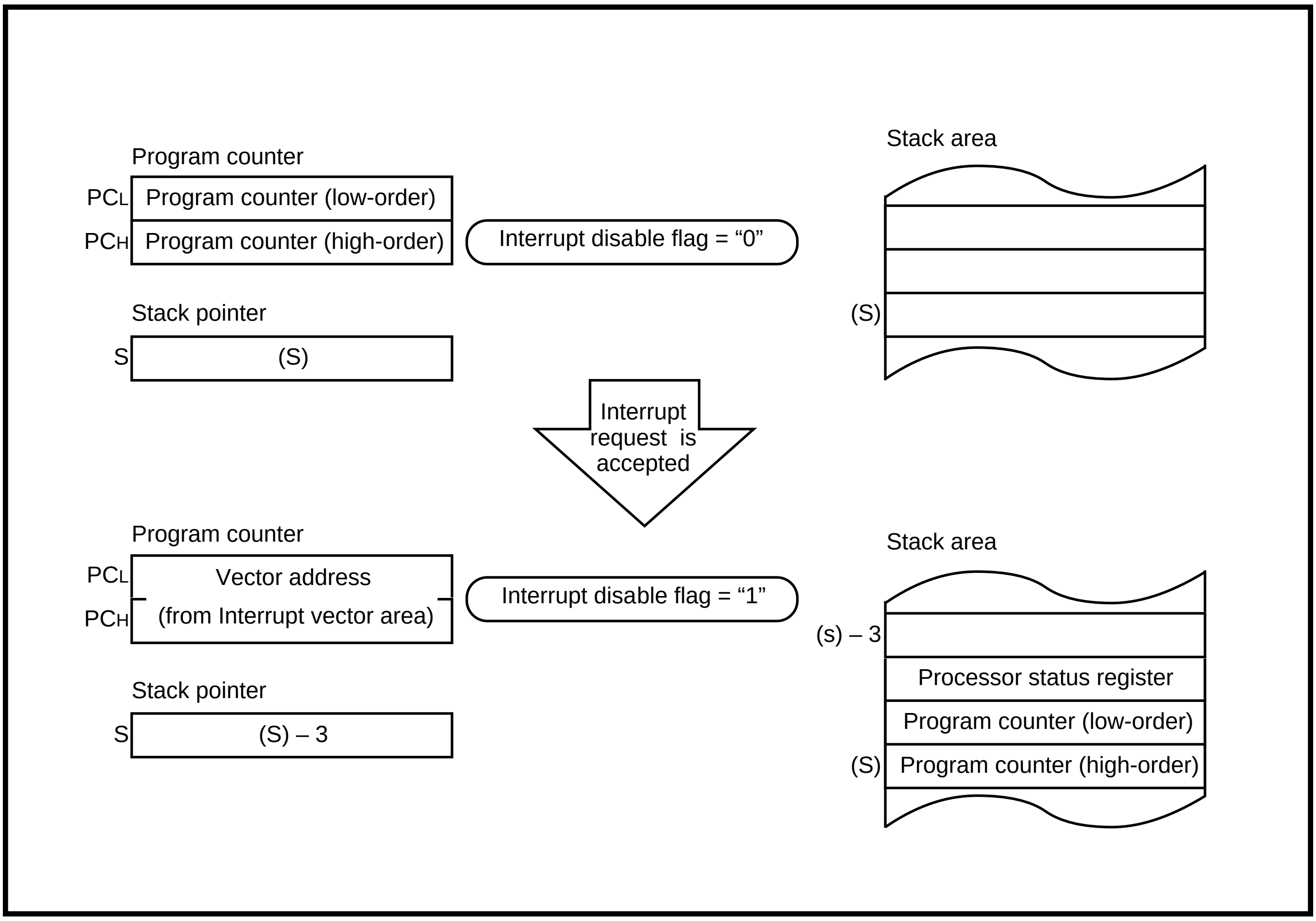

Program counter
PCL
Program counter (low-order)
PCH
Program counter (high-order)
Interrupt disable flag = “0”
Stack area
Stack pointer
S
(S)
(S)
Interrupt request is accepted
Program counter
PCL
Vector address
PCH
(from Interrupt vector area)
Interrupt disable flag = “1”
Stack area
(s) – 3
Processor status register
Program counter (low-order)
(S)
Program counter (high-order)
Stack pointer
S
(S) – 3

Fig. 2.2.8 Changes of stack pointer and program counter upon acceptance of interrupt request

(2) Timing after acceptance of interrupt request

The interrupt processing routine begins with the machine cycle following the completion of the instruction that is currently being executed.

Figure 2.2.9 shows the time up to execution of interrupt processing routine and Figure 2.2.10 shows the timing chart after acceptance of interrupt request.

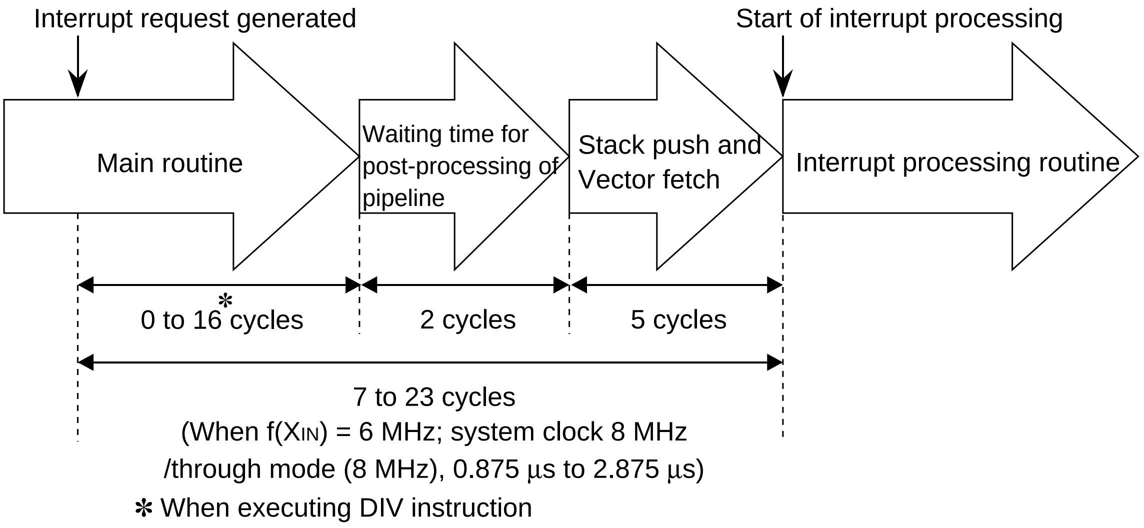
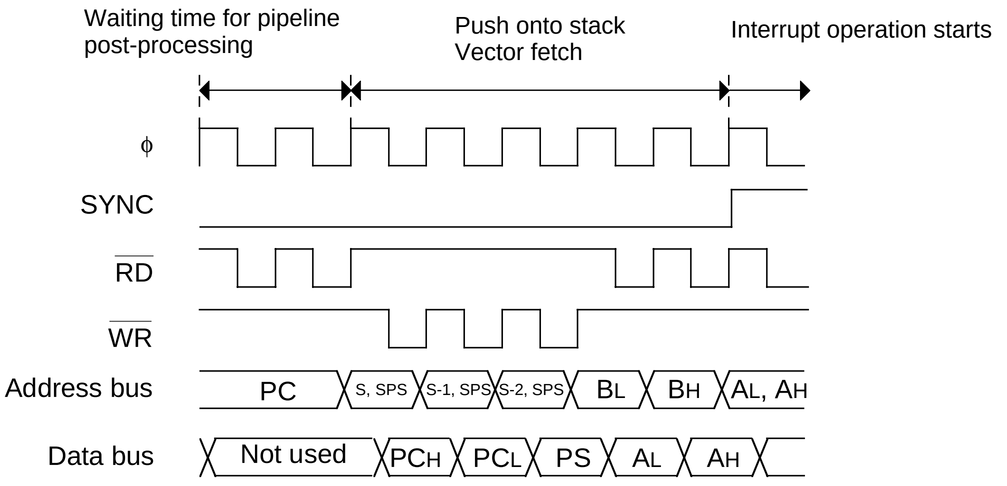

Interrupt request generated
Start of interrupt processing
Main routine
Waiting time for post-processing of pipeline
Stack push and Vector fetch
Interrupt processing routine
0 to 16✻ cycles
2 cycles
5 cycles
7 to 23 cycles
(When f(XIN) = 6 MHz; system clock 8 MHz /through mode (8 MHz), 0.875 µs to 2.875 µs)
✻ When executing DIV instruction

Fig. 2.2.9 Time up to execution of interrupt processing routine



SYNC : CPU operation code fetch cycle
(This is an internal signal that cannot be observed from the external unit.)
BL, BH : Vector address of each interrupt
AL, AH : Jump destination address of each interrupt
SPS : “00_{16}” or “01_{16}”

Fig. 2.2.10 Timing chart after acceptance of interrupt request

2.2.5 Interrupt control

The acceptance of all interrupts, excluding the BRK instruction interrupt, can be controlled by the interrupt request bit, interrupt enable bit, and an interrupt disable flag, as described in detail below. Figure 2.2.11 shows an interrupt control diagram.

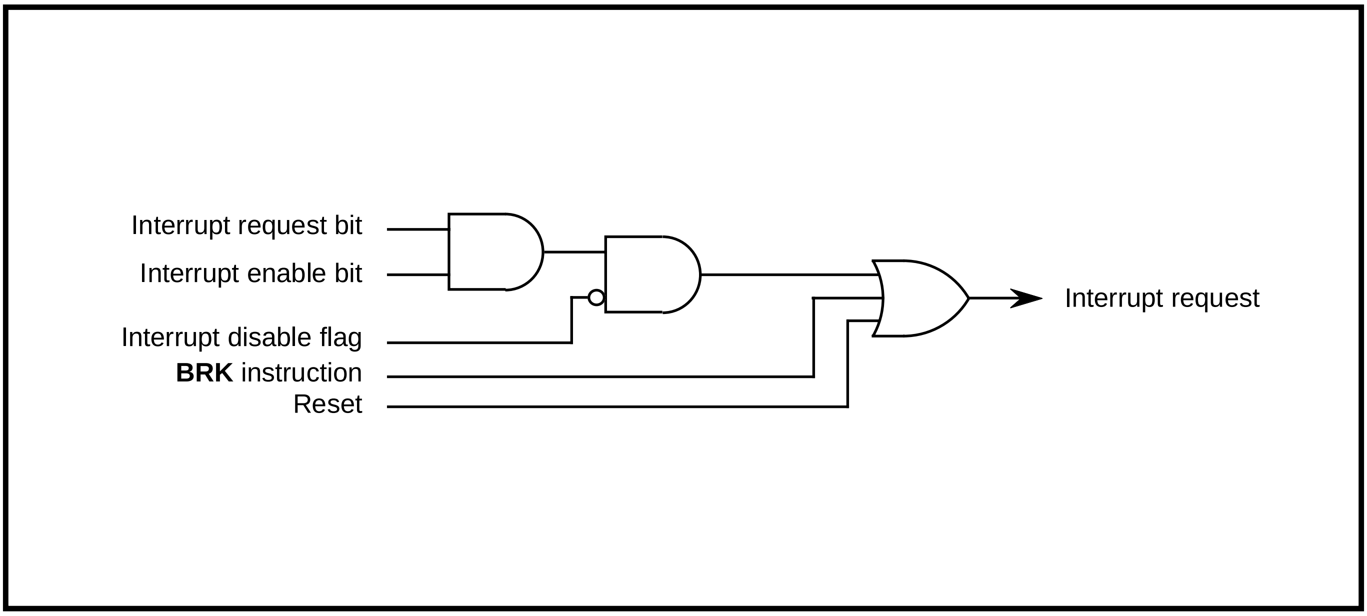


Fig. 2.2.11 Interrupt control diagram

The interrupt request bit, interrupt enable bit and interrupt disable flag function independently and do not affect each other. An interrupt is accepted when all the following conditions are satisfied.

●Interrupt request bit .... "1"
●Interrupt enable bit .... "1"
●Interrupt disable flag .... "0"

Though the interrupt priority is determined by hardware, a variety of priority processing can be performed by software using the above bits and flag. Table 2.2.2 shows a list of interrupt control bits according to the interrupt source.

(1) Interrupt request bits

The interrupt request bits are allocated to the interrupt request register 1 (address $3C_{16}$) and interrupt request register 2 (address $3D_{16}$).

The occurrence of an interrupt request causes the corresponding interrupt request bit to be set to "1". The interrupt request bit is held in the "1" state until the interrupt is accepted. When the interrupt is accepted, this bit is automatically cleared to "0".

Each interrupt request bit can be set to "0", but cannot be set to "1", by software.

(2) Interrupt enable bits

The interrupt enable bits are allocated to the interrupt control register 1 (address $003E_{16}$) and the interrupt control register 2 (address $3F_{16}$).

The interrupt enable bits control the acceptance of the corresponding interrupt request.

When an interrupt enable bit is "0", the corresponding interrupt request is disabled. If an interrupt request occurs when this bit is "0", the corresponding interrupt request bit is set to "1" but the interrupt is not accepted. In this case, unless the interrupt request bit is set to "0" by software, the interrupt request bit remains in the "1" state.

When an interrupt enable bit is "1", the corresponding interrupt is enabled. If an interrupt request occurs when this bit is "1", the interrupt is accepted (when interrupt disable flag = "0").

Each interrupt enable bit can be set to "0" or "1" by software.

(3) Interrupt disable flag

The interrupt disable flag is allocated to bit 2 of the processor status register. The interrupt disable flag controls the acceptance of interrupt request except BRK instruction.

When this flag is “1”, the acceptance of interrupt requests is disabled. When the flag is “0”, the acceptance of interrupt requests is enabled. This flag is set to “1” with the SEI instruction and is set to “0” with the CLI instruction.

When a main routine branches to an interrupt processing routine, this flag is automatically set to “1”, so that multiple interrupts are disabled. To use multiple interrupts, set this flag to “0” with the CLI instruction within the interrupt processing routine. Figure 2.2.12 shows an example of multiple interrupts.

Table 2.2.2 List of interrupt bits according to interrupt source

Interrupt source	Interrupt enable bit		Interrupt request bit	
	Address	Bit	Address	Bit
USB bus reset	$003E_{16}$	b0	$003C_{16}$	b0
USB SOF	$003E_{16}$	b1	$003C_{16}$	b1
USB device	$003E_{16}$	b2	$003C_{16}$	b2
External bus	$003E_{16}$	b3	$003C_{16}$	b3
INT_0	$003E_{16}$	b4	$003C_{16}$	b4
Timer X	$003E_{16}$	b5	$003C_{16}$	b5
Timer 1	$003E_{16}$	b6	$003C_{16}$	b6
Timer 2	$003E_{16}$	b7	$003C_{16}$	b7
INT_1	$003F_{16}$	b0	$003D_{16}$	b0
USB HUB	$003F_{16}$	b1	$003D_{16}$	b1
Serial I/O receive	$003F_{16}$	b2	$003D_{16}$	b2
Serial I/O transmit	$003F_{16}$	b3	$003D_{16}$	b3
$CNTR_0$	$003F_{16}$	b4	$003D_{16}$	b4
Key-on wake-up	$003F_{16}$	b5	$003D_{16}$	b5
A/D converter	$003F_{16}$	b6	$003D_{16}$	b6

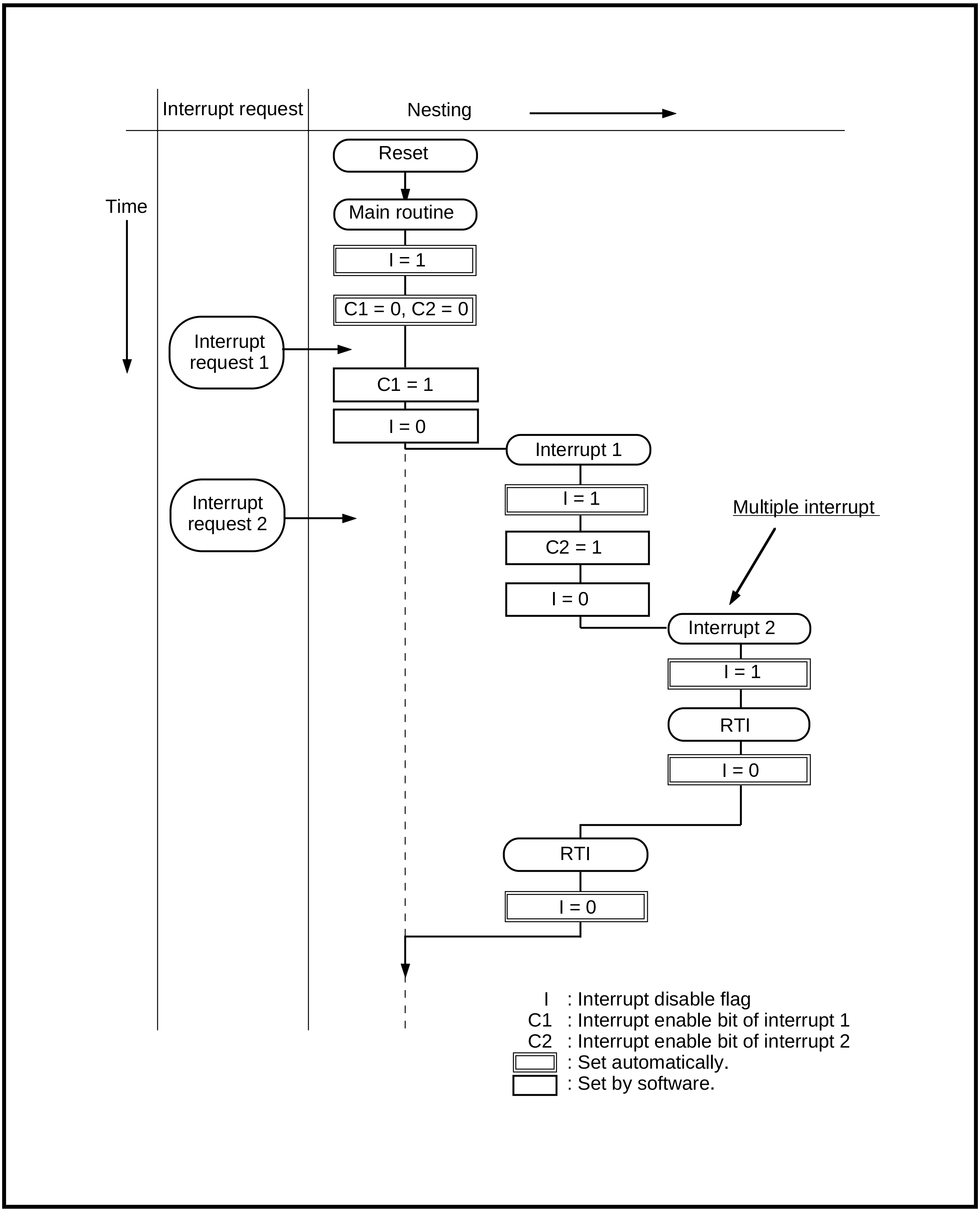


Fig. 2.2.12 Example of multiple interrupts

2.2.6 INT interrupt

The INT interrupt requests is generated when the microcomputer detects a level change of each INT pin (INT_0, INT_1).

(1) Active edge selection

INT_0 and INT_1 can be selected from either a falling edge or rising edge detection as an active edge by the interrupt edge selection register. In the “0” state, the falling edge of the corresponding pin is detected. In the “1” state, the rising edge of the corresponding pin is detected.

2.2.7 Key input interrupt

A key input interrupt request is generated by applying "L" level to any port P0 pin that has been set to the input mode. In other words, it is generated when AND of the input level goes from "1" to "0".

(1) Connection example when Key input interrupt is used

When using the Key input interrupt, compose an active-low key matrix which inputs to port P0. Figure 2.2.13 shows a connection example and the port P0 block diagram when using a key input interrupt. In the connection example in Figure 2.2.13, a key input interrupt request is generated by pressing one of the keys corresponding to ports $P0_0$ to $P0_3$.

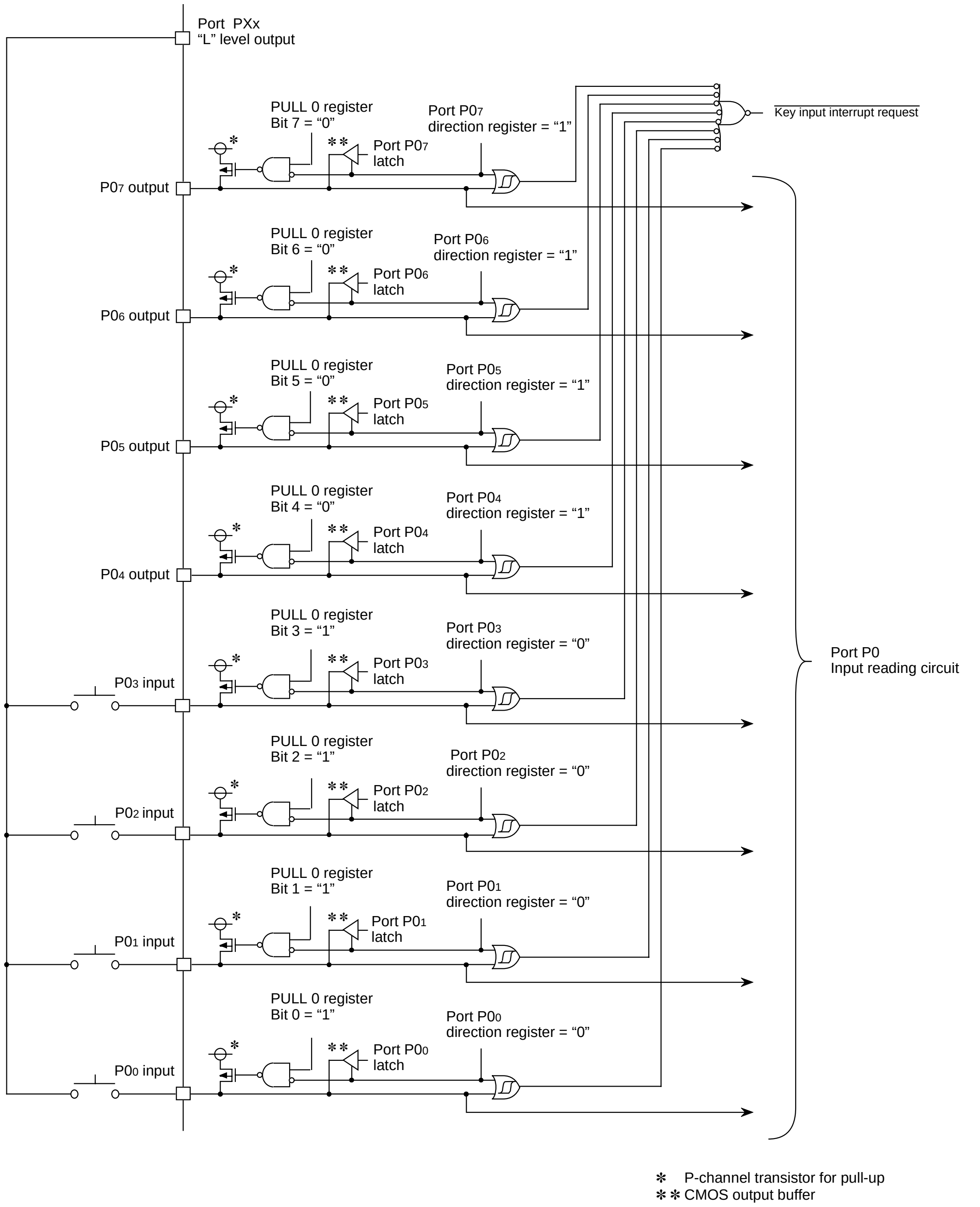


Fig. 2.2.13 Connection example and port P0 block diagram when using key input interrupt

(2) Related registers setting

Figure 2.2.14 shows the related registers setting (corresponding to Figure 2.2.13).

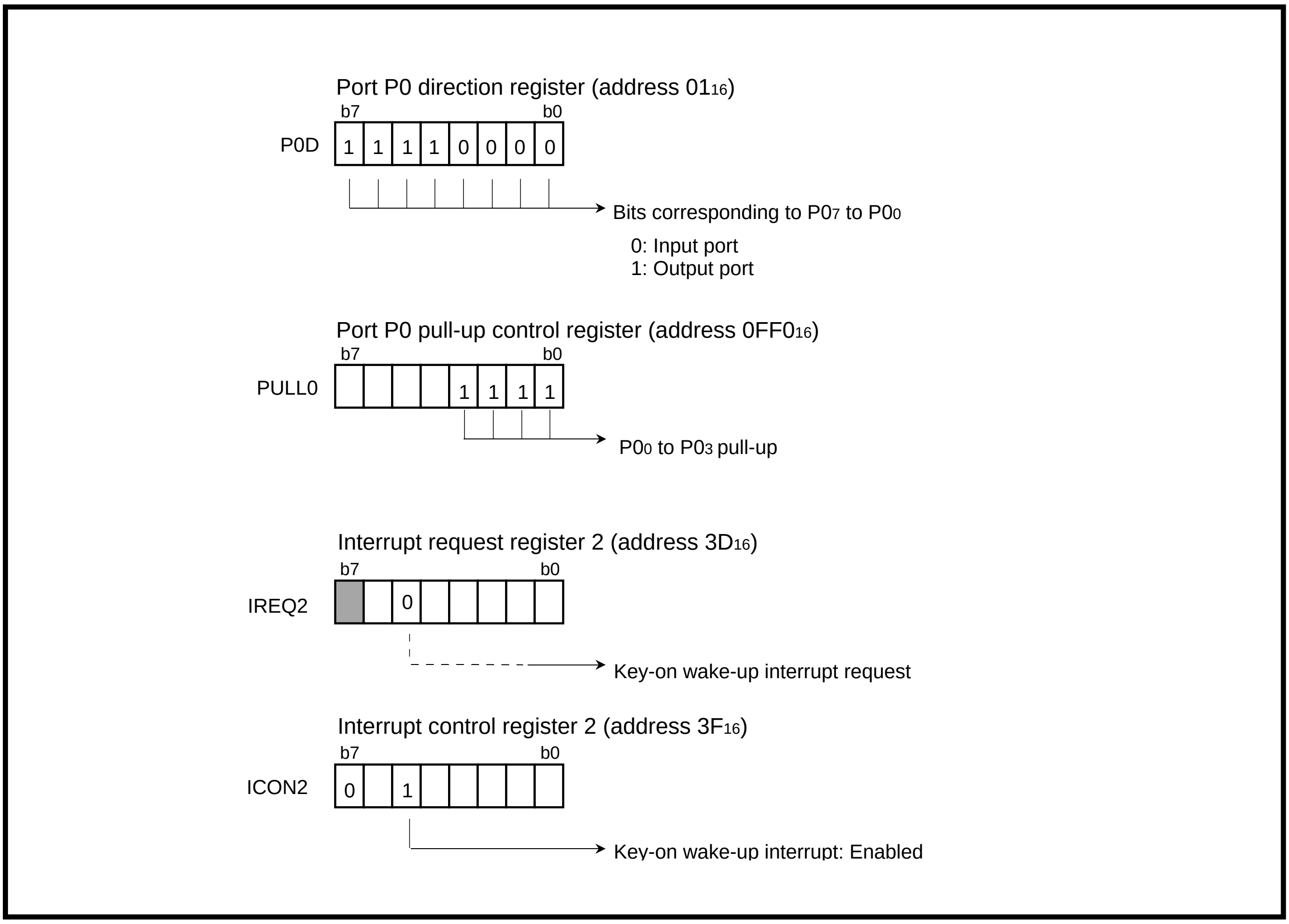

Port P0 direction register (address 01₁₆)
b7
b0
P0D
1 1 1 1 0 0 0 0
Bits corresponding to P07 to P00
0: Input port
1: Output port
Port P0 pull-up control register (address 0FF0₁₆)
b7
b0
PULL0
1 1 1 1
P00 to P03 pull-up
Interrupt request register 2 (address 3D₁₆)
b7
b0
IREQ2
0
Key-on wake-up interrupt request
Interrupt control register 2 (address 3F₁₆)
b7
b0
ICON2
0 1
Key-on wake-up interrupt: Enabled

Fig. 2.2.14 Registers setting related to key input interrupt (corresponding to Figure 2.2.13)

2.2.8 Notes on interrupts

(1) Change of relevant register settings

When the setting of the following registers or bits is changed, the interrupt request bit may be set to “1”. When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

•Interrupt edge selection register (address $0FF3_{16}$)

•Timer X mode register (address 23_{16})

Set the above listed registers or bits as the following sequence.

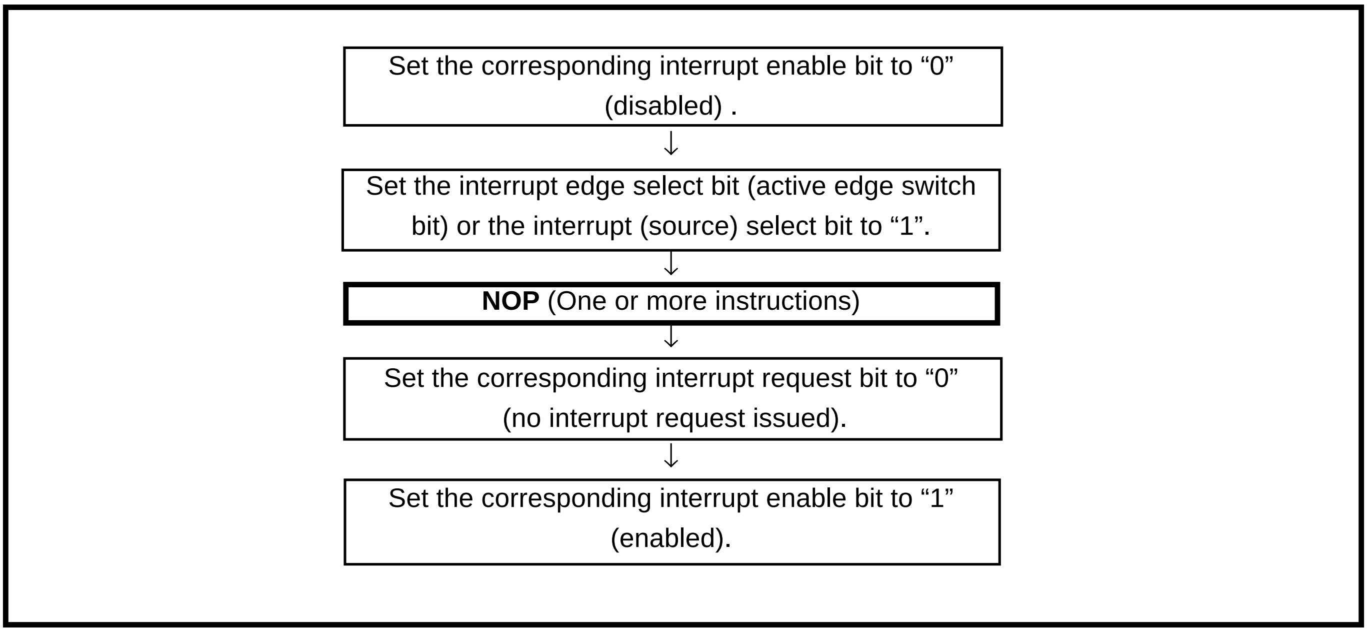

Set the corresponding interrupt enable bit to “0” (disabled) .
↓
Set the interrupt edge select bit (active edge switch bit) or the interrupt (source) select bit to “1”.
↓
NOP (One or more instructions)
↓
Set the corresponding interrupt request bit to “0” (no interrupt request issued).
↓
Set the corresponding interrupt enable bit to “1” (enabled).

Fig. 2.2.15 Sequence of changing relevant register

■ Reason

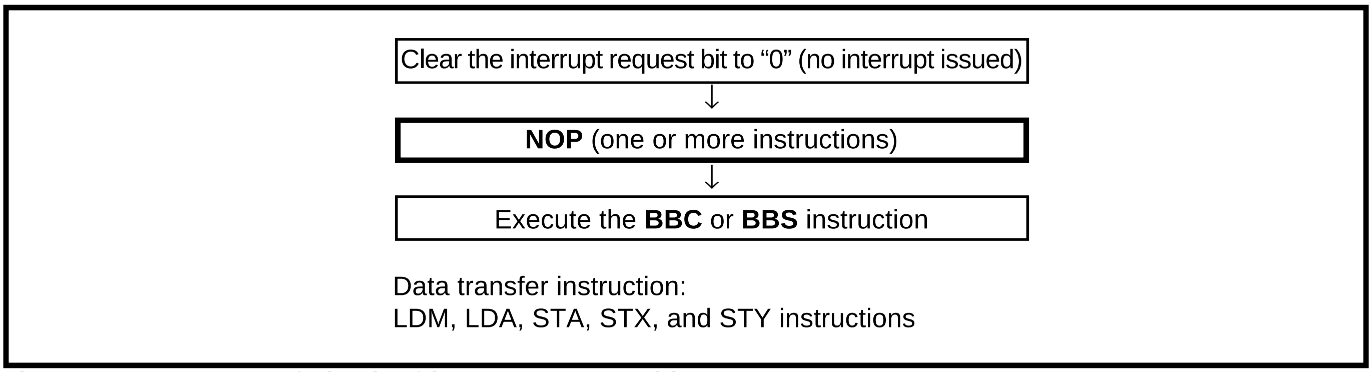
When setting the following, the interrupt request bit may be set to “1”.

•When setting external interrupt active edge

Concerned register: Interrupt edge selection register (address $0FF3_{16}$)
Timer X mode register (address 23_{16})

(2) Check of interrupt request bit

- When executing the **BBC** or **BBS** instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to “0” by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.


Clear the interrupt request bit to “0” (no interrupt issued)
NOP (one or more instructions)
Execute the BBC or BBS instruction

Data transfer instruction:
LDM, LDA, STA, STX, and STY instructions

Fig. 2.2.16 Sequence of check of interrupt request bit

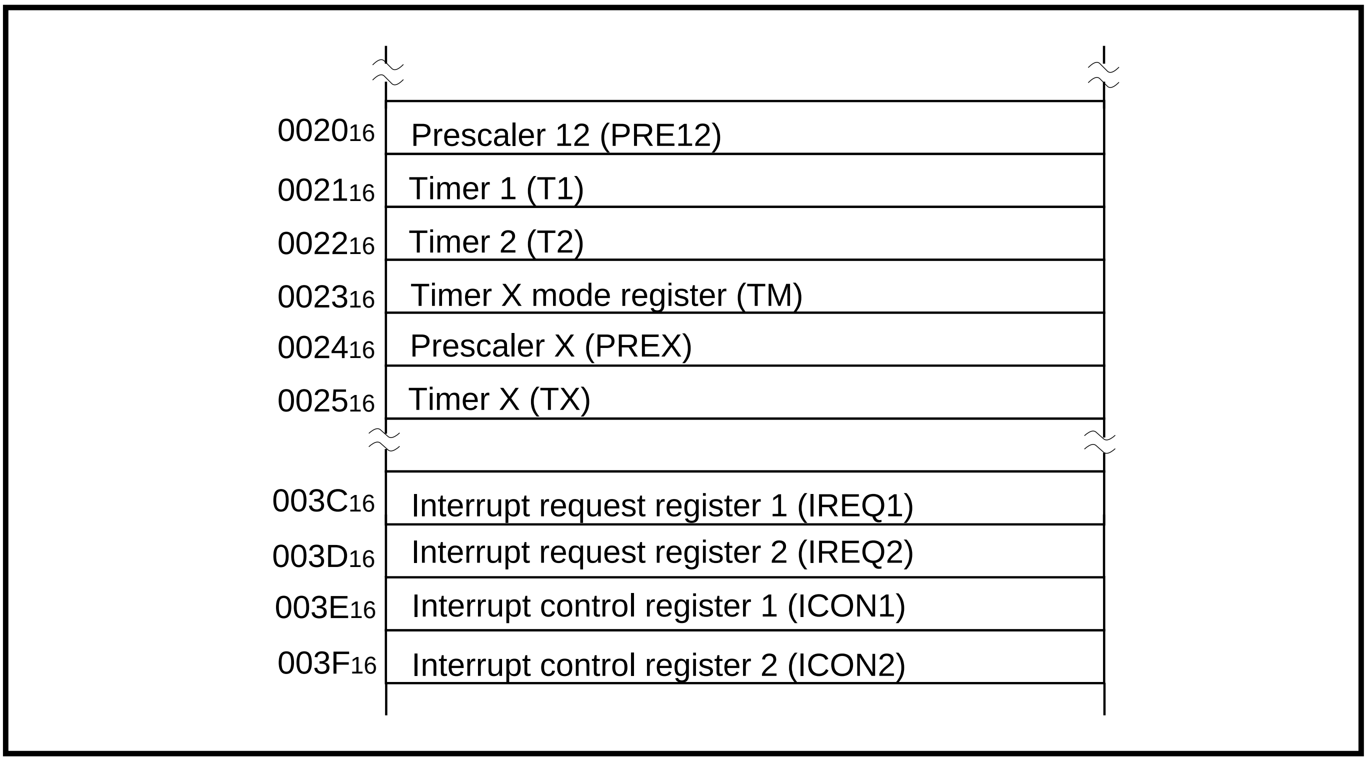
■ **Reason**

If the BBC or BBS instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to “0”, the value of the interrupt request bit before being cleared to “0” is read.

2.3 Timer

This paragraph explains the registers setting method and the notes related to the timers.

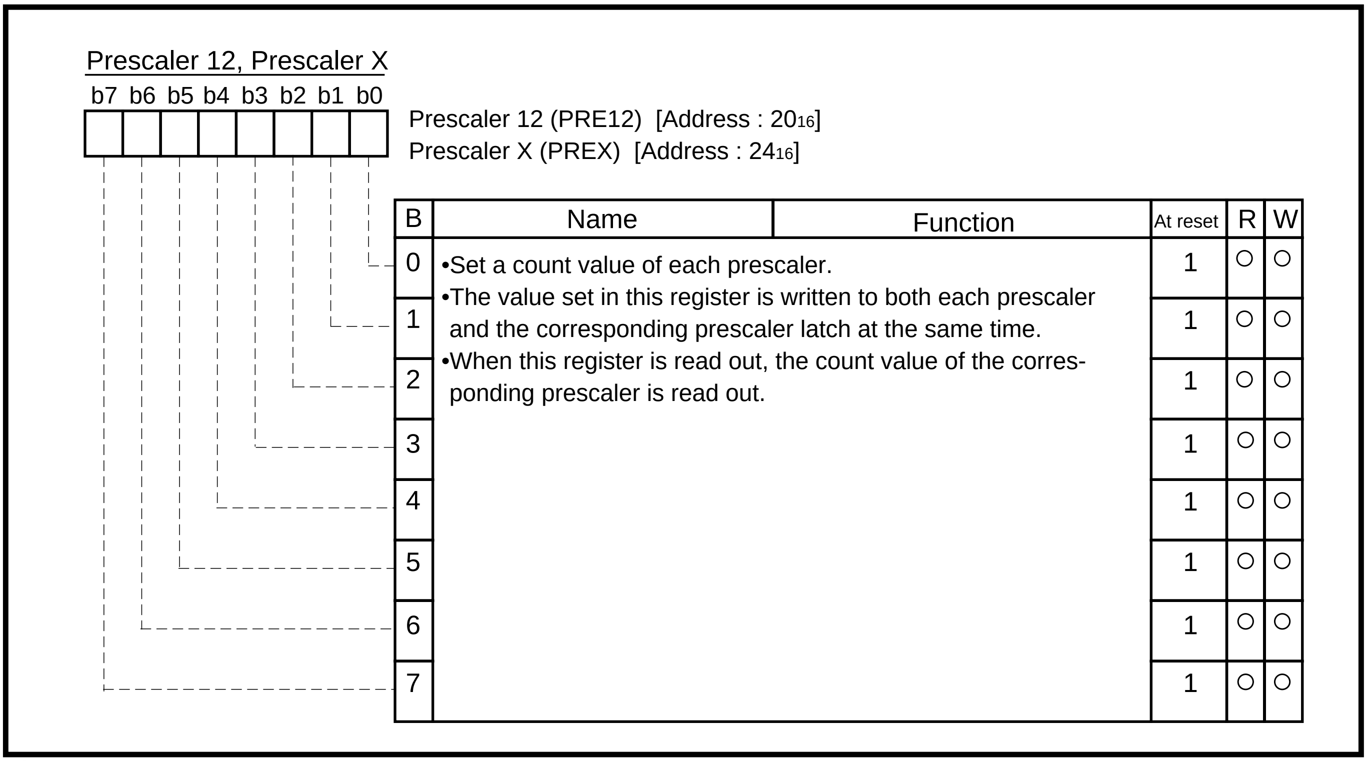
2.3.1 Memory map



Address	Register
0020_{16}	Prescaler 12 (PRE12)
0021_{16}	Timer 1 (T1)
0022_{16}	Timer 2 (T2)
0023_{16}	Timer X mode register (TM)
0024_{16}	Prescaler X (PREX)
0025_{16}	Timer X (TX)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)

Fig. 2.3.1 Memory map of registers related to timers

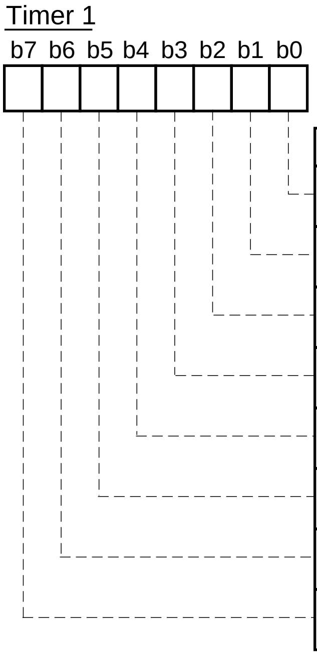
2.3.2 Related registers



Prescaler 12 (PRE12) [Address : 20_{16}]
Prescaler X (PREX) [Address : 24_{16}]

B	Name	Function	At reset	R	W
0		•Set a count value of each prescaler. •The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time. •When this register is read out, the count value of the corresponding prescaler is read out.	1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

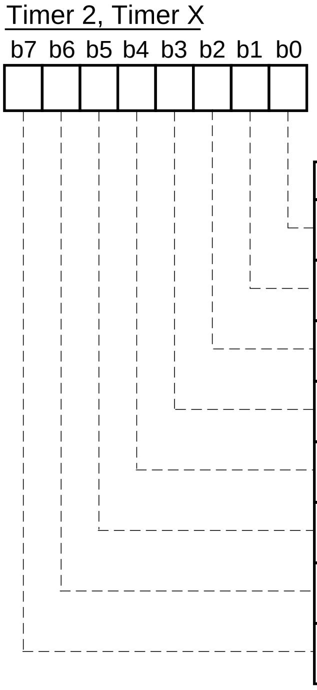
Fig. 2.3.2 Structure of Prescaler 12, Prescaler X



Timer 1 (T1) [Address : 21_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of timer 1. •The value set in this register is written to both timer 1 and timer 1 latch at the same time. •When this register is read out, the timer 1's count value is read out.		1	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

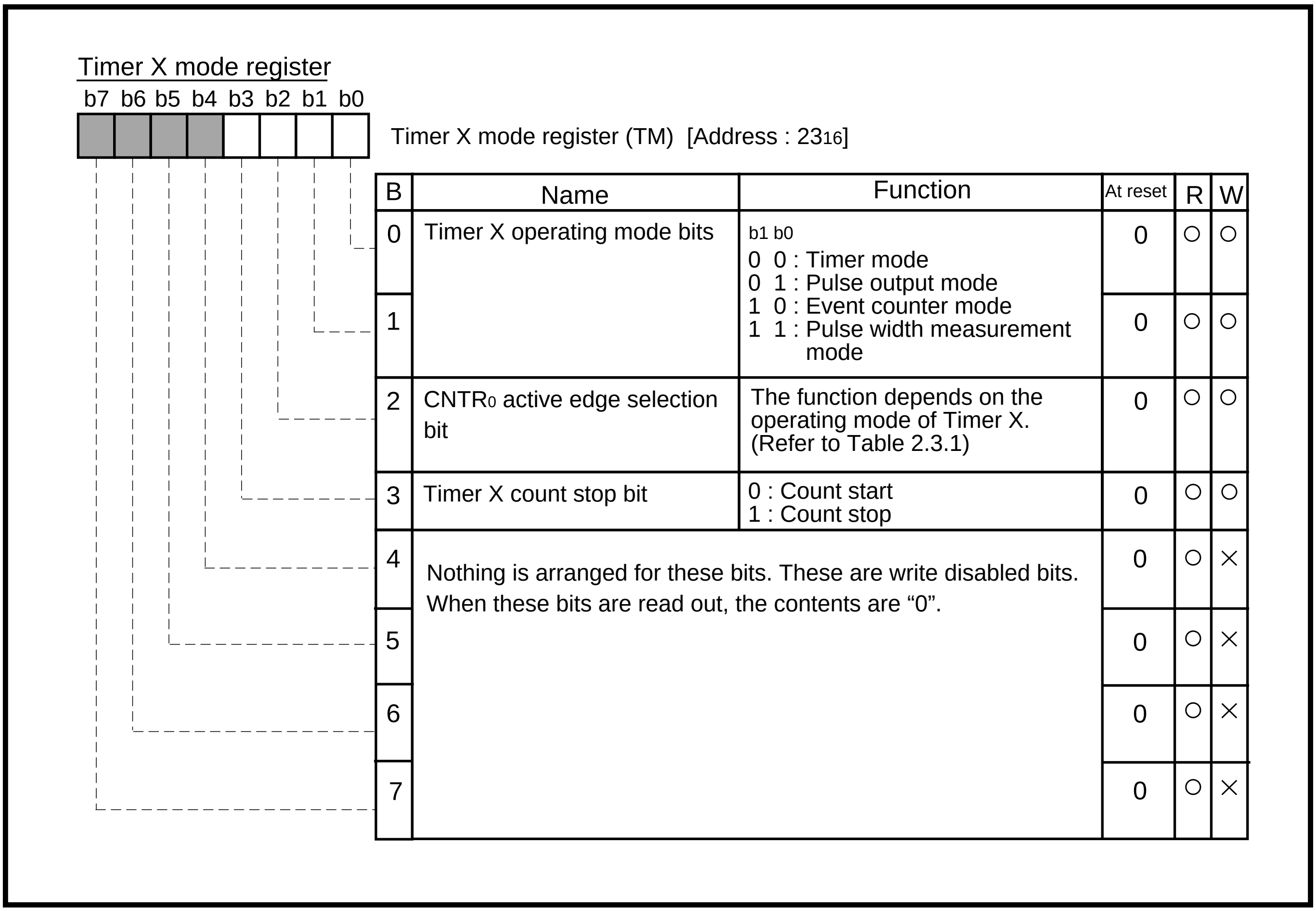
Fig. 2.3.3 Structure of Timer 1



Timer 2 (T2) [Address : 22_{16}]
Timer X (TX) [Address : 25_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of each timer. •The value set in this register is written to both each timer and each timer latch at the same time. •When this register is read out, each timer's count value is read out.		1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 2.3.4 Structure of Timer 2, Timer X



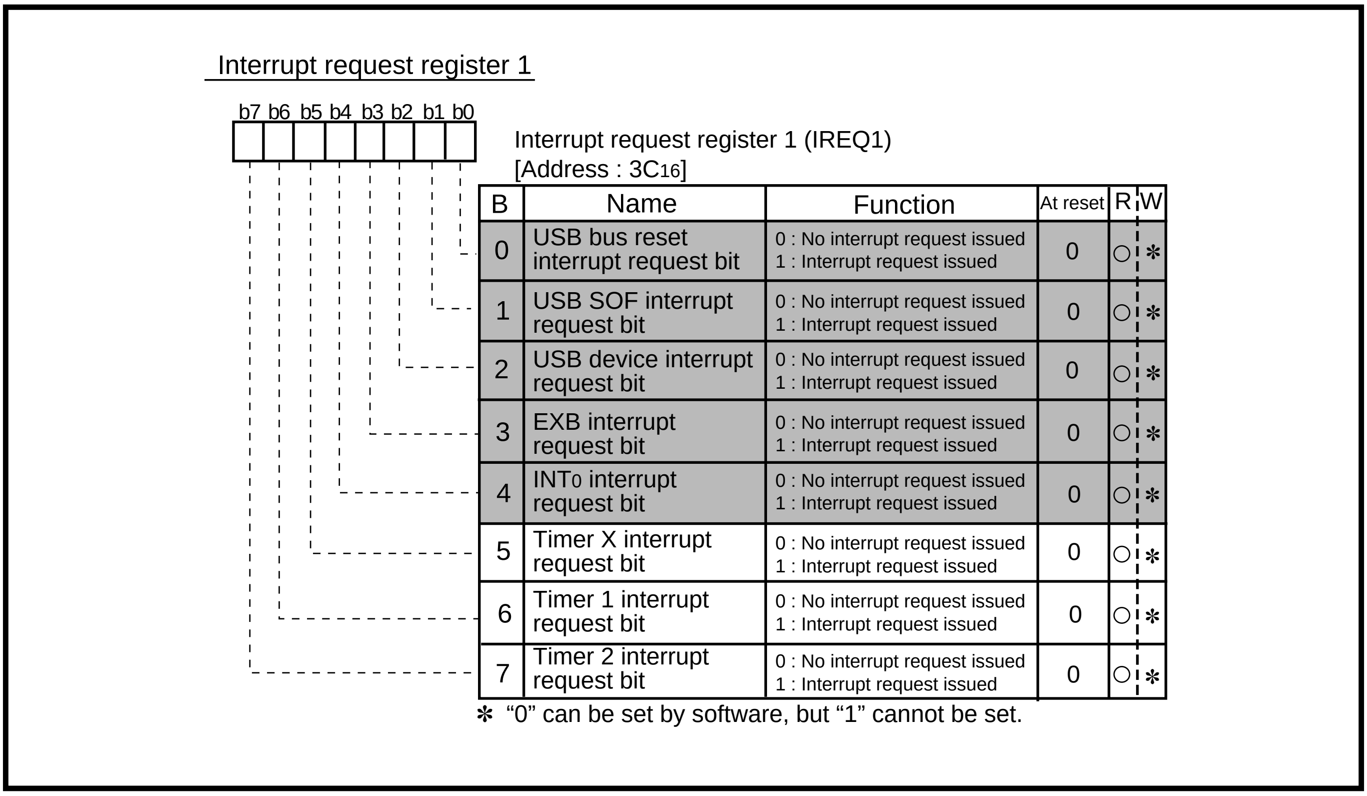
Timer X mode register (TM) [Address : 23_{16}]

B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	$CNTR_0$ active edge selection bit	The function depends on the operating mode of Timer X. (Refer to Table 2.3.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.3.5 Structure of Timer X mode register

Table 2.3.1 $CNTR_0$ active edge selection bit function

Timer X operation modes	$CNTR_0$ active edge selection bit (bits 2 of address 23_{16}) contents	
Timer mode	"0"	$CNTR_0$ interrupt request occurrence: Falling edge ; No influence to timer count
	"1"	$CNTR_0$ interrupt request occurrence: Rising edge ; No influence to timer count
Pulse output mode	"0"	Pulse output start: Beginning at "H" level $CNTR_0$ interrupt request occurrence: Falling edge
	"1"	Pulse output start: Beginning at "L" level $CNTR_0$ interrupt request occurrence: Rising edge
Event counter mode	"0"	Timer X: Rising edge count $CNTR_0$ interrupt request occurrence: Falling edge
	"1"	Timer X: Falling edge count $CNTR_0$ interrupt request occurrence: Rising edge
Pulse width measurement mode	"0"	Timer X: "H" level width measurement $CNTR_0$ interrupt request occurrence: Falling edge
	"1"	Timer X: "L" level width measurement $CNTR_0$ interrupt request occurrence: Rising edge



Interrupt request register 1

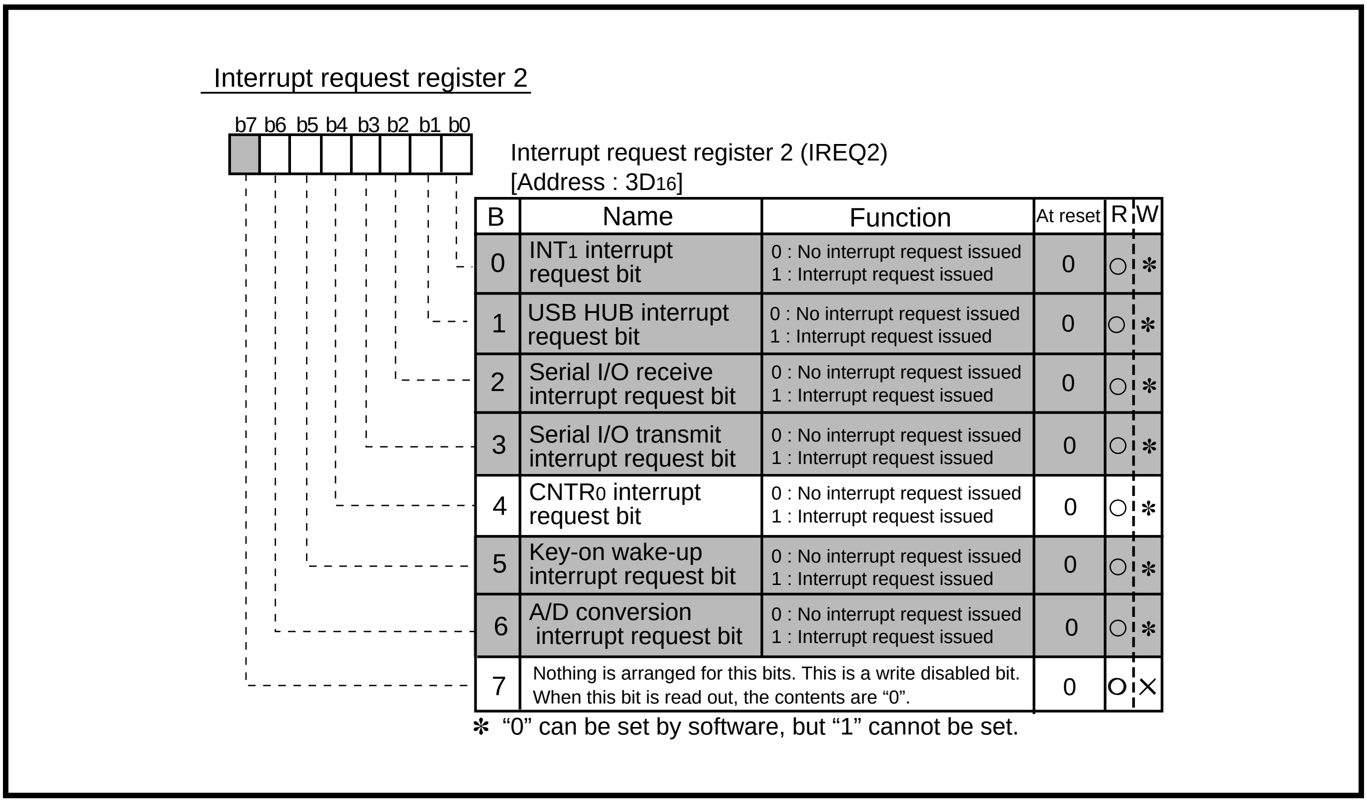
b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 1 (IREQ1)
[Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	USB bus reset interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB SOF interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	USB device interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	EXB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT_0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻ "0" can be set by software, but "1" cannot be set.

Fig. 2.3.6 Structure of Interrupt request register 1



Interrupt request register 2

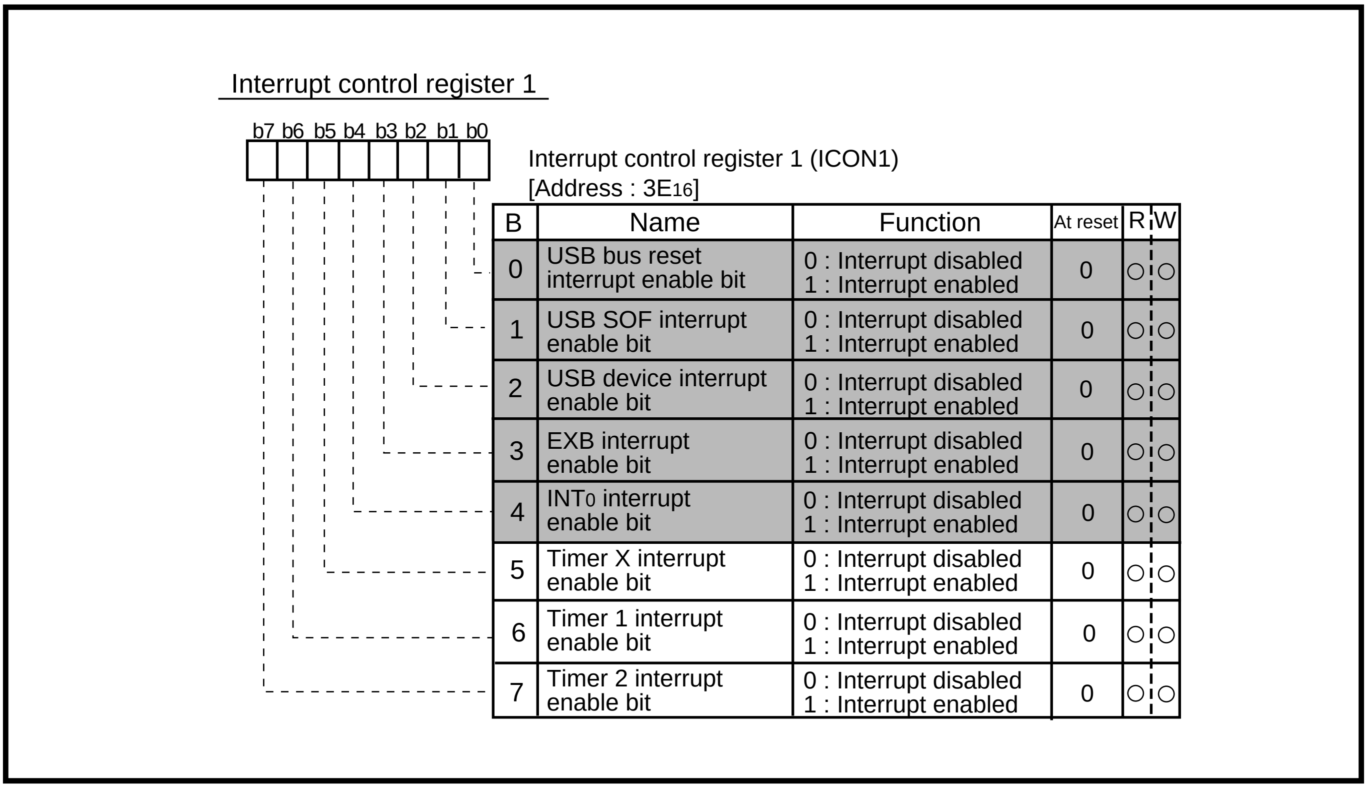
b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2)
[Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	INT_1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB HUB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	$CNTR_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A/D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are "0".		0	O	×

✻ "0" can be set by software, but "1" cannot be set.

Fig. 2.3.7 Structure of Interrupt request register 2

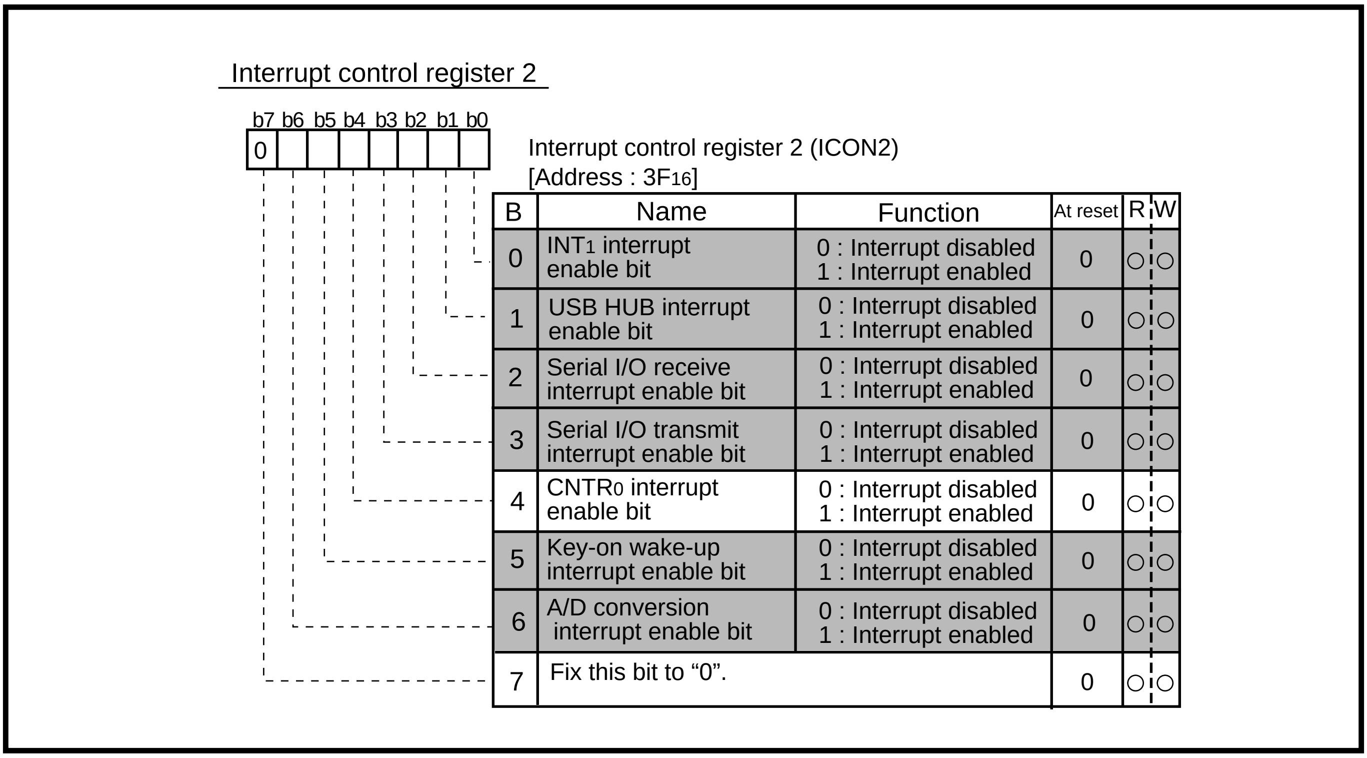


Interrupt control register 1

Interrupt control register 1 (ICON1)
[Address : 3E16]

B	Name	Function	At reset	R	W
0	USB bus reset interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB SOF interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	USB device interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	EXB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 2.3.8 Structure of Interrupt control register 1



Interrupt control register 2

Interrupt control register 2 (ICON2)
[Address : 3F16]

B	Name	Function	At reset	R	W
0	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB HUB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	CNTR0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A/D conversion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.3.9 Structure of Interrupt control register 2

2.3.3 Timer application examples

(1) Basic functions and uses

[Function 1] Control of Event interval (Timer X, Timer 1, Timer 2)

When a certain time, by setting a count value to each timer, has passed, the timer interrupt request occurs.

<Use>

- •Generation of an output signal timing
- •Generation of a wait time

[Function 2] Control of Cyclic operation (Timer X, Timer 1, Timer 2)

The value of the timer latch is automatically written to the corresponding timer each time the timer underflows, and each timer interrupt request occurs in cycles.

<Use>

- •Generation of cyclic interrupts
- •Clock function (measurement of 10 ms); see Application example 1
- •Control of a main routine cycle

[Function 3] Output of Rectangular waveform (Timer X)

The output level of the $CNTR_0$ pin is inverted each time the timer underflows (in the pulse output mode).

<Use>

- •Piezoelectric buzzer output; see Application example 2
- •Generation of the remote control carrier waveforms

[Function 4] Count of External pulses (Timer X)

External pulses input to the $CNTR_0$ pin are counted as the timer count source (in the event counter mode).

<Use>

- •Frequency measurement; see Application example 3
- •Division of external pulses
- •Generation of interrupts due to a cycle using external pulses as the count source; count of a reel pulse

[Function 5] Measurement of External pulse width (Timer X)

The “H” or “L” level width of external pulses input to $CNTR_0$ pin is measured (in the pulse width measurement mode).

<Use>

- •Measurement of external pulse frequency (measurement of pulse width of FG pulse* for a motor); see Application example 4
- •Measurement of external pulse duty (when the frequency is fixed)

FG pulse*: Pulse used for detecting the motor speed to control the motor speed.

(2) Timer application example 1: Clock function (measurement of 10 ms)

Outline: The input clock is divided by the timer so that the clock can count up at 10 ms intervals.

Specifications: •The clock $f(X_{IN})$ = 6 MHz is divided by the timer.

•The clock is counted up in the process routine of the timer X interrupt which occurs at 10 ms intervals.

Figure 2.3.10 shows the timers connection and setting of division ratios; Figure 2.3.11 shows the related registers setting; Figure 2.3.12 shows the control procedure.

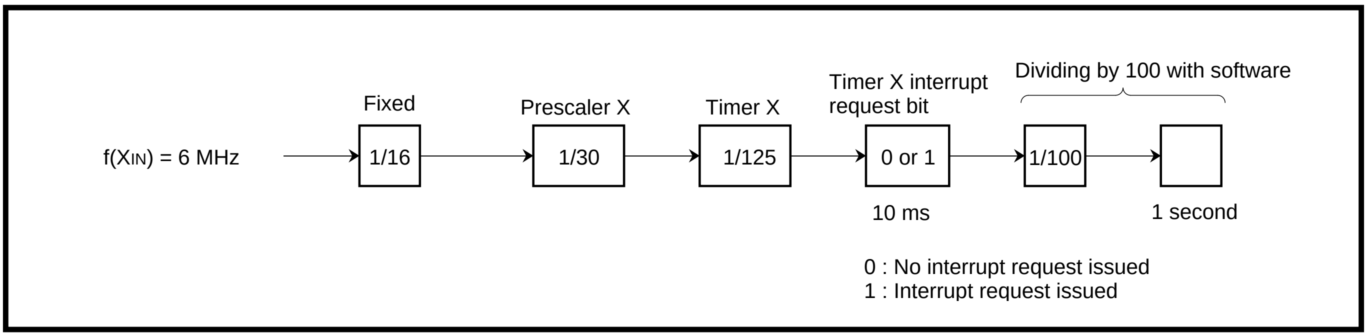

f(XIN) = 6 MHz
Fixed
1/16
Prescaler X
1/30
Timer X
1/125
Timer X interrupt request bit
0 or 1
10 ms
Dividing by 100 with software
1/100
1 second
0 : No interrupt request issued
1 : Interrupt request issued

Fig. 2.3.10 Timers connection and setting of division ratios

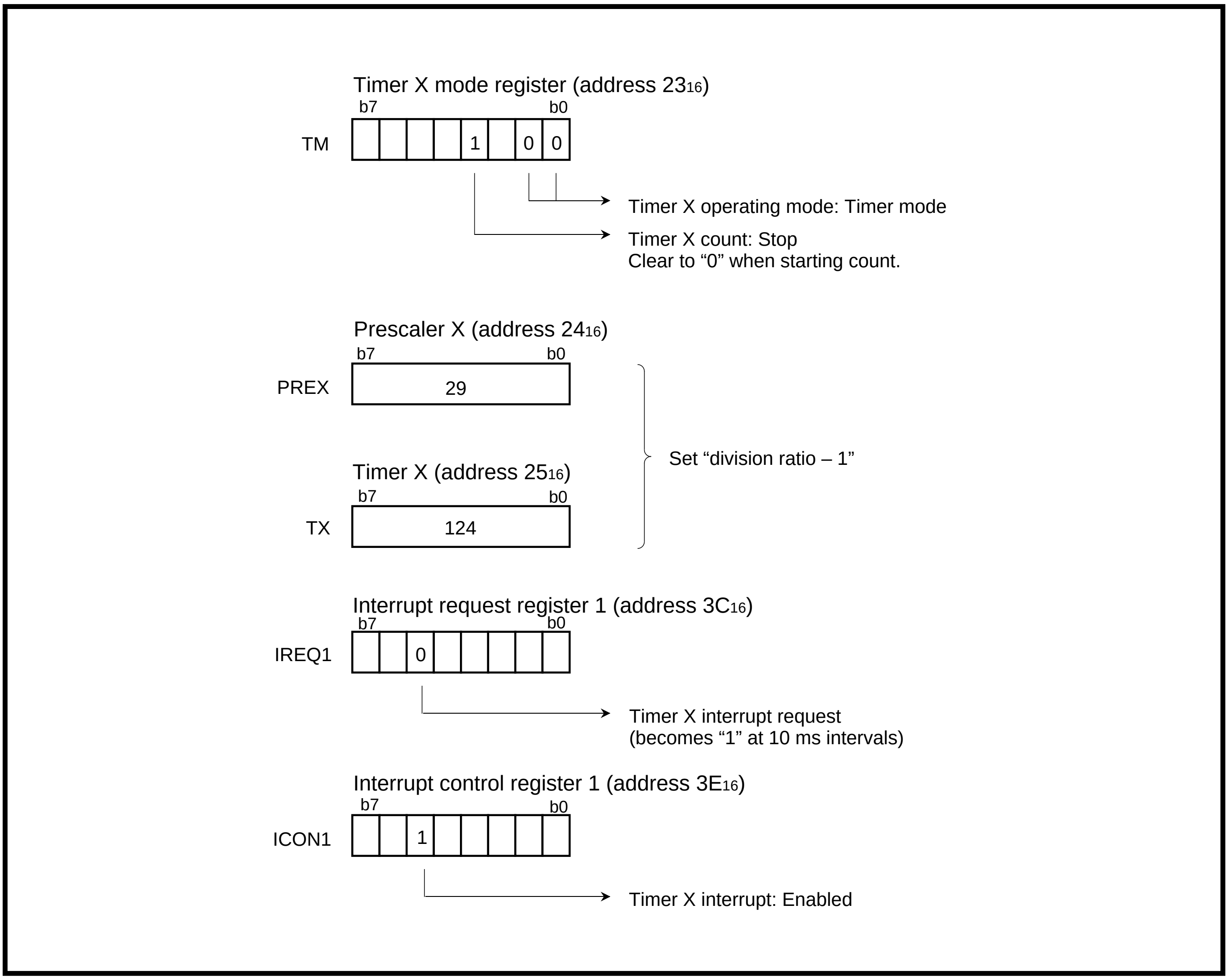

Timer X mode register (address 2316)
b7 b0
TM
1 0 0
Timer X operating mode: Timer mode
Timer X count: Stop
Clear to “0” when starting count.
Prescaler X (address 2416)
b7 b0
PREX
29
Timer X (address 2516)
b7 b0
TX
124
Set “division ratio – 1”
Interrupt request register 1 (address 3C16)
b7 b0
IREQ1
0
Timer X interrupt request
(becomes “1” at 10 ms intervals)
Interrupt control register 1 (address 3E16)
b7 b0
ICON1
1
Timer X interrupt: Enabled

Fig. 2.3.11 Related registers setting

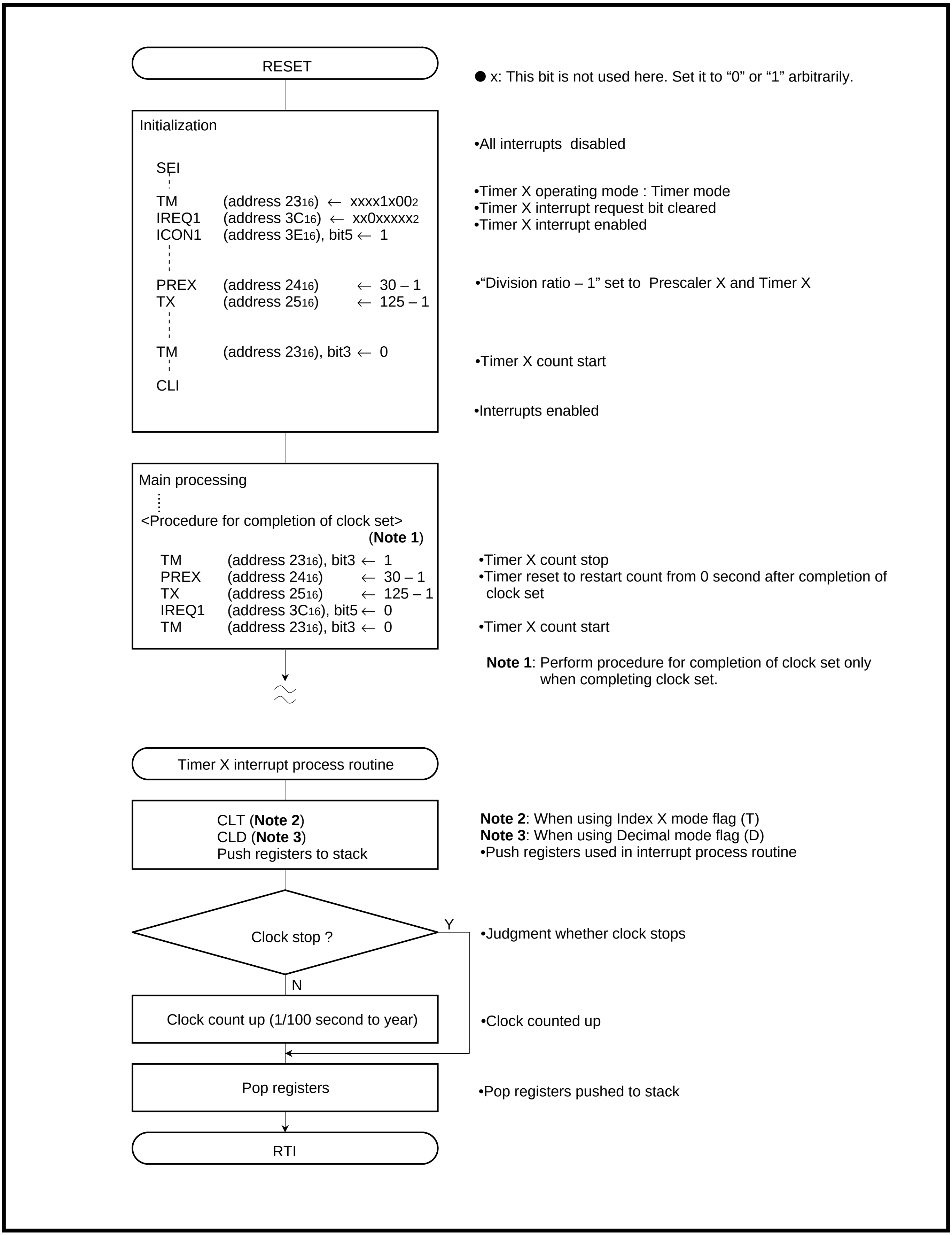

RESET
● x: This bit is not used here. Set it to “0” or “1” arbitrarily.
Initialization
SEI
•All interrupts disabled
TM (address 23₁₆) ← xxxx1x00₂
IREQ1 (address 3C₁₆) ← xx0xxxxx₂
ICON1 (address 3E₁₆), bit5 ← 1
•Timer X operating mode : Timer mode
•Timer X interrupt request bit cleared
•Timer X interrupt enabled
PREX (address 24₁₆) ← 30 – 1
TX (address 25₁₆) ← 125 – 1
•“Division ratio – 1” set to Prescaler X and Timer X
TM (address 23₁₆), bit3 ← 0
•Timer X count start
CLI
•Interrupts enabled
Main processing
<Procedure for completion of clock set>
(Note 1)
TM (address 23₁₆), bit3 ← 1
PREX (address 24₁₆) ← 30 – 1
TX (address 25₁₆) ← 125 – 1
IREQ1 (address 3C₁₆), bit5 ← 0
TM (address 23₁₆), bit3 ← 0
•Timer X count stop
•Timer reset to restart count from 0 second after completion of clock set
•Timer X count start
Note 1: Perform procedure for completion of clock set only when completing clock set.
Timer X interrupt process routine
CLT (Note 2)
CLD (Note 3)
Push registers to stack
Note 2: When using Index X mode flag (T)
Note 3: When using Decimal mode flag (D)
•Push registers used in interrupt process routine
Clock stop ?
Y
N
•Judgment whether clock stops
Clock count up (1/100 second to year)
•Clock counted up
Pop registers
•Pop registers pushed to stack
RTI

Fig. 2.3.12 Control procedure

(3) Timer application example 2: Piezoelectric buzzer output

Outline: The rectangular waveform output function of the timer is applied for a piezoelectric buzzer output.

Specifications: •The rectangular waveform, dividing the clock f(X_{IN}) = 6 MHz into about 2 kHz (2038 Hz), is output from the P5$_1$/CNTR$_0$ pin.

•The level of the P5$_1$/CNTR$_0$ pin is fixed to "H" while a piezoelectric buzzer output stops.

Figure 2.3.13 shows a peripheral circuit example, and Figure 2.3.14 shows the timers connection and setting of division ratios. Figures 2.3.15 shows the related registers setting, and Figure 2.3.16 shows the control procedure.

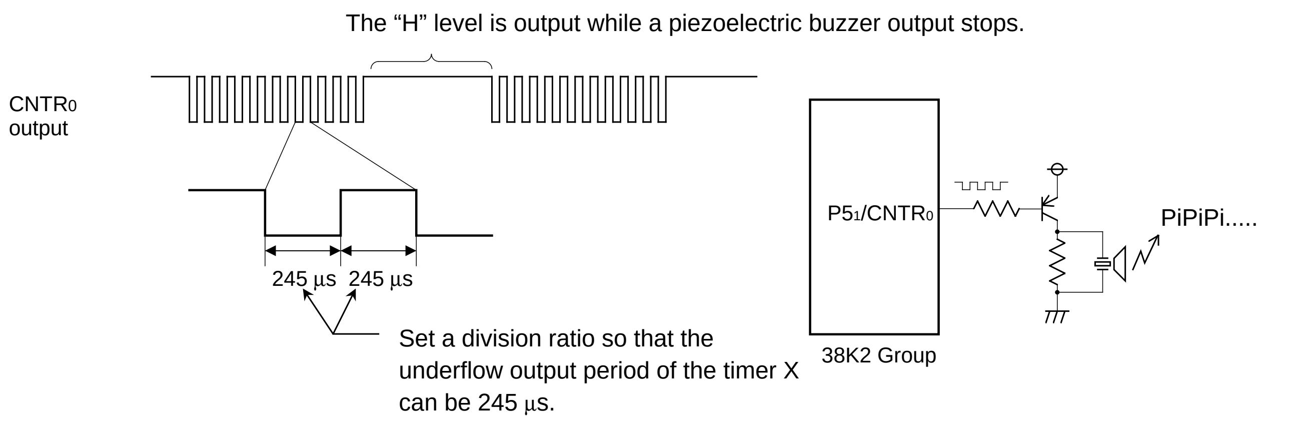

The "H" level is output while a piezoelectric buzzer output stops.
CNTR0 output
245 µs
245 µs
Set a division ratio so that the underflow output period of the timer X can be 245 µs.
P51/CNTR0
38K2 Group
PiPiPi.....

Fig. 2.3.13 Peripheral circuit example

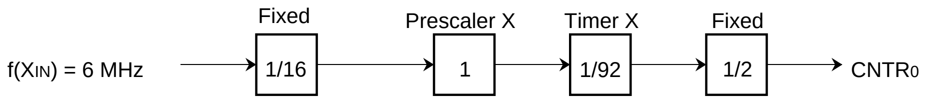


Fig. 2.3.14 Timers connection and setting of division ratios

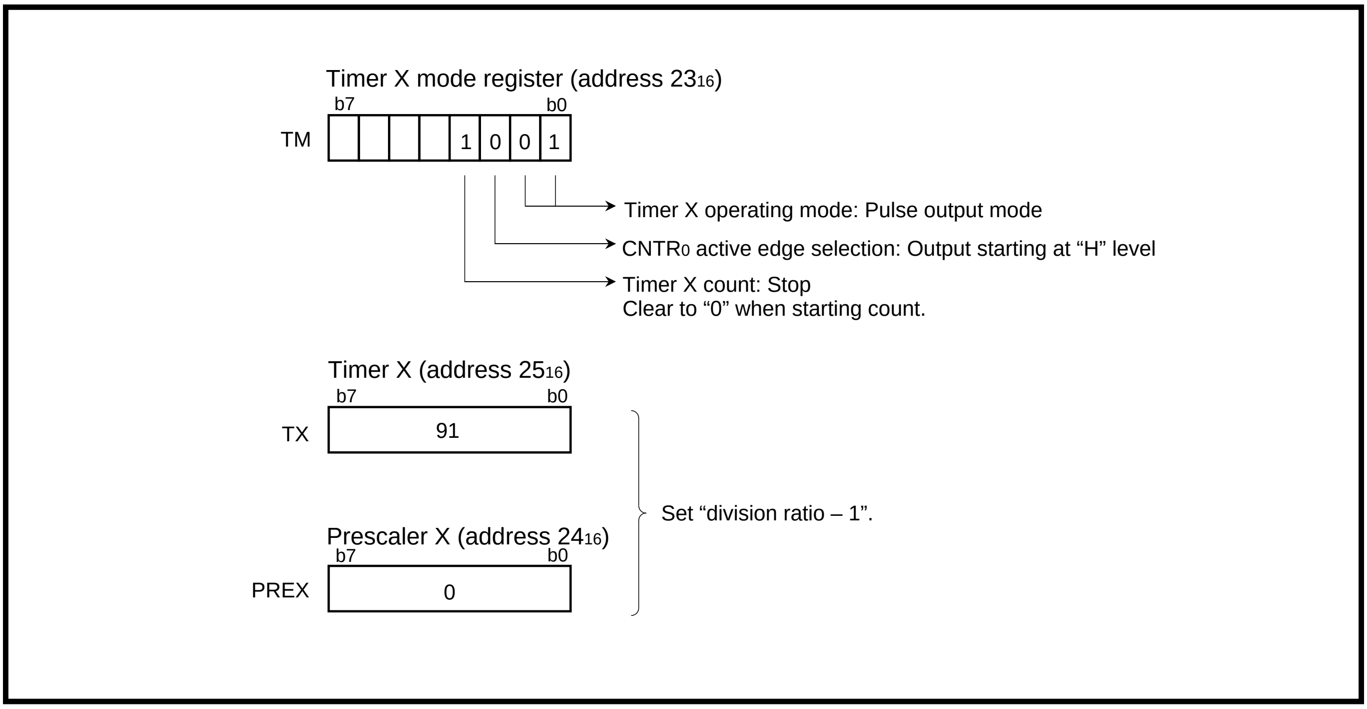

Timer X mode register (address 23$_{16}$)
b7
b0
TM
1
0
0
1
Timer X operating mode: Pulse output mode
CNTR$_0$ active edge selection: Output starting at "H" level
Timer X count: Stop
Clear to "0" when starting count.
Timer X (address 25$_{16}$)
b7
b0
TX
91
Prescaler X (address 24$_{16}$)
b7
b0
PREX
0
Set "division ratio – 1".

Fig. 2.3.15 Related registers setting

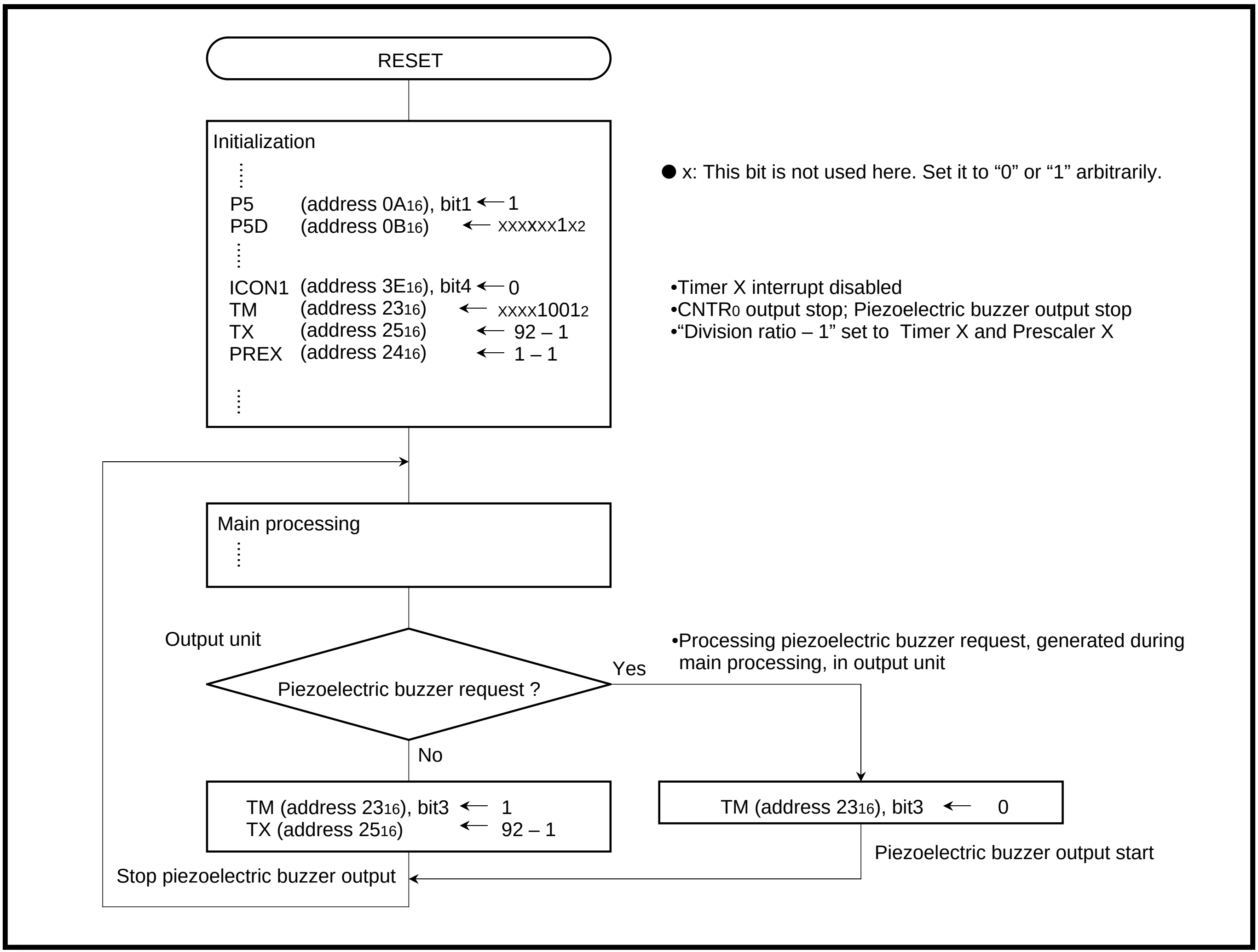

RESET
Initialization
P5 (address 0A$_{16}$), bit1 ← 1
P5D (address 0B$_{16}$) ← xxxxxx1x$_2$
ICON1 (address 3E$_{16}$), bit4 ← 0
TM (address 23$_{16}$) ← xxxx1001$_2$
TX (address 25$_{16}$) ← 92 – 1
PREX (address 24$_{16}$) ← 1 – 1
● x: This bit is not used here. Set it to "0" or "1" arbitrarily.
•Timer X interrupt disabled
•CNTR$_0$ output stop; Piezoelectric buzzer output stop
•"Division ratio – 1" set to Timer X and Prescaler X
Main processing
Output unit
Piezoelectric buzzer request ?
Yes
No
•Processing piezoelectric buzzer request, generated during main processing, in output unit
TM (address 23$_{16}$), bit3 ← 1
TX (address 25$_{16}$) ← 92 – 1
TM (address 23$_{16}$), bit3 ← 0
Piezoelectric buzzer output start
Stop piezoelectric buzzer output

Fig. 2.3.16 Control procedure

(4) Timer application example 3: Frequency measurement

Outline: The following two values are compared to judge whether the frequency is within a valid range.

•A value by counting pulses input to P5₁/CNTR₀ pin with the timer.

•A reference value

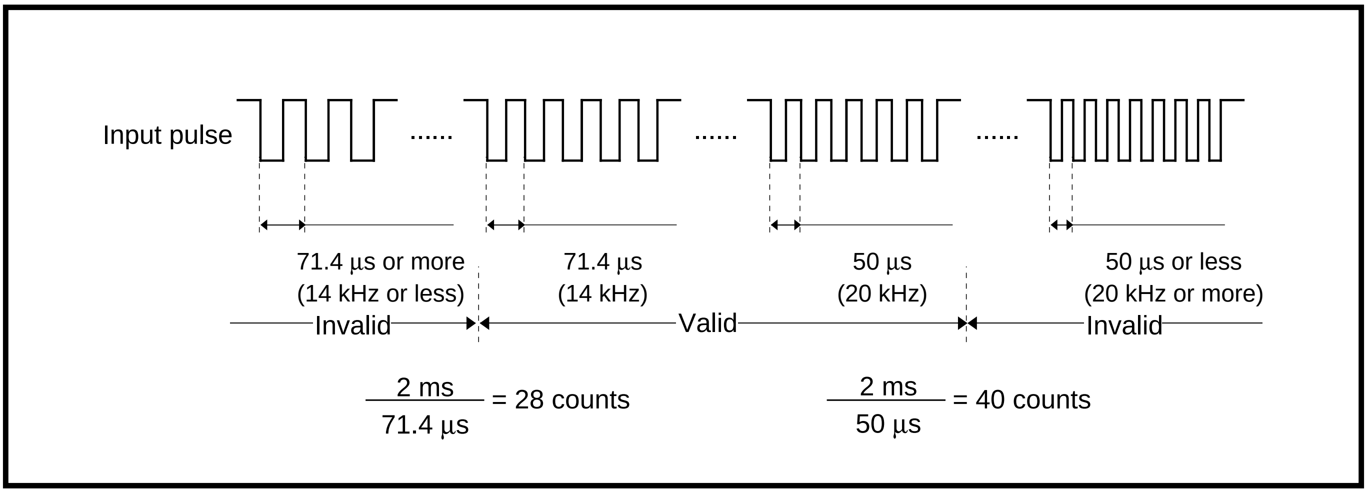
Specifications: •The pulse is input to the P5₁/CNTR₀ pin and counted by the timer X.

•A count value is read out at about 2 ms intervals, the timer 1 interrupt interval. When the count value is 28 to 40, it is judged that the input pulse is valid.

•Because the timer is a down-counter, the count value is compared with 227 to 215 (Note).

Note: 227 to 215 = {255 (initial value of counter) – 28} to {255 – 40}; 28 to 40 means the number of valid value.

Figure 2.3.17 shows the judgment method of valid/invalid of input pulses; Figure 2.3.18 shows the related registers setting; Figure 2.3.19 shows the control procedure.



$$\frac{2\text{ ms}}{71.4\ \mu\text{s}} = 28\text{ counts} \qquad \frac{2\text{ ms}}{50\ \mu\text{s}} = 40\text{ counts}$$

Fig. 2.3.17 Judgment method of valid/invalid of input pulses

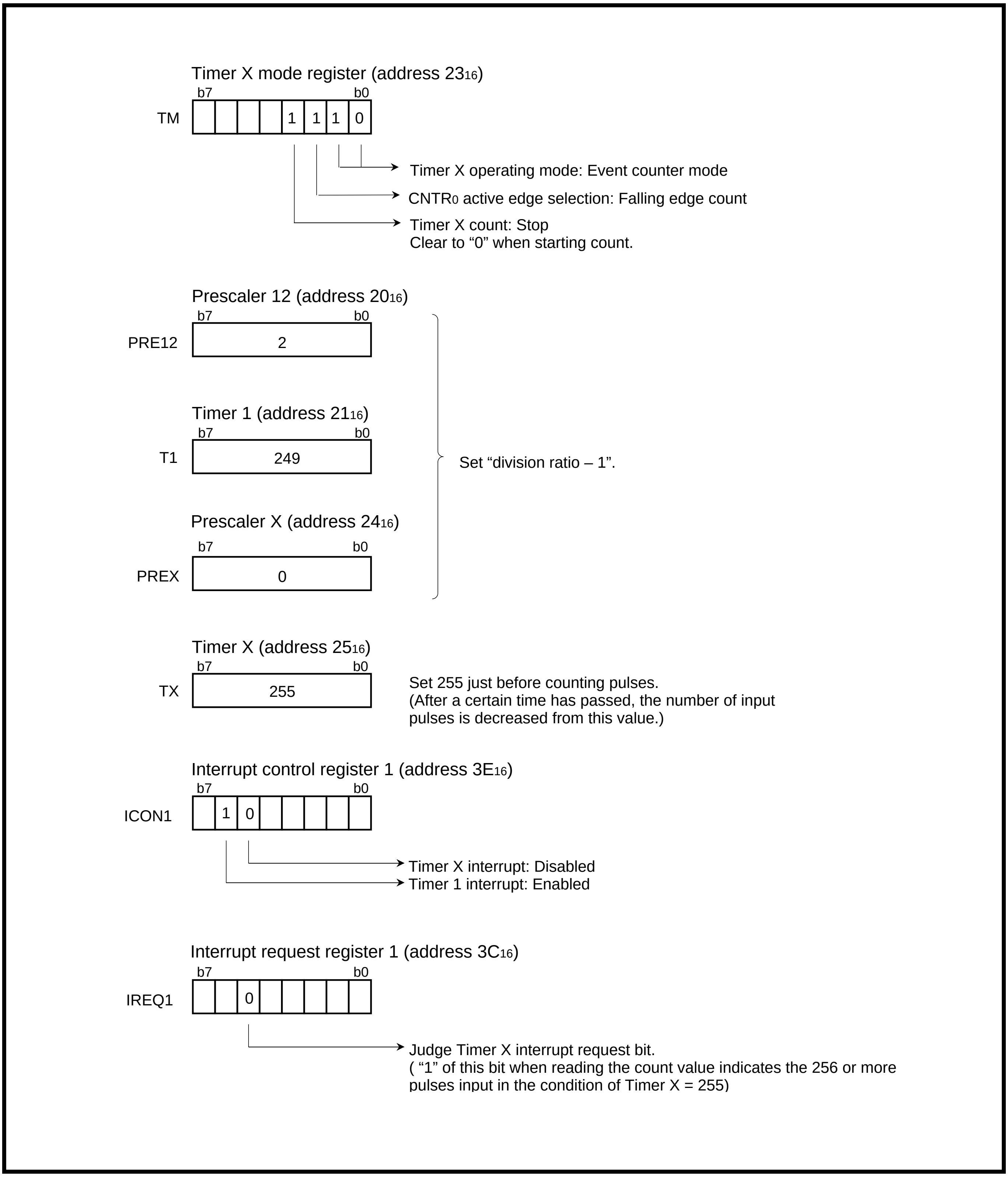

Timer X mode register (address 23₁₆)
b7
b0
TM
1
1
1
0
Timer X operating mode: Event counter mode
CNTR₀ active edge selection: Falling edge count
Timer X count: Stop
Clear to “0” when starting count.
Prescaler 12 (address 20₁₆)
b7
b0
PRE12
2
Timer 1 (address 21₁₆)
b7
b0
T1
249
Set “division ratio – 1”.
Prescaler X (address 24₁₆)
b7
b0
PREX
0
Timer X (address 25₁₆)
b7
b0
TX
255
Set 255 just before counting pulses.
(After a certain time has passed, the number of input
pulses is decreased from this value.)
Interrupt control register 1 (address 3E₁₆)
b7
b0
ICON1
1
0
Timer X interrupt: Disabled
Timer 1 interrupt: Enabled
Interrupt request register 1 (address 3C₁₆)
b7
b0
IREQ1
0
Judge Timer X interrupt request bit.
(“1” of this bit when reading the count value indicates the 256 or more
pulses input in the condition of Timer X = 255)

Fig. 2.3.18 Related registers setting

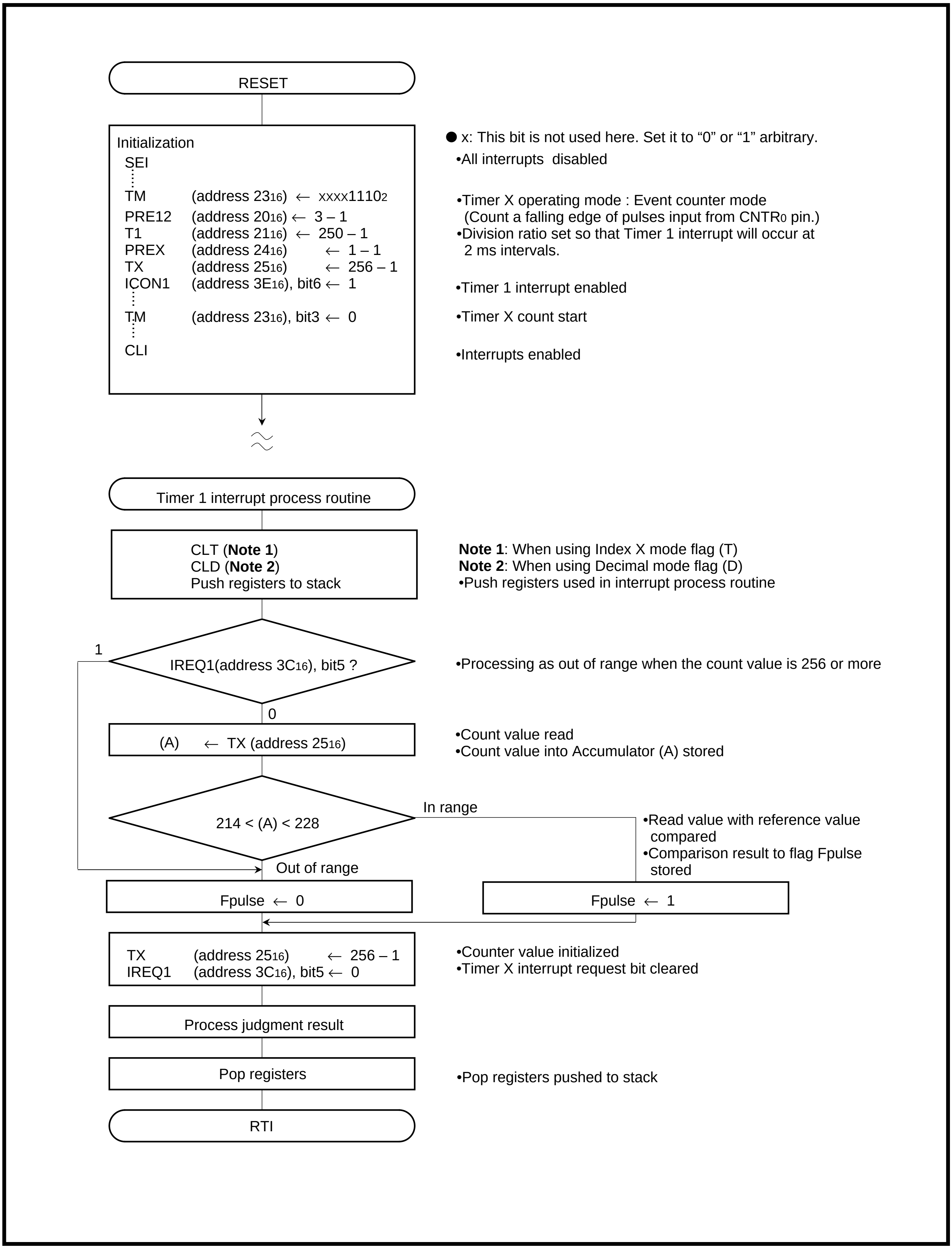

RESET
Initialization
SEI
TM (address 23_{16}) ← $xxxx1110_2$
PRE12 (address 20_{16}) ← 3 – 1
T1 (address 21_{16}) ← 250 – 1
PREX (address 24_{16}) ← 1 – 1
TX (address 25_{16}) ← 256 – 1
ICON1 (address $3E_{16}$), bit6 ← 1
TM (address 23_{16}), bit3 ← 0
CLI
● x: This bit is not used here. Set it to “0” or “1” arbitrary.
•All interrupts disabled
•Timer X operating mode : Event counter mode
(Count a falling edge of pulses input from $CNTR_0$ pin.)
•Division ratio set so that Timer 1 interrupt will occur at 2 ms intervals.
•Timer 1 interrupt enabled
•Timer X count start
•Interrupts enabled
Timer 1 interrupt process routine
CLT (**Note 1**)
CLD (**Note 2**)
Push registers to stack
Note 1: When using Index X mode flag (T)
Note 2: When using Decimal mode flag (D)
•Push registers used in interrupt process routine
IREQ1(address $3C_{16}$), bit5 ?
1
0
•Processing as out of range when the count value is 256 or more
(A) ← TX (address 25_{16})
•Count value read
•Count value into Accumulator (A) stored
214 < (A) < 228
In range
Out of range
•Read value with reference value compared
•Comparison result to flag Fpulse stored
Fpulse ← 0
Fpulse ← 1
TX (address 25_{16}) ← 256 – 1
IREQ1 (address $3C_{16}$), bit5 ← 0
•Counter value initialized
•Timer X interrupt request bit cleared
Process judgment result
Pop registers
•Pop registers pushed to stack
RTI

Fig. 2.3.19 Control procedure

(5) Timer application example 4: Measurement of FG pulse width for motor

Outline: The timer X counts the "H" level width of the pulses input to the $P5_1/CNTR_0$ pin. An underflow is detected by the timer X interrupt and an end of the input pulse "H" level is detected by the $CNTR_0$ interrupt.

Specifications: •The timer X counts the "H" level width of the FG pulse input to the $P5_1/CNTR_0$ pin.

<Example>

When the clock frequency is 6 MHz, the count source is 2.67 µs, which is obtained by dividing the clock frequency by 16. Measurement can be performed to 175 ms in the range of $FFFF_{16}$ to 0000_{16}.

Figure 2.3.20 shows the timers connection and setting of division ratio; Figure 2.3.21 shows the related registers setting; Figure 2.3.22 shows the control procedure.

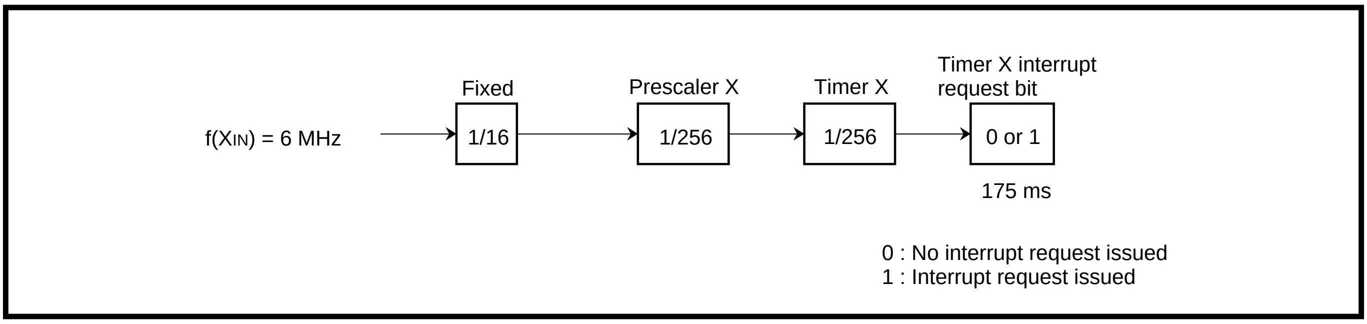

f(XIN) = 6 MHz
Fixed
1/16
Prescaler X
1/256
Timer X
1/256
Timer X interrupt request bit
0 or 1
175 ms
0 : No interrupt request issued
1 : Interrupt request issued

Fig. 2.3.20 Timers connection and setting of division ratios

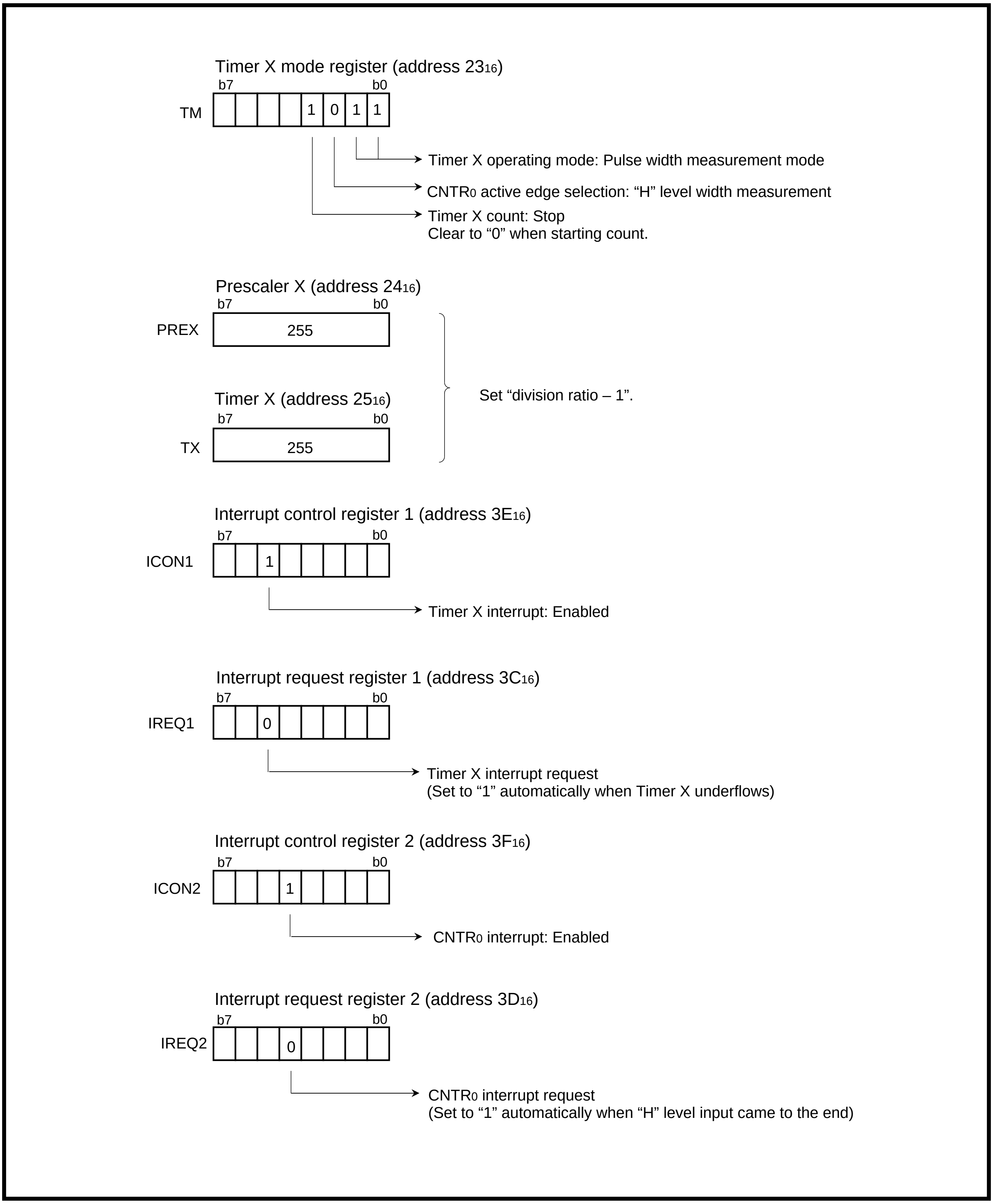
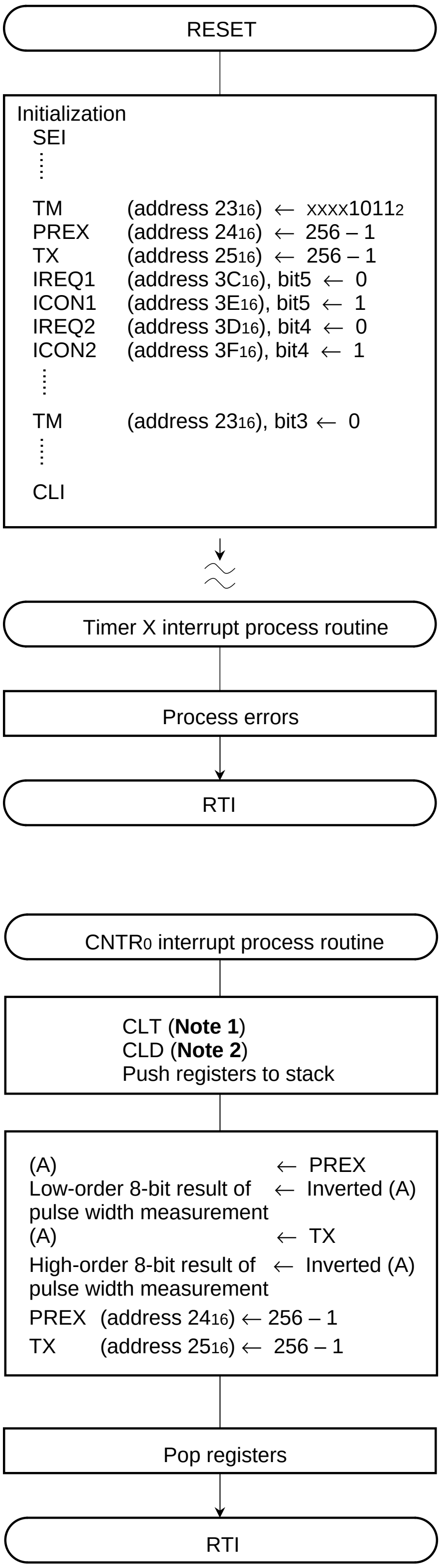

Timer X mode register (address 2316)
b7
b0
TM
1
0
1
1
Timer X operating mode: Pulse width measurement mode
CNTR0 active edge selection: "H" level width measurement
Timer X count: Stop
Clear to "0" when starting count.
Prescaler X (address 2416)
b7
b0
PREX
255
Timer X (address 2516)
b7
b0
TX
255
Set "division ratio – 1".
Interrupt control register 1 (address 3E16)
b7
b0
ICON1
1
Timer X interrupt: Enabled
Interrupt request register 1 (address 3C16)
b7
b0
IREQ1
0
Timer X interrupt request
(Set to "1" automatically when Timer X underflows)
Interrupt control register 2 (address 3F16)
b7
b0
ICON2
1
CNTR0 interrupt: Enabled
Interrupt request register 2 (address 3D16)
b7
b0
IREQ2
0
CNTR0 interrupt request
(Set to "1" automatically when "H" level input came to the end)

Fig. 2.3.21 Related registers setting


RESET
Initialization
SEI
TM (address 23_{16}) ← xxxx1011_2
PREX (address 24_{16}) ← 256 – 1
TX (address 25_{16}) ← 256 – 1
IREQ1 (address $3C_{16}$), bit5 ← 0
ICON1 (address $3E_{16}$), bit5 ← 1
IREQ2 (address $3D_{16}$), bit4 ← 0
ICON2 (address $3F_{16}$), bit4 ← 1
TM (address 23_{16}), bit3 ← 0
CLI
Timer X interrupt process routine
Process errors
RTI
CNTR$_0$ interrupt process routine
CLT (**Note 1**)
CLD (**Note 2**)
Push registers to stack
(A) ← PREX
Low-order 8-bit result of pulse width measurement ← Inverted (A)
(A) ← TX
High-order 8-bit result of pulse width measurement ← Inverted (A)
PREX (address 24_{16}) ← 256 – 1
TX (address 25_{16}) ← 256 – 1
Pop registers
RTI

● x: This bit is not used here. Set it to “0” or “1” arbitrarily.

•All interrupts disabled

•Timer X operating mode : Pulse width measurement mode
(Measure “H” level of pulses input from CNTR$_0$ pin.)
•Set division ratio so that Timer X interrupt will occur at 175 ms intervals.
•Timer X interrupt request bit cleared
•Timer X interrupt enabled
•CNTR$_0$ interrupt request bit cleared
•CNTR$_0$ interrupt enabled

•Timer X count start

•Interrupts enabled

•Error occurs

Note 1: When using Index X mode flag (T)
Note 2: When using Decimal mode flag (D)
•Push registers used in interrupt process routine

•Read the count value and store it to RAM

•Division ratio set so that Timer X interrupt will occur at 175 ms intervals.

•Pop registers pushed to stack

Fig. 2.3.22 Control procedure

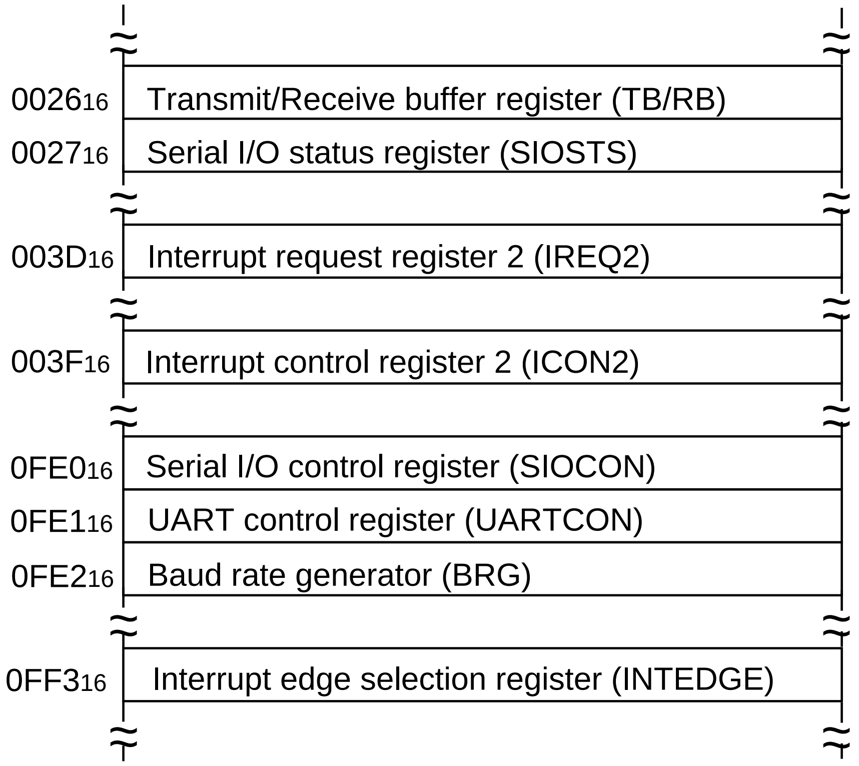
2.3.4 Notes on timer

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer X count source selection bit, the value of timer count is altered in unconsiderable amount owing to generating of a thin pulses in the count input signals. Therefore, select the timer count source before set the value to the prescaler and the timer.

2.4 Serial I/O

This paragraph explains the registers setting method and the notes related to the Serial I/O.

2.4.1 Memory map



0026_{16}	Transmit/Receive buffer register (TB/RB)
0027_{16}	Serial I/O status register (SIOSTS)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003F_{16}$	Interrupt control register 2 (ICON2)
$0FE0_{16}$	Serial I/O control register (SIOCON)
$0FE1_{16}$	UART control register (UARTCON)
$0FE2_{16}$	Baud rate generator (BRG)
$0FF3_{16}$	Interrupt edge selection register (INTEDGE)

Fig. 2.4.1 Memory map of registers related to Serial I/O

2.4.2 Related registers

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address : 26_{16}]

B	Name / Function	At reset	R	W
0	The transmission data is written to or the receive data is read out from this buffer register. • At writing: A data is written to the transmit buffer register. • At reading: The contents of the receive buffer register are read out.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Note: The contents of transmit buffer register cannot be read out.
The data cannot be written to the receive buffer register.

Fig. 2.4.2 Structure of Transmit/Receive buffer register

Serial I/O status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O status register (SIOSTS) [Address : 27_{16}]

B	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	○	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	○	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	○	×
6	Summing error flag (SE)	0 : (OE) U (PE) U (FE) = 0 1 : (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the contents are “1”.		1	○	×

Fig. 2.4.3 Structure of Serial I/O status register

Serial I/O control register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O control register (SIOCON) [Address : 0FE0$_{16}$]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : System clock 1 : System clock/4	0	○	○
1	Serial I/O synchronous clock selection bit (SCS)	• In clock synchronous serial I/O 0 : BRG output divided by 4 1 : External clock input • In UART 0 : BRG output divided by 16 1 : External clock input divided by 16	0	○	○
2	$\overline{S_{RDY}}$ output enable bit (SRDY)	0 : P4$_3$ pin operates as ordinary I/O pin 1 : P4$_3$ pin operates as $\overline{S_{RDY}}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt when transmit buffer has emptied 1 : Interrupt when transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	○	○
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	○	○
6	Serial I/O mode selection bit (SIOM)	0 : Clock asynchronous(UART) serial I/O 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O enable bit (SIOE)	0 : Serial I/O disabled (pins P4$_0$ to P4$_3$ operate as ordinary I/O pins) 1 : Serial I/O enabled (pins P4$_0$ to P4$_3$ operate as serial I/O pins)	0	○	○

Fig. 2.4.4 Structure of Serial I/O control register

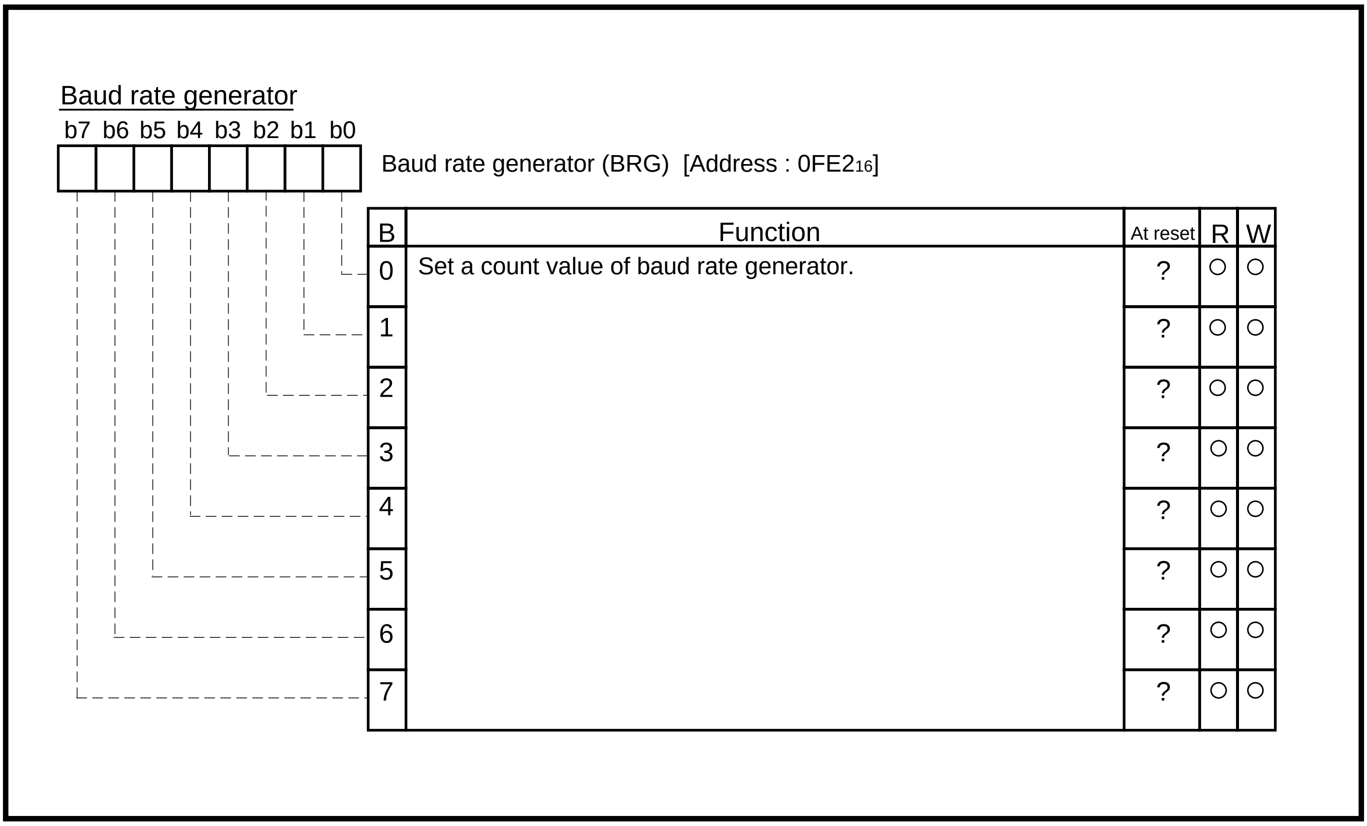
UART control register

b7 b6 b5 b4 b3 b2 b1 b0

UART control register (UARTCON) [Address : 0FE1$_{16}$]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the contents are “1”.		1	○	×
6			1	○	×
7			1	○	×

Fig. 2.4.5 Structure of UART control register

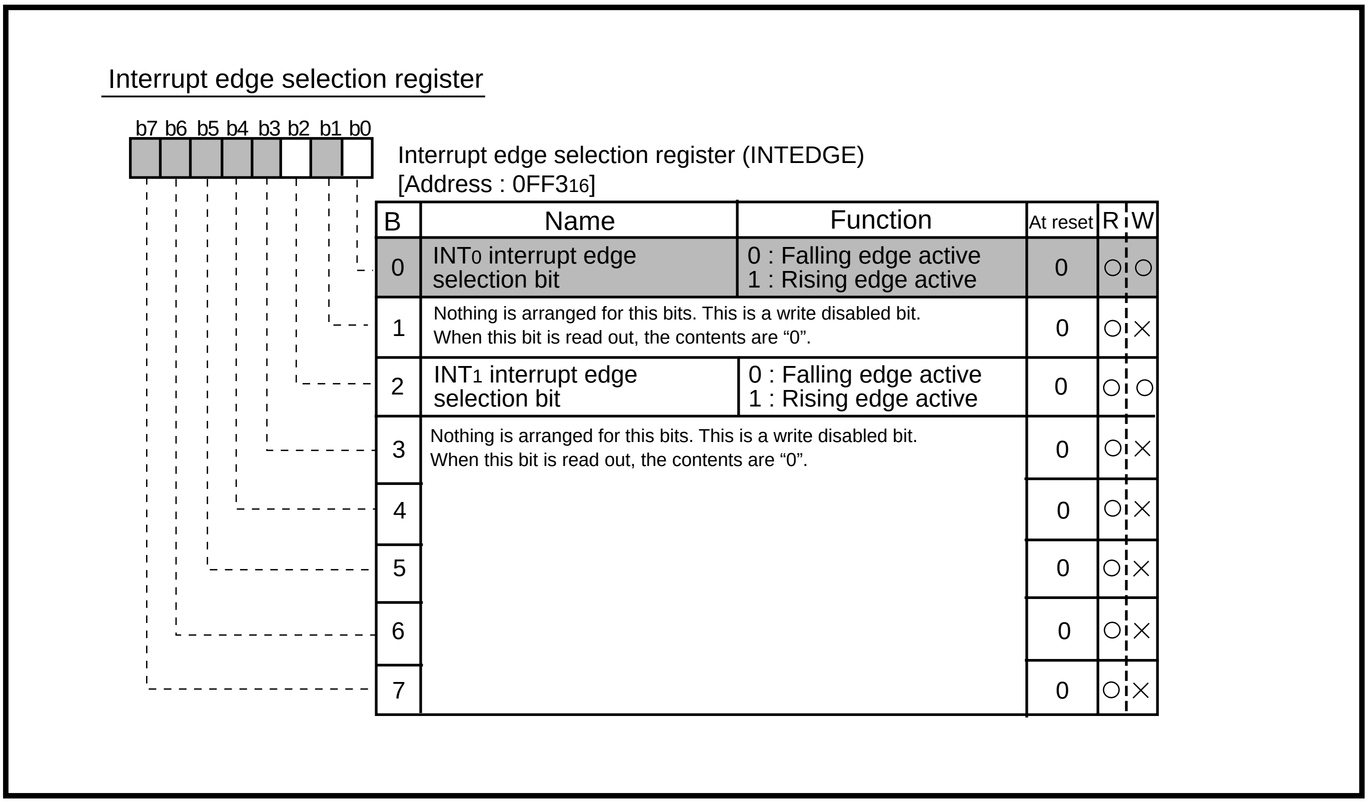


Baud rate generator

Baud rate generator (BRG) [Address : 0FE2$_{16}$]

B	Function	At reset	R	W
0	Set a count value of baud rate generator.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 2.4.6 Structure of Baud rate generator

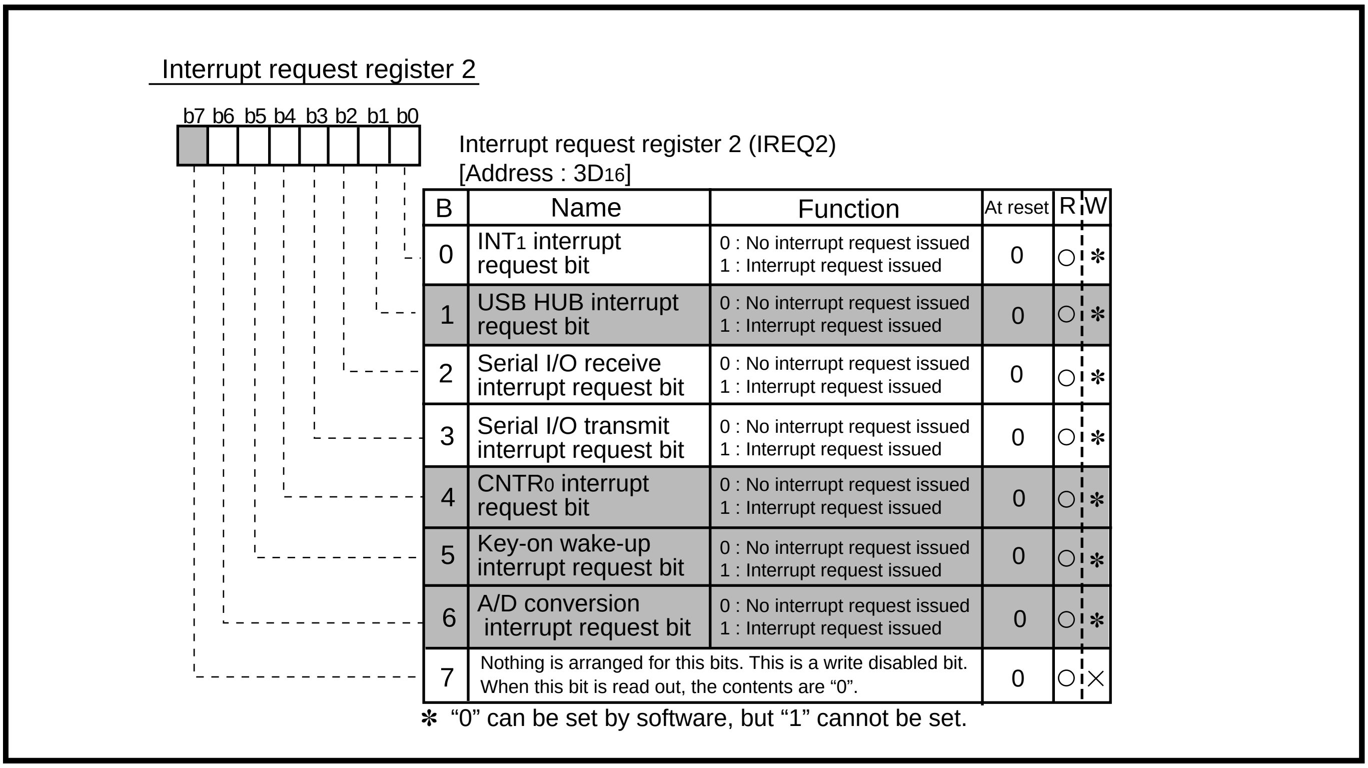


Interrupt edge selection register

Interrupt edge selection register (INTEDGE)
[Address : 0FF3$_{16}$]

B	Name	Function	At reset	R	W
0	INT$_0$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
2	INT$_1$ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 2.4.7 Structure of Interrupt edge selection register



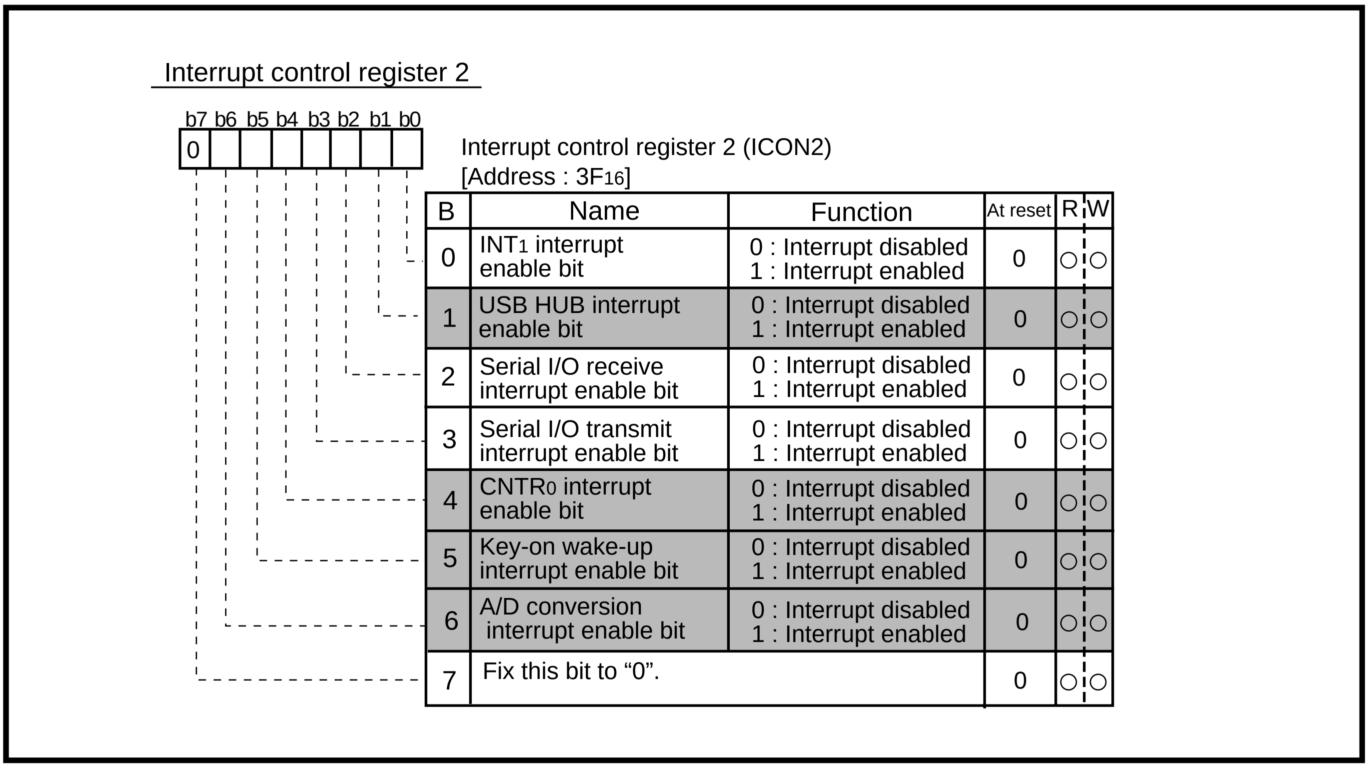
Interrupt request register 2

Interrupt request register 2 (IREQ2)
[Address : 3D16]

B	Name	Function	At reset	R	W
0	INT1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB HUB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	CNTR0 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A/D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

✻ “0” can be set by software, but “1” cannot be set.

Fig. 2.4.8 Structure of Interrupt request register 2



Interrupt control register 2

Interrupt control register 2 (ICON2)
[Address : 3F16]

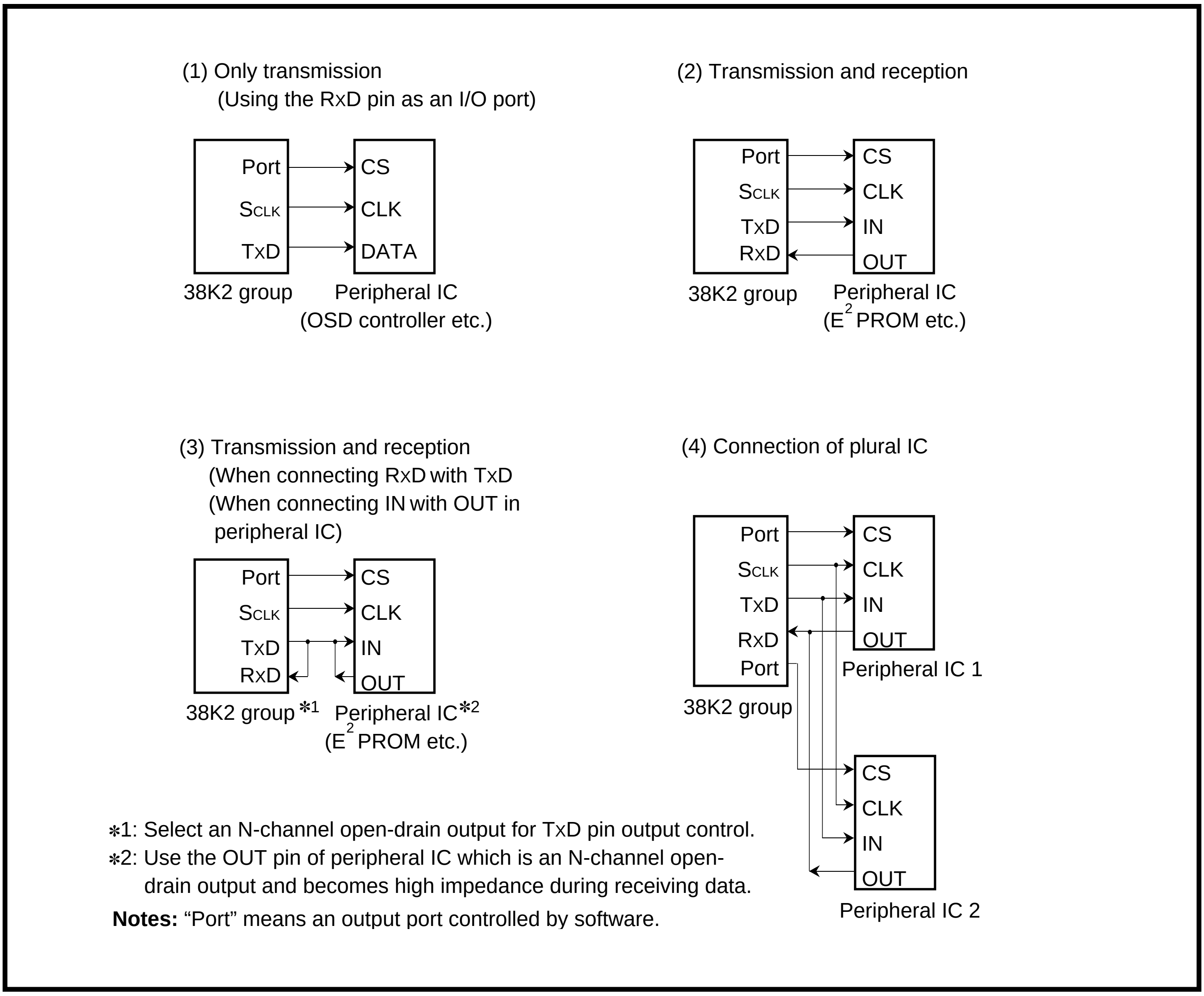
B	Name	Function	At reset	R	W
0	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB HUB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	CNTR0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A/D conversion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.4.9 Structure of Interrupt control register 2

2.4.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

Figure 2.4.10 shows connection examples of a peripheral IC equipped with the CS pin. There are connection examples using a clock synchronous serial I/O mode.



*1: Select an N-channel open-drain output for TxD pin output control.
*2: Use the OUT pin of peripheral IC which is an N-channel open-drain output and becomes high impedance during receiving data.

Notes: “Port” means an output port controlled by software.

Fig. 2.4.10 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.4.11 shows connection examples with another microcomputer.

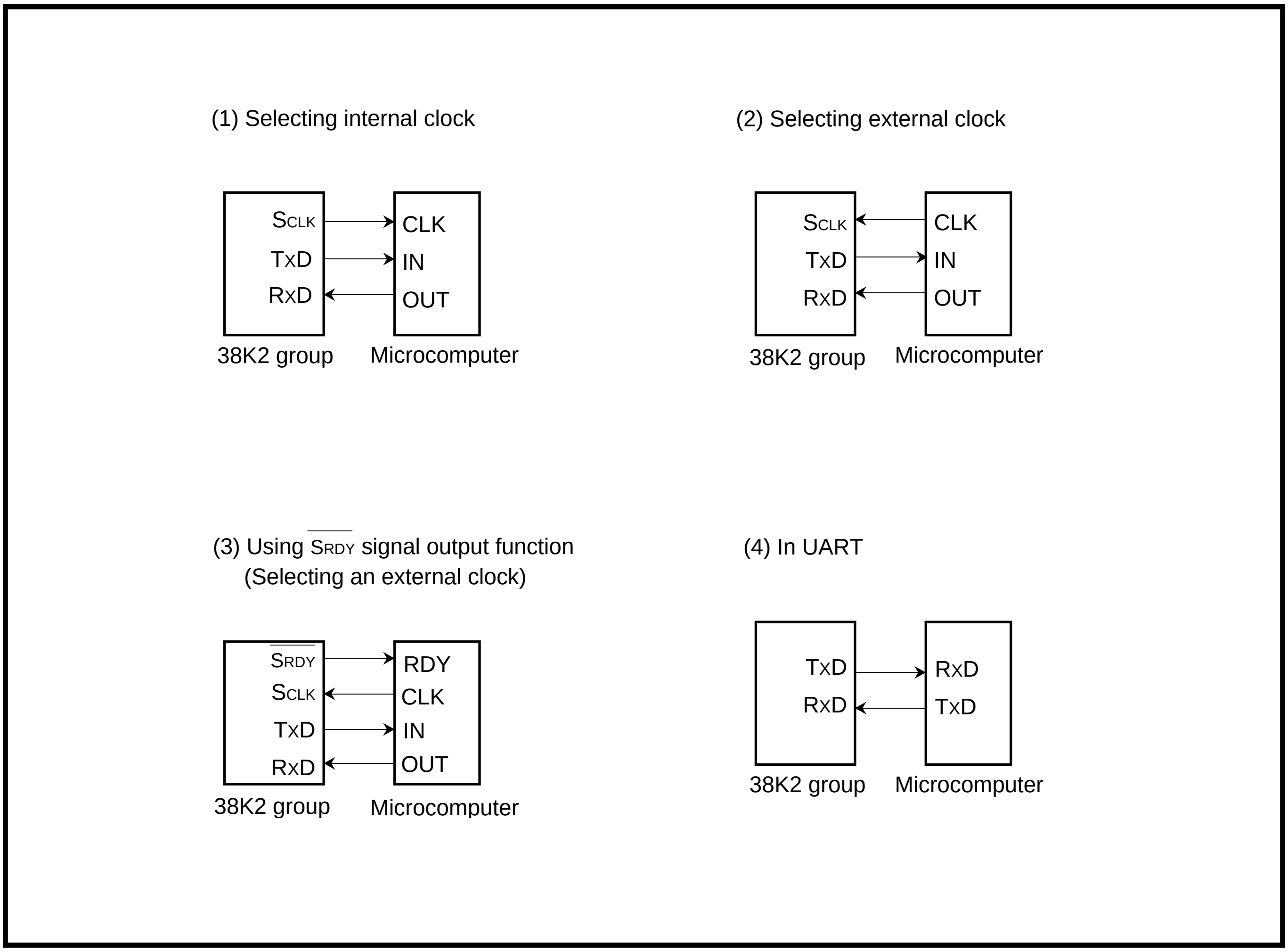


Fig. 2.4.11 Serial I/O connection examples (2)

2.4.4 Setting of serial I/O transfer data format

A clock synchronous or clock asynchronous (UART) can be selected as a data format of Serial I/O. Figure 2.4.12 shows the serial I/O transfer data format.

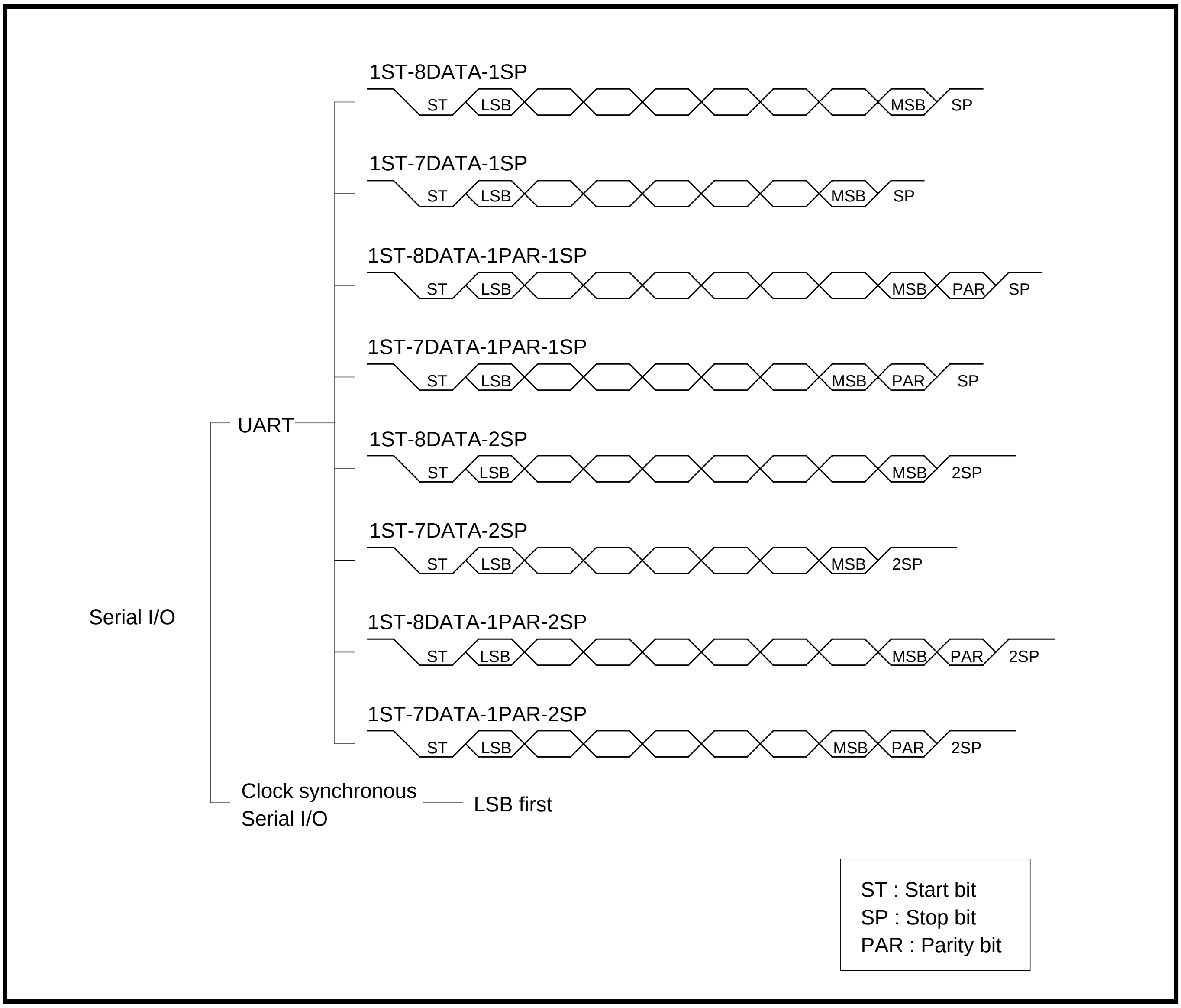


Fig. 2.4.12 Serial I/O transfer data format

2.4.5 Serial I/O application examples

(1) Communication using clock synchronous serial I/O (transmit/receive)

Outline : 2-byte data is transmitted and received, using the clock synchronous serial I/O. The $\overline{\text{SRDY}}$ signal is used for communication control.

Figure 2.4.13 shows a connection diagram, and Figure 2.4.14 shows a timing chart.
Figure 2.4.15 shows a registers setting related to the transmitting side, and Figure 2.4.16 shows registers setting related to the receiving side.

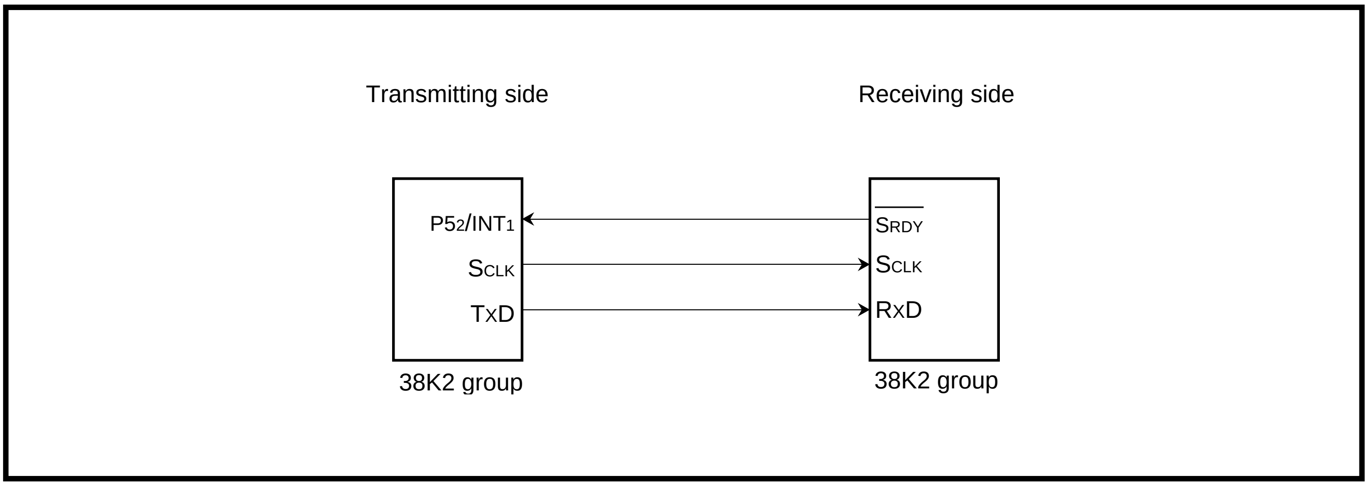


Fig. 2.4.13 Connection diagram

Specifications :
- The Serial I/O is used (clock synchronous serial I/O is selected.)
- Synchronous clock frequency : 125 kHz (f(XIN) = 6 MHz is divided by 48)
- The $\overline{\text{SRDY}}$ (receivable signal) is used.
- The receiving side outputs the $\overline{\text{SRDY}}$ signal at intervals of 2 ms (generated by timer), and 2-byte data is transferred from the transmitting side to the receiving side.

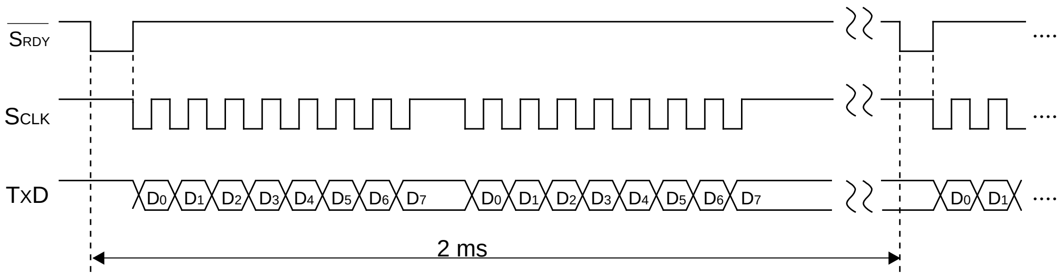
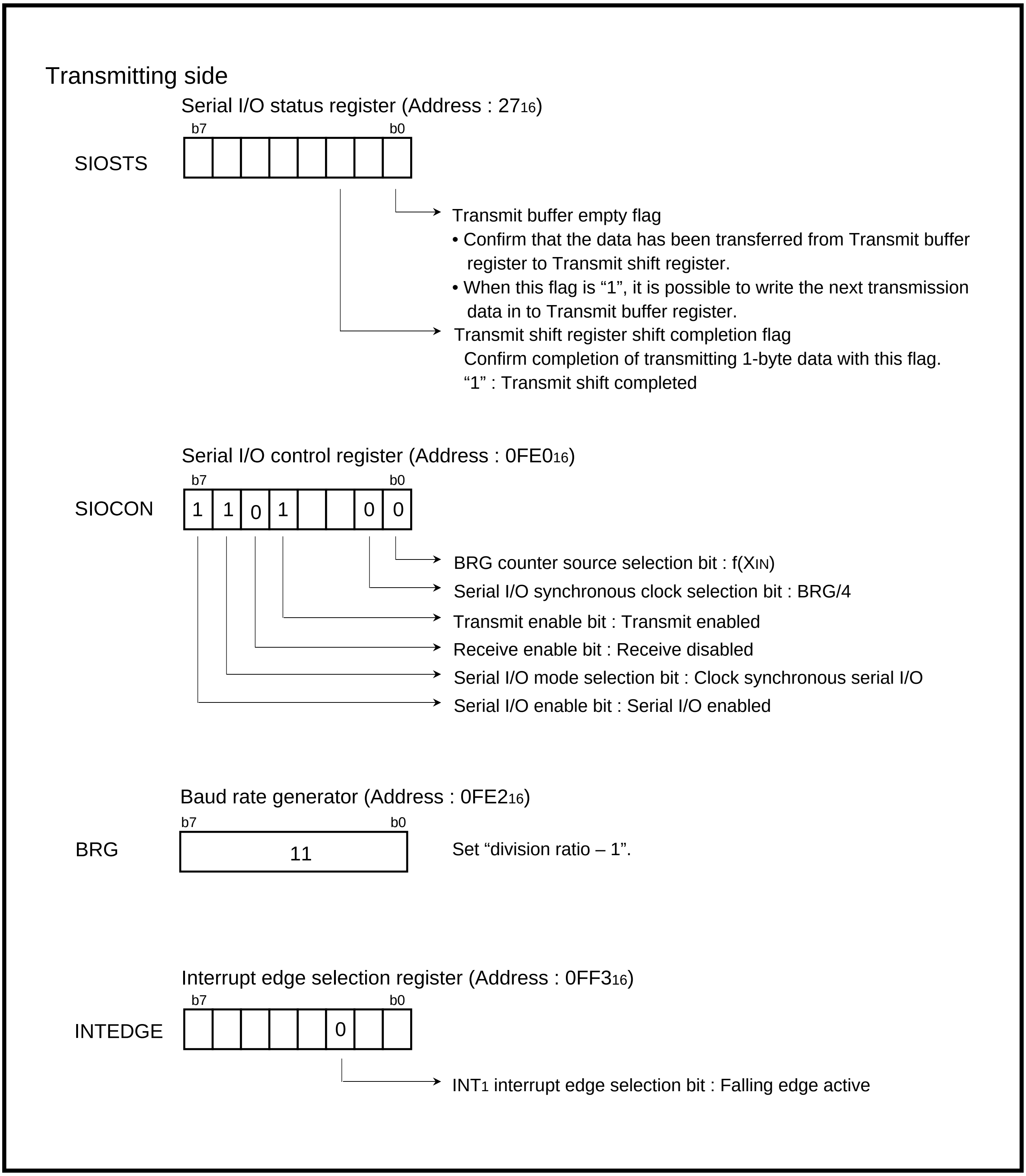


Fig. 2.4.14 Timing chart



Transmitting side

Serial I/O status register (Address : 27_{16})

Transmit buffer empty flag

- Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
- When this flag is “1”, it is possible to write the next transmission data in to Transmit buffer register.

Transmit shift register shift completion flag

Confirm completion of transmitting 1-byte data with this flag.

“1” : Transmit shift completed

Serial I/O control register (Address : $0FE0_{16}$)

BRG counter source selection bit : f(XIN)

Serial I/O synchronous clock selection bit : BRG/4

Transmit enable bit : Transmit enabled

Receive enable bit : Receive disabled

Serial I/O mode selection bit : Clock synchronous serial I/O

Serial I/O enable bit : Serial I/O enabled

Baud rate generator (Address : $0FE2_{16}$)

Set “division ratio – 1”.

Interrupt edge selection register (Address : $0FF3_{16}$)

INT_1 interrupt edge selection bit : Falling edge active

Fig. 2.4.15 Registers setting related to transmitting side

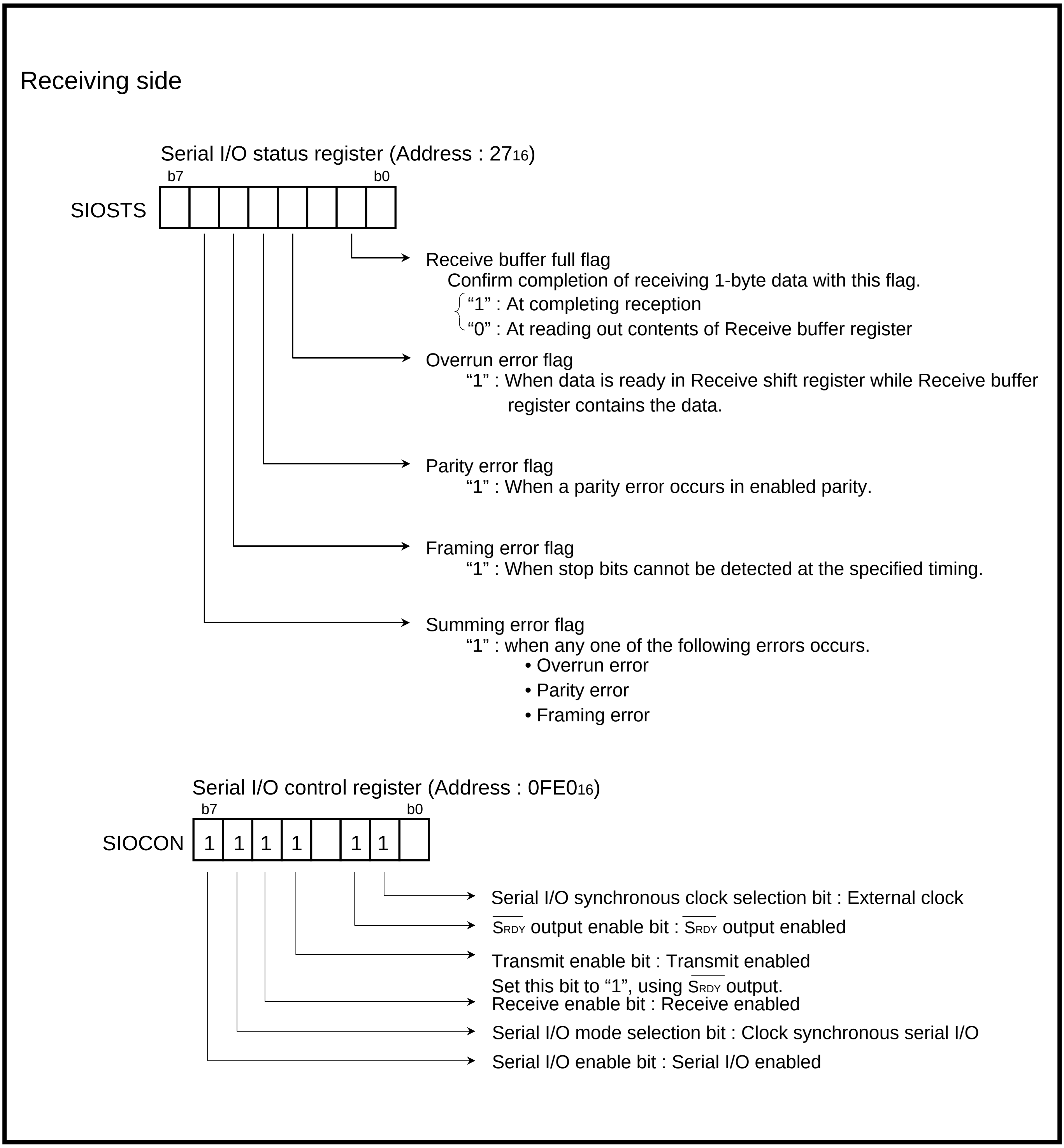

Receiving side
Serial I/O status register (Address : 27_{16})
b7
b0
SIOSTS
Receive buffer full flag
Confirm completion of receiving 1-byte data with this flag.
"1" : At completing reception
"0" : At reading out contents of Receive buffer register
Overrun error flag
"1" : When data is ready in Receive shift register while Receive buffer register contains the data.
Parity error flag
"1" : When a parity error occurs in enabled parity.
Framing error flag
"1" : When stop bits cannot be detected at the specified timing.
Summing error flag
"1" : when any one of the following errors occurs.
• Overrun error
• Parity error
• Framing error
Serial I/O control register (Address : $0FE0_{16}$)
b7
b0
SIOCON
1 1 1 1 1 1
Serial I/O synchronous clock selection bit : External clock
$\overline{\text{SRDY}}$ output enable bit : $\overline{\text{SRDY}}$ output enabled
Transmit enable bit : Transmit enabled
Set this bit to "1", using $\overline{\text{SRDY}}$ output.
Receive enable bit : Receive enabled
Serial I/O mode selection bit : Clock synchronous serial I/O
Serial I/O enable bit : Serial I/O enabled

Fig. 2.4.16 Registers setting related to receiving side

Figure 2.4.17 shows a control procedure of the transmitting side, and Figure 2.4.18 shows a control procedure of the receiving side.

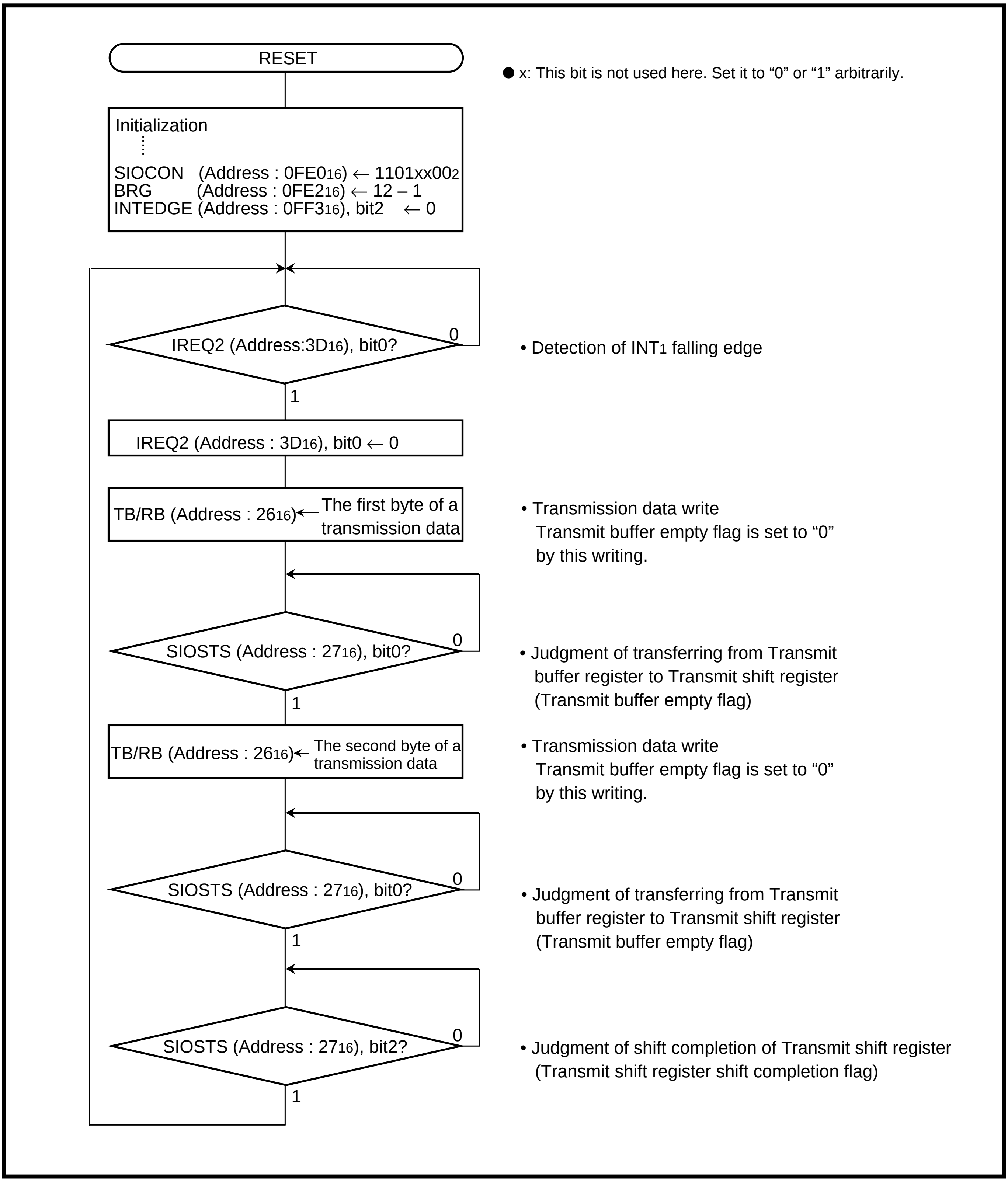

RESET
● x: This bit is not used here. Set it to "0" or "1" arbitrarily.
Initialization
SIOCON (Address : 0FE016) ← 1101xx002
BRG (Address : 0FE216) ← 12 – 1
INTEDGE (Address : 0FF316), bit2 ← 0
IREQ2 (Address:3D16), bit0?
0
1
• Detection of INT1 falling edge
IREQ2 (Address : 3D16), bit0 ← 0
TB/RB (Address : 2616) ← The first byte of a transmission data
• Transmission data write
Transmit buffer empty flag is set to "0" by this writing.
SIOSTS (Address : 2716), bit0?
0
1
• Judgment of transferring from Transmit buffer register to Transmit shift register (Transmit buffer empty flag)
TB/RB (Address : 2616) ← The second byte of a transmission data
• Transmission data write
Transmit buffer empty flag is set to "0" by this writing.
SIOSTS (Address : 2716), bit0?
0
1
• Judgment of transferring from Transmit buffer register to Transmit shift register (Transmit buffer empty flag)
SIOSTS (Address : 2716), bit2?
0
1
• Judgment of shift completion of Transmit shift register (Transmit shift register shift completion flag)

Fig. 2.4.17 Control procedure of transmitting side

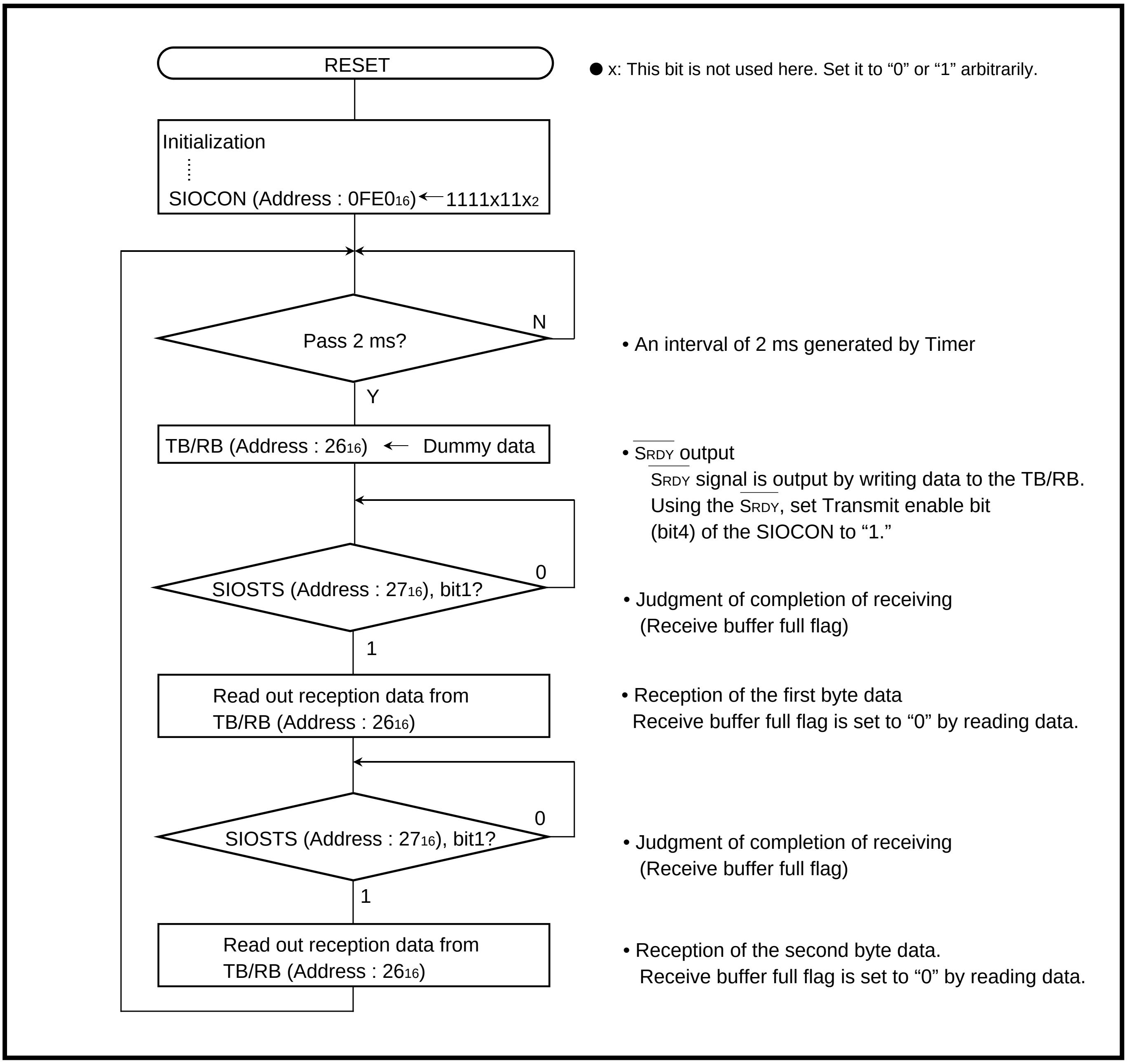

RESET
Initialization
SIOCON (Address : 0FE0₁₆) ← 1111x11x₂
Pass 2 ms?
N
Y
TB/RB (Address : 26₁₆) ← Dummy data
SIOSTS (Address : 27₁₆), bit1?
0
1
Read out reception data from TB/RB (Address : 26₁₆)
SIOSTS (Address : 27₁₆), bit1?
0
1
Read out reception data from TB/RB (Address : 26₁₆)
● x: This bit is not used here. Set it to "0" or "1" arbitrarily.
• An interval of 2 ms generated by Timer
• SRDY output
SRDY signal is output by writing data to the TB/RB.
Using the SRDY, set Transmit enable bit (bit4) of the SIOCON to "1."
• Judgment of completion of receiving (Receive buffer full flag)
• Reception of the first byte data
Receive buffer full flag is set to "0" by reading data.
• Judgment of completion of receiving (Receive buffer full flag)
• Reception of the second byte data.
Receive buffer full flag is set to "0" by reading data.

Fig. 2.4.18 Control procedure of receiving side

(2) Output of serial data (control of peripheral IC)

Outline : 4-byte data is transmitted and received, using the clock synchronous serial I/O.
The CS signal is output to a peripheral IC through port $P5_3$.

The example for using Serial I/O is shown.
Figure 2.4.19 shows a connection diagram, and Figure 2.4.20 shows a timing chart.

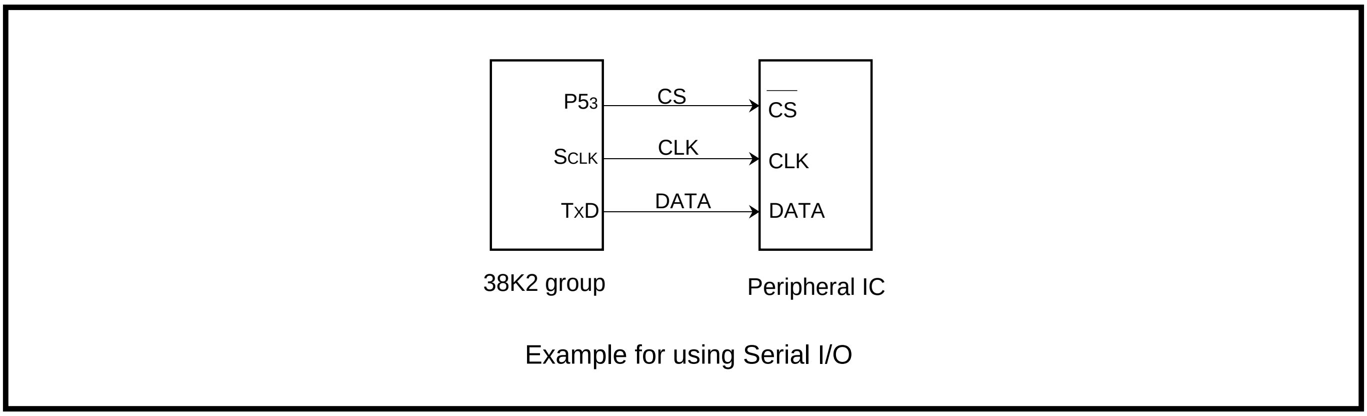


Fig. 2.4.19 Connection diagram

Specifications :
- The Serial I/O is used (clock synchronous serial I/O is selected.)
- Synchronous clock frequency : 125 kHz (f(X_{IN}) = 6 MHz is divided by 48)
- Transfer direction : LSB first
- The Serial I/O interrupt is not used.
- Port $P5_3$ is connected to the $\overline{CS}$ pin ("L" active) of the peripheral IC for transmission control; the output level of port $P5_3$ is controlled by software.

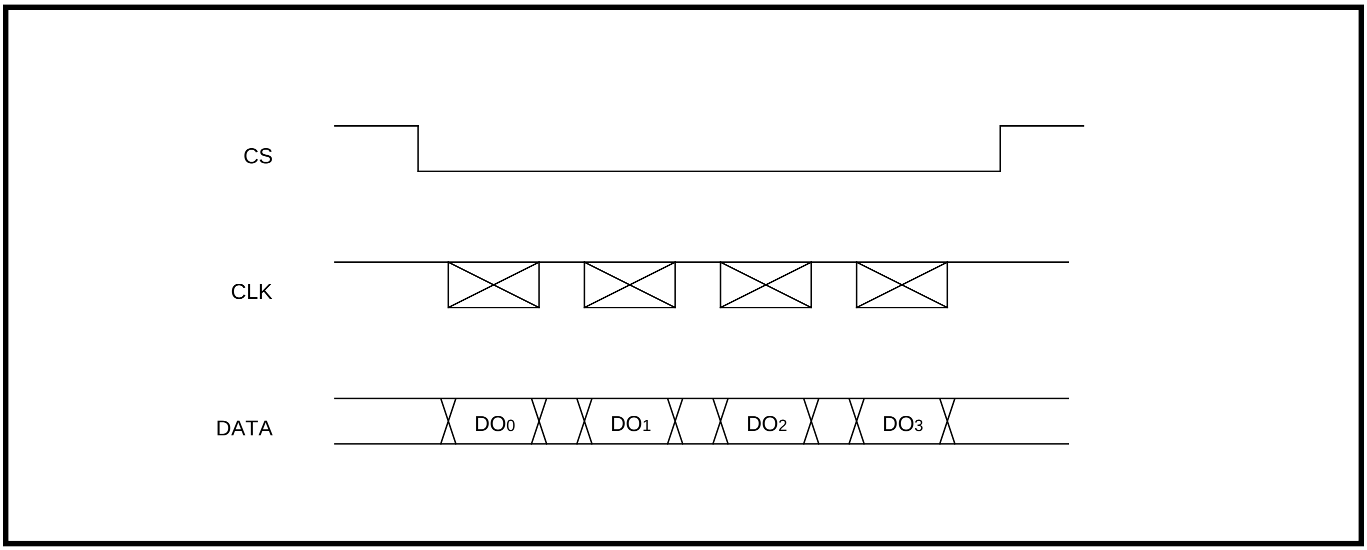


Fig. 2.4.20 Timing chart

Figure 2.4.21 shows registers setting related to Serial I/O, and Figure 2.4.22 shows a setting of serial I/O transmission data.

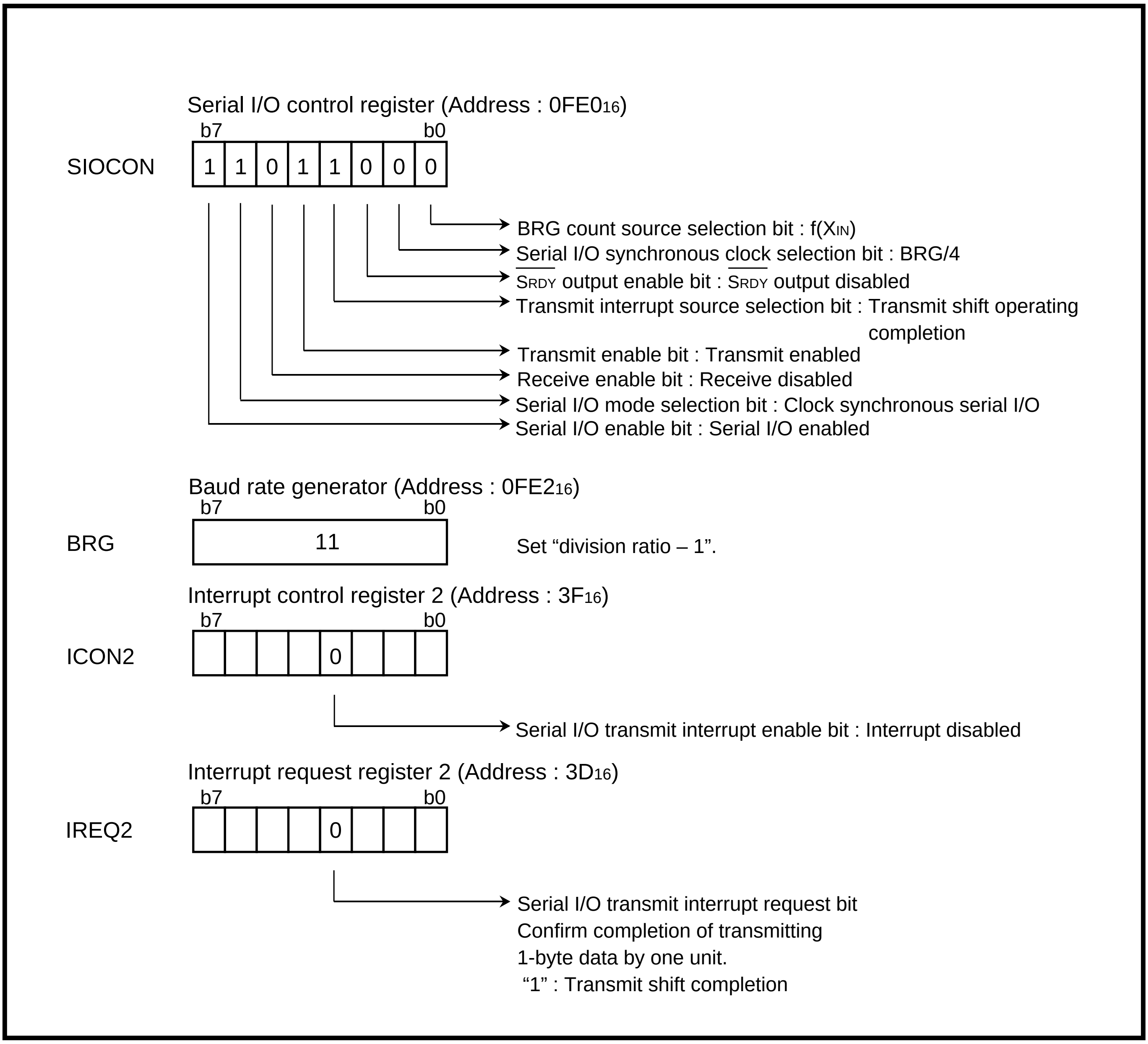

Serial I/O control register (Address : 0FE0$_{16}$)
b7
b0
SIOCON
1 1 0 1 1 0 0 0
BRG count source selection bit : f(XIN)
Serial I/O synchronous clock selection bit : BRG/4
SRDY output enable bit : SRDY output disabled
Transmit interrupt source selection bit : Transmit shift operating completion
Transmit enable bit : Transmit enabled
Receive enable bit : Receive disabled
Serial I/O mode selection bit : Clock synchronous serial I/O
Serial I/O enable bit : Serial I/O enabled
Baud rate generator (Address : 0FE2$_{16}$)
b7
b0
BRG
11
Set “division ratio – 1”.
Interrupt control register 2 (Address : 3F$_{16}$)
b7
b0
ICON2
0
Serial I/O transmit interrupt enable bit : Interrupt disabled
Interrupt request register 2 (Address : 3D$_{16}$)
b7
b0
IREQ2
0
Serial I/O transmit interrupt request bit
Confirm completion of transmitting
1-byte data by one unit.
“1” : Transmit shift completion

Fig. 2.4.21 Registers setting related to Serial I/O

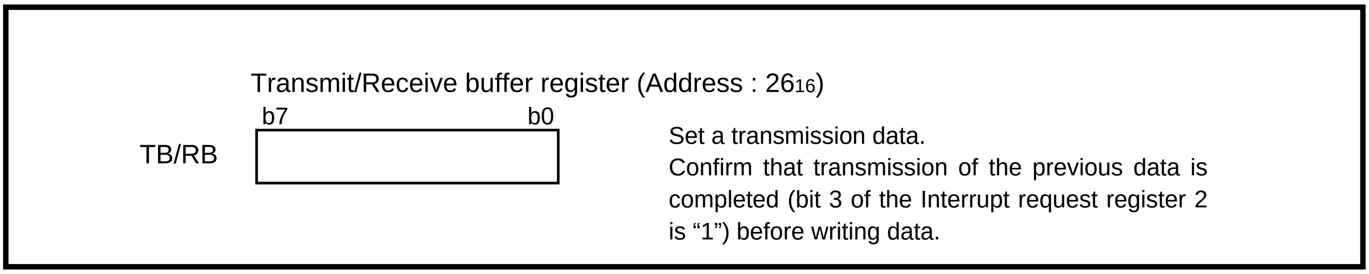

Transmit/Receive buffer register (Address : 26$_{16}$)
b7
b0
TB/RB
Set a transmission data.
Confirm that transmission of the previous data is completed (bit 3 of the Interrupt request register 2 is “1”) before writing data.

Fig. 2.4.22 Setting of serial I/O transmission data

When the registers are set as shown in Fig. 2.4.21, the Serial I/O can transmit 1-byte data by writing data to the transmit buffer register.
Thus, after setting the CS signal to “L”, write the transmission data to the transmit buffer register by each 1 byte, and return the CS signal to “H” when the target number of bytes has been transmitted. Figure 2.4.23 shows a control procedure of Serial I/O.

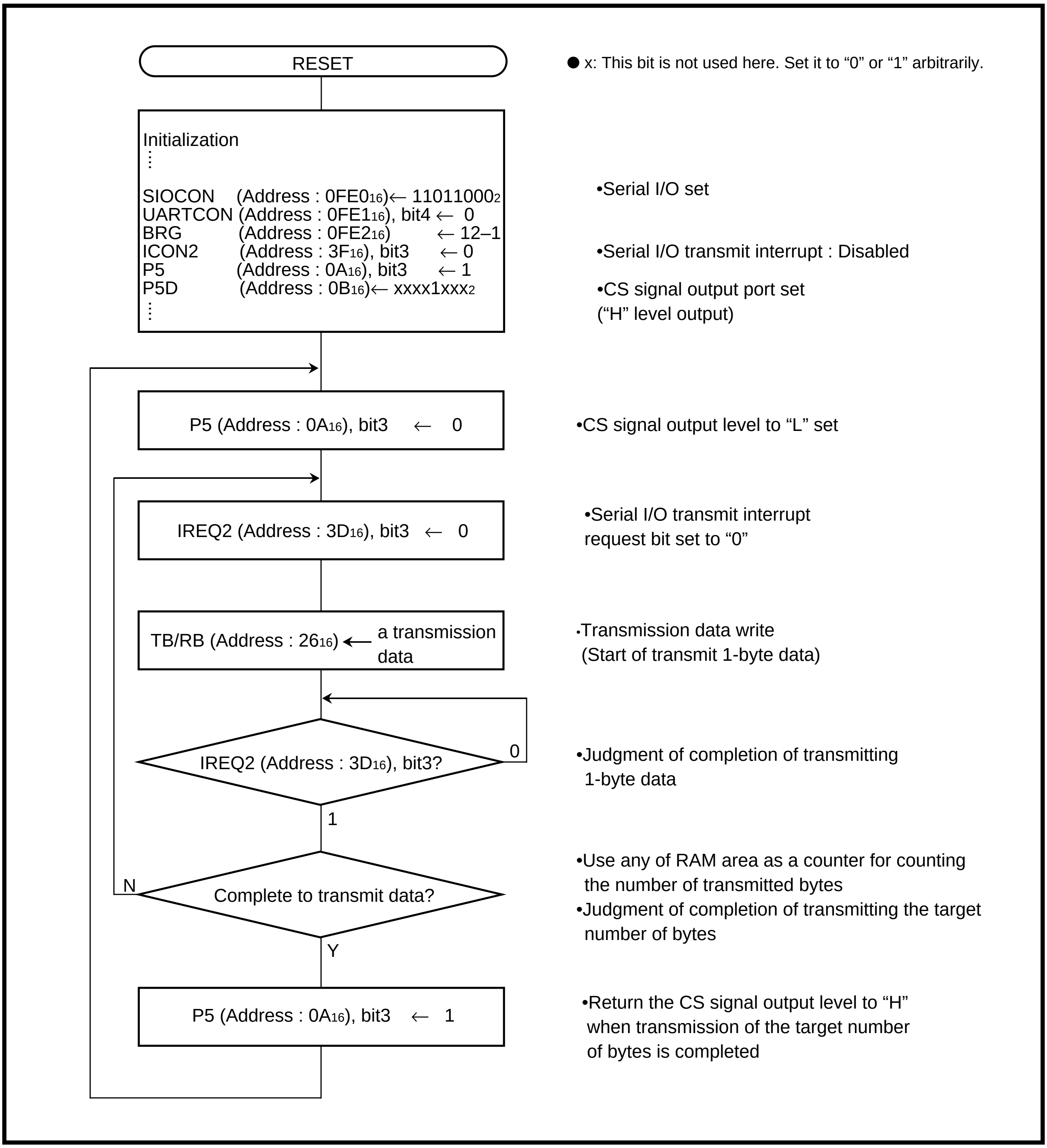

RESET
● x: This bit is not used here. Set it to “0” or “1” arbitrarily.
Initialization
SIOCON (Address : 0FE0$_{16}$)← 11011000$_2$
UARTCON (Address : 0FE1$_{16}$), bit4 ← 0
BRG (Address : 0FE2$_{16}$) ← 12–1
ICON2 (Address : 3F$_{16}$), bit3 ← 0
P5 (Address : 0A$_{16}$), bit3 ← 1
P5D (Address : 0B$_{16}$)← xxxx1xxx$_2$
•Serial I/O set
•Serial I/O transmit interrupt : Disabled
•CS signal output port set (“H” level output)
P5 (Address : 0A$_{16}$), bit3 ← 0
•CS signal output level to “L” set
IREQ2 (Address : 3D$_{16}$), bit3 ← 0
•Serial I/O transmit interrupt request bit set to “0”
TB/RB (Address : 26$_{16}$) ← a transmission data
•Transmission data write (Start of transmit 1-byte data)
IREQ2 (Address : 3D$_{16}$), bit3?
0
1
•Judgment of completion of transmitting 1-byte data
Complete to transmit data?
N
Y
•Use any of RAM area as a counter for counting the number of transmitted bytes
•Judgment of completion of transmitting the target number of bytes
P5 (Address : 0A$_{16}$), bit3 ← 1
•Return the CS signal output level to “H” when transmission of the target number of bytes is completed

Fig. 2.4.23 Control procedure of Serial I/O

(3) Cyclic transmission or reception of block data (data of specified number of bytes) between two microcomputers

Outline : When the clock synchronous serial I/O is used for communication, synchronization of the clock and the data between the transmitting and receiving sides may be lost because of noise included in the synchronous clock. It is necessary to correct that constantly, using "heading adjustment".

This "heading adjustment" is carried out by using the interval between blocks in this example.

Figure 2.4.24 shows a connection diagram.

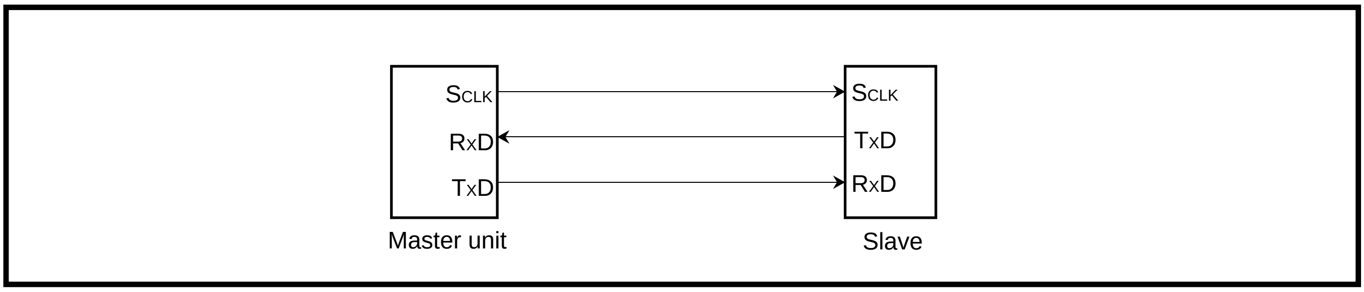


Fig. 2.4.24 Connection diagram

Specifications :

- The serial I/O is used (clock synchronous serial I/O is selected).
- Synchronous clock frequency : 125 kHz (f(X_{IN}) = 6 MHz is divided by 48)
- Byte cycle: 488 µs
- Number of bytes for transmission or reception : 8 byte/block
- Block transfer cycle : 16 ms
- Block transfer term : 3.5 ms
- Interval between blocks : 12.5 ms
- Heading adjustment time : 8 ms

Limitations of specifications :

- Reading of the reception data and setting of the next transmission data must be completed within the time obtained from "byte cycle – time for transferring 1-byte data" (in this example, the time taken from generating of the serial I/O receive interrupt request to input of the next synchronous clock is 431 µs).
- "Heading adjustment time < interval between blocks" must be satisfied.

The communication is performed according to the timing shown in Figure 2.4.25. In the slave unit, when a synchronous clock is not input within a certain time (heading adjustment time), the next clock input is processed as the beginning (heading) of a block.
When a clock is input again after one block (8 byte) is received, the clock is ignored.
Figure 2.4.26 shows related registers setting.

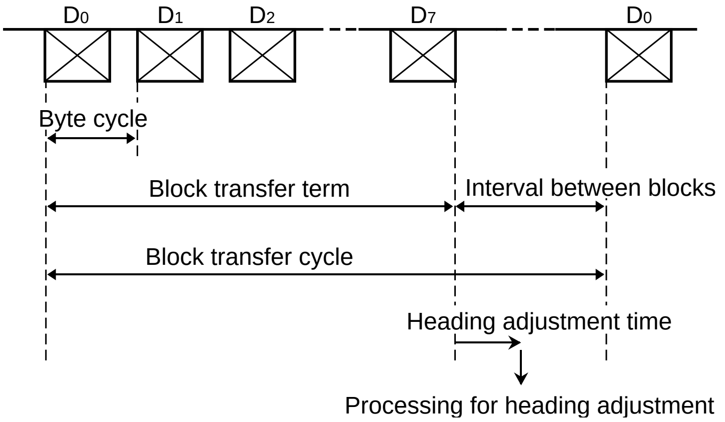


Fig. 2.4.25 Timing chart

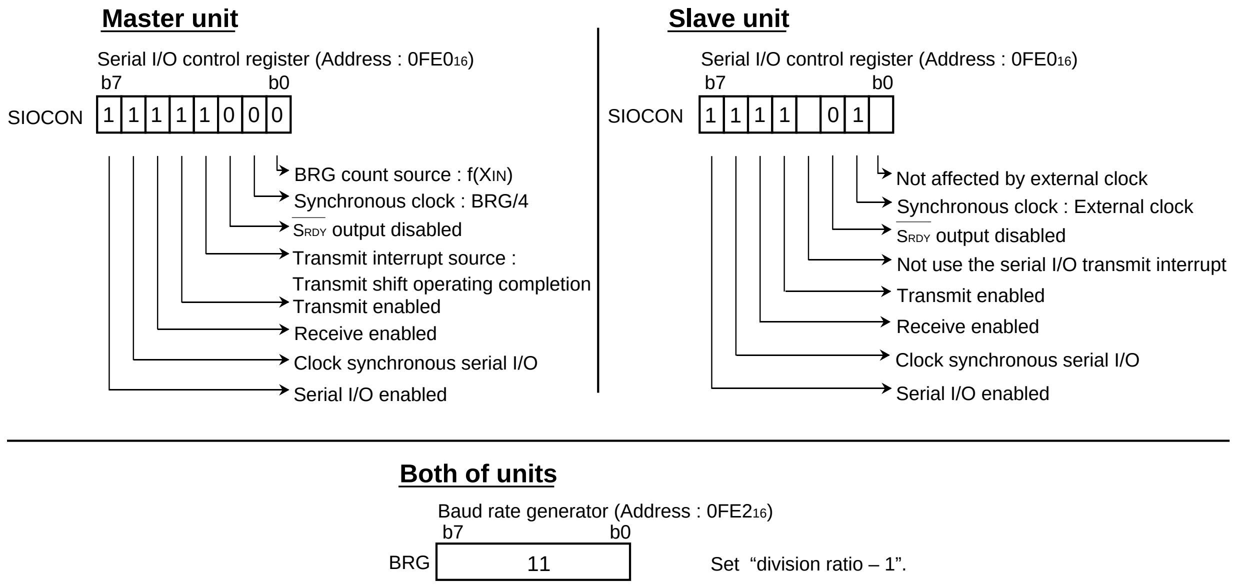

Master unit
Serial I/O control register (Address : 0FE016)
b7
b0
SIOCON
1 1 1 1 1 0 0 0
BRG count source : f(XIN)
Synchronous clock : BRG/4
SRDY output disabled
Transmit interrupt source :
Transmit shift operating completion
Transmit enabled
Receive enabled
Clock synchronous serial I/O
Serial I/O enabled
Slave unit
Serial I/O control register (Address : 0FE016)
b7
b0
SIOCON
1 1 1 1 0 1
Not affected by external clock
Synchronous clock : External clock
SRDY output disabled
Not use the serial I/O transmit interrupt
Transmit enabled
Receive enabled
Clock synchronous serial I/O
Serial I/O enabled
Both of units
Baud rate generator (Address : 0FE216)
b7
b0
BRG
11
Set "division ratio – 1".

Fig. 2.4.26 Related registers setting

Control procedure :

● Control in the master unit

After setting the related registers shown in Figure 2.4.26, the master unit starts transmission or reception of 1-byte data by writing transmission data to the transmit buffer register.

To perform the communication in the timing shown in Figure 2.4.25, take the timing into account and write transmission data. Additionally, read out the reception data when the serial I/O transmit interrupt request bit is set to "1," or before the next transmission data is written to the transmit buffer register.

Figure 2.4.27 shows a control procedure of the master unit using timer interrupts.

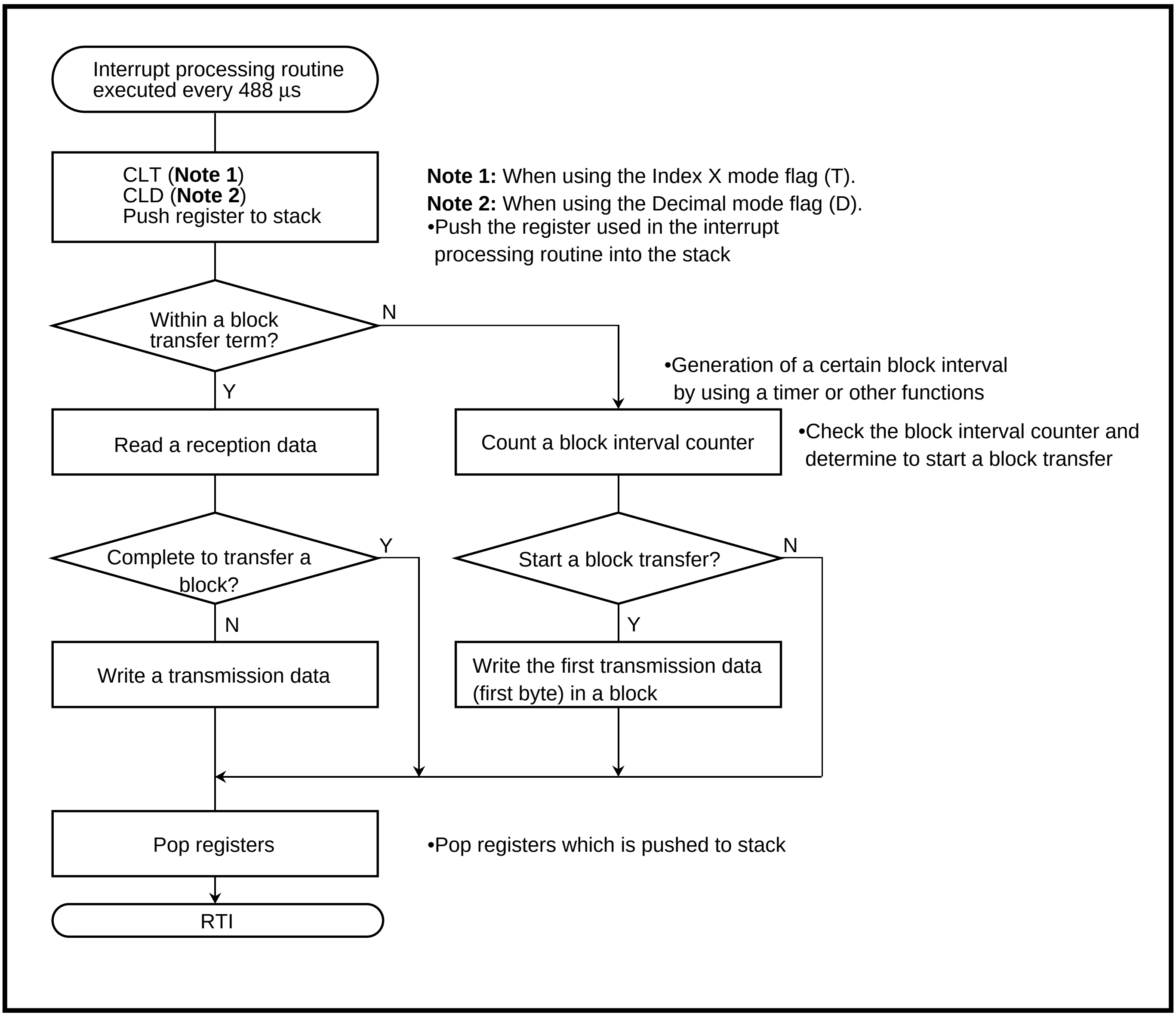

Interrupt processing routine executed every 488 µs
CLT (**Note 1**)
CLD (**Note 2**)
Push register to stack
Note 1: When using the Index X mode flag (T).
Note 2: When using the Decimal mode flag (D).
•Push the register used in the interrupt processing routine into the stack
Within a block transfer term?
N
Y
•Generation of a certain block interval by using a timer or other functions
Read a reception data
Count a block interval counter
•Check the block interval counter and determine to start a block transfer
Complete to transfer a block?
Y
N
Start a block transfer?
N
Y
Write a transmission data
Write the first transmission data (first byte) in a block
Pop registers
•Pop registers which is pushed to stack
RTI

Fig. 2.4.27 Control procedure of master unit

- Control in the slave unit

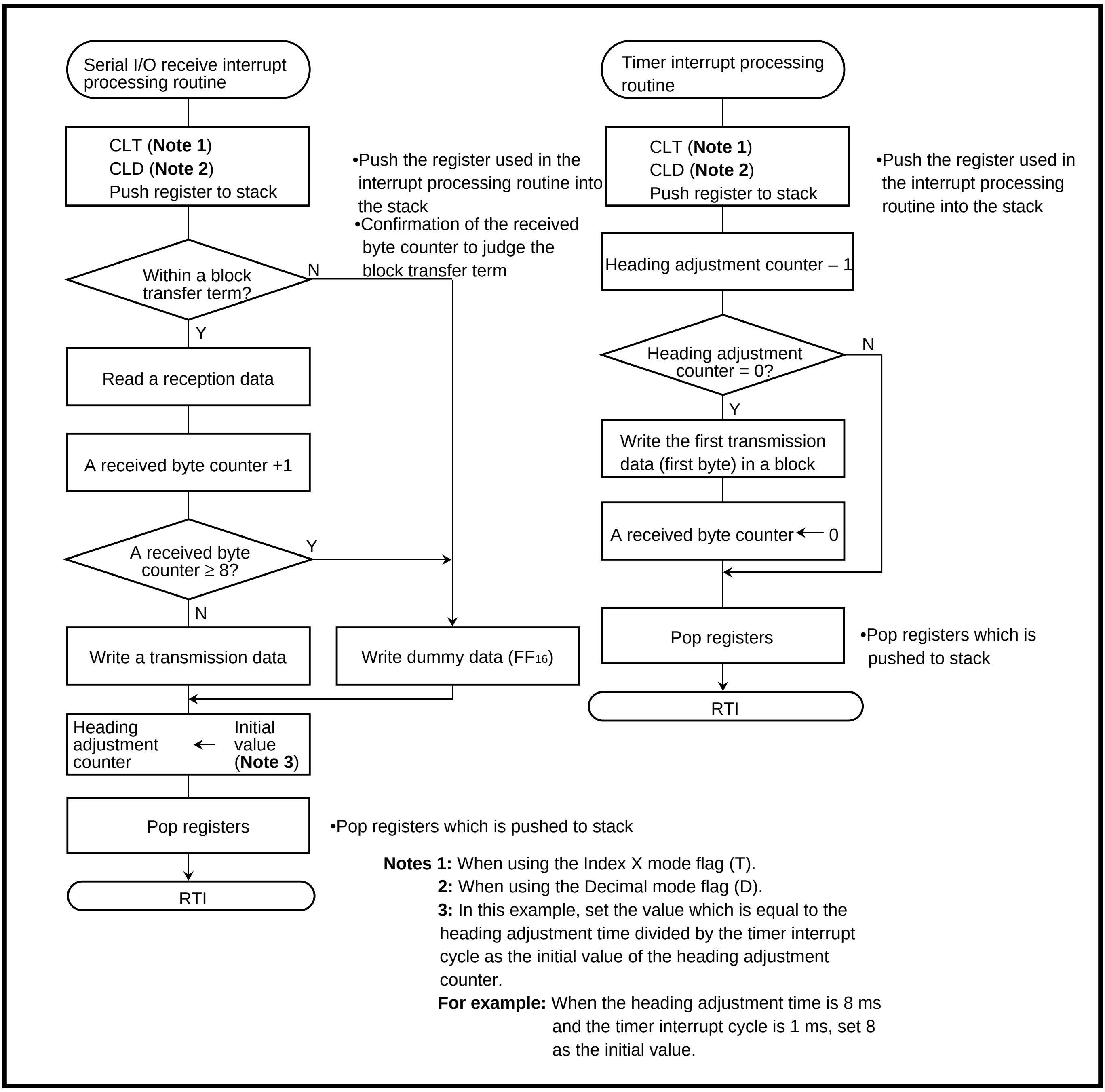
After setting the related registers as shown in Figure 2.4.26, the slave unit becomes the state where a synchronous clock can be received at any time, and the serial I/O receive interrupt request bit is set to “1” each time an 8-bit synchronous clock is received.

In the serial I/O receive interrupt processing routine, the data to be transmitted next is written to the transmit buffer register after the received data is read out.

However, if no serial I/O receive interrupt occurs for a certain time (heading adjustment time or more), the following processing will be performed.

1. The first 1-byte data of the transmission data in the block is written into the transmit buffer register.
2. The data to be received next is processed as the first 1 byte of the received data in the block.

Figure 2.4.28 shows a control procedure of the slave unit using the serial I/O receive interrupt and any timer interrupt (for heading adjustment).


Serial I/O receive interrupt processing routine
CLT (Note 1)
CLD (Note 2)
Push register to stack
•Push the register used in the interrupt processing routine into the stack
•Confirmation of the received byte counter to judge the block transfer term
Within a block transfer term?
N
Y
Read a reception data
A received byte counter +1
A received byte counter ≥ 8?
Y
N
Write a transmission data
Write dummy data (FF16)
Heading adjustment counter ← Initial value (Note 3)
Pop registers
•Pop registers which is pushed to stack
RTI
Timer interrupt processing routine
CLT (Note 1)
CLD (Note 2)
Push register to stack
•Push the register used in the interrupt processing routine into the stack
Heading adjustment counter – 1
Heading adjustment counter = 0?
N
Y
Write the first transmission data (first byte) in a block
A received byte counter ← 0
Pop registers
•Pop registers which is pushed to stack
RTI

Notes 1: When using the Index X mode flag (T).
2: When using the Decimal mode flag (D).
3: In this example, set the value which is equal to the heading adjustment time divided by the timer interrupt cycle as the initial value of the heading adjustment counter.
For example: When the heading adjustment time is 8 ms and the timer interrupt cycle is 1 ms, set 8 as the initial value.

Fig. 2.4.28 Control procedure of slave unit

(4) Communication (transmit/receive) using asynchronous serial I/O (UART)

Outline : 2-byte data is transmitted and received, using the asynchronous serial I/O. Port $P2_4$ is used for communication control.

Figure 2.4.29 shows a connection diagram, and Figure 2.4.30 shows a timing chart.

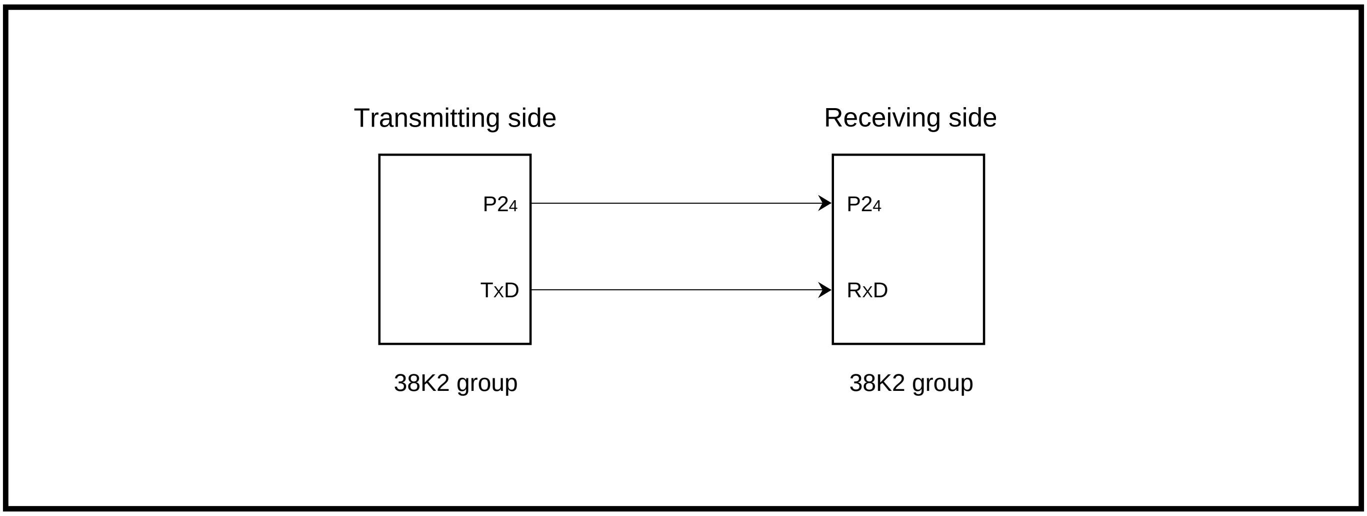


Fig. 2.4.29 Connection diagram (Communication using UART)

Specifications :
- The Serial I/O is used (UART is selected).
- Transfer bit rate : 9600 bps ($f(X_{IN})$ = 6 MHz is divided by 624)
- Communication control using port $P2_4$ (The output level of port $P2_4$ is controlled by software.)
- 2-byte data is transferred from the transmitting side to the receiving side at intervals of 10 ms generated by the timer.

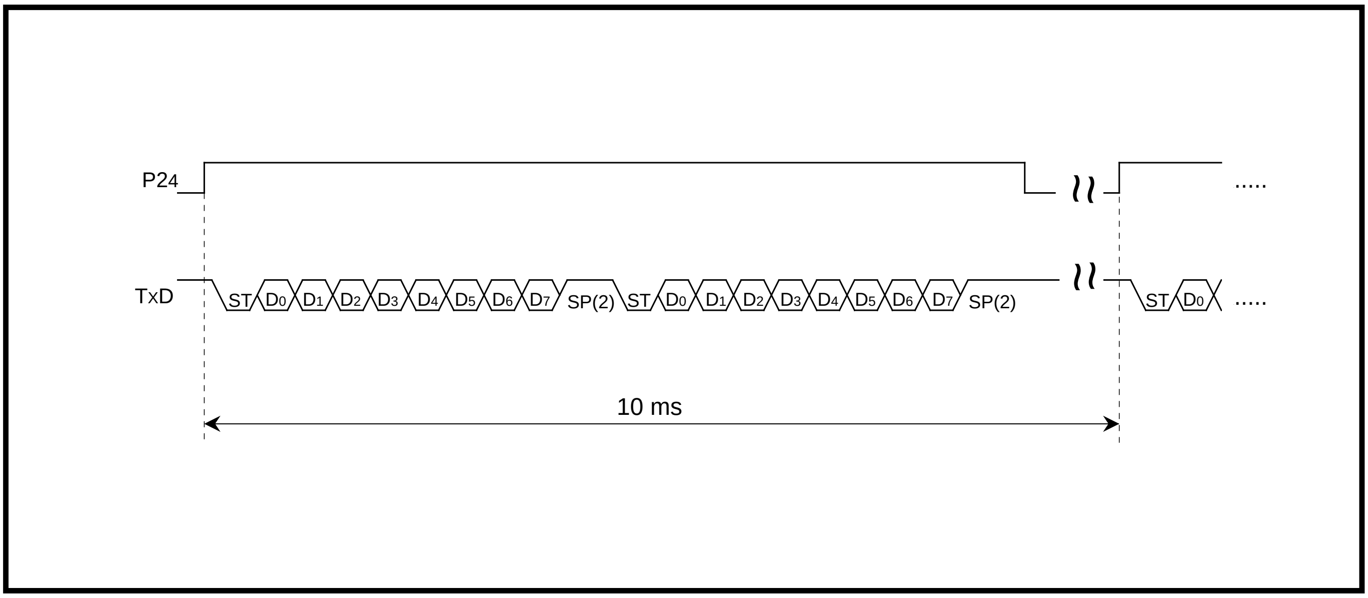

P24
TxD
ST D0 D1 D2 D3 D4 D5 D6 D7 SP(2) ST D0 D1 D2 D3 D4 D5 D6 D7 SP(2)
ST D0
.....
.....
10 ms

Fig. 2.4.30 Timing chart (using UART)

Table 2.4.1 shows setting examples of the baud rate generator (BRG) values and transfer bit rate values; Figure 2.4.31 shows registers setting related to the transmitting side; Figure 2.4.32 shows registers setting related to the receiving side.

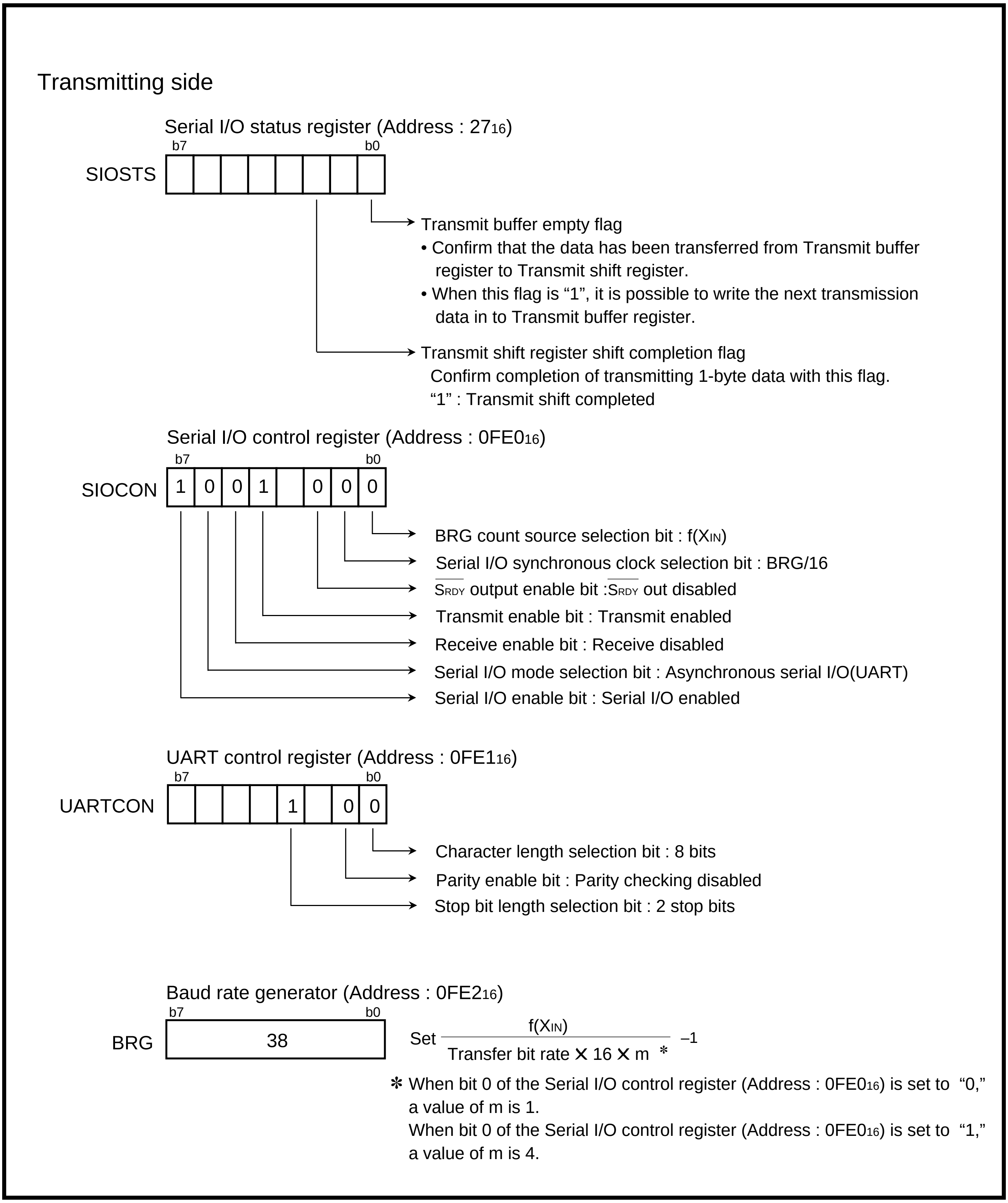
Table 2.4.1 Setting examples of Baud rate generator values and transfer bit rate values

Transfer bit rate (bps) **(Note 3)**	BRG count source **(Note 1)**	At $f(X_{IN})$ = 6 MHz BRG setting value **(Note 2)**	At $f(X_{IN})$ = 8 MHz BRG setting value **(Note 2)**
600	$f(X_{IN})/4$	155	207
1200	$f(X_{IN})/4$	77	103
2400	$f(X_{IN})$	155	207
4800	$f(X_{IN})$	77	103
9600	$f(X_{IN})$	38	51
14400	$f(X_{IN})$	25	34
19200	$f(X_{IN})$	19	25
38400	$f(X_{IN})$	9	12
57600	$f(X_{IN})$	–	8

Notes 1: Select the BRG count source with bit 0 of the serial I/O control register (Address : $0FE0_{16}$).
2: These are setting values with small errors.
3: Equation of transfer bit rate:

$$\text{Transfer bit rate (bps)} = \frac{f(X_{IN})}{(\text{BRG setting value} + 1) \times 16 \times m^{*}}$$

✻m: When bit 0 of the serial I/O control register (Address : $0FE0_{16}$) is set to “0”, a value of m is 1.
When bit 0 of the serial I/O control register (Address : $0FE0_{16}$) is set to “1”, a value of m is 4.



Transmitting side

Serial I/O status register (Address : 27_{16})

Transmit buffer empty flag
- Confirm that the data has been transferred from Transmit buffer register to Transmit shift register.
- When this flag is “1”, it is possible to write the next transmission data in to Transmit buffer register.

Transmit shift register shift completion flag
Confirm completion of transmitting 1-byte data with this flag.
“1” : Transmit shift completed

Serial I/O control register (Address : $0FE0_{16}$)

BRG count source selection bit : f(X_{IN})
Serial I/O synchronous clock selection bit : BRG/16
$\overline{S_{RDY}}$ output enable bit :$\overline{S_{RDY}}$ out disabled
Transmit enable bit : Transmit enabled
Receive enable bit : Receive disabled
Serial I/O mode selection bit : Asynchronous serial I/O(UART)
Serial I/O enable bit : Serial I/O enabled

UART control register (Address : $0FE1_{16}$)

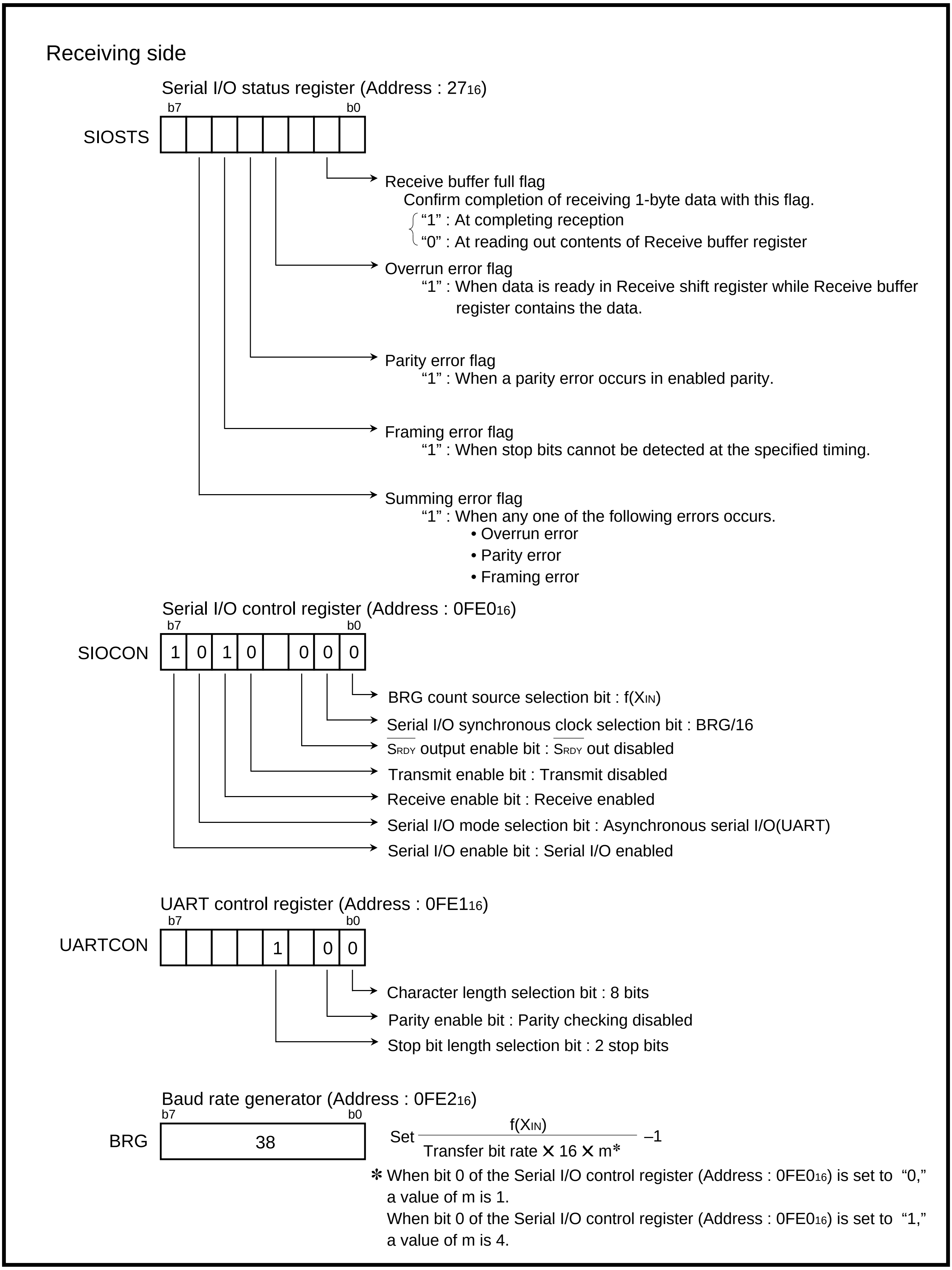
Character length selection bit : 8 bits
Parity enable bit : Parity checking disabled
Stop bit length selection bit : 2 stop bits

Baud rate generator (Address : $0FE2_{16}$)

Set $\frac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m^{*}} - 1$

✻ When bit 0 of the Serial I/O control register (Address : $0FE0_{16}$) is set to “0,” a value of m is 1.
When bit 0 of the Serial I/O control register (Address : $0FE0_{16}$) is set to “1,” a value of m is 4.

Fig. 2.4.31 Registers setting related to transmitting side



Receiving side

Serial I/O status register (Address : 27_{16})

Receive buffer full flag
Confirm completion of receiving 1-byte data with this flag.
"1" : At completing reception
"0" : At reading out contents of Receive buffer register

Overrun error flag
"1" : When data is ready in Receive shift register while Receive buffer register contains the data.

Parity error flag
"1" : When a parity error occurs in enabled parity.

Framing error flag
"1" : When stop bits cannot be detected at the specified timing.

Summing error flag
"1" : When any one of the following errors occurs.
- Overrun error
- Parity error
- Framing error

Serial I/O control register (Address : $0FE0_{16}$)

BRG count source selection bit : f(X_{IN})
Serial I/O synchronous clock selection bit : BRG/16
$\overline{\text{S}_{\text{RDY}}}$ output enable bit : $\overline{\text{S}_{\text{RDY}}}$ out disabled
Transmit enable bit : Transmit disabled
Receive enable bit : Receive enabled
Serial I/O mode selection bit : Asynchronous serial I/O(UART)
Serial I/O enable bit : Serial I/O enabled

UART control register (Address : $0FE1_{16}$)

Character length selection bit : 8 bits
Parity enable bit : Parity checking disabled
Stop bit length selection bit : 2 stop bits

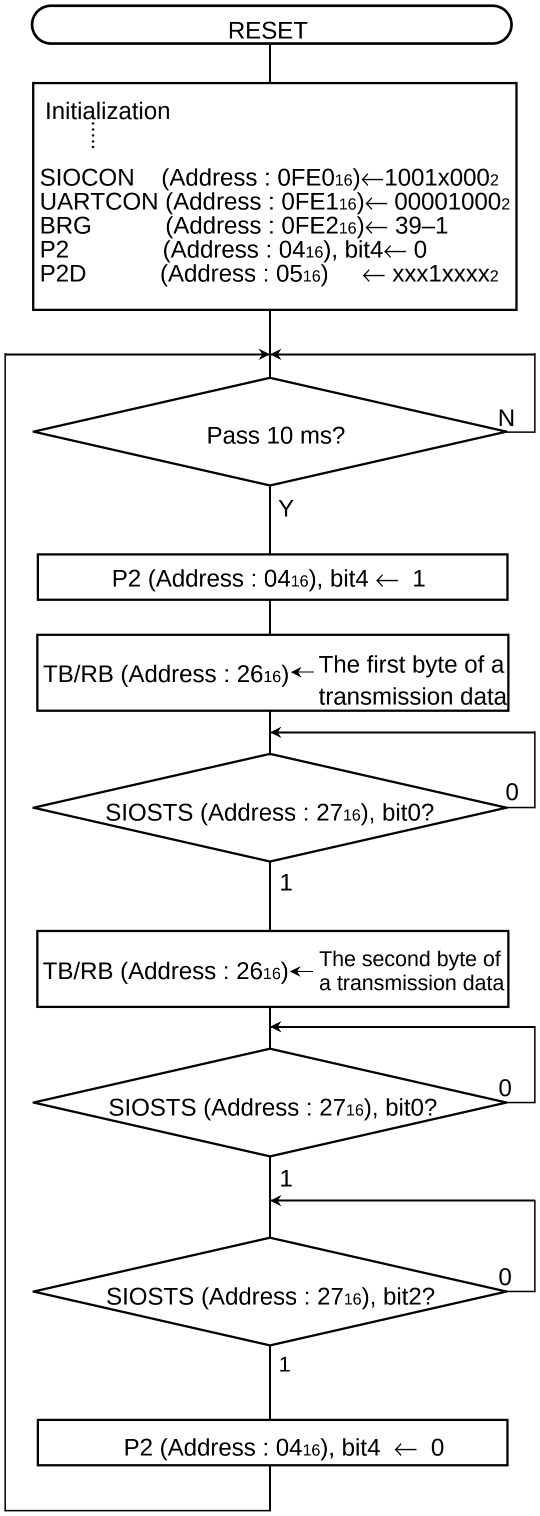
Baud rate generator (Address : $0FE2_{16}$)

Set $\frac{f(X_{IN})}{\text{Transfer bit rate} \times 16 \times m^{*}} - 1$

✻ When bit 0 of the Serial I/O control register (Address : $0FE0_{16}$) is set to "0," a value of m is 1.
When bit 0 of the Serial I/O control register (Address : $0FE0_{16}$) is set to "1," a value of m is 4.

Fig. 2.4.32 Registers setting related to receiving side

Figure 2.4.33 shows a control procedure of the transmitting side, and Figure 2.4.34 shows a control procedure of the receiving side.


RESET
Initialization
SIOCON (Address : $0FE0_{16}$)← $1001x000_2$
UARTCON (Address : $0FE1_{16}$)← 00001000_2
BRG (Address : $0FE2_{16}$)← 39–1
P2 (Address : 04_{16}), bit4← 0
P2D (Address : 05_{16}) ← $xxx1xxxx_2$
Pass 10 ms?
N
Y
P2 (Address : 04_{16}), bit4 ← 1
TB/RB (Address : 26_{16})← The first byte of a transmission data
SIOSTS (Address : 27_{16}), bit0?
0
1
TB/RB (Address : 26_{16})← The second byte of a transmission data
SIOSTS (Address : 27_{16}), bit0?
0
1
SIOSTS (Address : 27_{16}), bit2?
0
1
P2 (Address : 04_{16}), bit4 ← 0

● x: This bit is not used here. Set it to “0” or “1” arbitrarily.

- Port $P2_4$ set for communication control
- An interval of 10 ms generated by Timer
- Communication start
- Transmission data write
  Transmit buffer empty flag is set to “0” by this writing.
- Judgment of transferring data from Transmit buffer register to Transmit shift register (Transmit buffer empty flag)
- Transmission data write
  Transmit buffer empty flag is set to “0” by this writing.
- Judgment of transferring data from Transmit buffer register to Transmit shift register (Transmit buffer empty flag)
- Judgment of shift completion of Transmit shift register (Transmit shift register shift completion flag)
- Communication completion

Fig. 2.4.33 Control procedure of transmitting side

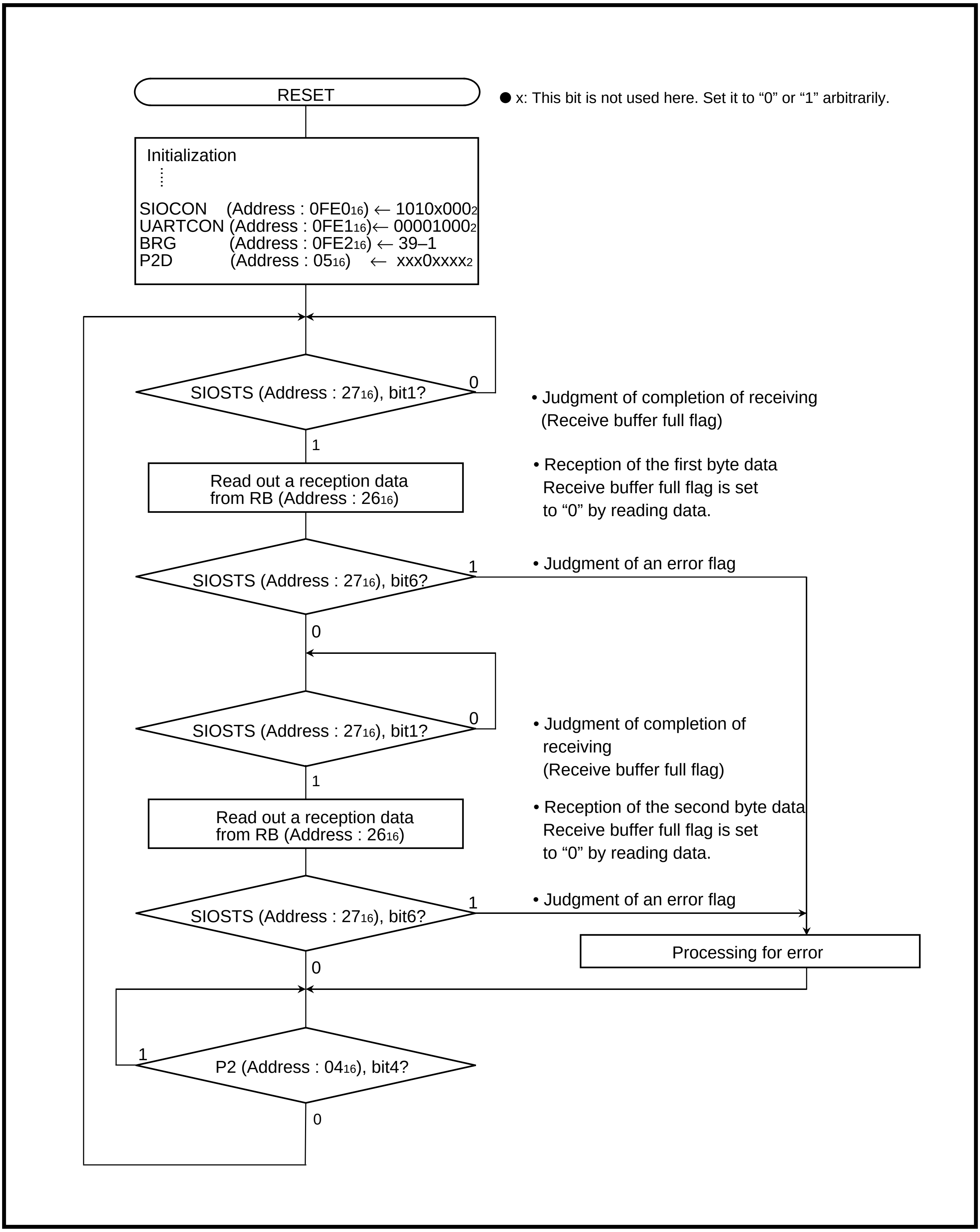

RESET
● x: This bit is not used here. Set it to “0” or “1” arbitrarily.
Initialization
SIOCON (Address : 0FE0$_{16}$) ← 1010x000$_2$
UARTCON (Address : 0FE1$_{16}$)← 00001000$_2$
BRG (Address : 0FE2$_{16}$) ← 39–1
P2D (Address : 05$_{16}$) ← xxx0xxxx$_2$
SIOSTS (Address : 27$_{16}$), bit1?
0
1
• Judgment of completion of receiving (Receive buffer full flag)
Read out a reception data from RB (Address : 26$_{16}$)
• Reception of the first byte data Receive buffer full flag is set to “0” by reading data.
SIOSTS (Address : 27$_{16}$), bit6?
1
• Judgment of an error flag
0
SIOSTS (Address : 27$_{16}$), bit1?
0
1
• Judgment of completion of receiving (Receive buffer full flag)
Read out a reception data from RB (Address : 26$_{16}$)
• Reception of the second byte data Receive buffer full flag is set to “0” by reading data.
SIOSTS (Address : 27$_{16}$), bit6?
1
• Judgment of an error flag
0
Processing for error
P2 (Address : 04$_{16}$), bit4?
1
0

Fig. 2.4.34 Control procedure of receiving side

2.4.6 Notes on serial I/O

(1) Notes when selecting clock synchronous serial I/O (Serial I/O)

① **Stop of transmission operation**
Clear the serial I/O enable bit and the transmit enable bit to “0” (Serial I/O and transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (Serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled), or clear the serial I/O enable bit to “0” (Serial I/O disabled).

③ **Stop of transmit/receive operation**
Clear the transmit enable bit and receive enable bit to “0” simultaneously (transmit and receive disabled).
(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● **Reason**
In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O enable bit to “0” (Serial I/O disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O (Serial I/O)

① **Stop of transmission operation**
Clear the transmit enable bit to “0” (transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (Serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled).

③ **Stop of transmit/receive operation**

Only transmission operation is stopped.
Clear the transmit enable bit to “0” (transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (Serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

Only receive operation is stopped.
Clear the receive enable bit to “0” (receive disabled).

(3) $\overline{\mathrm{SRDY}}$ output of reception side (Serial I/O)

When signals are output from the $\overline{\mathrm{SRDY}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\mathrm{SRDY}}$ output enable bit, and the transmit enable bit to “1” (transmit enabled).

(4) Setting serial I/O control register again (Serial I/O)

Set the serial I/O control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to “0.”

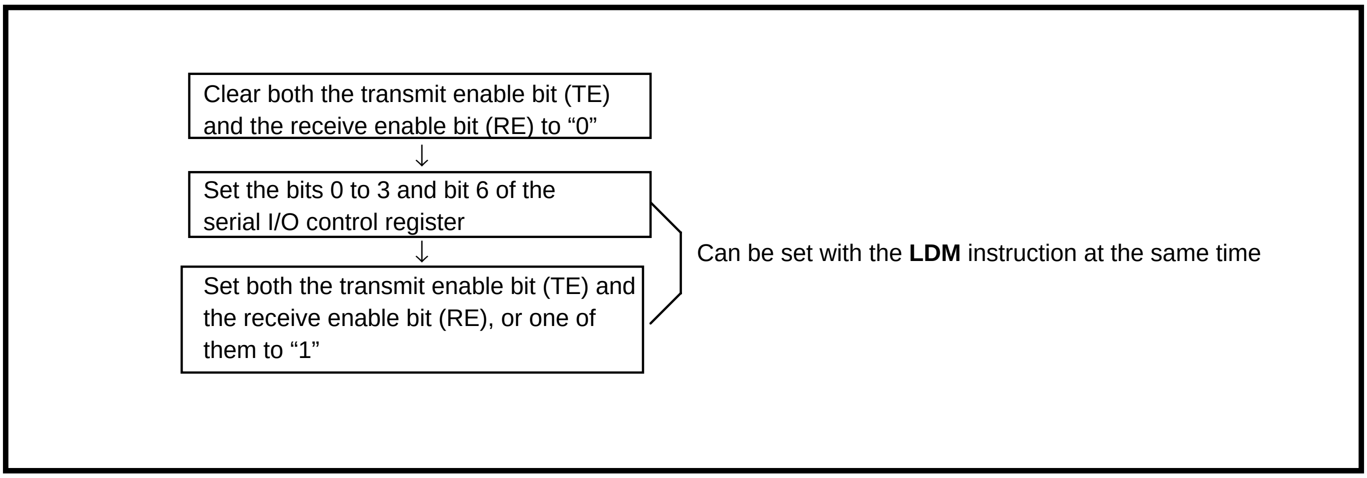

Clear both the transmit enable bit (TE) and the receive enable bit (RE) to “0”
Set the bits 0 to 3 and bit 6 of the serial I/O control register
Set both the transmit enable bit (TE) and the receive enable bit (RE), or one of them to “1”
Can be set with the **LDM** instruction at the same time

Fig. 2.4.35 Sequence of setting serial I/O control register again

(5) Data transmission control with referring to transmit shift register completion flag (Serial I/O)

The transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected (Serial I/O)

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK input level. Also, write the transmit data to the transmit buffer register (serial I/O shift register) at “H” of the SCLK input level.

(7) Transmit interrupt request when transmit enable bit is set (Serial I/O)

When the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as shown in the following sequence.

① Set the interrupt enable bit to “0” (disabled) with CLB instruction.
② Prepare serial I/O for transmission/reception.
③ Set the interrupt request bit to “0” with CLB instruction after 1 or more instruction has been executed.
④ Set the interrupt enable bit to “1” (enabled).

● Reason

When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”. The interrupt request is generated and the transmission interrupt bit is set regardless of which of the two timings listed below is selected as the timing for the transmission interrupt to be generated.

- Transmit buffer empty flag is set to “1”
- Transmit shift register completion flag is set to “1”

2.5 USB function

Some application notes are available on the Web site: “Renesas Technology Corp.” Homepage
USB Device
(http://www.renesas.com/en/usb)

Please refer to them for explanation and application of USB function.

2.6 HUB function

Some application notes are available on the Web site: “Renesas Technology Corp.” Homepage
USB Device
(http://www.renesas.com/en/usb)

Please refer to them for explanation and application of HUB function.

2.7 External bus interface(EXB)

Some application notes are available on the Web site: “Renesas Technology Corp.” Homepage
USB Device
(http://www.renesas.com/en/usb)

Please refer to them for explanation and application of external bus interface.

2.8 A/D converter

This paragraph explains the registers setting method and the notes related to the A/D converter.

2.8.1 Memory map

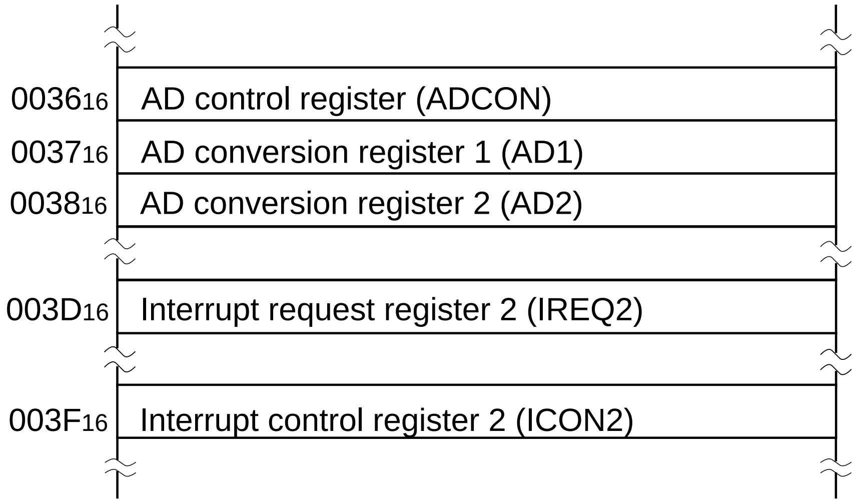
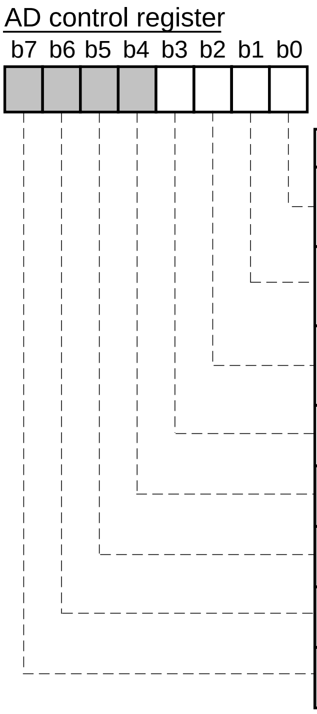

003616 AD control register (ADCON)
003716 AD conversion register 1 (AD1)
003816 AD conversion register 2 (AD2)
003D16 Interrupt request register 2 (IREQ2)
003F16 Interrupt control register 2 (ICON2)

Fig. 2.8.1 Memory map of registers related to A/D converter

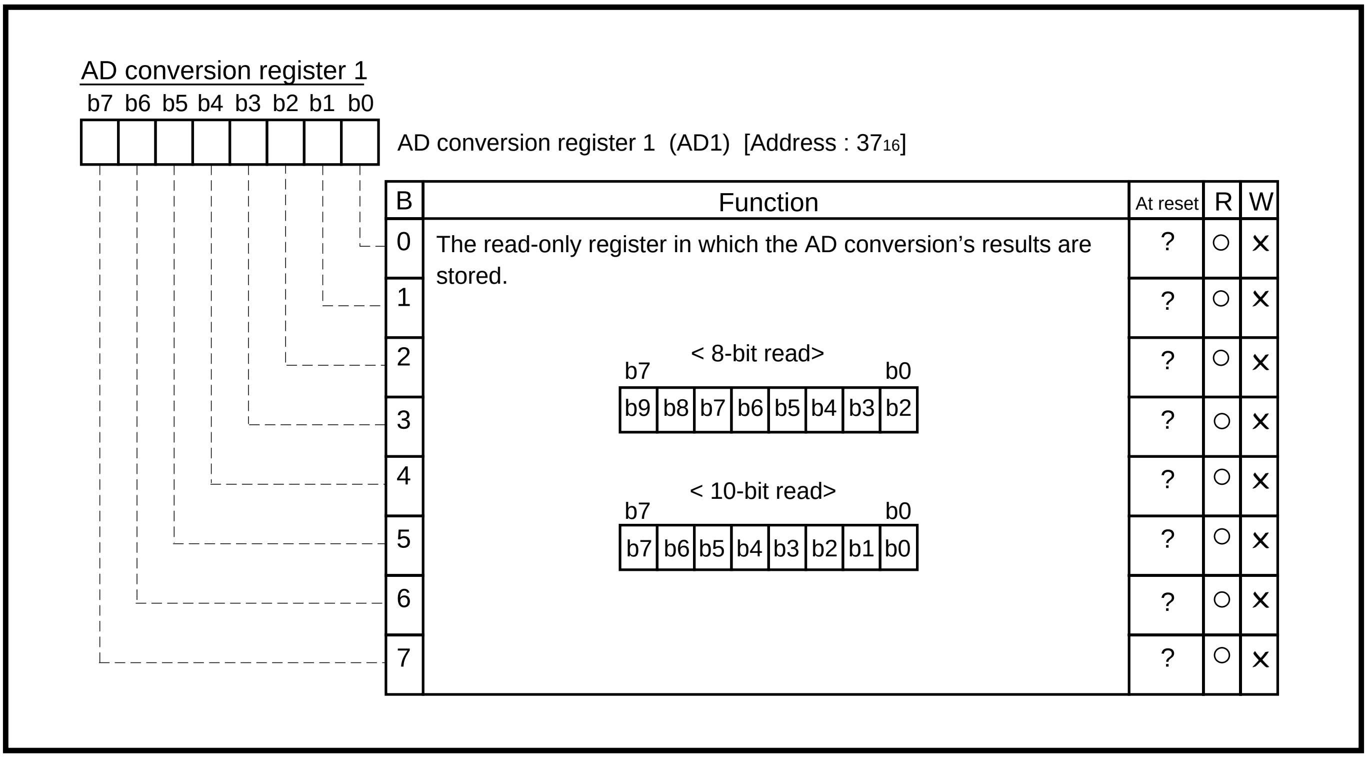
2.8.2 Related registers



AD control register (ADCON) [Address : 36_{16}]

B	Name	Function	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0 : P1$_0$/DQ$_0$/AN$_0$ 0 0 1 : P1$_1$/DQ$_1$/AN$_1$ 0 1 0 : P1$_2$/DQ$_2$/AN$_2$ 0 1 1 : P1$_3$/DQ$_3$/AN$_3$ 1 0 0 : P1$_4$/DQ$_4$/AN$_4$ 1 0 1 : P1$_5$/DQ$_5$/AN$_5$ 1 1 0 : P1$_6$/DQ$_6$/AN$_6$ 1 1 1 : P1$_7$/DQ$_7$/AN$_7$	0	○	○
1					
2					
3	AD conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	○	○
4	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are indefinite.		?	○	×
5			?	○	×
6			?	○	×
7			?	○	×

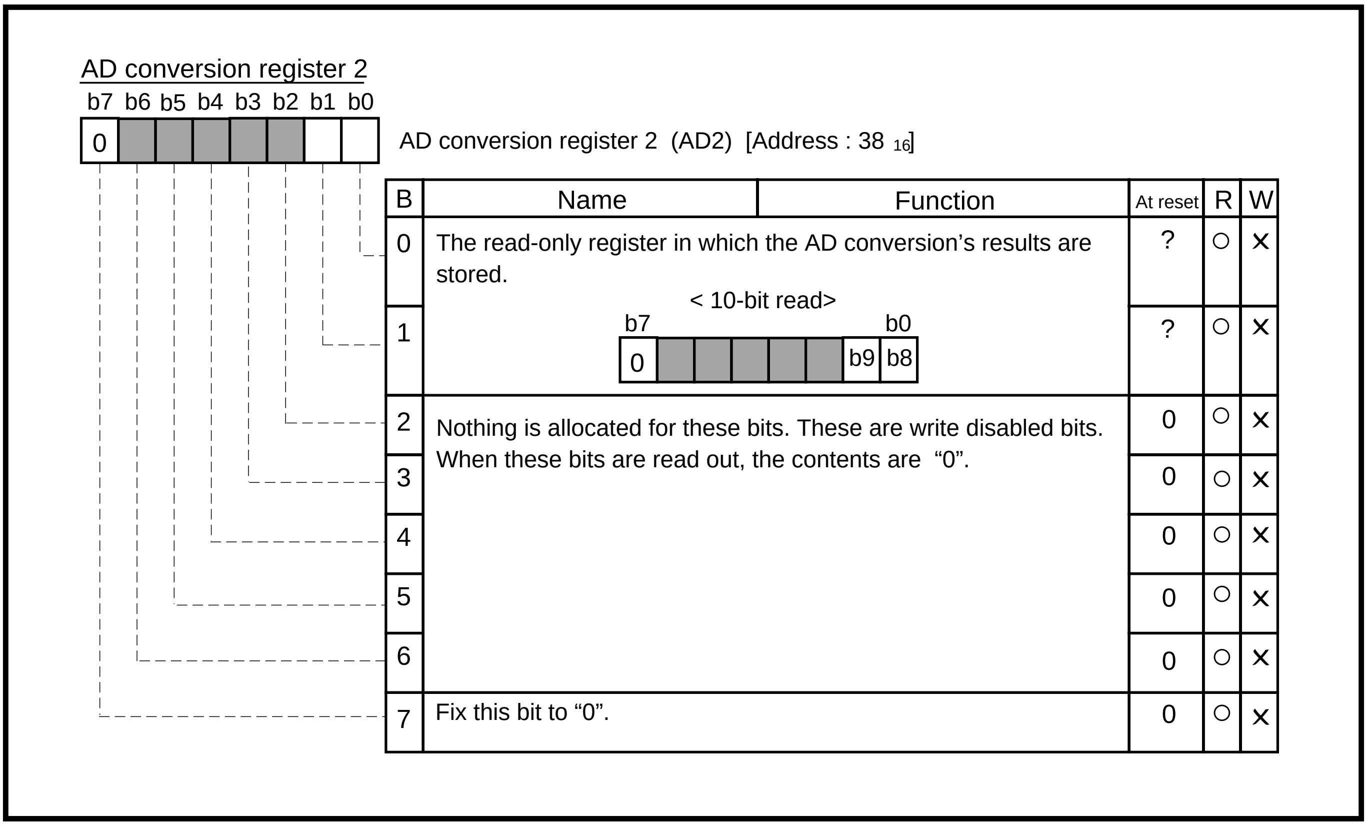
Fig. 2.8.2 Structure of AD control register



AD conversion register 1 (AD1) [Address : 37₁₆]

B	Function	At reset	R	W
0	The read-only register in which the AD conversion's results are stored. < 8-bit read> b7 b0 b9 b8 b7 b6 b5 b4 b3 b2 < 10-bit read> b7 b0 b7 b6 b5 b4 b3 b2 b1 b0	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

Fig. 2.8.3 Structure of AD conversion register 1



AD conversion register 2 (AD2) [Address : 38 16]

B	Name	Function	At reset	R	W
0		The read-only register in which the AD conversion's results are stored. < 10-bit read> b7 b0 0 ▨ ▨ ▨ ▨ ▨ b9 b8	?	○	×
1			?	○	×
2		Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the contents are "0".	0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7		Fix this bit to "0".	0	○	×

Fig. 2.8.4 Structure of AD conversion register 2

Interrupt request register 2

b7 b6 b5 b4 b3 b2 b1 b0

Interrupt request register 2 (IREQ2)
[Address : 3D$_{16}$]

B	Name	Function	At reset	R	W
0	INT$_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB HUB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	CNTR$_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A/D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

✻ “0” can be set by software, but “1” cannot be set.

Fig. 2.8.5 Structure of Interrupt request register 2

Interrupt control register 2

b7 b6 b5 b4 b3 b2 b1 b0
0

Interrupt control register 2 (ICON2)
[Address : 3F$_{16}$]

B	Name	Function	At reset	R	W
0	INT$_1$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB HUB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	CNTR$_0$ interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A/D conversion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 2.8.6 Structure of Interrupt control register 2

2.8.3 A/D converter application examples

(1) Conversion of analog input voltage

Outline : The analog input voltage input from a sensor is converted to digital values.

Figure 2.8.7 shows a connection diagram, and Figure 2.8.8 shows the related registers setting.

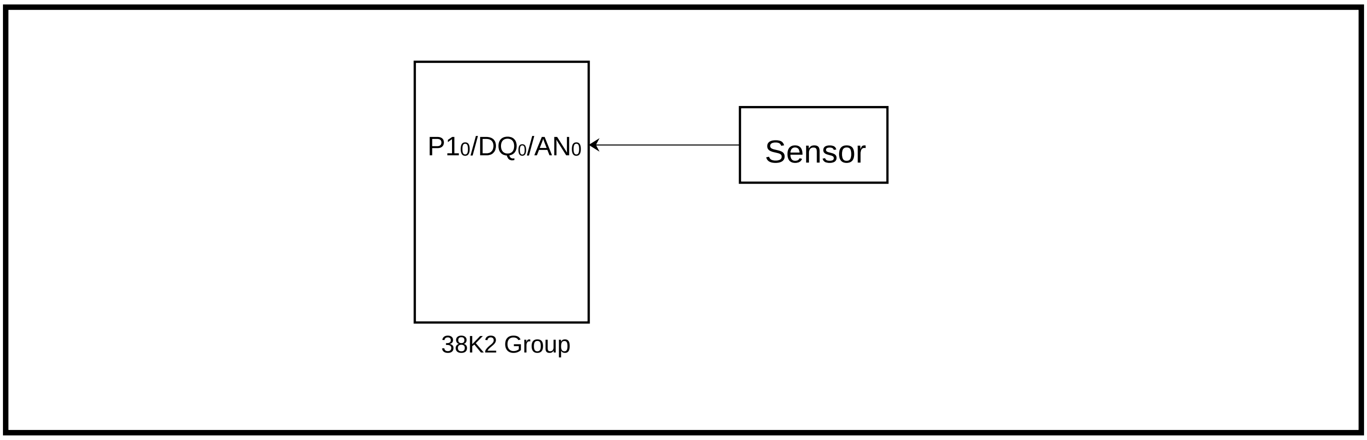
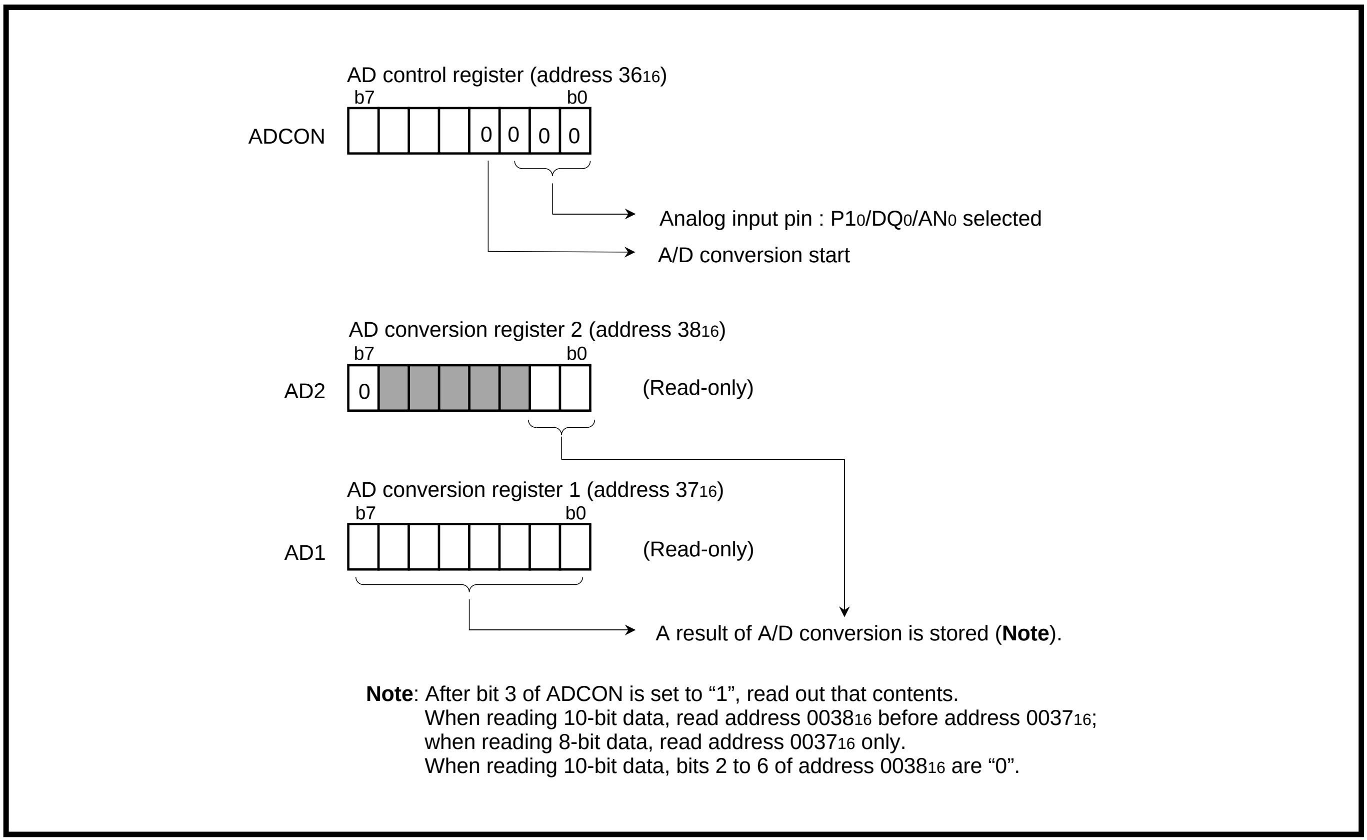


Fig. 2.8.7 Connection diagram

Specifications : •The analog input voltage input from a sensor is converted to digital values.
•P1₀/DQ₀/AN₀ pin is used as an analog input pin.


AD control register (address 36₁₆)
b7
b0
ADCON
0
0
0
0
Analog input pin : P1₀/DQ₀/AN₀ selected
A/D conversion start
AD conversion register 2 (address 38₁₆)
b7
b0
AD2
0
(Read-only)
AD conversion register 1 (address 37₁₆)
b7
b0
AD1
(Read-only)
A result of A/D conversion is stored (Note).

Note: After bit 3 of ADCON is set to “1”, read out that contents.
When reading 10-bit data, read address 0038₁₆ before address 0037₁₆;
when reading 8-bit data, read address 0037₁₆ only.
When reading 10-bit data, bits 2 to 6 of address 0038₁₆ are “0”.

Fig. 2.8.8 Related registers setting

An analog input signal from a sensor is converted to the digital value according to the related registers setting shown by Figure 2.8.8. Figure 2.8.9 shows the control procedure for 8-bit read, and Figure 2.8.10 shows the control procedure for 10-bit read.

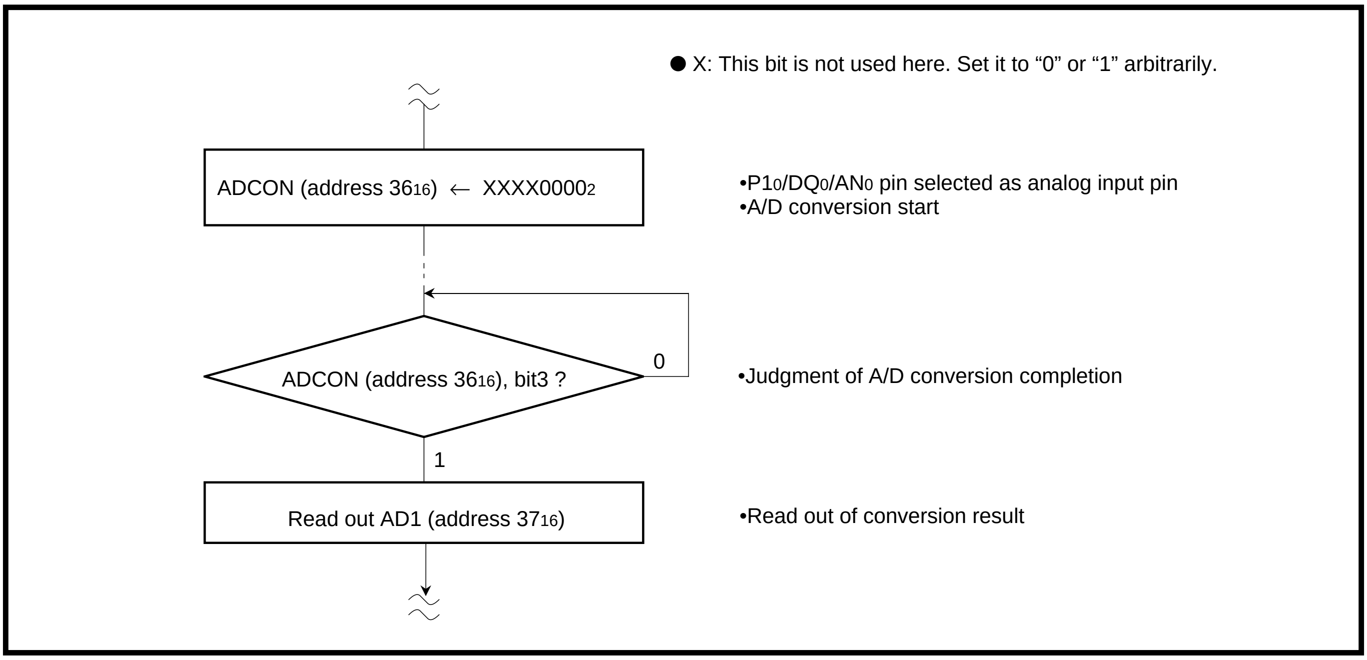

● X: This bit is not used here. Set it to “0” or “1” arbitrarily.
ADCON (address 3616) ← XXXX00002
•P10/DQ0/AN0 pin selected as analog input pin
•A/D conversion start
ADCON (address 3616), bit3 ?
0
1
•Judgment of A/D conversion completion
Read out AD1 (address 3716)
•Read out of conversion result

Fig. 2.8.9 Control procedure for 8-bit read

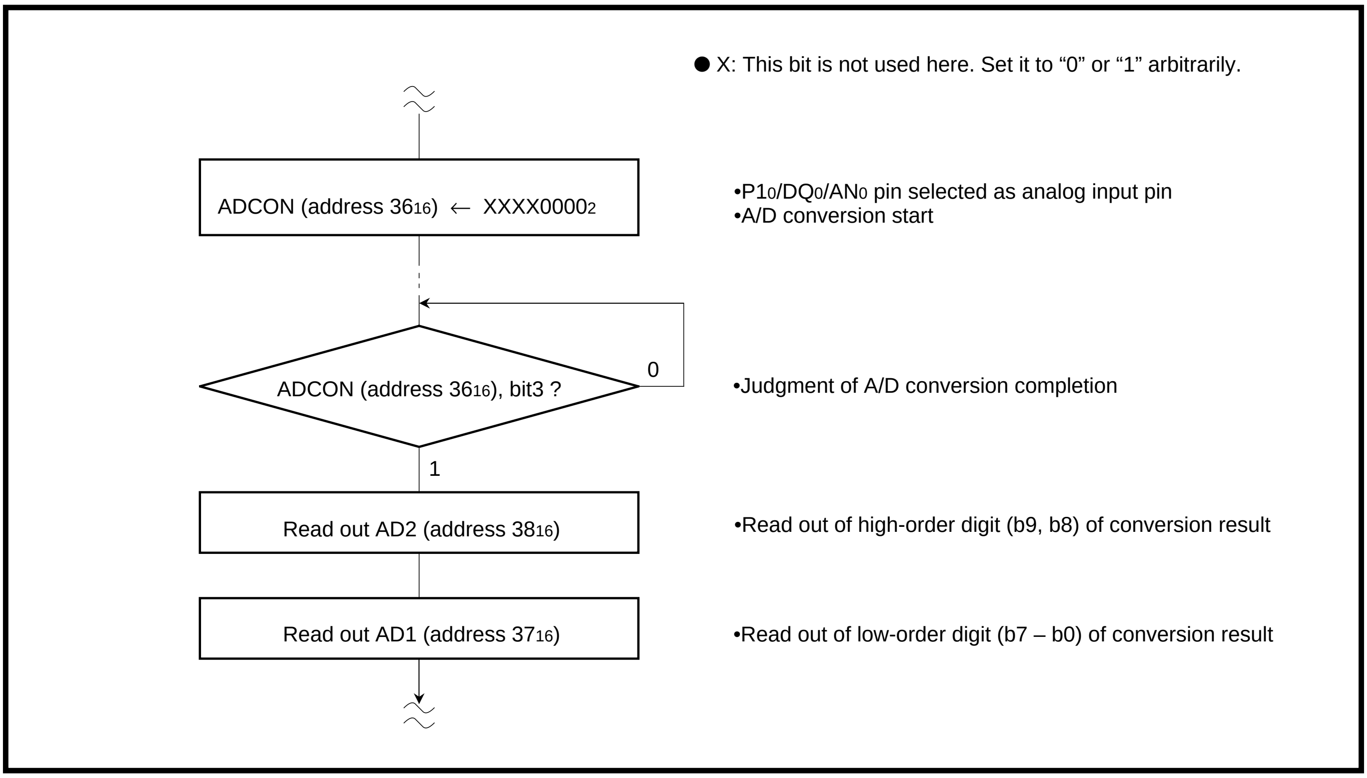

● X: This bit is not used here. Set it to “0” or “1” arbitrarily.
ADCON (address 3616) ← XXXX00002
•P10/DQ0/AN0 pin selected as analog input pin
•A/D conversion start
ADCON (address 3616), bit3 ?
0
1
•Judgment of A/D conversion completion
Read out AD2 (address 3816)
•Read out of high-order digit (b9, b8) of conversion result
Read out AD1 (address 3716)
•Read out of low-order digit (b7 – b0) of conversion result

Fig. 2.8.10 Control procedure for 10-bit read

2.8.4 Notes on A/D converter

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 μF to 1 μF. Further, be sure to verify the operation of application products on the user side.

● **Reason**

An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A/D conversion precision to be worse.

(2) Clock frequency during A/D conversion

The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A/D conversion.

- $f(X_{IN})$ is 500 kHz or more
- Do not execute the **STP** instruction

2.9 Watchdog timer

This paragraph explains the registers setting method and the notes related to the watchdog timer.

2.9.1 Memory map

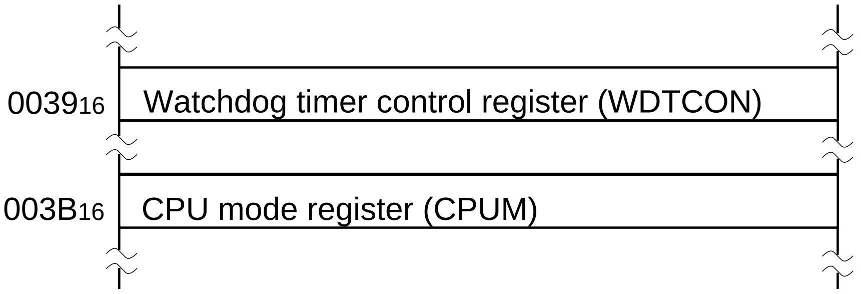
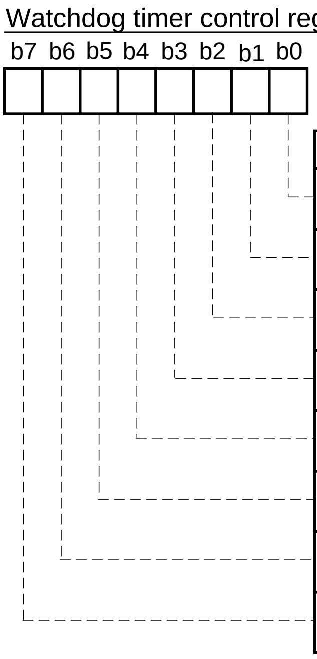


Fig. 2.9.1 Memory map of registers related to watchdog timer

2.9.2 Related registers

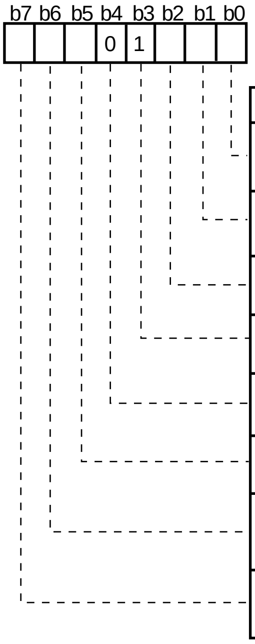


Watchdog timer control register (WDTCON) [Address : 39_{16}]

B	Name	Function	At reset	R	W
0	Watchdog timer H (for read-out of high-order 6 bits)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0 : STP instruction enabled 1 : STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0 : Watchdog timer L underflow 1 : System clock/16	0	○	○

Fig. 2.9.2 Structure of Watchdog timer control register

CPU mode register



CPU mode register
(CPUM: address $3B_{16}$)

B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Not available 1 0 : Not available 1 1 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Fix this bit to “0”.		0	○	○
5	System clock selection bit	0 : Main clock f(X_{IN}) 1 : f_{SYN}	0	○	○
6	System clock division ratio selection bits	b7 b6 0 0 : ϕ = f(system clock)/8 (8-divide mode) 0 1 : ϕ = f(system clock)/4 (4-divide mode) 1 0 : ϕ = f(system clock)/2 (2-divide mode) 1 1 : ϕ = f(system clock) (Through mode)	0	○	○
7					

*: The initial value of bit 1 depends on the CNVss level.

Fig. 2.9.3 Structure of CPU mode register

2.9.3 Watchdog timer application examples

(1) Detection of program runaway

Outline: If program runaway occurs, let the microcomputer reset, using the internal timer for detection of program runaway.

Specifications:
- •An underflow of watchdog timer H is judged to be program runaway, and the microcomputer is returned to the reset status.
- •Before the watchdog timer H underflows, “0” is set into bit 7 of the watchdog timer control register at every cycle in a main routine.
- •Through mode is used as a system clock division ratio.
- •An underflow signal of the watchdog timer L is supplied as the count source of watchdog timer H.

Figure 2.9.4 shows a watchdog timer connection and division ratio setting; Figure 2.9.5 shows the related registers setting; Figure 2.9.6 shows the control procedure.

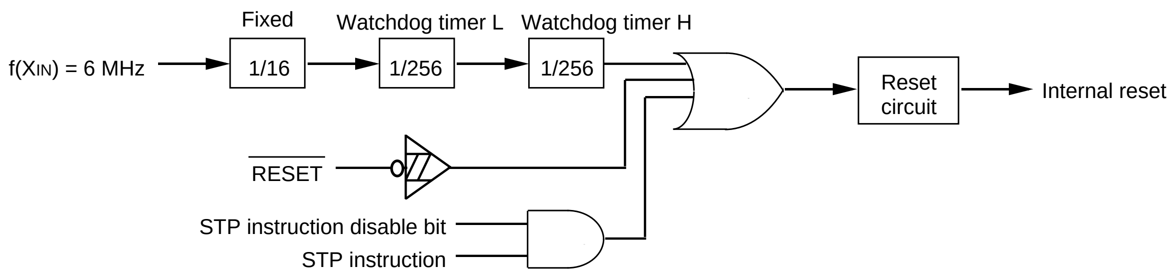


Fig. 2.9.4 Watchdog timer connection and division ratio setting

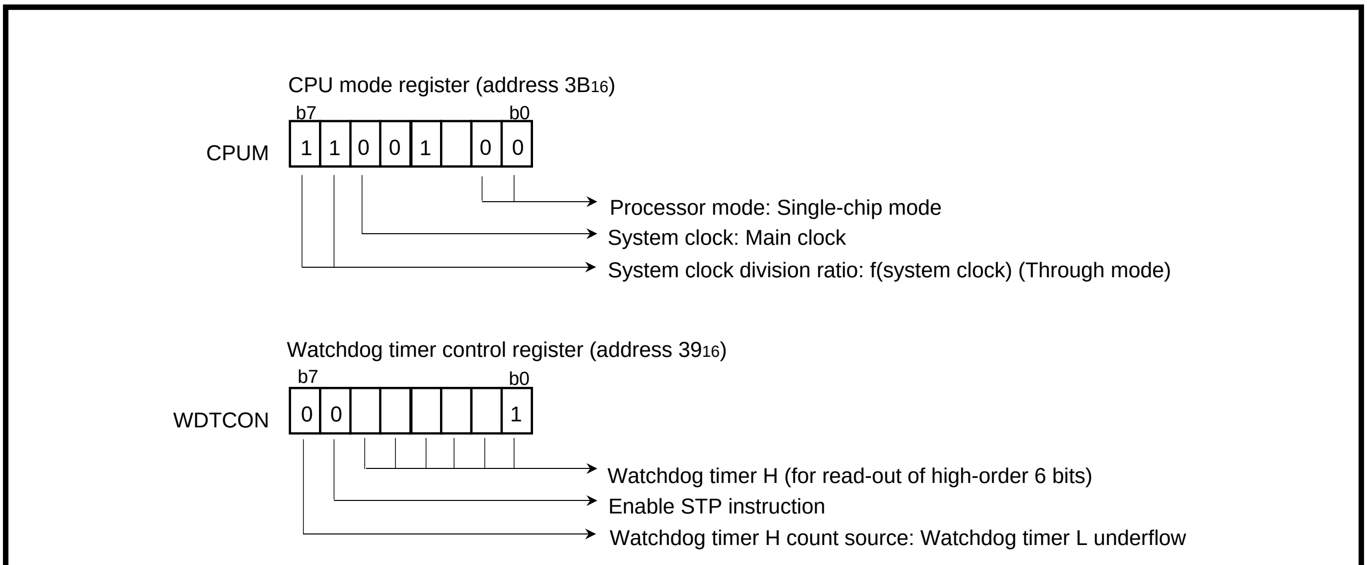

CPU mode register (address 3B16)
b7
b0
CPUM
1 1 0 0 1 0 0
Processor mode: Single-chip mode
System clock: Main clock
System clock division ratio: f(system clock) (Through mode)
Watchdog timer control register (address 3916)
b7
b0
WDTCON
0 0 1
Watchdog timer H (for read-out of high-order 6 bits)
Enable STP instruction
Watchdog timer H count source: Watchdog timer L underflow

Fig. 2.9.5 Related registers setting

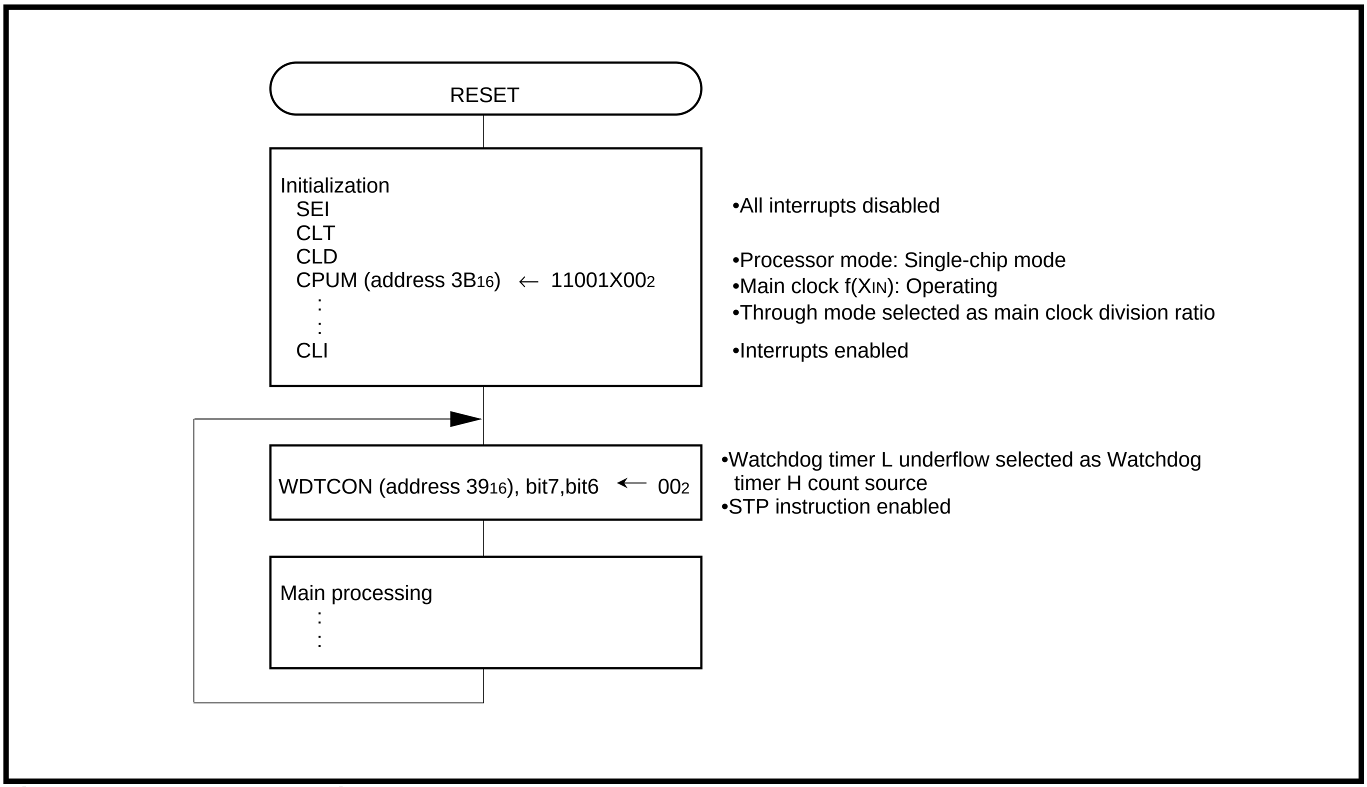

RESET
Initialization
SEI
CLT
CLD
CPUM (address 3B16) ← 11001X002
:
:
CLI
•All interrupts disabled
•Processor mode: Single-chip mode
•Main clock f(XIN): Operating
•Through mode selected as main clock division ratio
•Interrupts enabled
WDTCON (address 3916), bit7,bit6 ← 002
•Watchdog timer L underflow selected as Watchdog timer H count source
•STP instruction enabled
Main processing
:
:

Fig. 2.9.6 Control procedure

2.9.4 Notes on watchdog timer

- ●Make sure that the watchdog timer does not underflow while waiting Stop release, because the watchdog timer keeps counting during that term.
- ●When the STP instruction disable bit has been set to “1”, it is impossible to switch it to “0” by a program.

2.10 Reset

2.10.1 Connection example of reset IC

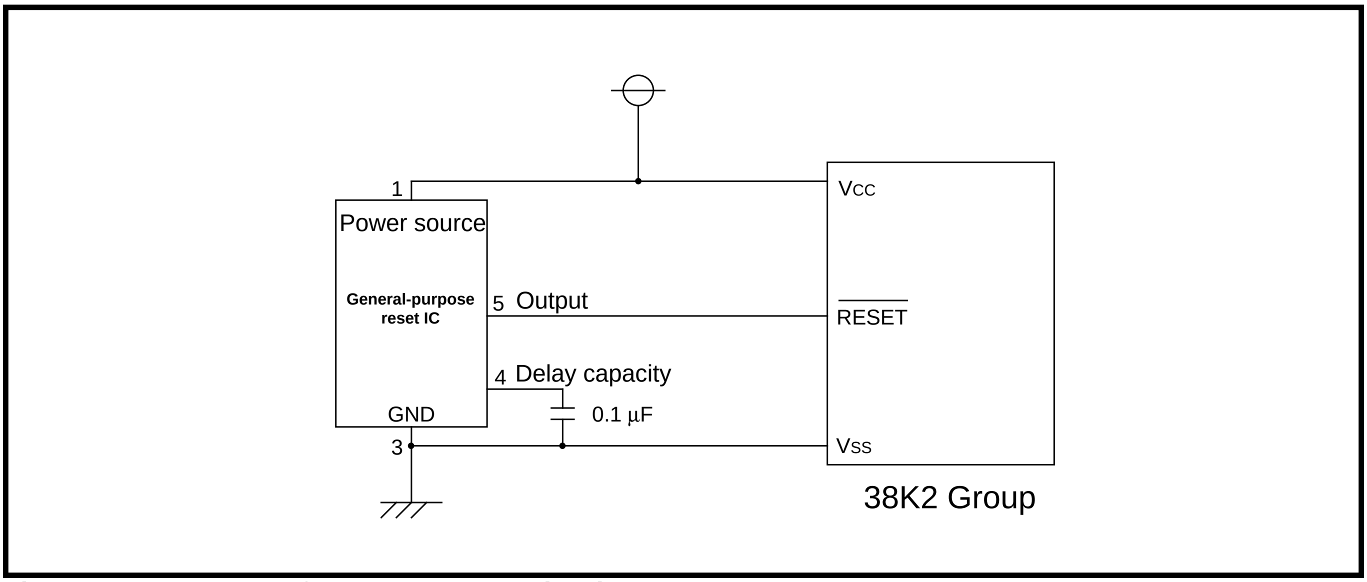


Fig. 2.10.1 Example of poweron reset circuit

Figure 2.10.2 shows the system example which switches to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

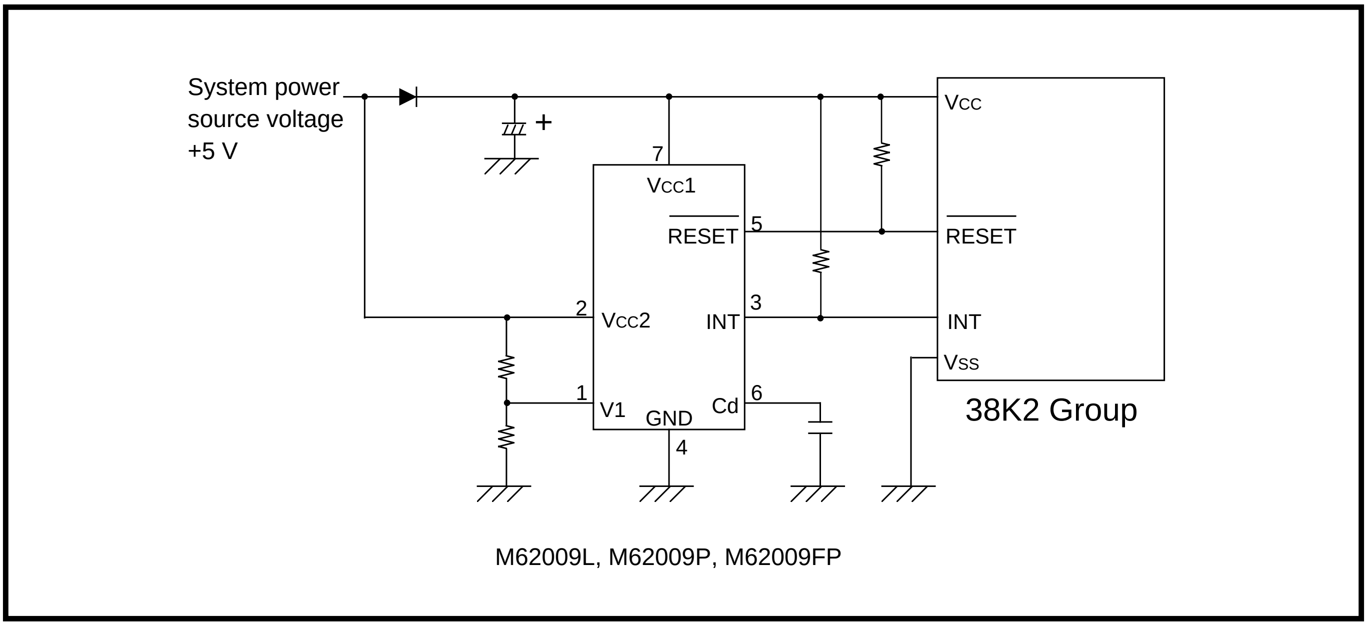


Fig. 2.10.2 RAM backup system

2.10.2 Notes on $\overline{\text{RESET}}$ pin

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the Vss pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

● **Reason**

If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, it may cause a microcomputer failure.

2.11 Frequency synthesizer (PLL)

This paragraph explains the registers setting method and the notes related to the frequency synthesizer (PLL circuit).

2.11.1 Memory map

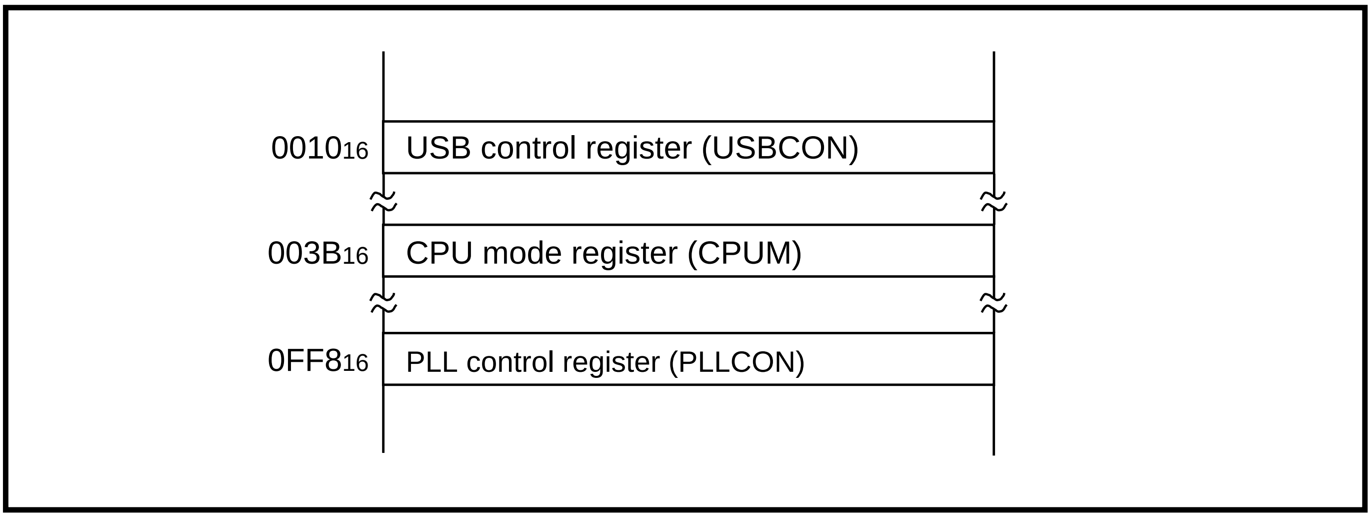
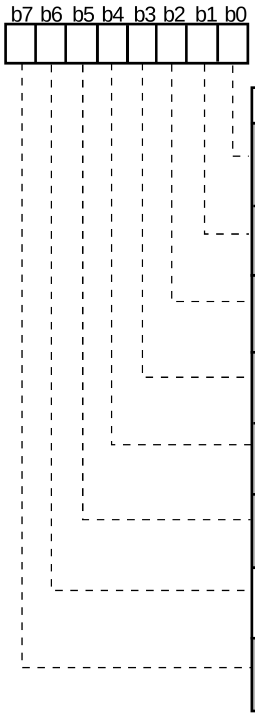


Fig. 2.11.1 Memory map of registers related to PLL

2.11.2 Related registers

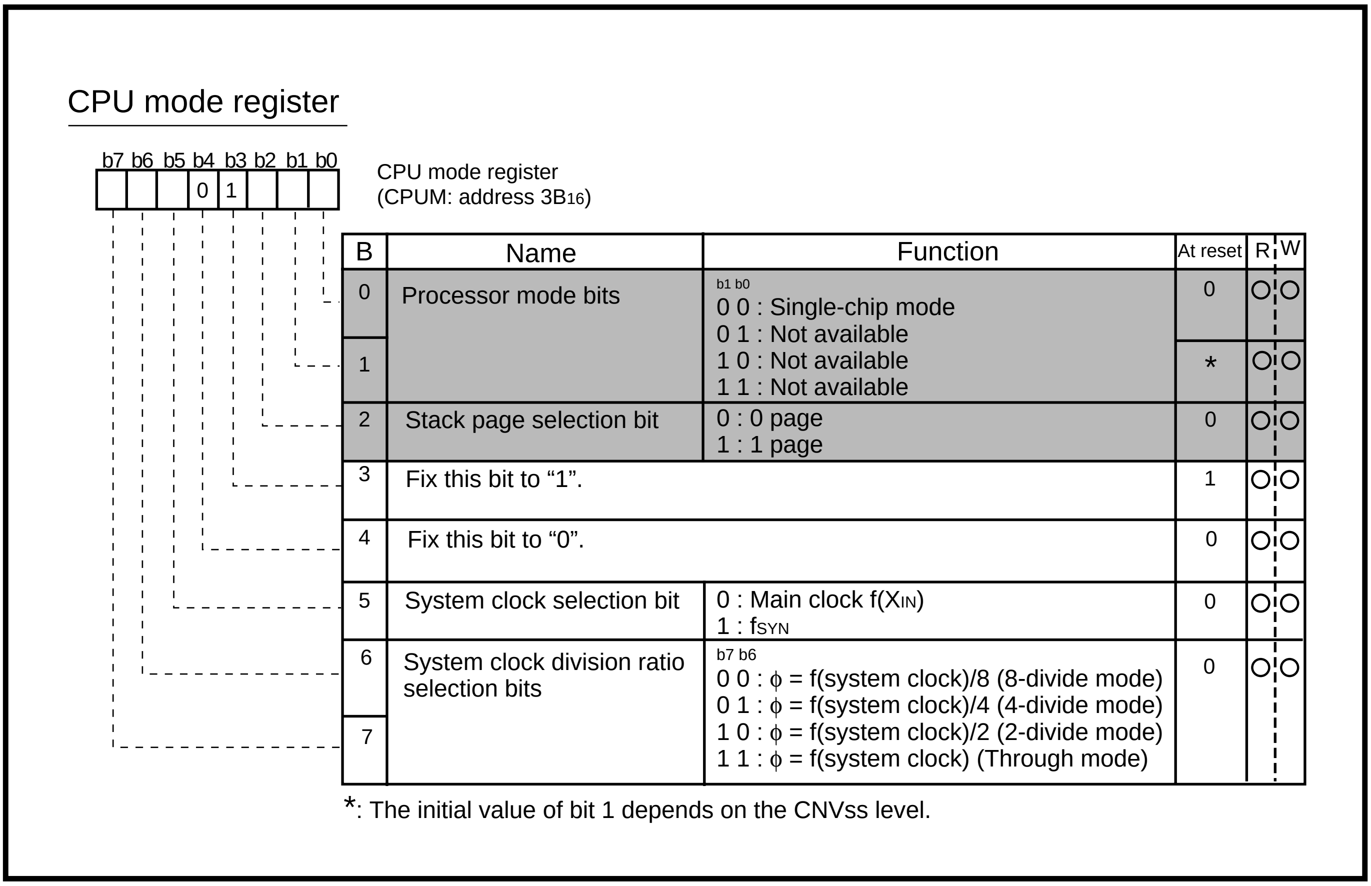
USB control register



USB control register (USBCON)
[Address 10$_{16}$]

B	Name	Function	At reset	R	W
0	Remote wakeup bit	0 : Returning to BUS idle state by writing "1" first and then "0". (Remote wakeup signal) 1 : K-state output	0	○	○
1	TrON output control bit	0 : "L" output mode (valid in TRONE = "1") 1 : "H" output mode (valid in TRONE = "1")	0	○	○
2	TrON output enable bit	0 : TrON port output disabled (Hi-Z state) 1 : TrON port output enabled	0	○	○
3	USB reference voltage control bit	0 : Normal mode (valid in VREFE = "1") 1 : Low current mode (valid in VREFE = "1")	0	○	○
4	USB reference voltage enable bit	0 : USB reference voltage circuit operation disabled 1 : USB reference voltage circuit operation enabled	0	○	○
5	USB difference input enable bit	0 : Upstream-port difference input circuit operation disabled 1 : Upstream--port difference input circuit operation enabled	0	○	○
6	USB clock select bit	0 : External oscillating clock f(XIN) 1 : PLL circuit output clock fVCO	0	○	○
7	USB module operation enable bit	0 : USB module reset 1 : USB module operation enabled	0	○	○

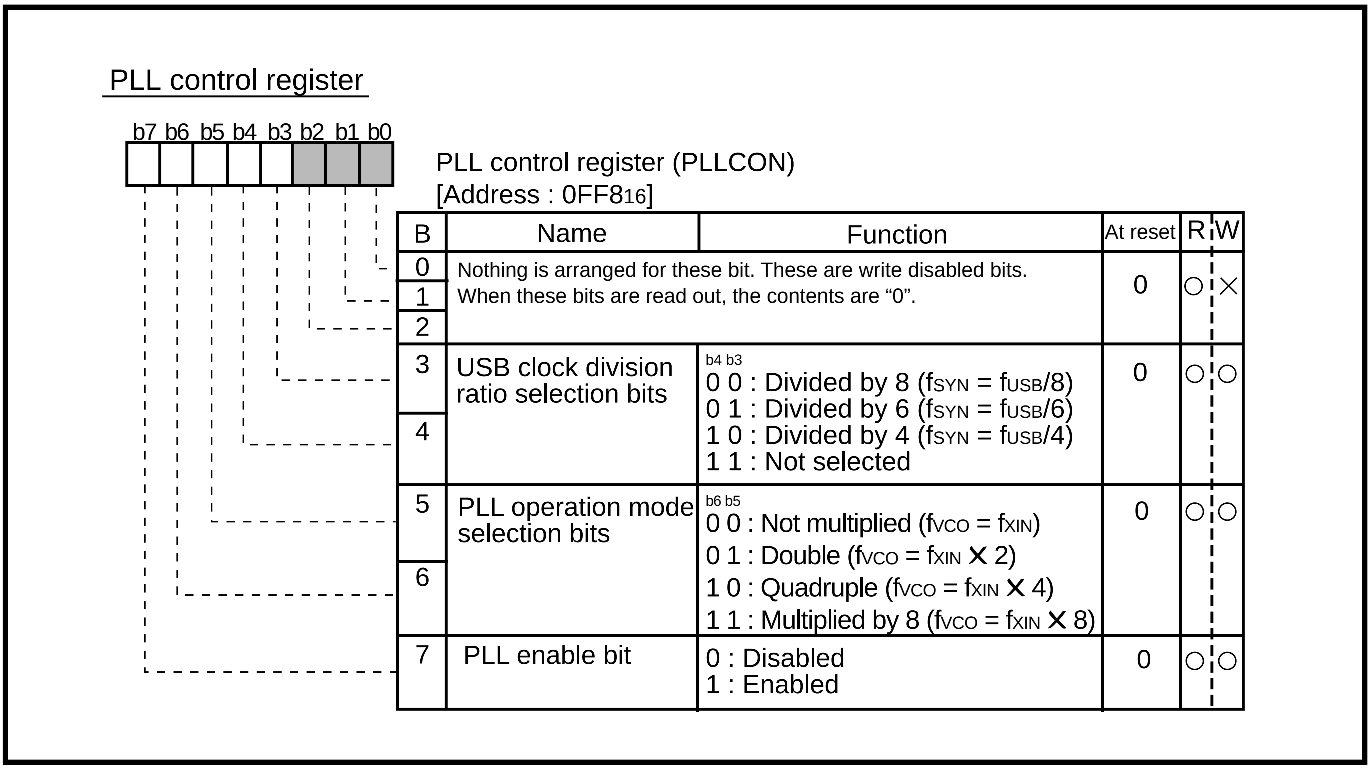
Fig. 2.11.2 Structure of USB control register



B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Not available 1 0 : Not available 1 1 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Fix this bit to “0”.		0	○	○
5	System clock selection bit	0 : Main clock $f(X_{IN})$ 1 : f_{SYN}	0	○	○
6	System clock division ratio selection bits	b7 b6 0 0 : ϕ = f(system clock)/8 (8-divide mode) 0 1 : ϕ = f(system clock)/4 (4-divide mode) 1 0 : ϕ = f(system clock)/2 (2-divide mode) 1 1 : ϕ = f(system clock) (Through mode)	0	○	○
7					

*: The initial value of bit 1 depends on the CNVss level.

Fig. 2.11.3 Structure of CPU mode register



B	Name	Function	At reset	R	W
0	Nothing is arranged for these bit. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
1					
2					
3	USB clock division ratio selection bits	b4 b3 0 0 : Divided by 8 ($f_{SYN} = f_{USB}/8$) 0 1 : Divided by 6 ($f_{SYN} = f_{USB}/6$) 1 0 : Divided by 4 ($f_{SYN} = f_{USB}/4$) 1 1 : Not selected	0	○	○
4					
5	PLL operation mode selection bits	b6 b5 0 0 : Not multiplied ($f_{VCO} = f_{XIN}$) 0 1 : Double ($f_{VCO} = f_{XIN} \times 2$) 1 0 : Quadruple ($f_{VCO} = f_{XIN} \times 4$) 1 1 : Multiplied by 8 ($f_{VCO} = f_{XIN} \times 8$)	0	○	○
6					
7	PLL enable bit	0 : Disabled 1 : Enabled	0	○	○

Fig. 2.11.4 Structure of PLL control register

2.11.3 Functional description

The frequency synthesizer generates the 48 MHz clock which is multiples of the external input reference $f(X_{IN})$ and is needed for operating USB function. When using the USB function, set PLL enable bit of PLL control register (PLLCON: address $0FF8_{16}$) to “1” (enabled) to send the 48 MHz PLL output clock (f_{VCO}) into USB function control unit. Figure 2.11.5 shows the block diagram for the frequency synthesizer circuit.

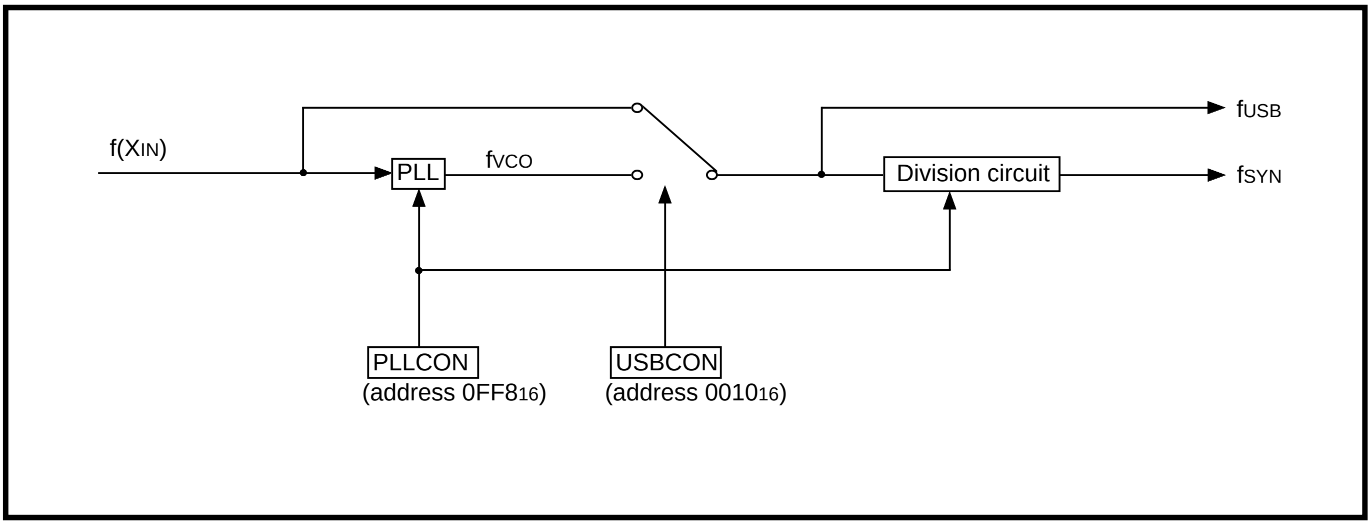


Fig. 2.11.5 Block diagram for frequency synthesizer circuit

- **f_{VCO} (PLL output clock)**

f_{VCO} is generated by multiplying PLL input clock according to the contents of PLL operation mode selection bits (bits 6, 5 of PLLCON), where
$f_{VCO} = f(X_{IN}) \times n$, n:value selected by PLL operation mode selection bits

Set PLL operation mode selection bits so that f_{VCO} may be set to 48 MHz.
While the PLL enable bit is “0” (disabled), f_{VCO} retains “L” level (except when PLL operation mode selection bits are set to “00_2”).
Table 2.11.1 shows the example of PLL operation mode selection bits setting.

Table 2.11.1 PLL operation mode selection bits setting example

$f(X_{IN})$	PLL operation mode selection bits *	f_{VCO}
6 MHz	11	48 MHz
12 MHz	10	48 MHz

***:** PLL control register (bits 6,5)

Furthermore, when PLL operation mode selection bits are set to “00_2”, the clock input into PLL is used as f_{VCO}, which is not multiplied, regardless of PLL operation enabled or disabled.

- **f_{USB} (USB clock)**

Either $f(X_{IN})$ (main clock) or f_{VCO} (PLL output clock) can be selected for f_{USB} by USB clock select bit of USB control register (bit6 of USBCON: address 0010_{16}), and it is supplied to the USB function control circuit. When supplying f_{VCO} to the USB function control circuit, after setting PLL enable bit to “1” (enabled) and then set USB clock select bit to “1” (USB clock).

- **f_{SYN} (f_{USB} division clock)**

According to the setting of the USB clock division ratio selection bits (bits 4, 3 of PLLCON), the division clock of f_{USB} is supplied to f_{SYN}.

f_{SYN} =f_{USB} / m, m:value selected by USB clock division ratio selection bits

Set the USB clock division ratio selection bits so that f_{SYN} may be set to 6 MHz, 8 MHz or 12 MHz. When using f_{SYN} as internal system clock, set the system clock selection bit of CPU mode register (bit 5 of CPUM: address $003B_{16}$) to "1" (f_{SYN}).

Table 2.11.2 shows the example of USB clock division ratio selection bits setting.

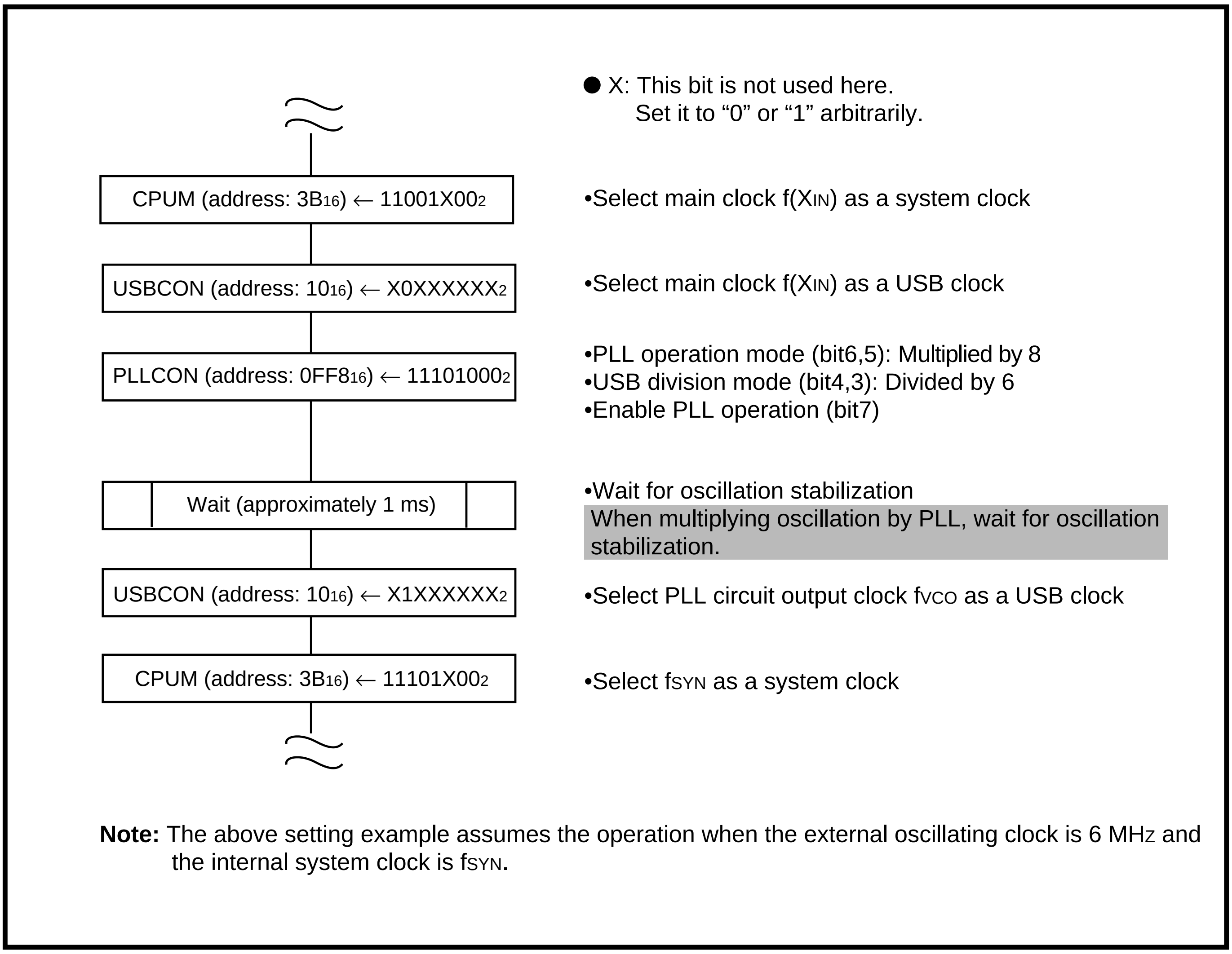
Table 2.11.2 USB clock division ratio selection bits setting example

f_{USB}	USB clock division ratio selection bits *	f_{SYN}
48 MHz	00	6 MHz
	01	8 MHz
	10	12 MHz

***:** PLL control register (bit4,3)

- **Setting for starting up PLL circuit when hardware reset**

Figure 2.11.6 shows the example of related registers setting.



● X: This bit is not used here.
Set it to "0" or "1" arbitrarily.

CPUM (address: $3B_{16}$) ← $11001X00_2$ — •Select main clock f(X_{IN}) as a system clock

USBCON (address: 10_{16}) ← $X0XXXXXX_2$ — •Select main clock f(X_{IN}) as a USB clock

PLLCON (address: $0FF8_{16}$) ← 11101000_2 — •PLL operation mode (bit6,5): Multiplied by 8
•USB division mode (bit4,3): Divided by 6
•Enable PLL operation (bit7)

Wait (approximately 1 ms) — •Wait for oscillation stabilization
When multiplying oscillation by PLL, wait for oscillation stabilization.

USBCON (address: 10_{16}) ← $X1XXXXXX_2$ — •Select PLL circuit output clock f_{VCO} as a USB clock

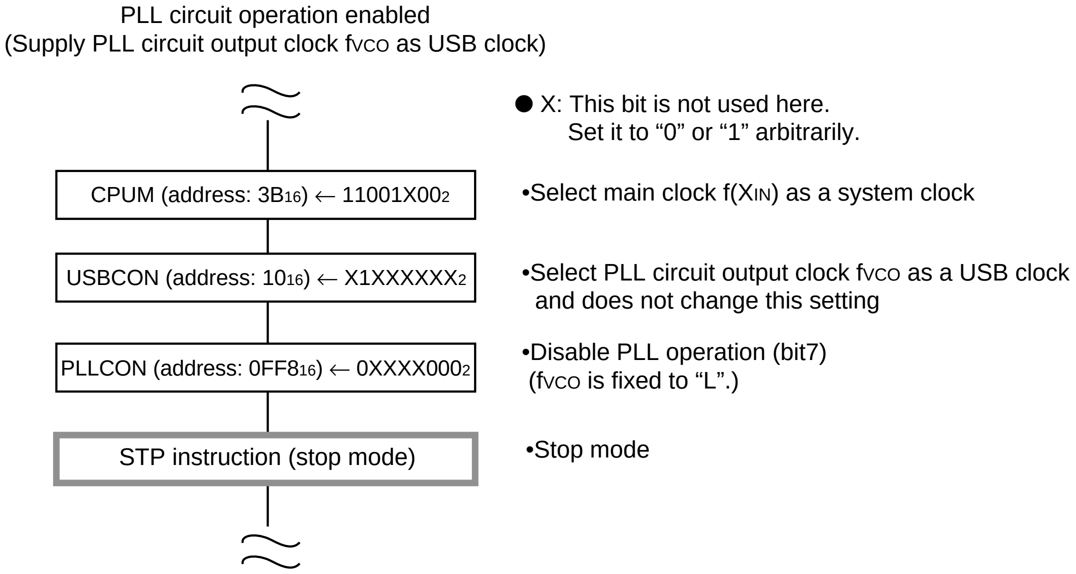
CPUM (address: $3B_{16}$) ← $11101X00_2$ — •Select f_{SYN} as a system clock

Note: The above setting example assumes the operation when the external oscillating clock is 6 MHz and the internal system clock is f_{SYN}.

Fig. 2.11.6 Related registers setting when hardware reset

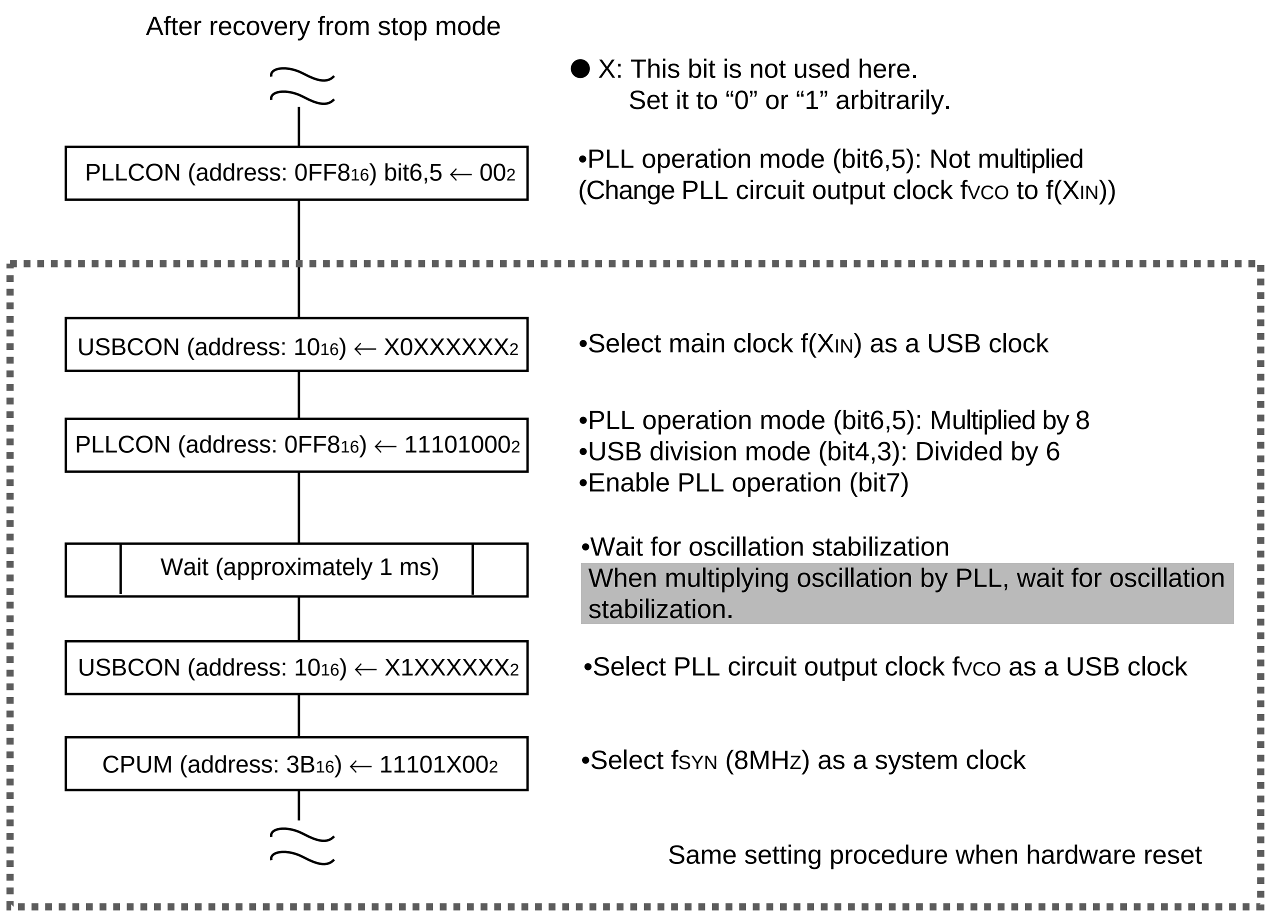
● Procedure for stop and return of PLL circuit when stop mode

Figure 2.11.7 shows the stop procedure of PLL circuit, and figure 2.11.8 shows the return procedure of PLL circuit.


PLL circuit operation enabled
(Supply PLL circuit output clock f_{VCO} as USB clock)
● X: This bit is not used here.
Set it to “0” or “1” arbitrarily.
CPUM (address: $3B_{16}$) ← $11001X00_2$
•Select main clock $f(X_{IN})$ as a system clock
USBCON (address: 10_{16}) ← $X1XXXXXX_2$
•Select PLL circuit output clock f_{VCO} as a USB clock and does not change this setting
PLLCON (address: $0FF8_{16}$) ← $0XXXX000_2$
•Disable PLL operation (bit7)
(f_{VCO} is fixed to “L”.)
STP instruction (stop mode)
•Stop mode

Note: The above setting example assumes the operation when the external oscillating clock is 6 MHz and the internal system clock is f_{SYN}.

Fig. 2.11.7 Related registers setting when stop mode


After recovery from stop mode
● X: This bit is not used here.
Set it to “0” or “1” arbitrarily.
PLLCON (address: 0FF8$_{16}$) bit6,5 ← 00$_2$
•PLL operation mode (bit6,5): Not multiplied
(Change PLL circuit output clock fVCO to f(XIN))
USBCON (address: 10$_{16}$) ← X0XXXXXX$_2$
•Select main clock f(XIN) as a USB clock
PLLCON (address: 0FF8$_{16}$) ← 11101000$_2$
•PLL operation mode (bit6,5): Multiplied by 8
•USB division mode (bit4,3): Divided by 6
•Enable PLL operation (bit7)
Wait (approximately 1 ms)
•Wait for oscillation stabilization
When multiplying oscillation by PLL, wait for oscillation stabilization.
USBCON (address: 10$_{16}$) ← X1XXXXXX$_2$
•Select PLL circuit output clock fVCO as a USB clock
CPUM (address: 3B$_{16}$) ← 11101X00$_2$
•Select fSYN (8MHz) as a system clock
Same setting procedure when hardware reset

Note: The above setting example assumes the operation when the external oscillating clock is 6 MHz and the internal system clock is f_{SYN}.

Fig. 2.11.8 Related registers setting when recovery from stop mode

2.11.4 Notes on PLL

●6 MHz or 12 MHz external oscillator can be connected as an input reference clock ($f(X_{IN})$). When using the frequency synthesized clock function, we recommend using the fastest frequency possible of $f(X_{IN})$ as an input clock reference for the PLL.

●When enabling PLL operation from PLL disabled status (disabled when reset), set the USB clock select bit of USBCON to “0” ($f(X_{IN})$) to operate with the main clock ($f(X_{IN})$).

●When supplying f_{VCO} to the USB block after setting PLL operation enable bit to “1” (PLL enabled), wait for the oscillation stable time (1 ms or less) of PLL to avoid any instability caused by the clock, then set USB clock select bit to “1” (USB clock).

●When selecting f_{SYN} as an internal system clock, f_{USB} must be 48 MHz.

●When selecting f_{SYN} as an internal system clock, change the system clock selection bit to main clock ($f(X_{IN})$) before executing STP instruction. It is because the following are needed for the low-power consumption:

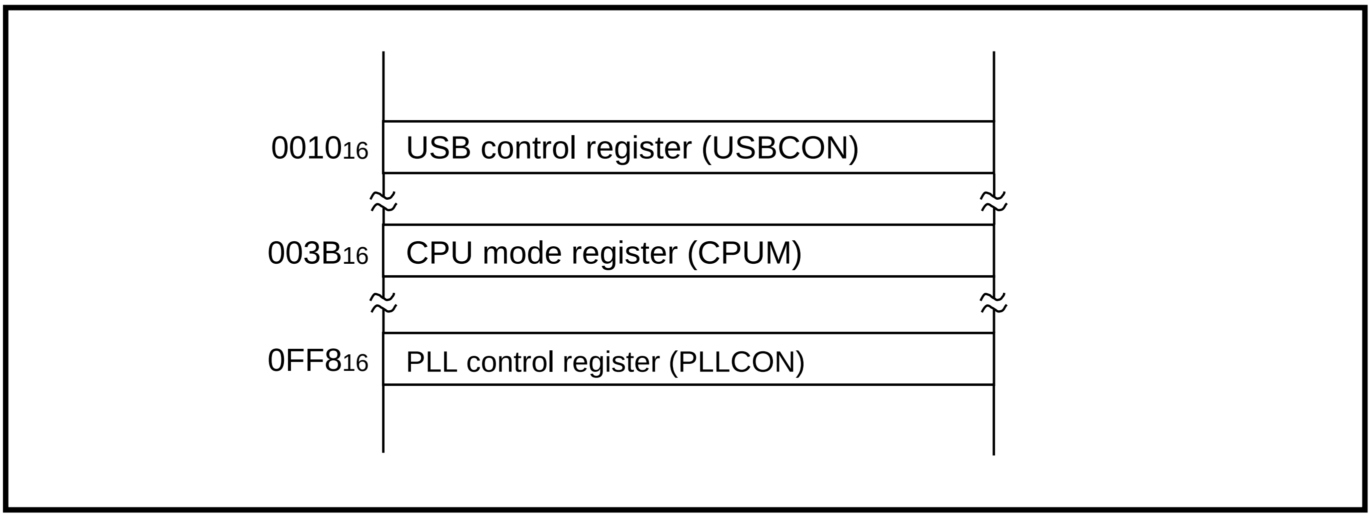
•f_{USB} must be stopped by disabling PLL operation in Stop mode.

•The taimer 1 for waiting oscillation stabilization when returning from Stop mode will require the input count source.

2.12 Clock generating circuit

This paragraph explains the registers setting method and the notes related to the clock generating circuit.

2.12.1 Memory map



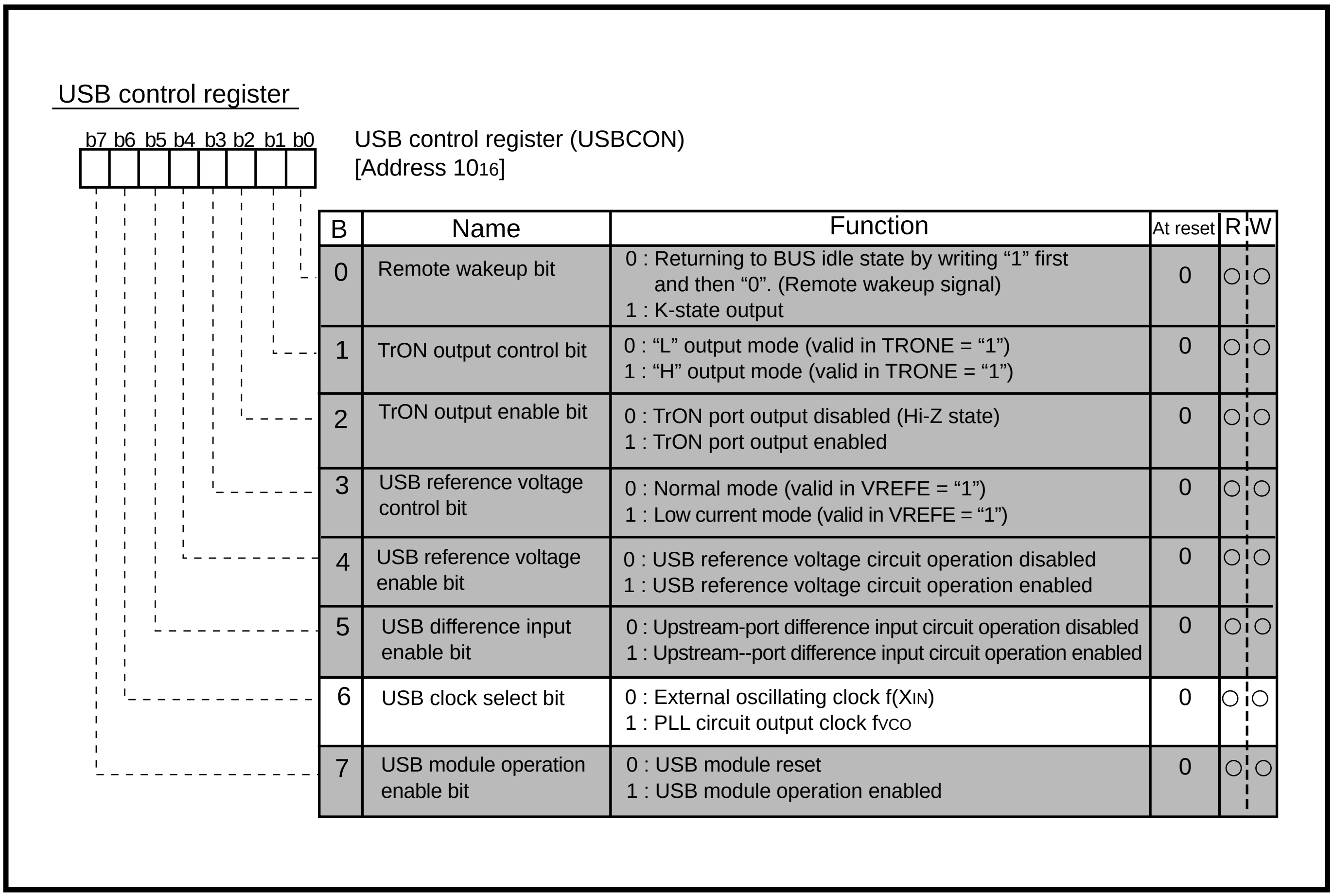
0010_{16} USB control register (USBCON)

$003B_{16}$ CPU mode register (CPUM)

$0FF8_{16}$ PLL control register (PLLCON)

Fig. 2.12.1 Memory map of registers related to clock generating circuit

2.12.2 Related registers

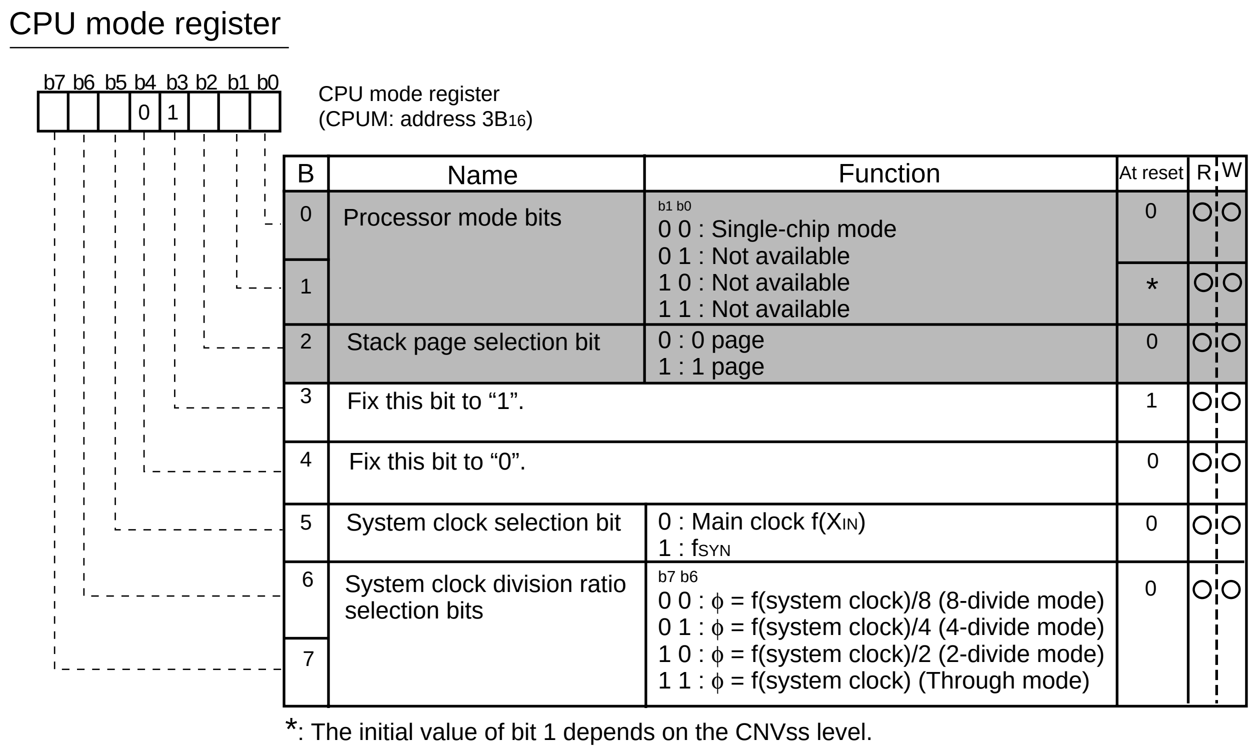


USB control register

USB control register (USBCON)
[Address 10_{16}]

B	Name	Function	At reset	R	W
0	Remote wakeup bit	0 : Returning to BUS idle state by writing “1” first and then “0”. (Remote wakeup signal) 1 : K-state output	0	○	○
1	TrON output control bit	0 : “L” output mode (valid in TRONE = “1”) 1 : “H” output mode (valid in TRONE = “1”)	0	○	○
2	TrON output enable bit	0 : TrON port output disabled (Hi-Z state) 1 : TrON port output enabled	0	○	○
3	USB reference voltage control bit	0 : Normal mode (valid in VREFE = “1”) 1 : Low current mode (valid in VREFE = “1”)	0	○	○
4	USB reference voltage enable bit	0 : USB reference voltage circuit operation disabled 1 : USB reference voltage circuit operation enabled	0	○	○
5	USB difference input enable bit	0 : Upstream-port difference input circuit operation disabled 1 : Upstream--port difference input circuit operation enabled	0	○	○
6	USB clock select bit	0 : External oscillating clock f(X_{IN}) 1 : PLL circuit output clock f_{VCO}	0	○	○
7	USB module operation enable bit	0 : USB module reset 1 : USB module operation enabled	0	○	○

Fig. 2.12.2 Structure of USB control register

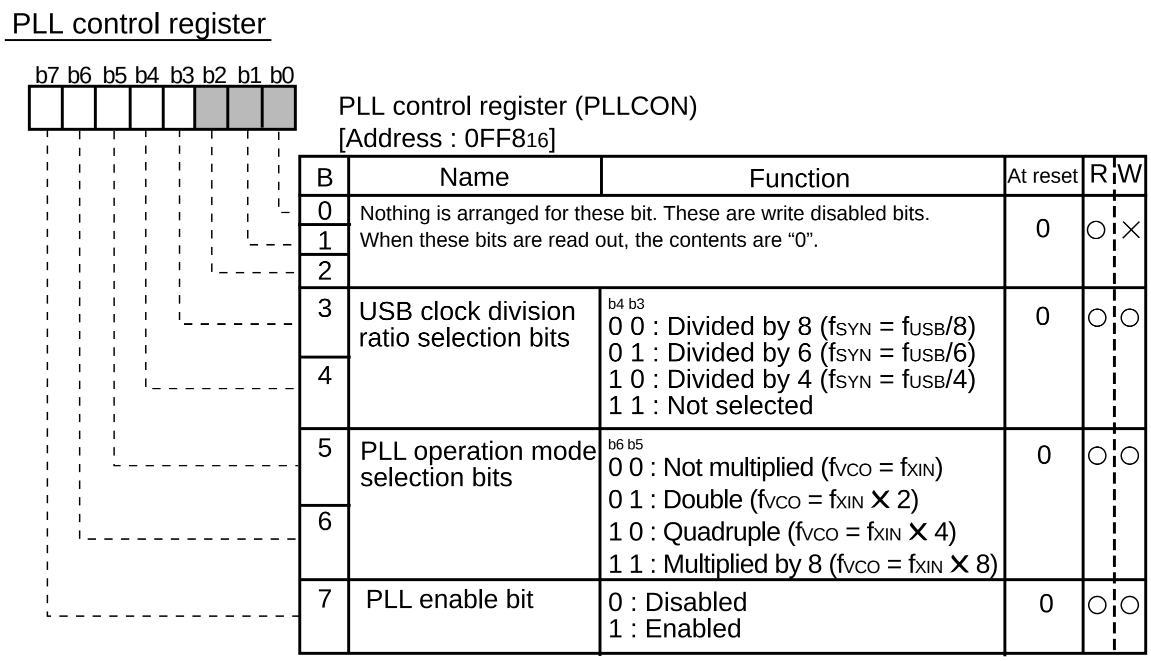

CPU mode register
b7 b6 b5 b4 b3 b2 b1 b0
0
1

CPU mode register
(CPUM: address 3B$_{16}$)

B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Not available 1 0 : Not available 1 1 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Fix this bit to “0”.		0	○	○
5	System clock selection bit	0 : Main clock f(X$_{IN}$) 1 : f$_{SYN}$	0	○	○
6	System clock division ratio selection bits	b7 b6 0 0 : ϕ = f(system clock)/8 (8-divide mode) 0 1 : ϕ = f(system clock)/4 (4-divide mode) 1 0 : ϕ = f(system clock)/2 (2-divide mode) 1 1 : ϕ = f(system clock) (Through mode)	0	○	○
7					

*: The initial value of bit 1 depends on the CNVss level.

Fig. 2.12.3 Structure of CPU mode register


PLL control register
b7 b6 b5 b4 b3 b2 b1 b0

PLL control register (PLLCON)
[Address : 0FF8$_{16}$]

B	Name	Function	At reset	R	W
0	Nothing is arranged for these bit. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
1					
2					
3	USB clock division ratio selection bits	b4 b3 0 0 : Divided by 8 (f$_{SYN}$ = f$_{USB}$/8) 0 1 : Divided by 6 (f$_{SYN}$ = f$_{USB}$/6) 1 0 : Divided by 4 (f$_{SYN}$ = f$_{USB}$/4) 1 1 : Not selected	0	○	○
4					
5	PLL operation mode selection bits	b6 b5 0 0 : Not multiplied (f$_{VCO}$ = f$_{XIN}$) 0 1 : Double (f$_{VCO}$ = f$_{XIN}$ × 2) 1 0 : Quadruple (f$_{VCO}$ = f$_{XIN}$ × 4) 1 1 : Multiplied by 8 (f$_{VCO}$ = f$_{XIN}$ × 8)	0	○	○
6					
7	PLL enable bit	0 : Disabled 1 : Enabled	0	○	○

Fig. 2.12.4 Structure of PLL control register

2.12.3 Oscillation control

Either can be selected as an internal system clock between the following two by system clock selection bit.

- Main clock $f(X_{IN})$
- f_{SYN} (f_{USB} division clock)

Any one can be selected as an internal clock ϕ among the following four by system clock division ratio selection bits.

- $f(X_{IN})$ or $f_{SYN}/8$ (8-divide mode)
- $f(X_{IN})$ or $f_{SYN}/4$ (4-divide mode)
- $f(X_{IN})$ or $f_{SYN}/2$ (2-divide mode)
- $f(X_{IN})$ or f_{SYN} (Through mode)

(1) Generation of internal clock f(ϕ) using main clock $f(X_{IN})$

Table 2.12.1 shows the example of internal clock f(ϕ) generation using main clock $f(X_{IN})$; Figure 2.12.5 shows the related registers setting.

Table 2.12.1 Example of internal clock f(ϕ) generation using main clock $f(X_{IN})$

System clock	System clock division ratio selection bits *	f(ϕ)	Power source voltage V_{CC} [V]
6 MHz	0 0	0.75 MHz	3.00 to 5.25
	0 1	1.5 MHz	
	1 0	3 MHz	
	1 1	6 MHz	
8 MHz	0 0	1 MHz	4.00 to 5.25
	0 1	2 MHz	
	1 0	4 MHz	
	1 1	8 MHz	
12 MHz	0 0	1.5 MHz	
	0 1	3 MHz	
	1 0	6 MHz	

***:** CPU mode register (bits 7,6)

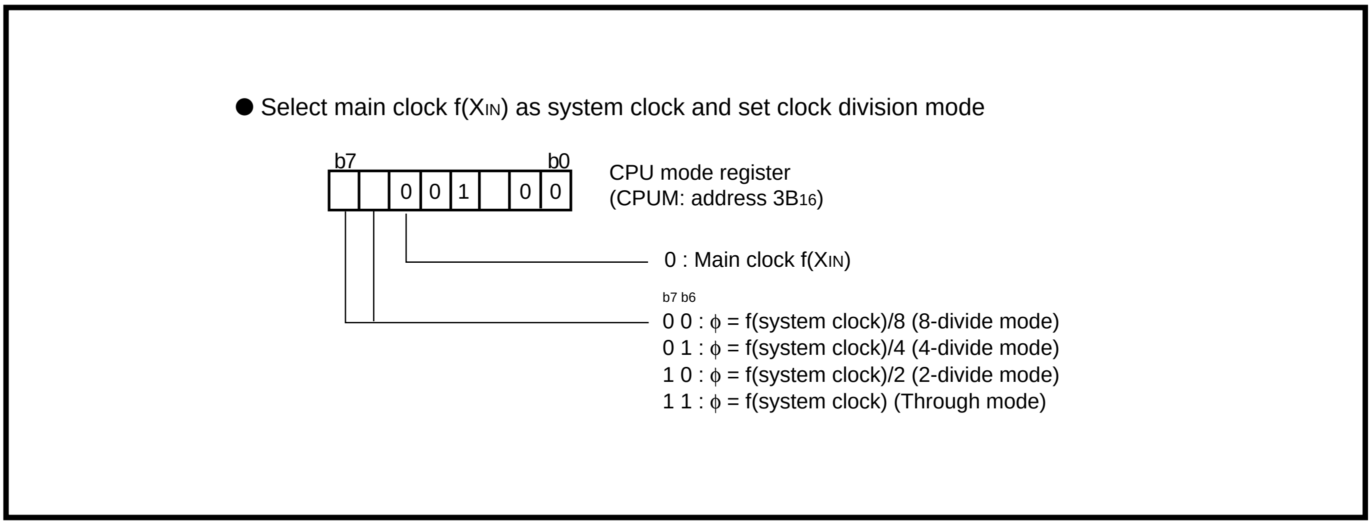

● Select main clock f(XIN) as system clock and set clock division mode
b7
b0
0 0 1 0 0
CPU mode register
(CPUM: address 3B16)
0 : Main clock f(XIN)
b7 b6
0 0 : ϕ = f(system clock)/8 (8-divide mode)
0 1 : ϕ = f(system clock)/4 (4-divide mode)
1 0 : ϕ = f(system clock)/2 (2-divide mode)
1 1 : ϕ = f(system clock) (Through mode)

Fig. 2.12.5 Related registers setting

(2) Generation of internal clock $f(\phi)$ using f_{SYN} (f_{USB} division clock)

Table 2.12.2 shows the example of internal clock $f(\phi)$ generation using f_{SYN}; Figure 2.12.6 shows the related registers setting.

Table 2.12.2 Example of internal clock $f(\phi)$ generation using f_{SYN}

f_{USB}	USB clock division ratio selection bits *1	f_{SYN}	System clock division ratio selection bits *2	$f(\phi)$	Power source voltage V_{CC} [V]
48 MHz	0 0	6 MHz	0 0	0.75 MHz	3.00 to 5.25
			0 1	1.5 MHz	
			1 0	3 MHz	
			1 1	6 MHz	
	0 1	8 MHz	0 0	1 MHz	4.00 to 5.25
			0 1	2 MHz	
			1 0	4 MHz	
			1 1	8 MHz	
	1 1	12 MHz	0 0	1.5 MHz	
			0 1	3 MHz	
			1 0	6 MHz	

***1:** PLL control register (bits 4,3)
***2:** CPU mode register (bits 7,6)

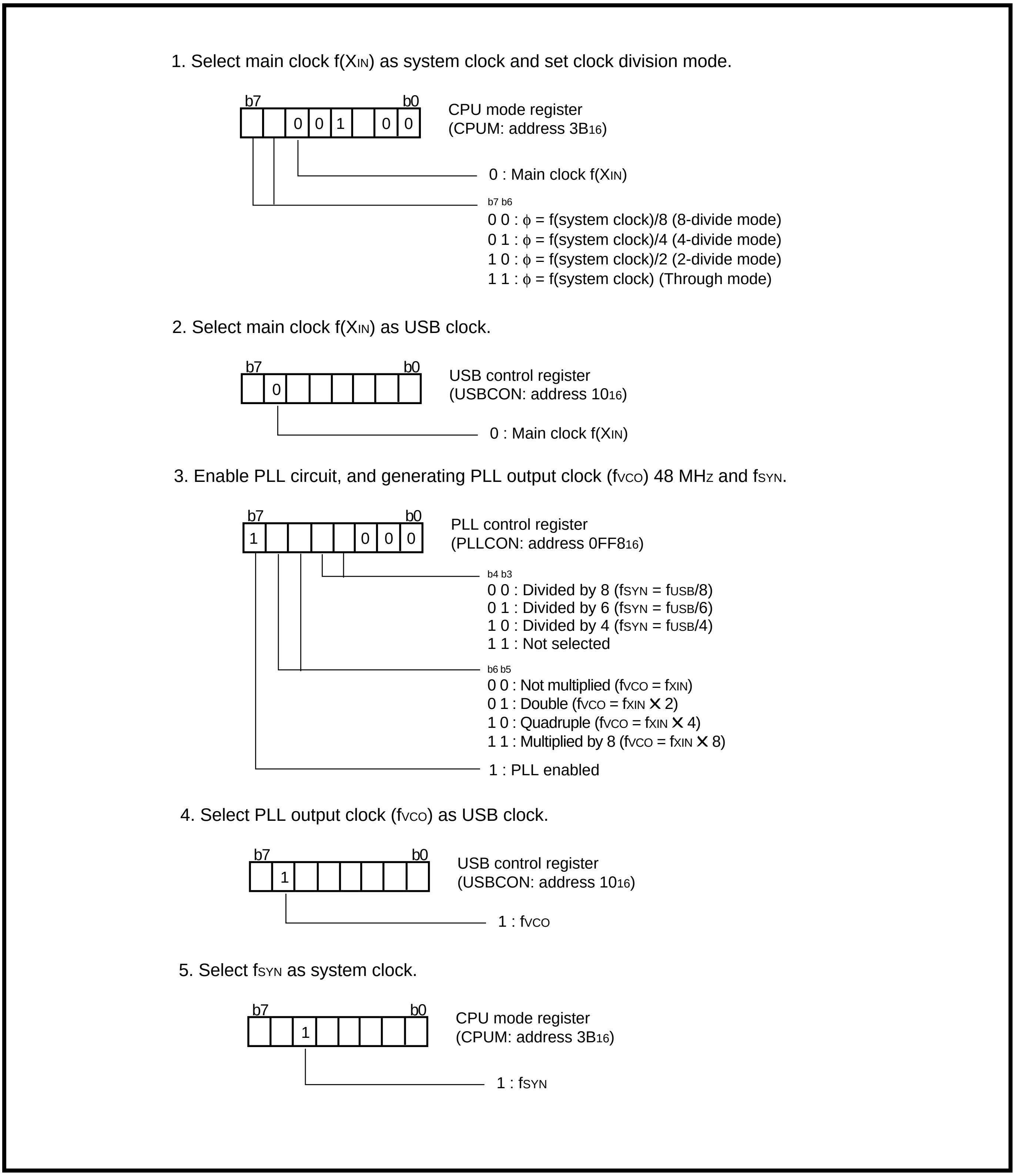

1. Select main clock $f(X_{IN})$ as system clock and set clock division mode.
b7 b0
0 0 1 0 0
CPU mode register
(CPUM: address $3B_{16}$)
0 : Main clock $f(X_{IN})$
b7 b6
0 0 : ϕ = f(system clock)/8 (8-divide mode)
0 1 : ϕ = f(system clock)/4 (4-divide mode)
1 0 : ϕ = f(system clock)/2 (2-divide mode)
1 1 : ϕ = f(system clock) (Through mode)
2. Select main clock $f(X_{IN})$ as USB clock.
b7 b0
0
USB control register
(USBCON: address 10_{16})
0 : Main clock $f(X_{IN})$
3. Enable PLL circuit, and generating PLL output clock (f_{VCO}) 48 MHz and f_{SYN}.
b7 b0
1 0 0 0
PLL control register
(PLLCON: address $0FF8_{16}$)
b4 b3
0 0 : Divided by 8 ($f_{SYN} = f_{USB}/8$)
0 1 : Divided by 6 ($f_{SYN} = f_{USB}/6$)
1 0 : Divided by 4 ($f_{SYN} = f_{USB}/4$)
1 1 : Not selected
b6 b5
0 0 : Not multiplied ($f_{VCO} = f_{XIN}$)
0 1 : Double ($f_{VCO} = f_{XIN} \times 2$)
1 0 : Quadruple ($f_{VCO} = f_{XIN} \times 4$)
1 1 : Multiplied by 8 ($f_{VCO} = f_{XIN} \times 8$)
1 : PLL enabled
4. Select PLL output clock (f_{VCO}) as USB clock.
b7 b0
1
USB control register
(USBCON: address 10_{16})
1 : f_{VCO}
5. Select f_{SYN} as system clock.
b7 b0
1
CPU mode register
(CPUM: address $3B_{16}$)
1 : f_{SYN}

Fig. 2.12.6 Related registers setting

Note: When selecting f_{SYN} as an internal system clock, refer to “2.11 Frequency synthesizer (PLL)” for details concerning how to generate f_{USB} (USB clock) from $f(X_{IN})$ and the notes on PLL circuit.

2.13 Standby function

The 38K2 group is provided with standby functions to stop the CPU by software and put the CPU into the low-power operation.

The following two types of standby functions are available.

•Stop mode using STP instruction
•Wait mode using WIT instruction

2.13.1 Memory map

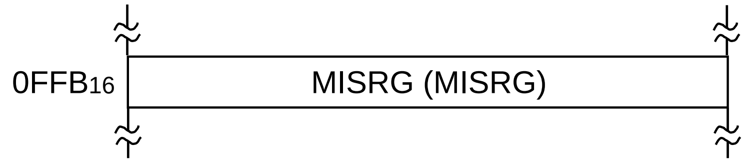
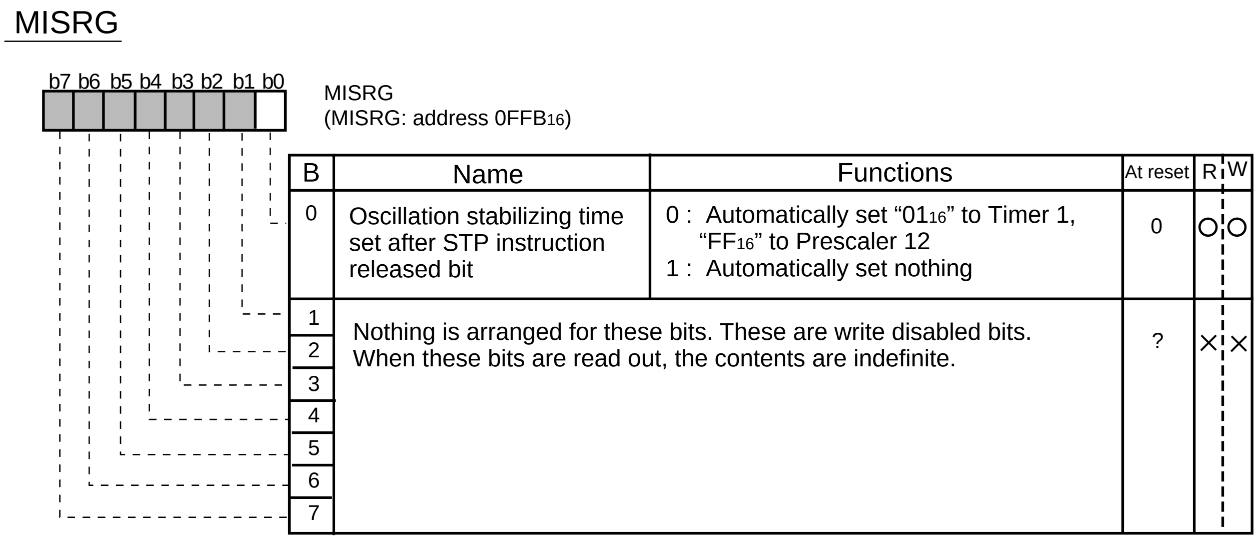


Fig. 2.13.1 Memory map of registers related to standby function

2.13.2 Related registers



MISRG
(MISRG: address 0FFB16)

B	Name	Functions	At reset	R	W
0	Oscillation stabilizing time set after STP instruction released bit	0 : Automatically set “0116” to Timer 1, “FF16” to Prescaler 12 1 : Automatically set nothing	0	○	○
1	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are indefinite.		?	×	×
2					
3					
4					
5					
6					
7					

Fig. 2.13.2 Structure of MISRG

2.13.3 Stop mode

The stop mode is set by executing the STP instruction. In the stop mode, the oscillation of clock (X_{IN}–X_{OUT}) stops and the internal clock ϕ stops at the "H" level. The CPU stops and peripheral units stop operating. As a result, power dissipation is reduced.

(1) State in stop mode

Table 2.13.1 shows the state in the stop mode.

Table 2.13.1 State in stop mode

Item	State in stop mode
Oscillation	Stopped.
CPU	Stopped.
Internal clock ϕ	Stopped at "H" level.
I/O ports P0–P6	Retains the state at the STP instruction execution.
Timer	Stopped. (Timers 1, 2, X) However, Timers X can be operated in the event counter mode.
Watchdog timer	Stopped.
Serial I/O	Stopped. However, these can be operated only when an external clock is selected.
USB function	Stopped.
HUB function	Stopped.
External BUS interface	Stopped.
A/D converter	Stopped.
Comparator	Stopped.

(2) Release of stop mode

The stop mode is released by a reset input or by the occurrence of an interrupt request. Note the differences in the restoration process according to reset input or interrupt request, as described below.

■Restoration by reset input

The stop mode is released by holding the $\overline{\text{RESET}}$ pin to the "L" input level during the stop mode. Oscillation is started when all ports are in the input state and the stop mode of the main clock (X_{IN}-X_{OUT}) is released.

Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. The input of the $\overline{\text{RESET}}$ pin should be held at the "L" level until oscillation stabilizes.

When the $\overline{\text{RESET}}$ pin is held at the "L" level for 16 cycles or more of X_{IN} after the oscillation has stabilized, the microcomputer will go to the reset state. After the input level of the $\overline{\text{RESET}}$ pin is returned to "H", the reset state is released in approximately 10.5 to 18.5 cycles of the X_{IN} input.

Figure 2.13.3 shows the oscillation stabilizing time at restoration by reset input.

At release of the stop mode by reset input, the internal RAM retains its contents previous to the reset. However, the previous contents of the CPU register and SFR are not retained.

For more details concerning reset, refer to "2.10 Reset".

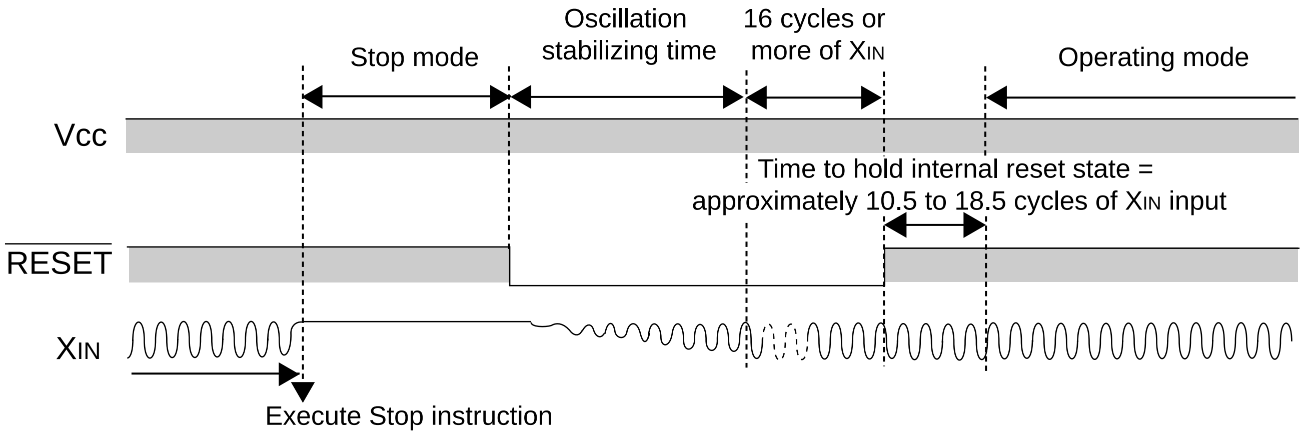

Oscillation stabilizing time
16 cycles or more of XIN
Stop mode
Operating mode
Vcc
Time to hold internal reset state = approximately 10.5 to 18.5 cycles of XIN input
RESET
XIN
Execute Stop instruction

Fig. 2.13.3 Oscillation stabilizing time at restoration by reset input

■Restoration by interrupt request

The occurrence of an interrupt request in the stop mode releases the stop mode. As a result, oscillation is resumed. The interrupts available for restoration are:

- INT_0, INT_1
- $CNTR_0$
- Serial I/O using an external clock
- Timer X using an external event count
- Key input (key-on wake-up)
- USB function (resume)

However, when using any of these interrupt requests for restoration from the stop mode, in order to enable the selected interrupt, you must execute the STP instruction after setting the following conditions.

[Necessary register setting]
① Interrupt disable flag I = “0” (interrupt enabled)
② Timer 1 interrupt enable bit = “0” (interrupt disabled)
③ Interrupt request bit of interrupt source to be used for restoration = “0” (no interrupt request issued)
④ Interrupt enable bit of interrupt source to be used for restoration = “1” (interrupts enabled)

For more details concerning interrupts, refer to “2.2 Interrupts”.

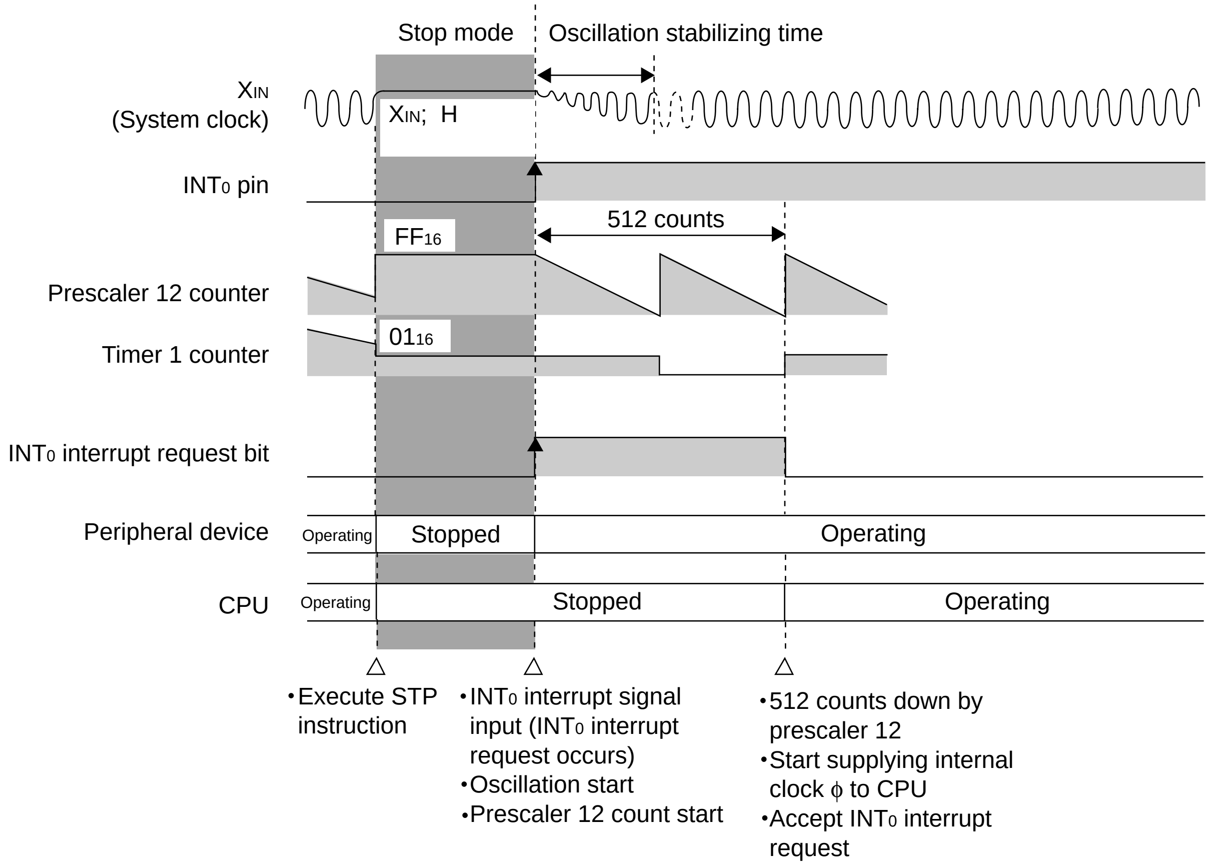
Oscillation is unstable when restarted. For this reason, time for stabilizing of oscillation (oscillation stabilizing time) is required. For restoration by an interrupt request, waiting time prior to supplying internal clock ϕ to the CPU is automatically generated*2 by Prescaler 12 and Timer 1*1. This waiting time is reserved as the oscillation stabilizing time on the system clock side. The supply of internal clock ϕ to the CPU is started at the Timer 1 underflow.

Figure 2.13.4 shows an execution sequence example at restoration by the occurrence of an INT_0 interrupt request.

✻1: If the STP instruction is executed when the oscillation stabilizing time set after STP instruction released bit is “0”, “FF_{16}” and “01_{16}” are automatically set in the Prescaler 12 counter/latch and Timer 1 counter/latch, respectively. When the oscillation stabilizing time set after STP instruction released bit is “1”, nothing is automatically set to either Prescaler 12 or Timer 1. For this reason, any suitable value can be set to Prescaler 12 and Timer 1 for the oscillation stabilizing time.

✻2: Immediately after the oscillation is started, the count source is supplied to the prescaler 12 so that a count operation is started.

●When restoring microcomputer from stop mode by INT0 interrupt (rising edge selected)


Stop mode
Oscillation stabilizing time
XIN
(System clock)
XIN; H
INT0 pin
512 counts
FF16
Prescaler 12 counter
Timer 1 counter
0116
INT0 interrupt request bit
Peripheral device
Operating
Stopped
Operating
CPU
Operating
Stopped
Operating
•Execute STP instruction
•INT0 interrupt signal input (INT0 interrupt request occurs)
•Oscillation start
•Prescaler 12 count start
•512 counts down by prescaler 12
•Start supplying internal clock φ to CPU
•Accept INT0 interrupt request

Note: f(XIN)/16 is input as the prescaler 12 count source.

Fig. 2.13.4 Execution sequence example at restoration by occurrence of INT0 interrupt request

(3) Notes on using stop mode

■Register setting

Since values of the prescaler 12 and Timer 1 are automatically reloaded when returning from the stop mode, set them again, respectively. (When the oscillation stabilizing time set after STP instruction released bit is “0”)

■Clock restoration

When the main clock side is set as a system clock, the oscillation stabilizing time for approximately 8,000 cycles of the XIN input is reserved at restoration from the stop mode.

2.13.4 Wait mode

The wait mode is set by execution of the WIT instruction. In the wait mode, oscillation continues, but the internal clock ϕ stops at the “H” level.

The CPU stops, but most of the peripheral units continue operating.

(1) State in wait mode

The continuation of oscillation permits clock supply to the peripheral units. Table 2.13.2 shows the state in the wait mode.

Table 2.13.2 State in wait mode

Item	State in wait mode
Oscillation	Operating.
CPU	Stopped.
Internal clock ϕ	Stopped at “H” level.
I/O ports P0–P6	Retains the state at the WIT instruction execution.
Timer	Operating.
Watchdog timer	Operating.
Serial I/O	Operating.
USB function	Operating.
HUB function	Operating.
External BUS interface	Stopped.
A/D converter	Operating.
Comparator	Operating.

(2) Release of wait mode

The wait mode is released by reset input or by the occurrence of an interrupt request. Note the differences in the restoration process according to reset input or interrupt request, as described below.

In the wait mode, oscillation is continued, so an instruction can be executed immediately after the wait mode is released.

■Restoration by reset input

The wait mode is released by holding the input level of the $\overline{\text{RESET}}$ pin at "L" in the wait mode. Upon release of the wait mode, all ports are in the input state, and supply of the internal clock ϕ to the CPU is started. To reset the microcomputer, the $\overline{\text{RESET}}$ pin should be held at an "L" level for 16 cycles or more of X_{IN}. The reset state is released in approximately 10.5 cycles to 18.5 cycles of the X_{IN} input after the input of the $\overline{\text{RESET}}$ pin is returned to the "H" level.

At release of wait mode, the internal RAM retains its contents previous to the reset. However, the previous contents of the CPU register and SFR are not retained.

Figure 2.13.5 shows the reset input time.

For more details concerning reset, refer to "2.10 Reset".

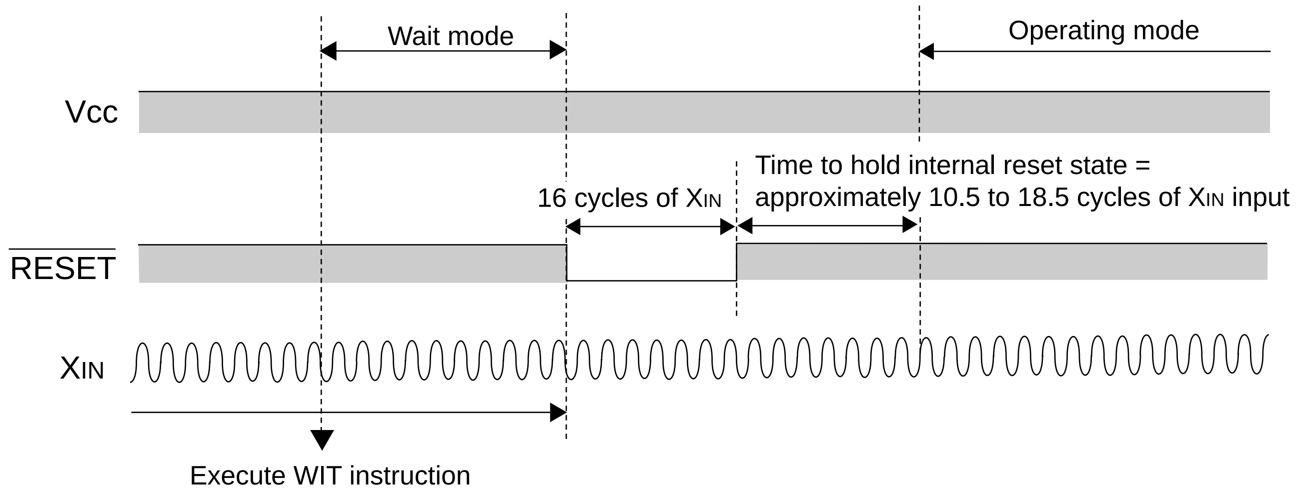

Wait mode
Operating mode
Vcc
Time to hold internal reset state =
approximately 10.5 to 18.5 cycles of XIN input
16 cycles of XIN
RESET
XIN
Execute WIT instruction

Fig. 2.13.5 Reset input time

■Restoration by interrupt request

In the wait mode, the occurrence of an interrupt request releases the wait mode and supply of the internal clock ϕ to the CPU is started. At the same time, the interrupt request used for restoration is accepted, so the interrupt processing routine is executed.

However, when using an interrupt request for restoration from the wait mode, in order to enable the selected interrupt, you must execute the STP instruction after setting the following conditions.

[Necessary register setting]
① Interrupt disable flag I = “0” (interrupt enabled)
② Interrupt request bit of interrupt source to be used for restoration = “0” (no interrupt request issued)
③ Interrupt enable bit of interrupt source to be used for restoration = “1” (interrupts enabled)

For more details concerning interrupts, refer to “2.2 Interrupts”.

2.13.5 Notes on stand-by function

In stand-by state*[1] for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”.

Pull-up (connect the port to V_{CC}) these ports through a resistor.

When determining a resistance value, note the following points:

- External circuit
- Variation of output levels during the ordinary operation

When using built-in pull-up resistor, note on varied current values.

- When setting as an input port: Fix its input level
- When setting as an output port: Prevent current from flowing out to external

● Reason

The potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of an input port and an I/O port are “undefined”. This may cause power source current.

*[1] stand-by state : the stop mode by executing the **STP** instruction
the wait mode by executing the **WIT** instruction

2.14 Flash memory

This paragraph explains the registers setting method and the notes related to the flash memory version.

2.14.1 Overview

The functions of the flash memory version are similar to those of the mask ROM version except that the flash memory is built-in and some of the SFR area differ from that of the mask ROM version (refer to “2.14.2 Memory map”).

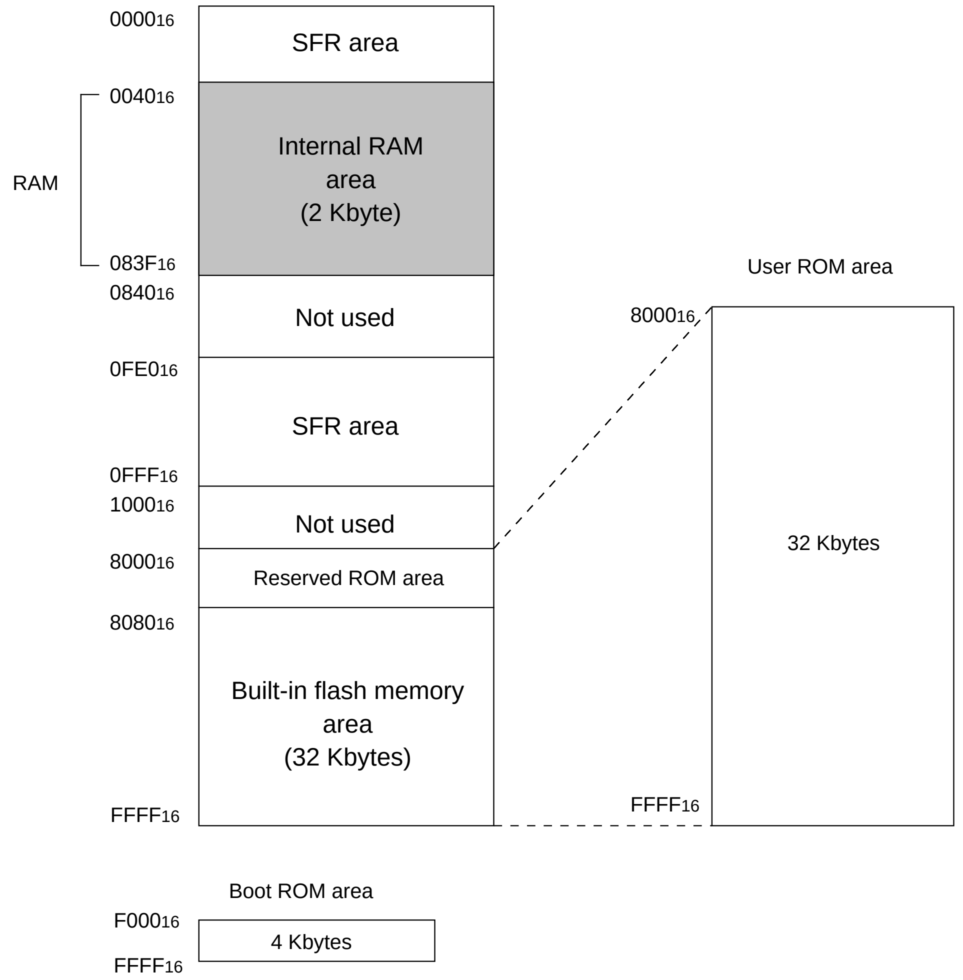
In the flash memory version, the built-in flash memory can be programmed or erased by using the following three modes.

- CPU rewrite mode
- Parallel I/O mode
- Standard serial I/O mode

2.14.2 Memory map

38K2 group flash memory version has 32 Kbytes of built-in flash memory.

Figure 2.14.1 shows the memory map of the flash memory version.



Note: Access to boot ROM area
Pararell I/O mode---------Read/Write avilable
CPU rewrite mode --------Read only available
Standard serial mode------Read only available

Fig. 2.14.1 Memory map of flash memory version for 38K2 Group

2.14.3 Related registers

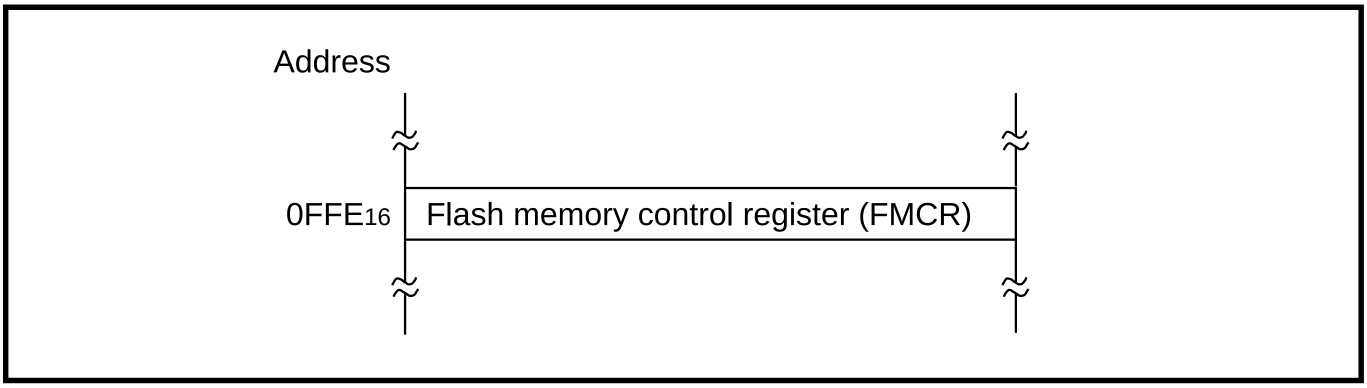
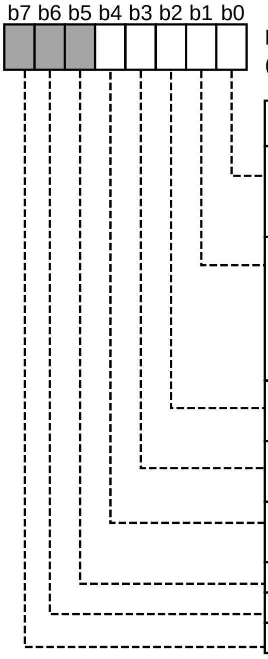


Fig. 2.14.2 Memory map of registers related to flash memory

Flash memory control register



Flash memory control register
(FMCR : address 0FFE16) **(Note 1)**

b	Name	Functions	At reset	R	W
0	RY/$\overline{\text{BY}}$ status flag	0 : Busy (being written or erased) 1 : Ready	1	○	×
1	CPU rewrite mode select bit **(Note 2)**	0 : Normal mode (Software commands invalid) 1 : CPU rewrite mode (Software commands acceptable)	0	○	○
2	CPU rewrite mode entry flag	0: Normal mode 1: CPU rewrite mode	0	○	×
3	Flash memory reset bit **(Note 3)**	0: Normal operation 1: Reset	0	○	○
4	User area/Boot area selection bit **(Note 4)**	0: User ROM area 1: Boot ROM area	0	○	○
5	Nothing is arranged for these bits. If writing, set “0”. When these bits are read out, the contents are undefined.		Undefined	×	○
6			Undefined	×	○
7			Undefined	×	○

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. 2.14.3 Structure of Flash memory control register

2.14.4 Parallel I/O mode

In the parallel I/O mode, program/erase to the built-in flash memory can be performed by a flash programmer (MFW-1).

The memory area of program/erase is from $0F000_{16}$ to $0FFFF_{16}$ (boot ROM area) or from 08000_{16} to $0FFFF_{16}$ (user ROM area). Be especially careful when erasing; if the memory area is not set correctly, the products will be damaged eternally.

Table 2.14.1 shows the setting of programmers when programming in the parallel I/O mode.

•MFW-1 provided by Sunny Giken Inc. (http://www.sunnygiken.co.jp/english/index.html)

Table 2.14.1 Setting of programmers when parallel programming

Products	Parallel adapter	Boot ROM area	User ROM area
M38K29F8HP/LHP	MFW-S18	$0F000_{16}$ to $0FFFF_{16}$	08000_{16} to $0FFFF_{16}$
M38K29F8FP/LFP	MFW-S19		

2.14.5 Standard serial I/O mode

Table 2.14.2 shows a pin connection example (4 wires) between the programmer (MFW-1) and the microcomputer when programming in the standard serial I/O mode.

•MFW-1 provided by Sunny Giken Inc. (http://www.sunnygiken.co.jp/english/index.html)

Table 2.14.2 Connection example to flash programmer when serial programming (4 wires)

Function	MFW-1		38K2 Group flash memory version	
	Signal name	MFW-1 side connector Line number	Pin name	Pin number
Transfer clock input	CLK	3	$P4_2$/ExTC/S_{CLK}	53
Serial data input	R_XD	10	$P4_0$/ExDREQ/R_XD	51
Serial data output	T_XD	4	$P4_1$/ExDACK/TxD	52
Transmit/Receive enable output	BUSY	2	$P4_3$/ExA1/$\overline{S_{RDY}}$	54
V_{PP} input	CNV_{SS}	1	CNV_{SS}	7
Reset input	RESET	8	$\overline{RESET}$	8
Target board power source monitor input	V_{CC} **(Note 2)**	1	V_{CC}, PV_{CC}, DV_{CC} (**Note 2**)	14, 21, 22
GND	GND **(Note 1)**	7	V_{SS}, PV_{SS} (**Note 1**)	11, 20

Notes 1: When connecting a serial programmer, first connect both GNDs to the same GND level.

2: V_{CC} power of MFW-1 is supplied from a target board. Power consumption of MFW-1 is Max. 200 mA when serial programming. Therefore, when the current capacity of target borad is short, connect AC adapter and supply power source to MFW-1.

2.14.6 CPU rewrite mode

In the CPU rewrite mode, issuing software commands through the Central Processing Unit (CPU) can rewrite the built-in flash memory. Accordingly, the contents of the built-in flash memory can be rewritten with the microcomputer itself mounted on board, without using the programmer.

Store the rewrite control program to the built-in flash memory in advance. The built-in flash memory cannot be read in the CPU rewrite mode. Accordingly, after transferring the rewrite control program to the internal RAM, execute it on the RAM.

The following commands can be used in the CPU rewrite mode: read array, read status register, clear status register, program, erase all block, and block erase. For details concerning each command, refer to "CHAPTER 1 Flash memory mode (CPU rewrite mode)".

(1) CPU rewrite mode beginning/release procedures

Operation procedure in the CPU rewrite mode for the built-in flash memory is described below. As for the control example, refer to "2.14.7 (2) Control example in the CPU rewrite mode."

[Beginning procedure]

① Apply 4.50 to 5.25 V to the CNV$_{SS}$/V$_{PP}$ pin (at selecting boot ROM area).
② Release reset.
③ Set bits 6 and 7 (main clock division ratio selection bits) of the CPU mode register.
④ After CPU rewrite mode control program is transferred to internal RAM, jump to this control program on RAM. (The following operations are controlled by this control program).
⑤ Apply 4.50 to 5.25 to the CNV$_{SS}$/V$_{PP}$ pin (in single-chip mode).
⑥ Set "1" to the CPU rewrite mode select bit (bit 1 of address 0FFE$_{16}$).
⑦ Read the CPU rewrite mode entry flag (bit 2 of address 0FFE$_{16}$) to confirm that the CPU rewrite mode is set to "1".
⑧ Flash memory operations are executed by using software commands.

Note: The following procedures are also necessary.

- Control for data which is input from the external (serial I/O etc.) and to be programmed to the flash memory.
- Initial setting for ports, etc.
- Writing to the watchdog timer

[Release procedure]

① Execute the read command or set the flash memory reset bit (bit 3 of address 0FFE$_{16}$).
② Set the CPU rewrite mode select bit (bit 0 of address 0FFE$_{16}$) to "0".

2.14.7 Flash memory mode application examples

The control pin processing example on the system board in the standard serial I/O mode and the control example in the CPU rewrite mode are described below.

(1) Control pin connection example on the system board in standard serial I/O mode

As shown in Figure 2.14.4, in the standard serial I/O mode, the built-in flash memory can be rewritten with the microcomputer mounted on board. Connection examples of control pins ($P4_0$/E_XDREQ/R_XD, $P4_1$/E_XDACK/T_XD, $P4_2$/E_XTC/S_{CLK}, $P4_3$/E_XA1/$\overline{S_{RDY}}$, $P1_6$, CNV_{SS}, and $\overline{RESET}$ pin) in the standard serial I/O mode are described below.

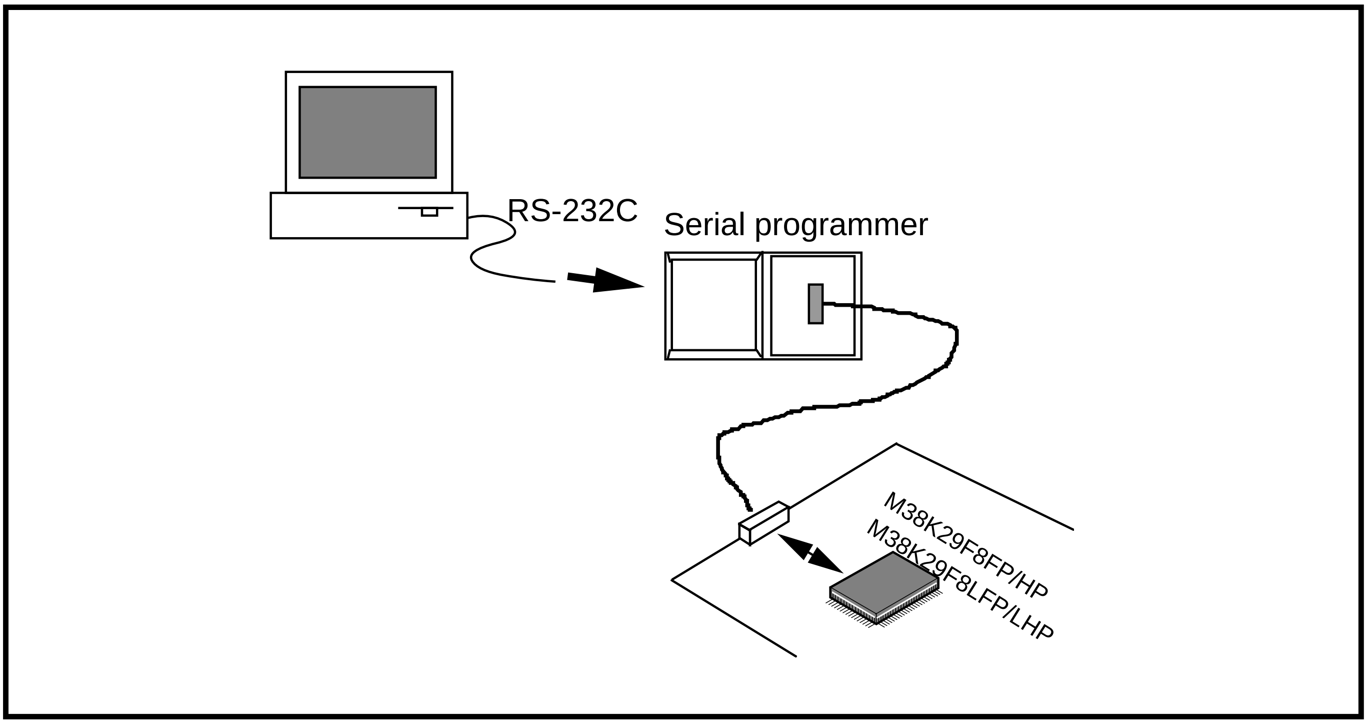


Fig. 2.14.4 Rewrite example of built-in flash memory in standard serial I/O mode

Table 2.14.3 shows the setting condition in the standard serial I/O mode.

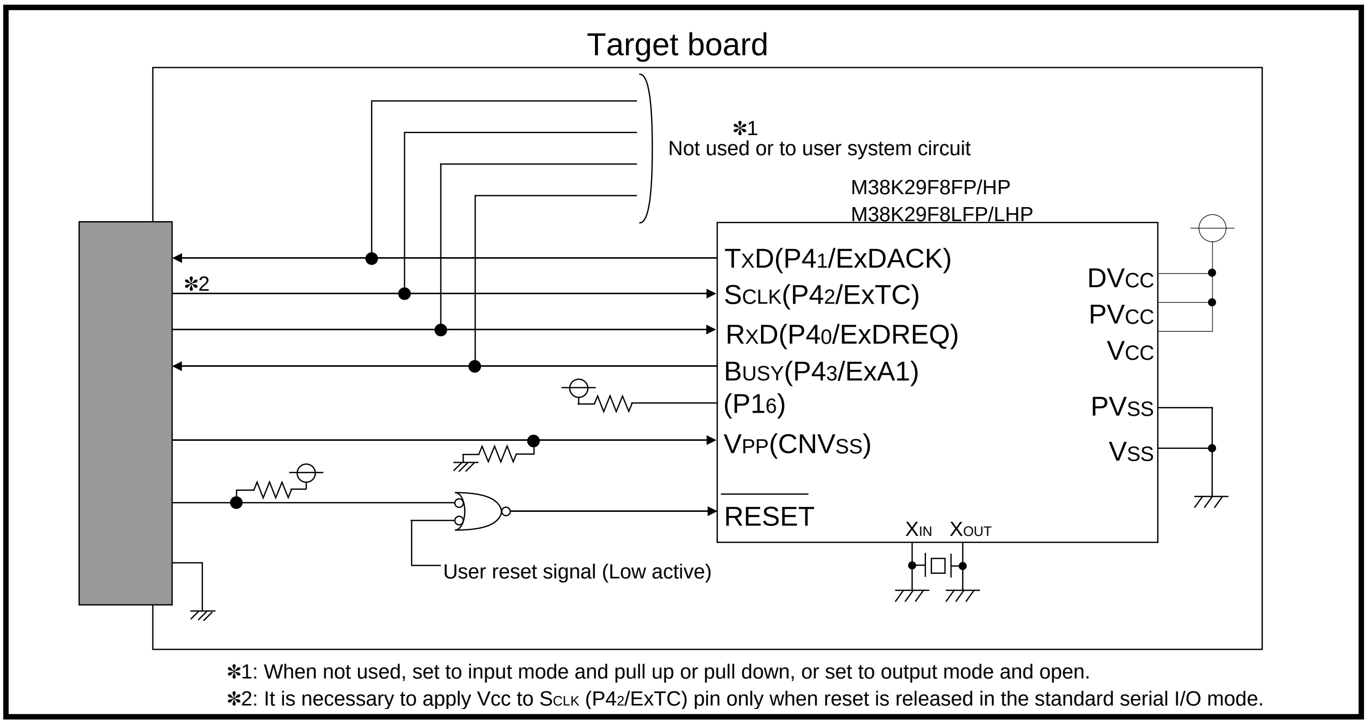
Table 2.14.3 Setting condition in serial I/O mode

38K2 Group flash memory version		Value
Pin name	Pin number	
CNV_{SS}/V_{PP} (**Note**)	7	4.50 to 5.25 V
$P1_6$	5	V_{CC}
$P4_2$/E_XTC/S_{CLK}	53	V_{CC}
$\overline{RESET}$	8	Edge from V_{SS} to V_{CC}

Note: CNV_{SS}/V_{PP} is not V_{CC} but a voltage when programming.

① When control signals are not affected to user system circuit

When the control signals in the standard serial I/O mode are not used or not affected to the user system circuit, they can be connected as shown in Figure 2.14.5.

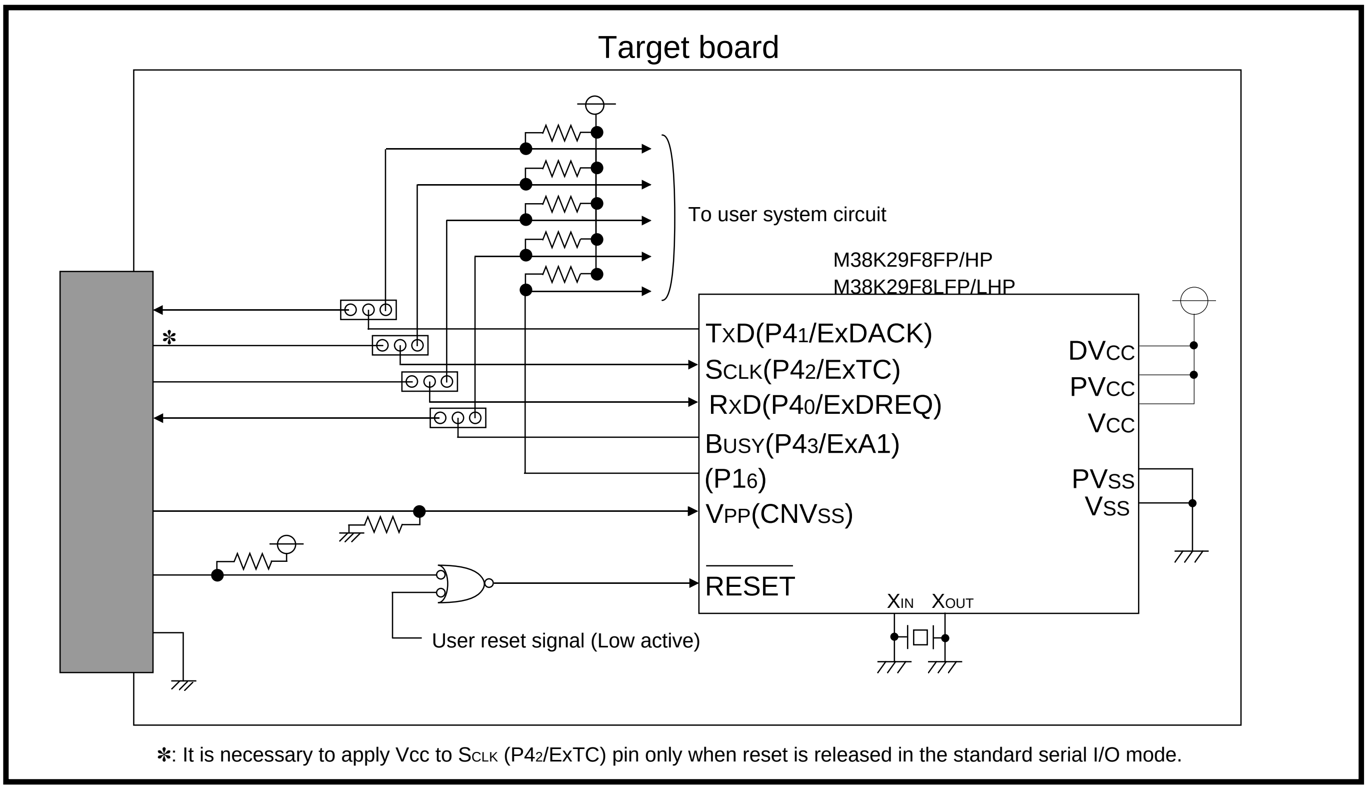


✻1: When not used, set to input mode and pull up or pull down, or set to output mode and open.
✻2: It is necessary to apply Vcc to SCLK (P42/ExTC) pin only when reset is released in the standard serial I/O mode.

Fig. 2.14.5 Connection example in standard serial I/O mode (1)

② When control signals are affected to user system circuit-1

Figure 2.14.6 shows an example that the jumper switch cut-off the control signals not to supply to the user system circuit in the standard serial I/O mode.

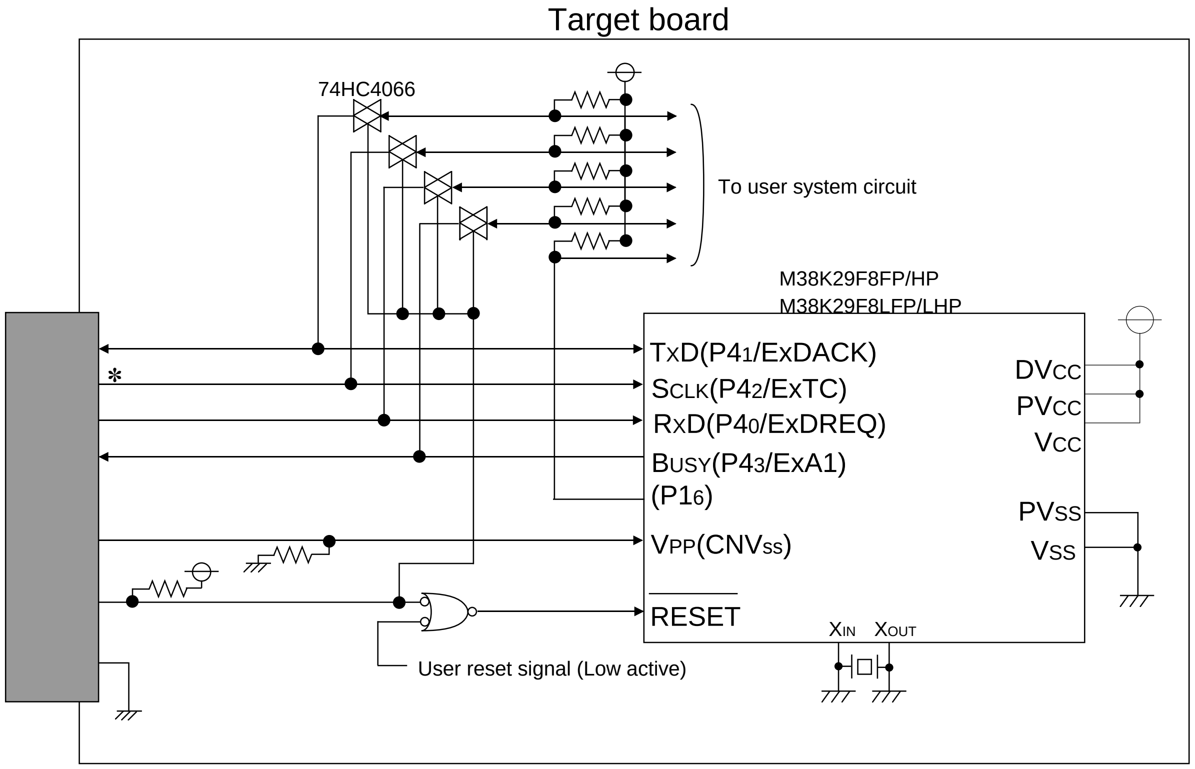


✻: It is necessary to apply Vcc to SCLK (P42/ExTC) pin only when reset is released in the standard serial I/O mode.

Fig. 2.14.6 Connection example in standard serial I/O mode (2)

③ When control signals are affected to user system circuit-2

Figure 2.14.7 shows an example that the analog switch (74HC4066) cut-off the control signals not to supply to the user system circuit in the standard serial I/O mode.



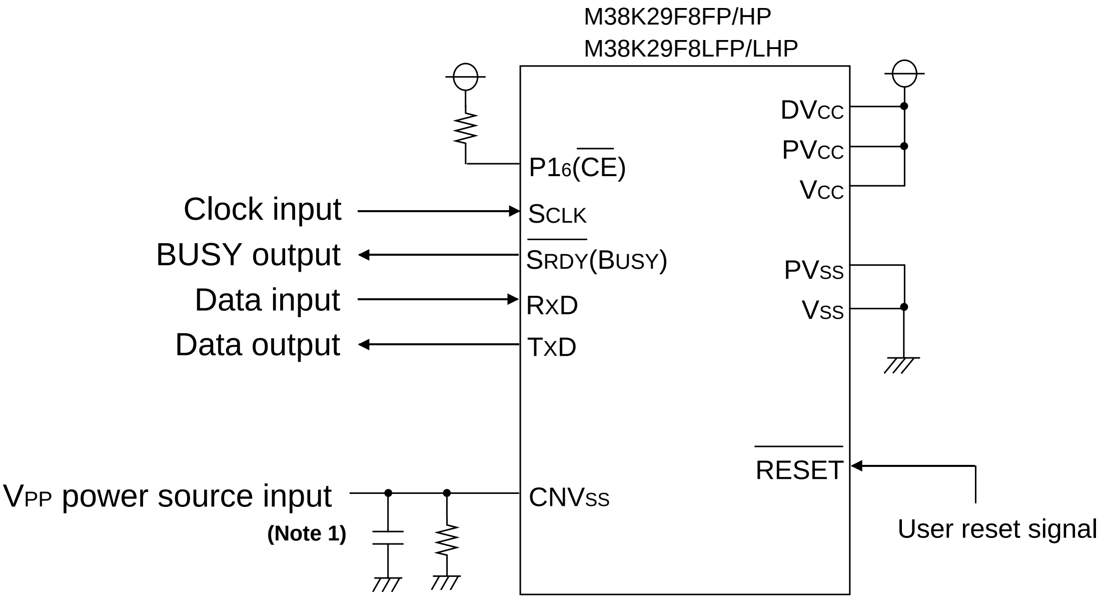
✻: It is necessary to apply Vcc to SCLK (P42/ExTC) pin only when reset is released in the standard serial I/O mode.

Fig. 2.14.7 Connection example in standard serial I/O mode (3)

(2) Control example in CPU rewrite mode

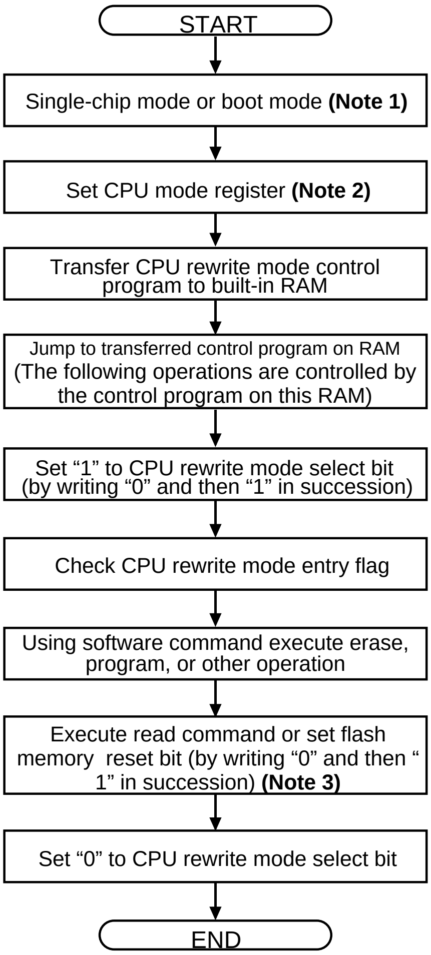
In this example, data is received by using serial I/O, and the data is programmed to the built-in flash memory in the CPU rewrite mode.

Figure 2.14.8 shows an example of the reprogramming system for the built-in flash memory in the CPU rewrite mode. Figure 2.14.9 shows the CPU rewrite mode beginning/release flowchart.



Note 1: Apply 4.50 to 5.25 V to the V_{PP} power source.

Fig. 2.14.8 Example of rewrite system for built-in flash memory in CPU rewrite mode


START
Single-chip mode or boot mode **(Note 1)**
Set CPU mode register **(Note 2)**
Transfer CPU rewrite mode control program to built-in RAM
Jump to transferred control program on RAM (The following operations are controlled by the control program on this RAM)
Set “1” to CPU rewrite mode select bit (by writing “0” and then “1” in succession)
Check CPU rewrite mode entry flag
Using software command execute erase, program, or other operation
Execute read command or set flash memory reset bit (by writing “0” and then “ 1” in succession) **(Note 3)**
Set “0” to CPU rewrite mode select bit
END

Notes 1: When MCU starts in the single-chip mode, it is necessary to apply 4.50 to 5.25 V to dhe CNVss pin until confirming of the CPU rewrite mode entry flag.

2: Set bits 6 and 7 (system clock division ratio selection bits) of the CPU mode register (address $003B_{16}$).

3: Before releasing 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. 2.14.9 CPU rewrite mode beginning/release flowchart

2.14.8 Notes on CPU rewrite mode

(1) Operation speed

During CPU rewrite mode, set the internal clock ϕ 1.5 MHz or less using the system clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

(2) Instructions inhibited against use

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

(3) Interrupts inhibited against use

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

(4) Watchdog timer

In case of the watchdog timer has been running already, the internal reset generated by watchdog timer underflow does not happen, because of watchdog timer is always clearing during program or erase operation.

(5) Reset

Reset is always valid. In case of CNV_{SS} = “H” when reset is released, boot mode is active. So the program starts from the address contained in address $FFFC_{16}$ and $FFFD_{16}$ in boot ROM area.

CHAPTER 3

APPENDIX

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 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
AV_{CC}	Analog power source voltage	$V_{CC}E$, V_{REF}, PV_{CC}, DV_{CC}, $USBV_{REF}$			–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_4$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	$\overline{\text{RESET}}$, X_{IN}, CNV_{SS2}			–0.3 to V_{CC} + 0.3	V
V_I	Input voltage	CNV_{SS}	Mask ROM version		–0.3 to V_{CC} + 0.3	V
			Flash memory version		–0.3 to 6.5	V
V_I	Input voltage	D0+, D0-, D1+, D1-, D2+, D2-			–0.5 to 3.8	V
V_O	Output voltage	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_4$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$, $P5_0$–$P5_7$, $P6_0$–$P6_3$, X_{OUT}			–0.3 to V_{CC} + 0.3	V
V_O	Output voltage	D0+, D0-, D1+, D1-, D2+, D2-, 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.

3.1.2 Recommended operating conditions (L.Ver)

Table 3.1.2 Recommended operating conditions (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.	
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_{CCE}			V_{CC}		V
V_{REF}	Analog reference voltage	V_{REF}		2.0		V_{CC}	V
V_{REF}	Analog reference voltage	$USBV_{REF}$	V_{cc} = 3.6 to 4.0 V	3.0		3.6	V
			V_{cc} = 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		$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$	$0.8V_{CC}$		V_{CC}	V V
V_{IH}	"H" input voltage		$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$	$0.8V_{CCE}$		V_{CCE}	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-, D1+, D1-, D2+, D2-	2.0		3.6	V
V_{IL}	"L" input voltage		$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$	0		$0.2V_{CC}$	V
V_{IL}	"L" input voltage		$P1_0$–$P1_7$, $P3_0$–$P3_7$, $P4_0$–$P4_3$	0		$0.2V_{CCE}$	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-, D1+, D1-, D2+, D2-	0		0.8	V

Table 3.1.3 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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$			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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$			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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$			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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_3$			–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**)	$P0_0$–$P0_7$, $P2_4$–$P2_7$, $P5_0$–$P5_7$			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.

3.1.3 Electrical characteristics (L.Ver)

Table 3.1.4 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$_4$–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_{CCE} = 4.00 to 5.25 V)	V_{CCE}–2.0			V
		I_{OH} = –1 mA	V_{CCE}–1.0			V
V_{OH}	“H” output voltage D0+, D0-, D1+, D1-, D2+, D2-	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$_4$–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_{CCE} = 4.00 to 5.25 V)			2.0	V
		I_{OL} = 1 mA (V_{CCE} = 3.00 to 5.25 V)			1.0	V
V_{OL}	“L” output voltage D0+, D0-, D1+, D1-, D2+, D2-	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$_1$/Ex$\overline{\text{DACK}}$/TxD, P4$_2$/ExTC/S_{CLK}, P4$_3$/ExA1/$\overline{S_{RDY}}$			0.6		V
V_{T+}–V_{T}	Hysteresis D0+, D0-, D1+, D1-, D2+, D2-			0.25		V
V_{T+}–V_{T-}	Hysteresis $\overline{\text{RESET}}$			0.5		V
I_{IH}	“H” input current P0$_0$–P0$_7$, P2$_4$–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_{CCE}			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$_4$–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 3.1.5 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.	
Icc	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

3.1.4 A/D converter characteristics (L.Ver)

Table 3.1.6 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

3.1.5 Timing Requirements (L.Ver)

Table 3.1.7 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
$t_{C}(X_{IN})$	Main clock input cycle time	83			ns
$t_{WH}(X_{IN})$	Main clock input “H” pulse width	35			ns
$t_{WL}(X_{IN})$	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
$t_{C}(S_{CLK})$	Serial I/O clock input cycle time (**Note**)	800			ns
$t_{WH}(S_{CLK})$	Serial I/O clock input “H” pulse width (**Note**)	370			ns
$t_{WL}(S_{CLK})$	Serial I/O clock input “L” pulse width (**Note**)	370			ns
$t_{su}(RxD–S_{CLK})$	Serial I/O input set up time	220			ns
$t_{h}(S_{CLK}–RxD)$	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 3.1.8 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
$t_{C}(X_{IN})$	Main clock input cycle time	166			ns
$t_{WH}(X_{IN})$	Main clock input “H” pulse width	70			ns
$t_{WL}(X_{IN})$	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
$t_{C}(S_{CLK})$	Serial I/O clock input cycle time (**Note**)	2000			ns
$t_{WH}(S_{CLK})$	Serial I/O clock input “H” pulse width (**Note**)	950			ns
$t_{WL}(S_{CLK})$	Serial I/O clock input “L” pulse width (**Note**)	950			ns
$t_{su}(RxD–S_{CLK})$	Serial I/O input set up time	400			ns
$t_{h}(S_{CLK}–RxD)$	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 3.1.9 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}(ϕ)	CPU clock cycle time		125			ns
t_{w}(cycle)	Burst mode access cycle time	USB function not operating	t_{C}(ϕ)•3+10			ns
		USB function operating	t_{C}(ϕ)•5+10			ns

Table 3.1.10 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

3.1.6 Switching Characteristics (L.Ver)

Table 3.1.11 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 XOUT, D0+, D0-, D1+, D2-, D2+, D2- are excluded.

Table 3.1.12 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 XOUT, D0+, D0-, D1+, D2-, D2+, D2- are excluded.

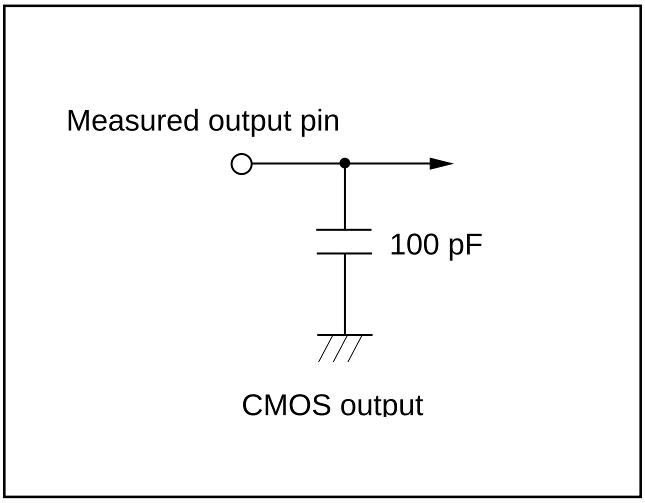


Fig. 3.1.1 Output switching characteristics measurement circuit

Table 3.1.13 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 3.1.14 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 3.1.15 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	C_L = 50 pF	4		20	ns
t_{ff}(D+/D-)	USB full-speed output rising time	C_L = 50 pF	4		20	ns
t_{lr}(D+/D-)	USB low-speed output rising time	C_L = 200 to 600 pF Ta = 0 to 85 °C	75		300	ns
		C_L = 250 to 600 pF Ta = –20 to 85 °C	75		300	ns
		C_L = 200 to 600 pF Ta = –20 to 85 °C	65		300	ns
t_{lf}(D+/D-)	USB low-speed output falling time	C_L = 200 to 600 pF Ta = 0 to 85 °C	75		300	ns
		C_L = 250 to 600 pF Ta = –20 to 85 °C	75		300	ns
		C_L = 200 to 600 pF Ta = –20 to 85 °C	65		300	ns
t_{frfm}(D+/D-)	USB full-speed ports rising/falling ratio	t_{fr}(D+/D-)/t_{ff}(D+/D-)	90		111.11	%
t_{lrfm}(D+/D-)	USB low-speed ports rising/falling ratio	t_{lr}(D+/D-)/t_{ff}(D+/D-)	80		125	%
V_{crs}(D+/D-)	USB output signal cross-over voltage		1.3		2.0	V

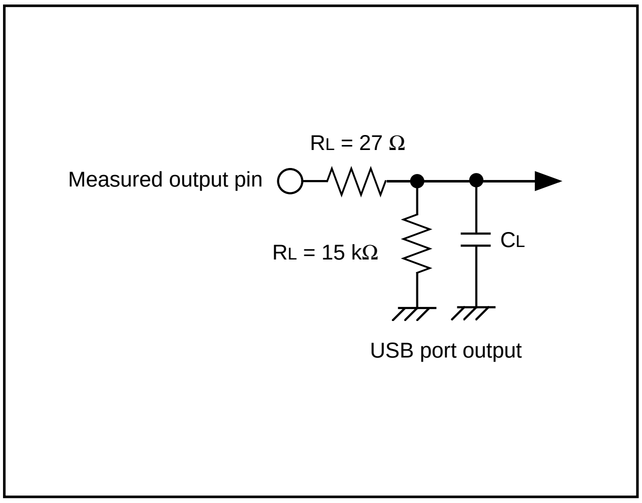


Fig. 3.1.2 USB output switching characteristics measurement circuit (1) for D0-, D1+/D2+ (low-speed), D1-/D2- (full-speed)

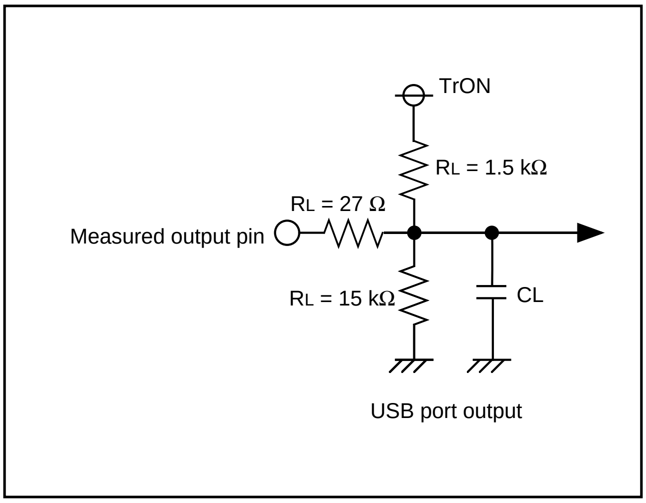


Fig. 3.1.3 USB output switching characteristics measurement circuit (2) for D0+, D1+/D2+ (full-speed), D1-/D2- (low-speed)

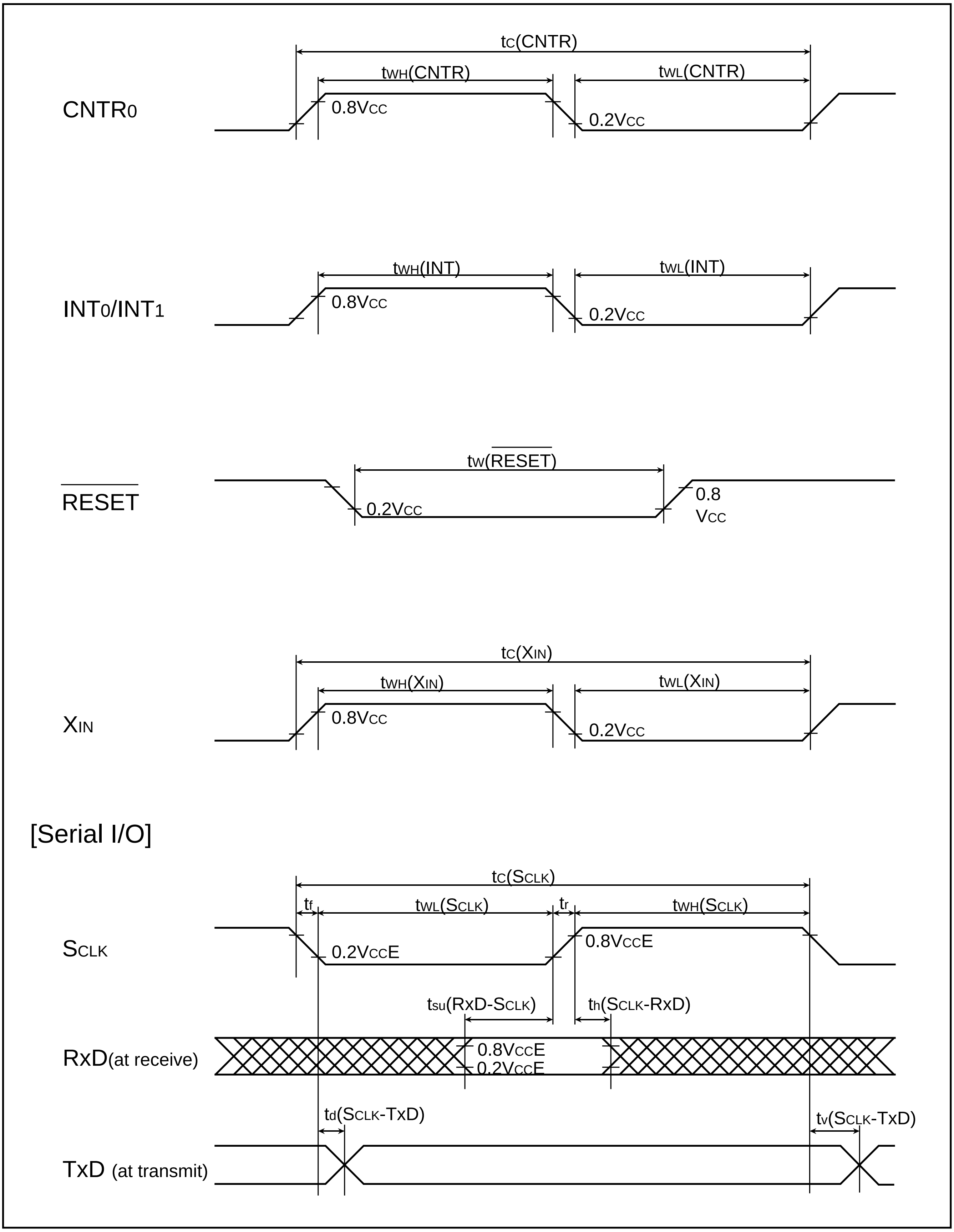


Fig. 3.1.4 Timing chart (1)

● Timing chart

[EXB <CPU channel mode>]

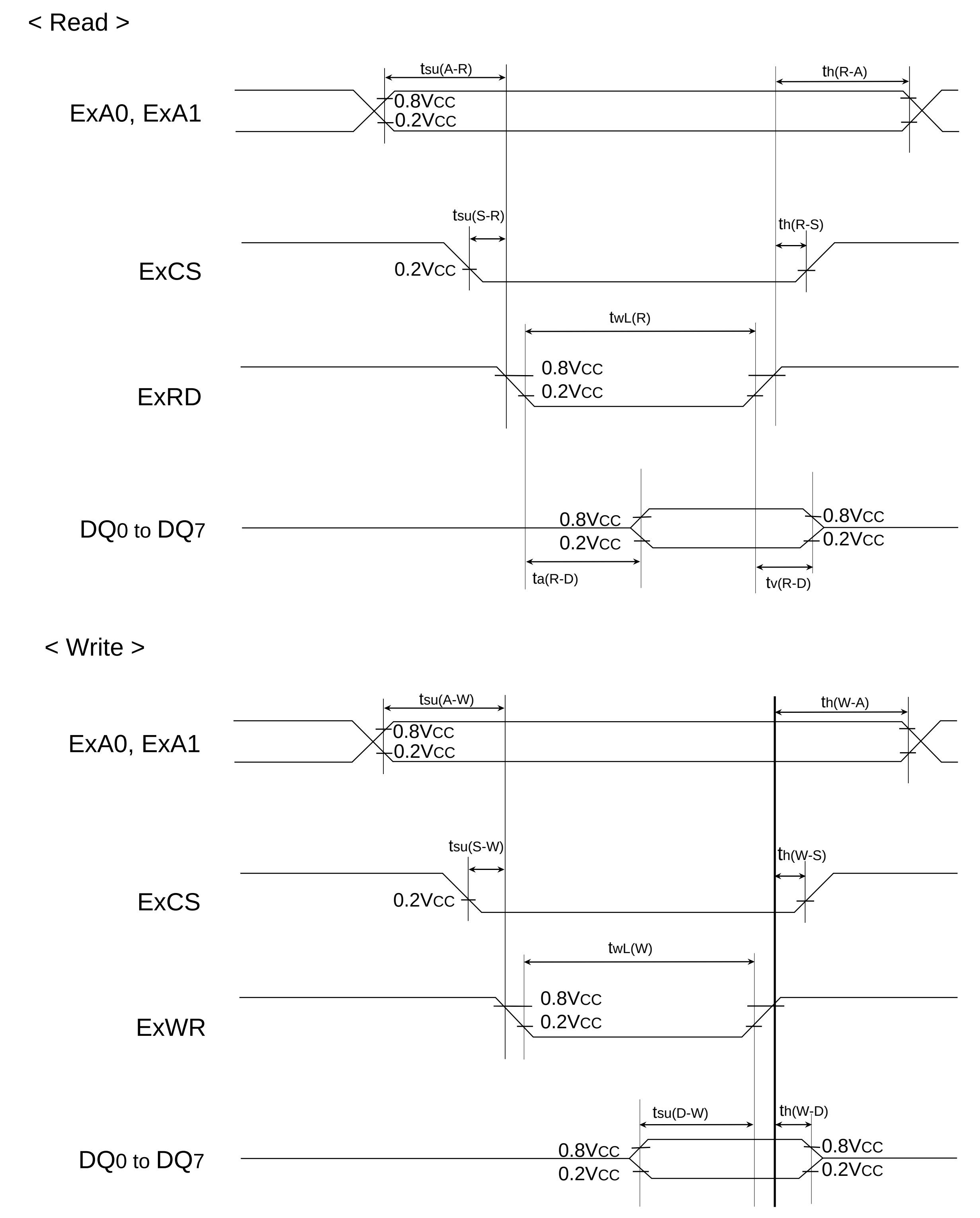


Fig. 3.1.5 Timing chart (2)

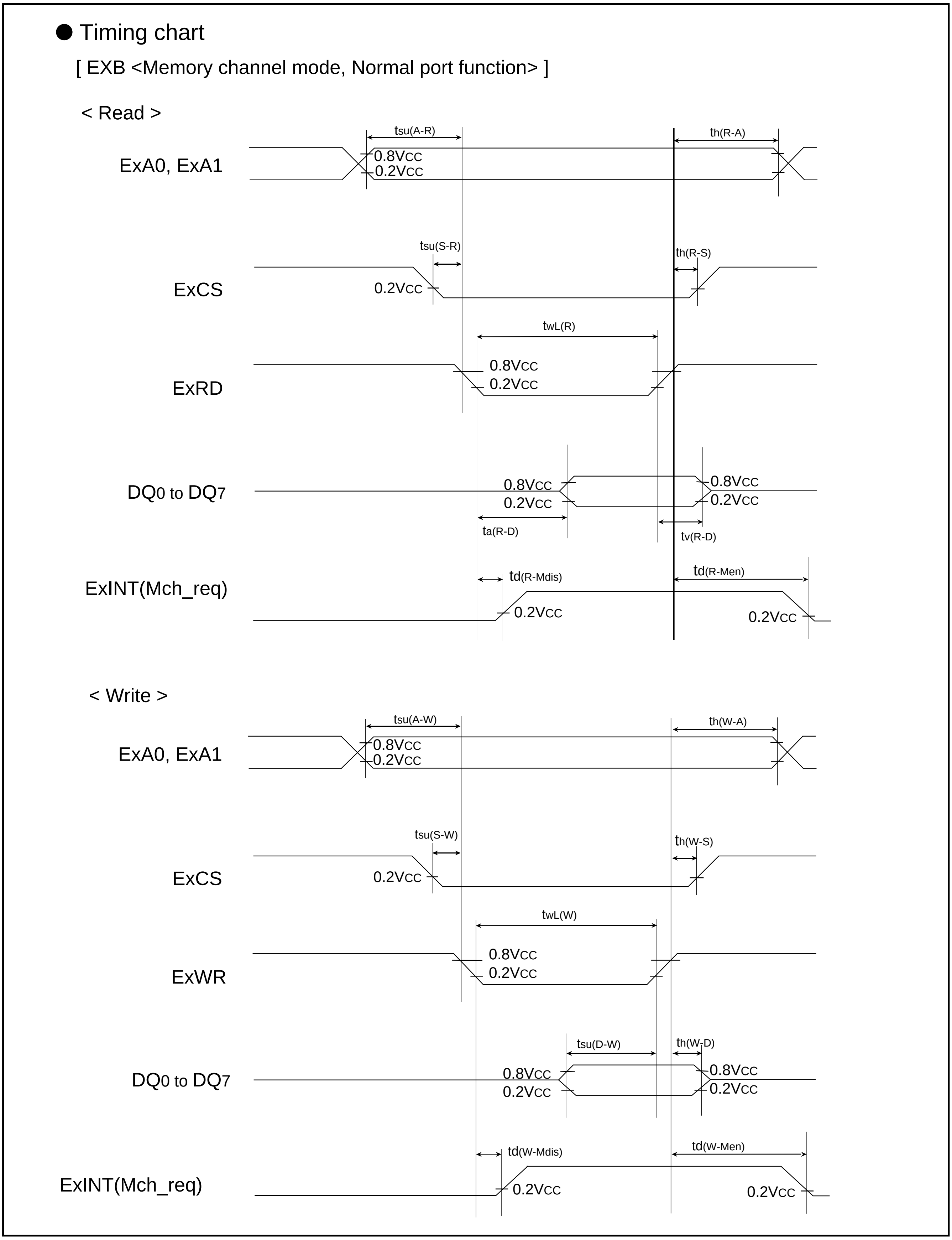


Fig. 3.1.6 Timing chart (3)

● Timing chart

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

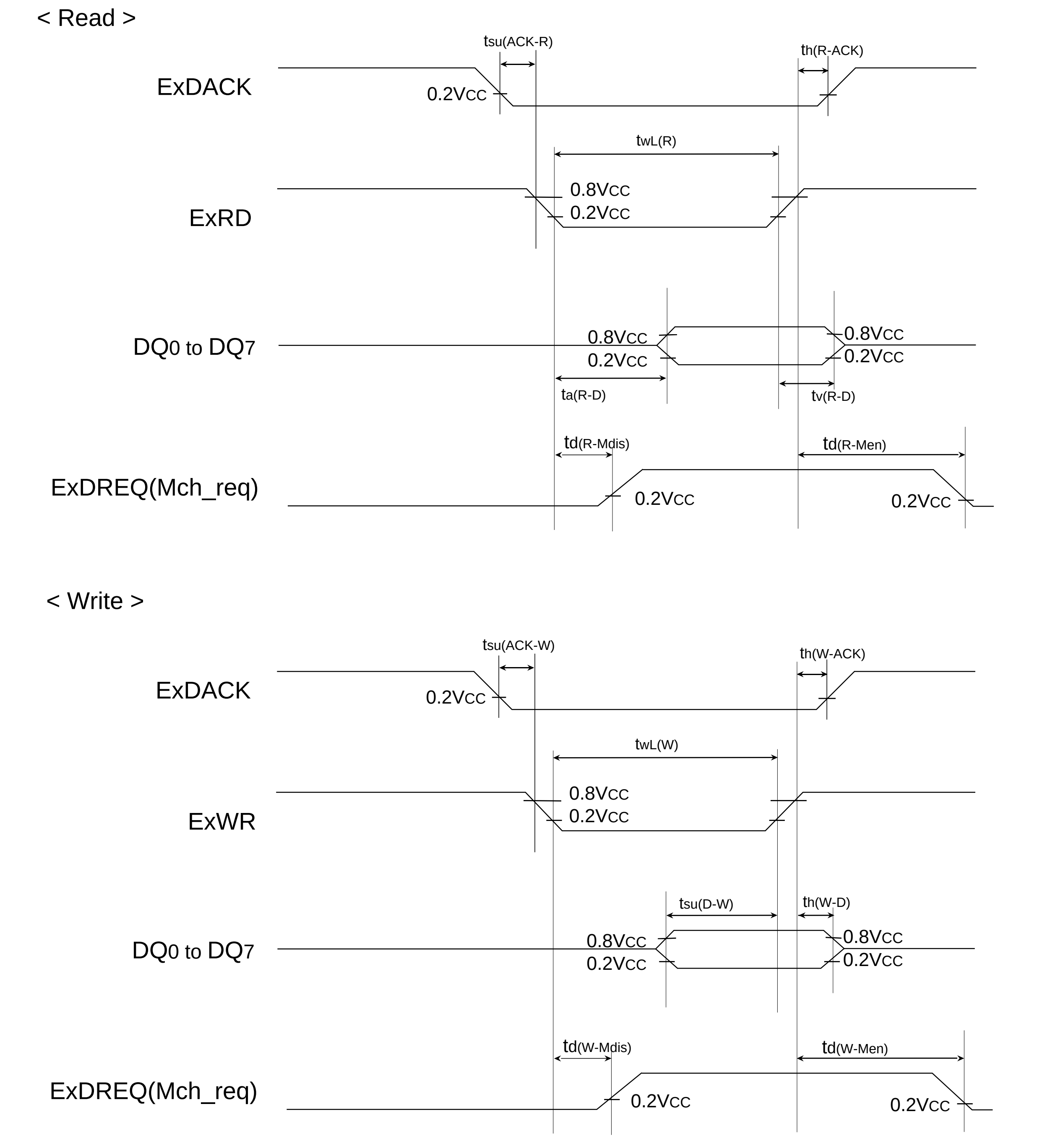


Fig. 3.1.7 Timing chart (4)

● Timing chart

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

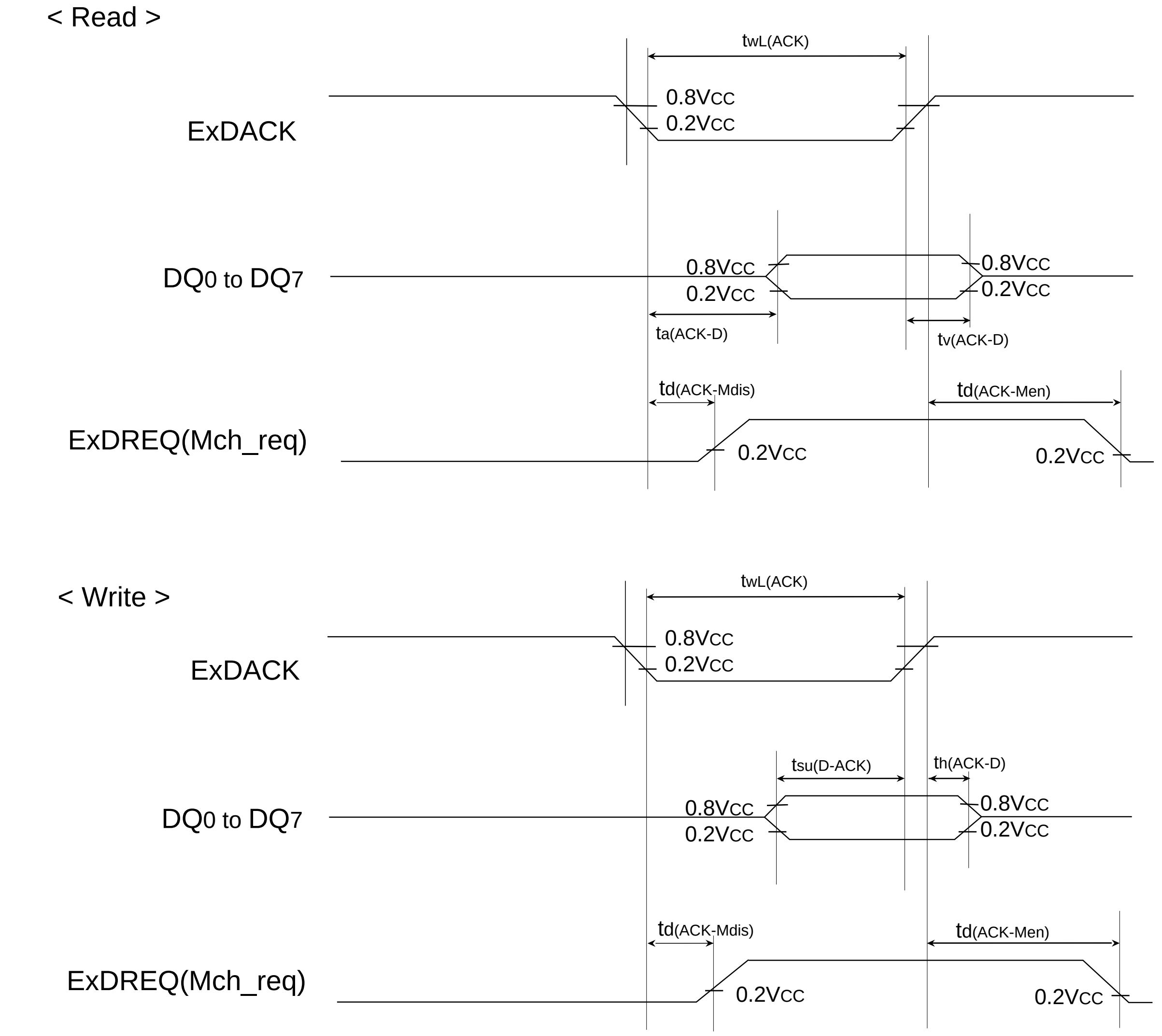


Fig. 3.1.8 Timing chart (5)

● Timing chart

[EXB <Memory channel mode, Burst transfer>]

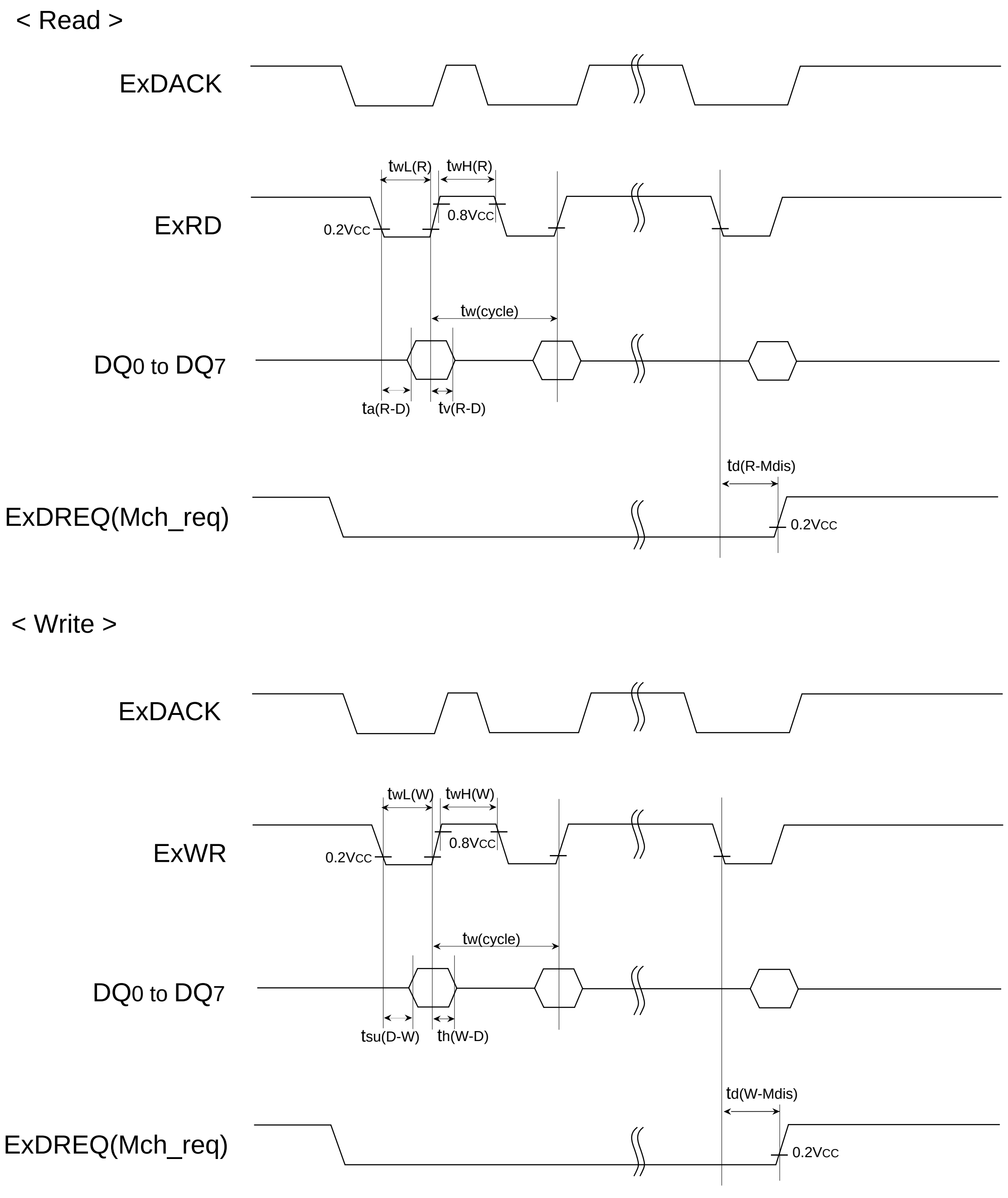


Fig. 3.1.9 Timing chart (6)

3.2 Notes on use

3.2.1 Notes on input and output ports

(1) Modifying output data with bit managing instruction

When the port latch of an I/O port is modified with the bit managing instruction*1, the value of the unspecified bit may be changed.

● **Reason**

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the port latch.

•As for bit which is set for input port:
The pin state is read in the CPU, and is written to this bit after bit managing.

•As for bit which is set for output port:
The bit value is read in the CPU, and is written to this bit after bit managing.

Note the following:

•Even when a port which is set as an output port is changed for an input port, its port latch holds the output data.

•As for a bit of which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its port latch contents.

*1 Bit managing instructions: **SEB** and **CLB** instructions

3.2.2 Termination of unused pins

(1) Terminate unused pins

① I/O ports :

- Set the I/O ports for the input mode and connect them to Vcc or Vss through each resistor of 1 kΩ to 10 kΩ.
  Set the I/O ports for the output mode and open them at “L” or “H”.
- When opening them in the output mode, the input mode of the initial status remains until the mode of the ports is switched over to the output mode by the program after reset. Thus, the potential at these pins is undefined and the power source current may increase in the input mode. With regard to an effects on the system, thoroughly perform system evaluation on the user side.
- Since the direction register setup may be changed because of a program runaway or noise, set direction registers by program periodically to increase the reliability of program.

(2) Termination remarks

① I/O ports :

Do not open in the input mode.

● **Reason**

- The power source current may increase depending on the first-stage circuit.
- An effect due to noise may be easily produced as compared with proper termination shown on the above.

② I/O ports :

When setting for the input mode, do not connect to Vcc or Vss directly.

● **Reason**

If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between a port and Vcc (or Vss).

③ I/O ports :

When setting for the input mode, do not connect multiple ports in a lump to Vcc or Vss through a resistor.

● **Reason**

If the direction register setup changes for the output mode because of a program runaway or noise, a short circuit may occur between ports.

- At the termination of unused pins, perform wiring at the shortest possible distance (20 mm or less) from microcomputer pins.

3.2.3 Notes on interrupts

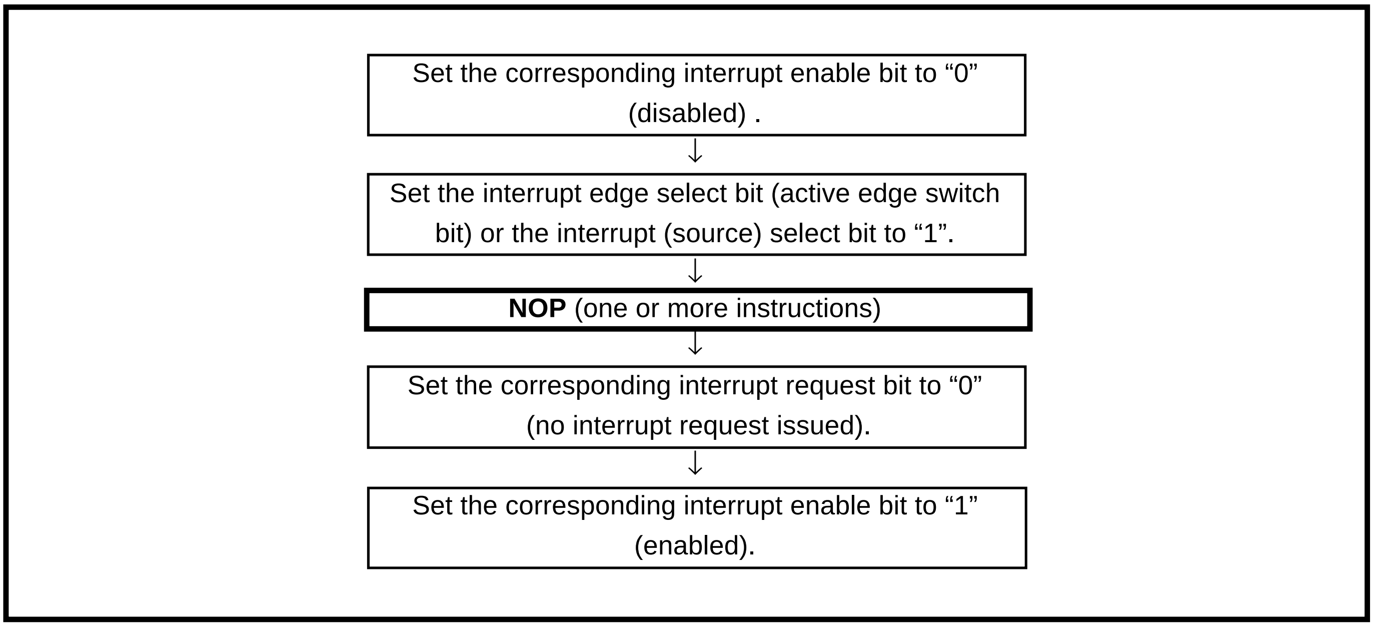
(1) Change of relevant register settings

When the setting of the following registers or bits is changed, the interrupt request bit may be set to “1”. When not requiring the interrupt occurrence synchronized with these setting, take the following sequence.

•Interrupt edge selection register (address $0FF3_{16}$)

•Timer X mode register (address 23_{16})

Set the above listed registers or bits as the following sequence.



Set the corresponding interrupt enable bit to “0” (disabled) .

↓

Set the interrupt edge select bit (active edge switch bit) or the interrupt (source) select bit to “1”.

↓

NOP (one or more instructions)

↓

Set the corresponding interrupt request bit to “0” (no interrupt request issued).

↓

Set the corresponding interrupt enable bit to “1” (enabled).

Fig. 3.2.1 Sequence of changing relevant register

■ **Reason**

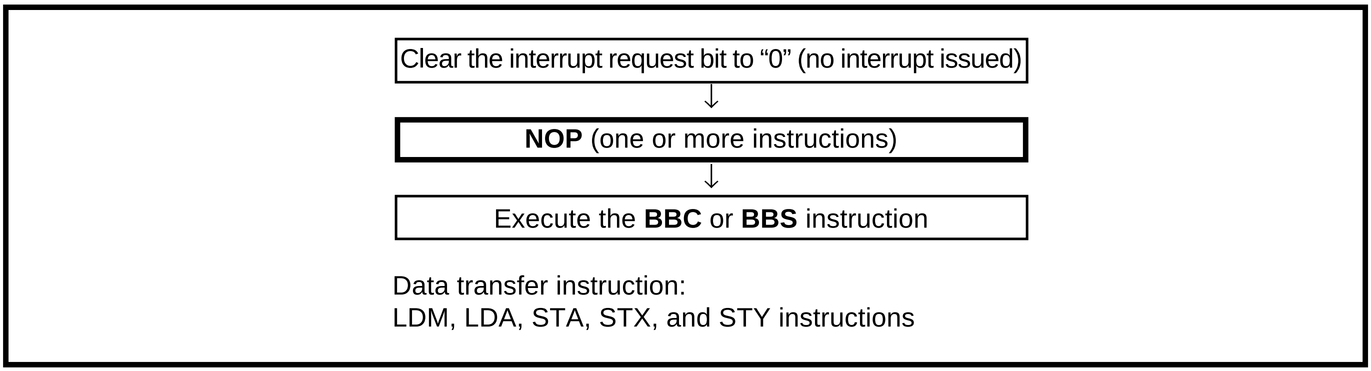
When setting the following, the interrupt request bit may be set to “1”.

•When setting external interrupt active edge

Concerned register: Interrupt edge selection register (address $0FF3_{16}$)
Timer X mode register (address 23_{16})

(2) Check of interrupt request bit

- When executing the **BBC** or **BBS** instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to "0" by using a data transfer instruction, execute one or more instructions before executing the **BBC** or **BBS** instruction.


Clear the interrupt request bit to "0" (no interrupt issued)
NOP (one or more instructions)
Execute the BBC or BBS instruction

Data transfer instruction:
LDM, LDA, STA, STX, and STY instructions

Fig. 3.2.2 Sequence of check of interrupt request bit

■ **Reason**

If the BBC or BBS instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to "0", the value of the interrupt request bit before being cleared to "0" is read.

3.2.4 Notes on timer

- If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).
- When switching the count source by the timer 12 and X count source selection bits, the value of timer count is altered in unconsiderable amount owing to generating of thin pulses in the count input signals.

  Therefore, select the timer count source before set the value to the prescaler and the timer.

3.2.5 Notes on serial I/O

(1) Notes when selecting clock synchronous serial I/O (Serial I/O)

① **Stop of transmission operation**
Clear the serial I/O enable bit and the transmit enable bit to “0” (Serial I/O and transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (Serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled), or clear the serial I/O enable bit to “0” (Serial I/O disabled).

③ **Stop of transmit/receive operation**
Clear the transmit enable bit and receive enable bit to “0” simultaneously (transmit and receive disabled).
(when data is transmitted and received in the clock synchronous serial I/O mode, any one of data transmission and reception cannot be stopped.)

● **Reason**
In the clock synchronous serial I/O mode, the same clock is used for transmission and reception. If any one of transmission and reception is disabled, a bit error occurs because transmission and reception cannot be synchronized.
In this mode, the clock circuit of the transmission circuit also operates for data reception. Accordingly, the transmission circuit does not stop by clearing only the transmit enable bit to “0” (transmit disabled). Also, the transmission circuit is not initialized by clearing the serial I/O enable bit to “0” (Serial I/O disabled) (refer to (1) ①).

(2) Notes when selecting clock asynchronous serial I/O (Serial I/O)

① **Stop of transmission operation**
Clear the transmit enable bit to “0” (transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (Serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

② **Stop of receive operation**
Clear the receive enable bit to “0” (receive disabled).

③ **Stop of transmit/receive operation**
Only transmission operation is stopped.
Clear the transmit enable bit to “0” (transmit disabled).

● **Reason**
Since transmission is not stopped and the transmission circuit is not initialized even if only the serial I/O enable bit is cleared to “0” (Serial I/O disabled), the internal transmission is running (in this case, since pins TxD, RxD, SCLK, and SRDY function as I/O ports, the transmission data is not output). When data is written to the transmit buffer register in this state, data starts to be shifted to the transmit shift register. When the serial I/O enable bit is set to “1” at this time, the data during internally shifting is output to the TxD pin and an operation failure occurs.

Only receive operation is stopped.
Clear the receive enable bit to “0” (receive disabled).

(3) $\overline{\text{SRDY}}$ output of reception side (Serial I/O)

When signals are output from the $\overline{\text{SRDY}}$ pin on the reception side by using an external clock in the clock synchronous serial I/O mode, set all of the receive enable bit, the $\overline{\text{SRDY}}$ output enable bit, and the transmit enable bit to “1” (transmit enabled).

(4) Setting serial I/O control register again (Serial I/O)

Set the serial I/O control register again after the transmission and the reception circuits are reset by clearing both the transmit enable bit and the receive enable bit to “0.”

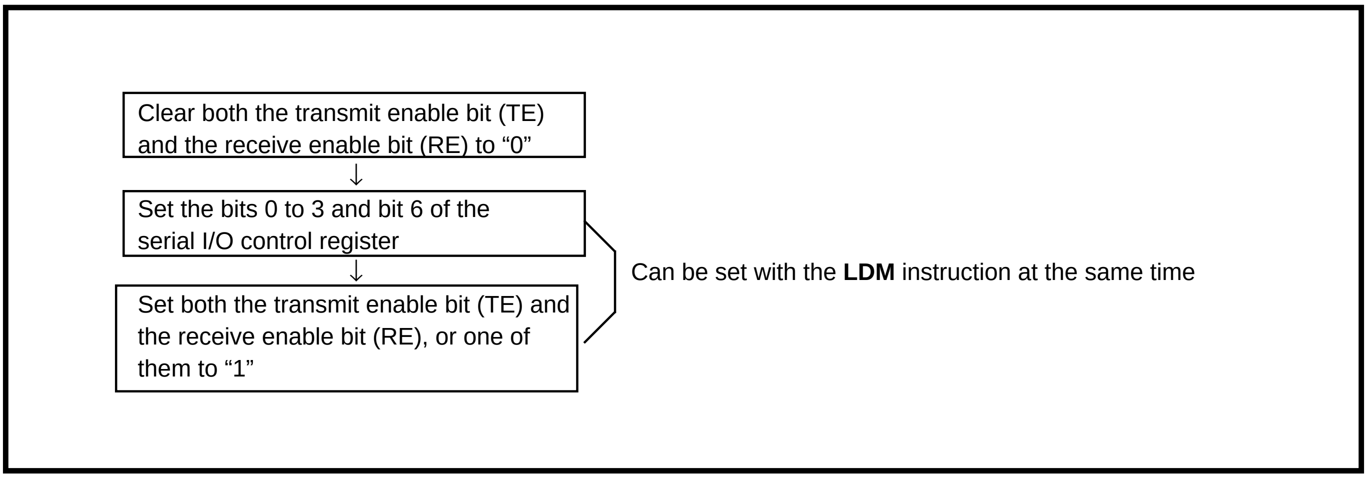

Clear both the transmit enable bit (TE) and the receive enable bit (RE) to “0”
↓
Set the bits 0 to 3 and bit 6 of the serial I/O control register
↓
Set both the transmit enable bit (TE) and the receive enable bit (RE), or one of them to “1”
Can be set with the **LDM** instruction at the same time

Fig. 3.2.3 Sequence of setting serial I/O control register again

(5) Data transmission control with referring to transmit shift register completion flag (Serial I/O)

The transmit shift register completion flag changes from “1” to “0” with a delay of 0.5 to 1.5 shift clocks. When data transmission is controlled with referring to the flag after writing the data to the transmit buffer register, note the delay.

(6) Transmission control when external clock is selected (Serial I/O)

When an external clock is used as the synchronous clock for data transmission, set the transmit enable bit to “1” at “H” of the SCLK input level. Also, write the transmit data to the transmit buffer register (serial I/O shift register) at “H” of the SCLK input level.

(7) Transmit interrupt request when transmit enable bit is set (Serial I/O)

When the transmit interrupt is used, set the transmit interrupt enable bit to transmit enabled as shown in the following sequence.

① Set the interrupt enable bit to “0” (disabled) with CLB instruction.
② Prepare serial I/O for transmission/reception.
③ Set the interrupt request bit to “0” with CLB instruction after 1 or more instruction has been executed.
④ Set the interrupt enable bit to “1” (enabled).

● Reason

When the transmission enable bit is set to “1”, the transmit buffer empty flag and transmit shift register completion flag are set to “1”. The interrupt request is generated and the transmission interrupt bit is set regardless of which of the two timings listed below is selected as the timing for the transmission interrupt to be generated.

- Transmit buffer empty flag is set to “1”
- Transmit shift register completion flag is set to “1”

3.2.6 Notes on USB function

(1) Port pins (D0+, D0-, D1+, D1-, D2+, D2-) 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 W 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.

(2) USBV$_{REF}$ pin treatment (Noise Elimination)

- Connect a capacitor between the USBV$_{REF}$ 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 USBV$_{REF}$ 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 USBV$_{REF}$ pin. Use the built-in USB reference voltage circuit.

(3) 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.

3.2.7 Notes on A/D converter

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 µF to 1 µF. Further, be sure to verify the operation of application products on the user side.

● **Reason**

An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A/D conversion precision to be worse.

(2) Clock frequency during A/D conversion

The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A/D conversion.

- f(X_{IN}) is 500 kHz or more
- Do not execute the **STP** instruction

3.2.8 Notes on watchdog timer

●Make sure that the watchdog timer does not underflow while waiting Stop release, because the watchdog timer keeps counting during that term.

●When the STP instruction disable bit has been set to “1”, it is impossible to switch it to “0” by a program

3.2.9 Notes on $\overline{\text{RESET}}$ pin

Connecting capacitor

In case where the $\overline{\text{RESET}}$ signal rise time is long, connect a ceramic capacitor or others across the $\overline{\text{RESET}}$ pin and the Vss pin. Use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to verify the operation of application products on the user side.

● **Reason**

If the several nanosecond or several ten nanosecond impulse noise enters the $\overline{\text{RESET}}$ pin, it may cause a microcomputer failure.

3.2.10 Notes on PLL

●6 MHz or 12 MHz external oscillator can be connected as an input reference clock (f(X_{IN})). When using the frequency synthesized clock function, we recommend using the fastest frequency possible of f(X_{IN}) as an input clock reference for the PLL.

●When enabling PLL operation from PLL disabled status (disabled when reset), set the USB clock select bit of USBCON to “0” (f(X_{IN})) to operate with the main clock (f(X_{IN})).

●When supplying f_{VCO} to the USB block after setting PLL operation enable bit to “1” (PLL enabled), wait for the oscillation stable time (1 ms or less) of PLL to avoid any instability caused by the clock, then set USB clock select bit to “1” (USB clock).

●When selecting f_{SYN} as an internal system clock, f_{USB} must be 48 MHz.

●When selecting f_{SYN} as an internal system clock, change the system clock selection bit to main clock (f(X_{IN})) before executing STP instruction. It is because the following are needed for the low-power consumption:

•f_{USB} must be stopped by disabling PLL operation in Stop mode.

•The taimer 1 for waiting oscillation stabilization when returning from Stop mode will require the input count source.

3.2.11 Notes on stand-by function

(1) Notes on using stop mode

■Register setting

Since values of the prescaler 12 and Timer 1 are automatically reloaded when returning from the stop mode, set them again, respectively. (When the oscillation stabilizing time set after STP instruction released bit is “0”)

■Clock restoration

When the main clock side is set as a system clock, the oscillation stabilizing time for approximately 8,000 cycles of the X_{IN} input is reserved at restoration from the stop mode.

(2) Notes on stand-by function

In stand-by state*[1] for low-power dissipation, do not make input levels of an input port and an I/O port “undefined”.
Pull-up (connect the port to V_{CC}) these ports through a resistor.
When determining a resistance value, note the following points:

- External circuit
- Variation of output levels during the ordinary operation

When using built-in pull-up resistor, note on varied current values.

- When setting as an input port: Fix its input level
- When setting as an output port: Prevent current from flowing out to external

● Reason

The potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of an input port and an I/O port are “undefined”. This may cause power source current.

*[1] stand-by state : the stop mode by executing the **STP** instruction
the wait mode by executing the **WIT** instruction

3.2.12 Notes on CPU rewrite mode

(1) Operation speed

During CPU rewrite mode, set the internal clock ϕ 1.5 MHz or less using the system clock division ratio selection bits (bits 6 and 7 of address $003B_{16}$).

(2) Instructions inhibited against use

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

(3) Interrupts inhibited against use

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

(4) Watchdog timer

In case of the watchdog timer has been running already, the internal reset generated by watchdog timer underflow does not happen, because of watchdog timer is always clearing during program or erase operation.

(5) Reset

Reset is always valid. In case of CNV_{SS} = “H” when reset is released, boot mode is active. So the program starts from the address contained in address $FFFC_{16}$ and $FFFD_{16}$ in boot ROM area.

3.2.13 Notes on programming

(1) Processor status register

① Initializing of processor status register

Flags which affect program execution must be initialized after a reset.
In particular, it is essential to initialize the T and D flags because they have an important effect on calculations.

● Reason

After a reset, the contents of the processor status register (PS) are undefined except for the I flag which is "1".

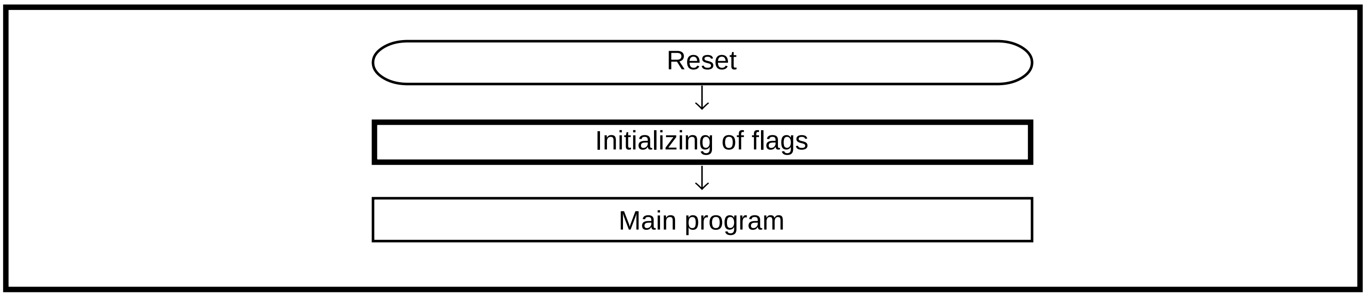


Fig. 3.2.4 Initialization of processor status register

② How to reference the processor status register

To reference the contents of the processor status register (PS), execute the **PHP** instruction once then read the contents of (S+1). If necessary, execute the **PLP** instruction to return the PS to its original status.
A **NOP** instruction should be executed after every **PLP** instruction.

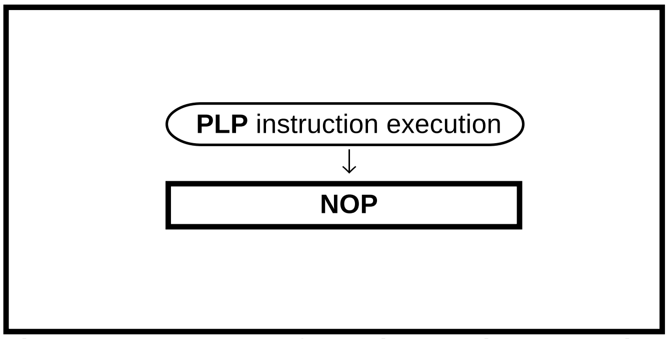


Fig. 3.2.5 Sequence of PLP instruction execution

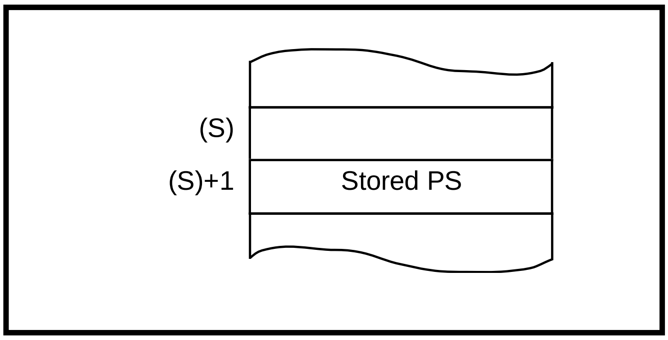


Fig. 3.2.6 Stack memory contents after PHP instruction execution

(2) BRK instruction

① Interrupt priority level

When the BRK instruction is executed with the following conditions satisfied, the interrupt execution is started from the address of interrupt vector which has the highest priority.

- Interrupt request bit and interrupt enable bit are set to "1".
- Interrupt disable flag (I) is set to "1" to disable interrupt.

(3) Decimal calculations

① **Execution of decimal calculations**
The **ADC** and **SBC** are the only instructions which will yield proper decimal notation, set the decimal mode flag (D) to “1” with the **SED** instruction. After executing the **ADC** or **SBC** instruction, execute another instruction before executing the **SEC**, **CLC**, or **CLD** instruction.

② **Notes on status flag in decimal mode**
When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after a **ADC** or **SBC** instruction is executed.
The carry flag (C) is set to “1” if a carry is generated as a result of the calculation, or is cleared to “0” if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to “0” before each calculation. To check for a borrow, the C flag must be initialized to “1” before each calculation.

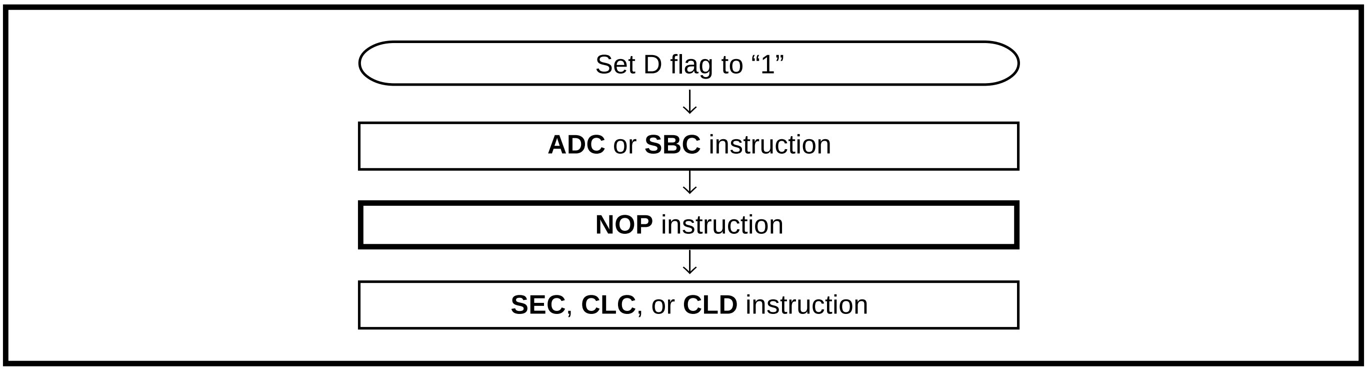


Fig. 3.2.7 Status flag at decimal calculations

(4) JMP instruction

When using the **JMP** instruction in indirect addressing mode, do not specify the last address on a page as an indirect address.

(5) 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.

(6) 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.

(7) Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock *f* by the number of cycles needed to execute an instruction.
The number of cycles required to execute an instruction is shown in the list of machine instructions.
The frequency of the internal clock *f* is half of the X_{IN} frequency in high-speed mode.

3.2.14 Notes on 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 (V_{PP} 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.

3.2.15 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.

3.3 Countermeasures against noise

Countermeasures against noise are described below. The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

3.3.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Package

Select the smallest possible package to make the total wiring length short.

● **Reason**

The wiring length depends on a microcomputer package. Use of a small package, for example QFP and not DIP, makes the total wiring length short to reduce influence of noise.

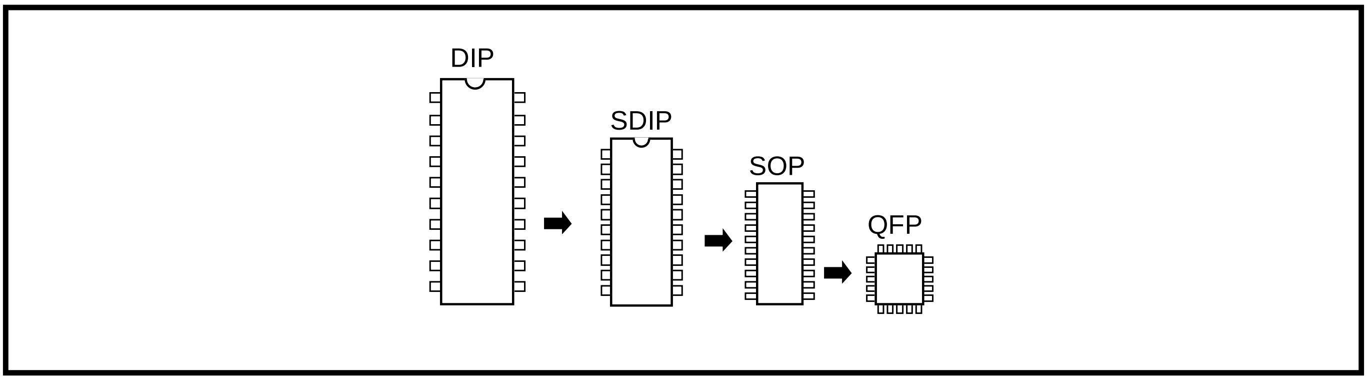


Fig. 3.3.1 Selection of packages

(2) Wiring for $\overline{\text{RESET}}$ pin

Make the length of wiring which is connected to the $\overline{\text{RESET}}$ pin as short as possible. Especially, connect a capacitor across the RESET pin and the Vss pin with the shortest possible wiring (within 20mm).

● **Reason**

The width of a pulse input into the $\overline{\text{RESET}}$ pin is determined by the timing necessary conditions. If noise having a shorter pulse width than the standard is input to the RESET pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

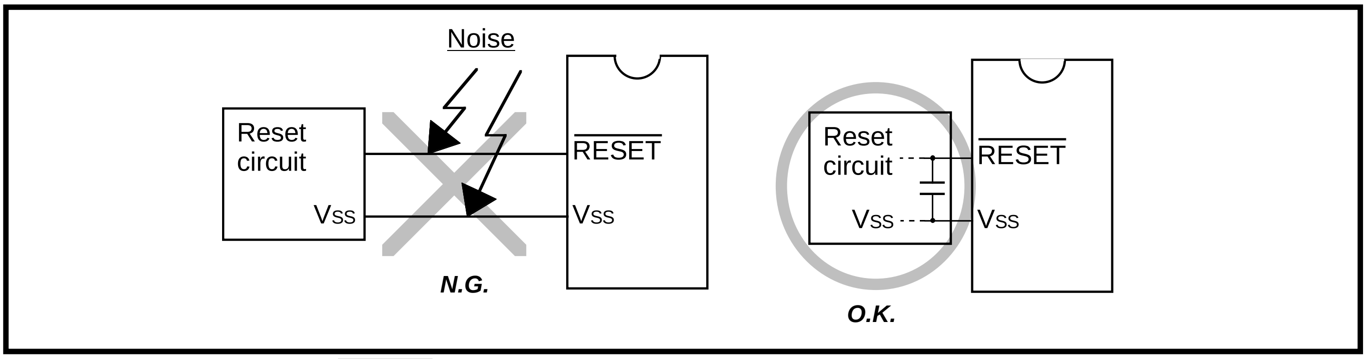


Fig. 3.3.2 Wiring for the $\overline{\text{RESET}}$ pin

(3) Wiring for clock input/output pins

- Make the length of wiring which is connected to clock I/O pins as short as possible.
- Make the length of wiring (within 20 mm) across the grounding lead of a capacitor which is connected to an oscillator and the Vss pin of a microcomputer as short as possible.
- Separate the Vss pattern only for oscillation from other Vss patterns.

● **Reason**

If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a program failure or program runaway. Also, if a potential difference is caused by the noise between the Vss level of a microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

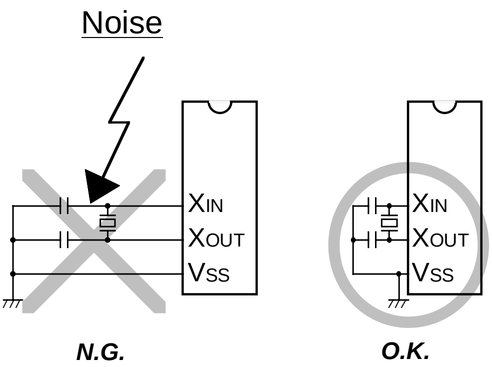


Fig. 3.3.3 Wiring for clock I/O pins

(4) Wiring to CNVss pin

Connect the CNVss pin to the Vss pin with the shortest possible wiring.

● **Reason**

The processor mode of a microcomputer is influenced by a potential at the CNVss pin. If a potential difference is caused by the noise between pins CNVss and Vss, the processor mode may become unstable. This may cause a microcomputer malfunction or a program runaway.

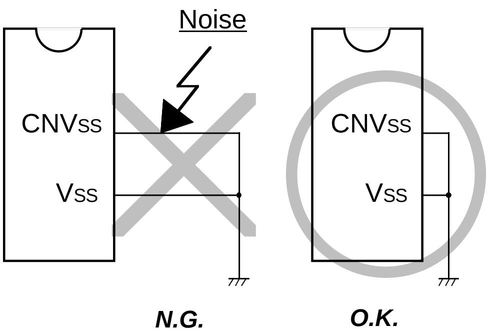


Fig. 3.3.4 Wiring for CNVss pin

(5) Wiring to VPP pin of Flash memory version

Connect an approximately 5 kΩ resistor to the VPP pin the shortest possible in series and also to the VSS pin. When not connecting the resistor, make the length of wiring between the VPP pin and the VSS pin the shortest possible.

Note: Even when a circuit which included an approximately 5 kΩ resistor is used in the Mask ROM version, the microcomputer operates correctly.

● **Reason**

The VPP pin of the flash memory version is the power source input pin for the built-in flash memory. When programming in the built-in flash memory, the impedance of the VPP pin is low to allow the electric current for writing flow into the flash memory. Because of this, noise can enter easily. If noise enters the VPP pin, abnormal instruction codes or data are read from the built-in flash memory, which may cause a program runaway.

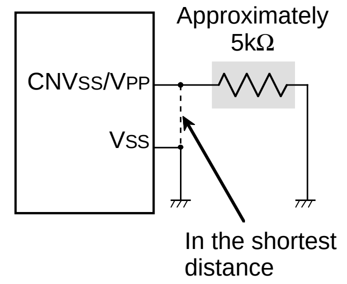


Fig. 3.3.5 Wiring for the VPP pin of the flash memory version

3.3.2 Connection of bypass capacitor across Vss line and Vcc line

Connect an approximately 0.1 μF bypass capacitor across the Vss line and the Vcc line as follows:

- Connect a bypass capacitor across the Vss pin and the Vcc pin at equal length.
- Connect a bypass capacitor across the Vss pin and the Vcc pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for Vss line and Vcc line.
- Connect the power source wiring via a bypass capacitor to the Vss pin and the Vcc pin.

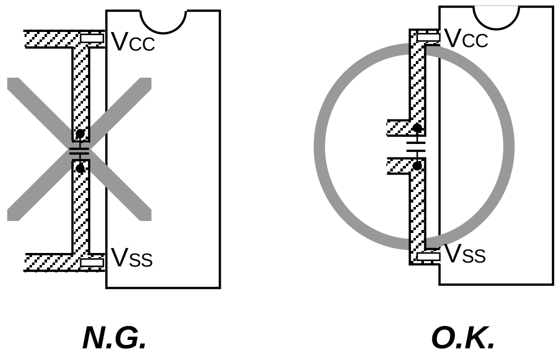


Fig. 3.3.6 Bypass capacitor across the Vss line and the Vcc line

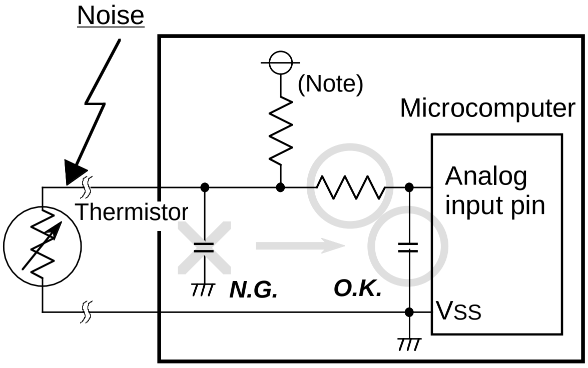
3.3.3 Wiring to analog input pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 1000 pF capacitor across the VSS pin and the analog input pin. Besides, connect the capacitor to the VSS pin as close as possible. Also, connect the capacitor across the analog input pin and the VSS pin at equal length.

● Reason

Signals which is input in an analog input pin (such as an A/D converter/comparator input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

If a capacitor between an analog input pin and the V_{SS} pin is grounded at a position far away from the V_{SS} pin, noise on the GND line may enter a microcomputer through the capacitor.



Note : The resistor is used for dividing resistance with a thermistor.

Fig. 3.3.7 Analog signal line and a resistor and a capacitor

3.3.4 Oscillator concerns

Take care to prevent an oscillator that generates clocks for a microcomputer operation from being affected by other signals.

(1) Keeping oscillator away from large current signal lines

Install a microcomputer (and especially an oscillator) as far as possible from signal lines where a current larger than the tolerance of current value flows.

● **Reason**

In the system using a microcomputer, there are signal lines for controlling motors, LEDs, and thermal heads or others. When a large current flows through those signal lines, strong noise occurs because of mutual inductance.

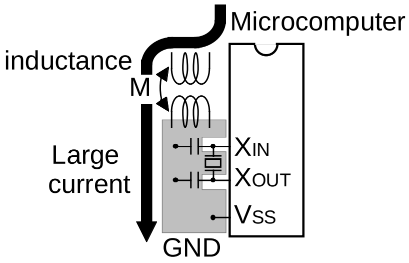


Fig. 3.3.8 Wiring for a large current signal line

(2) Installing oscillator away from signal lines where potential levels change frequently

Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

● **Reason**

Signal lines where potential levels change frequently (such as the CNTR pin signal line) may affect other lines at signal rising edge or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

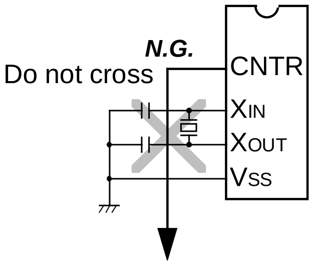


Fig. 3.3.9 Wiring of RESET pin

(3) Oscillator protection using Vss pattern

As for a two-sided printed circuit board, print a Vss pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.

Connect the Vss pattern to the microcomputer Vss pin with the shortest possible wiring. Besides, separate this Vss pattern from other Vss patterns.

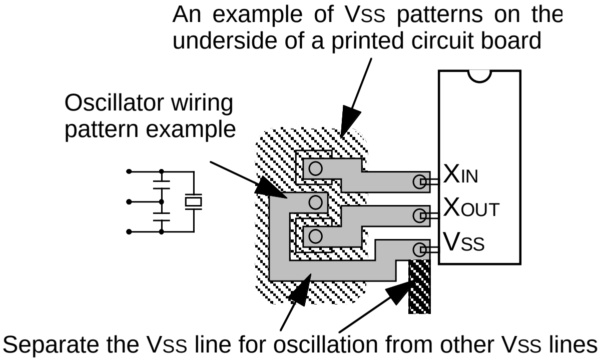

An example of Vss patterns on the underside of a printed circuit board
Oscillator wiring pattern example
XIN
XOUT
VSS
Separate the Vss line for oscillation from other Vss lines

Fig. 3.3.10 Vss pattern on the underside of an oscillator

3.3.5 Setup for I/O ports

Setup I/O ports using hardware and software as follows:

<Hardware>

- Connect a resistor of 100 Ω or more to an I/O port in series.

<Software>

- As for an input port, read data several times by a program for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its port latch at fixed periods.
- Rewrite data to direction registers at fixed periods.

Note: When a direction register is set for input port again at fixed periods, a several-nanosecond short pulse may be output from this port. If this is undesirable, connect a capacitor to this port to remove the noise pulse.

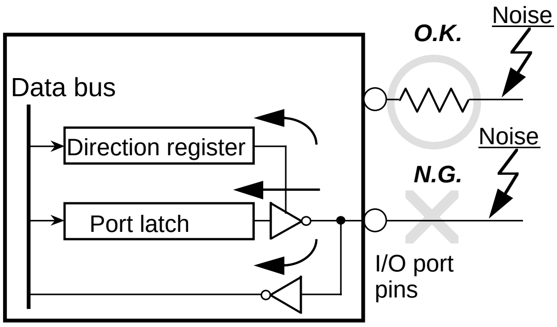


Fig. 3.3.11 Setup for I/O ports

3.3.6 Providing of watchdog timer function by software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine. This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:
  N+1 ≥ (Counts of interrupt processing executed in each main routine)
  As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.
- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
  If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

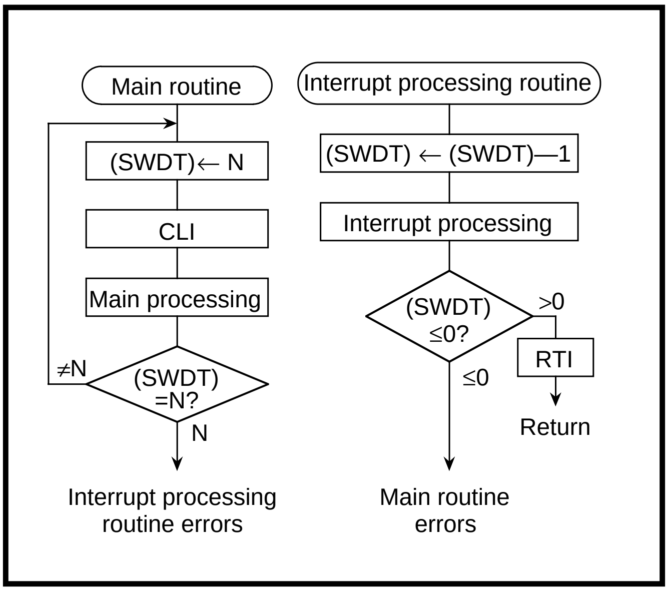


Fig. 3.3.12 Watchdog timer by software

3.4 List of registers

Port Pi

b7 b6 b5 b4 b3 b2 b1 b0

Port Pi (Pi) (i = 0, 1, 2, 3, 4, 5, 6) **(Note)**
[Address : 00$_{16}$, 02$_{16}$, 04$_{16}$, 06$_{16}$, 08$_{16}$, 0A$_{16}$, 0C$_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi_0	● In output mode Write } Port latch Read } ● In input mode Write : Port latch Read : Value of pins	?	○	○
1	Port Pi_1		?	○	○
2	Port Pi_2		?	○	○
3	Port Pi_3		?	○	○
4	Port Pi_4		?	○	○
5	Port Pi_5		?	○	○
6	Port Pi_6		?	○	○
7	Port Pi_7		?	○	○

Note: Since the following ports are not allocated, the corrrsponding bits can not be used.
- P2$_0$ to P2$_3$
- P4$_4$ to P4$_7$
- P6$_4$ to P6$_7$

Fig. 3.4.1 Structure of Port Pi

Port Pi direction register

b7 b6 b5 b4 b3 b2 b1 b0

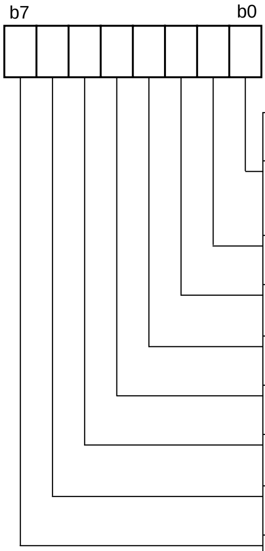
Port Pi direction register (PiD) (i = 0, 1, 2, 3, 4, 5, 6) **(Note)**
[Address : 01$_{16}$, 03$_{16}$, 05$_{16}$, 07$_{16}$, 09$_{16}$, 0B$_{16}$, 0D$_{16}$]

B	Name	Function	At reset	R	W
0	Port Pi direction register	0 : Port Pi_0 input mode 1 : Port Pi_0 output mode	0	×	○
1		0 : Port Pi_1 input mode 1 : Port Pi_1 output mode	0	×	○
2		0 : Port Pi_2 input mode 1 : Port Pi_2 output mode	0	×	○
3		0 : Port Pi_3 input mode 1 : Port Pi_3 output mode	0	×	○
4		0 : Port Pi_4 input mode 1 : Port Pi_4 output mode	0	×	○
5		0 : Port Pi_5 input mode 1 : Port Pi_5 output mode	0	×	○
6		0 : Port Pi_6 input mode 1 : Port Pi_6 output mode	0	×	○
7		0 : Port Pi_7 input mode 1 : Port Pi_7 output mode	0	×	○

Note: Since the following ports are not allocated, the corrrsponding bits can not be used.
- P2$_0$ to P2$_3$
- P4$_4$ to P4$_7$
- P6$_4$ to P6$_7$

Do not set bits of the direction register corresponding to ports P2$_0$–P2$_3$ (bits 0–3 of port P2 direction register (address 05$_{16}$)) to output mode (“1”).
If writing to these bits, write “0”.

Fig. 3.4.2 Structure of Port Pi direction register

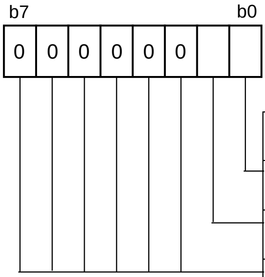


USB control register (USBCON) [address 0010_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.3 Structure of USB control register

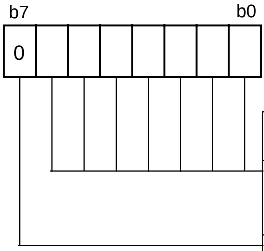


USB function/HUB enable register (USBAE) [address 0011_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
AD0E	USB function enable bit	0: USB function address register invalidated 1: USB function address register validated	0	–	O	O
AD1E	USB HUB enable bit	0: USB HUB address register invalidated 1: USB HUB address register validated	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.4 Structure of USB function/HUB enable register

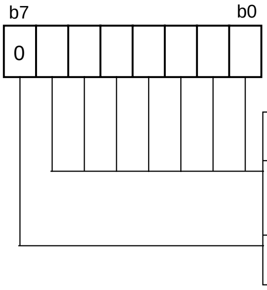


USB function address register (USBA0) [address 0012_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.5 Structure of USB function address register

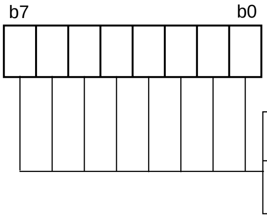


USB HUB address register (USBA1) [address 0013$_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
USBADD1 [6:0]	USB HUB address bit	In AD1E = “0”, this value changes after writing. In AD1E = “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

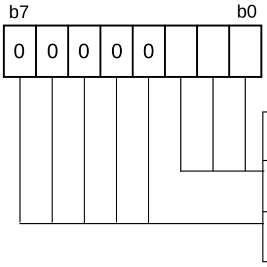
Fig. 3.4.6 Structure of USB HUB address register



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. 3.4.7 Structure of Frame number register Low

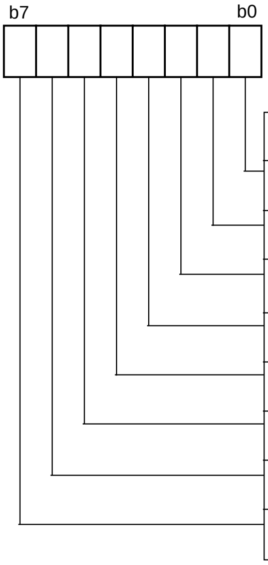


Frame number register High (FNUMH) [address 0015$_{16}$]

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

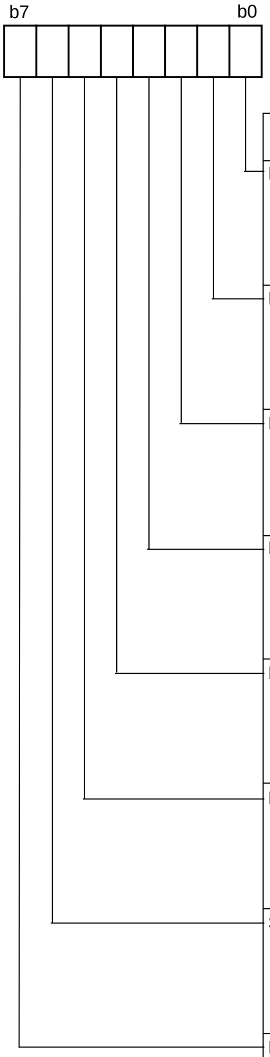
Fig. 3.4.8 Structure of Frame number register High



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
EP10E	USB HUB/Endpoint 0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	O	O
EP11E	USB HUB/Endpoint 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	0	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

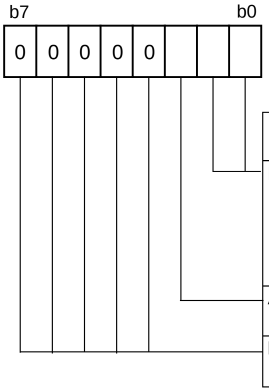
Fig. 3.4.9 Structure of USB interrupt source enable register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	×
EP10	USB HUB/Endpoint 0 interrupt bit	This bit is set to “1” when any one of EP10 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP10 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	O	×
EP11	USB HUB/Endpoint 1 interrupt bit	This bit is set to “1” when any one of EP11 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing EP11 interrupt source register to “00_{16}”. Writing to this bit causes no state change.	0	0	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	×

Fig. 3.4.10 Structure of USB interrupt source register

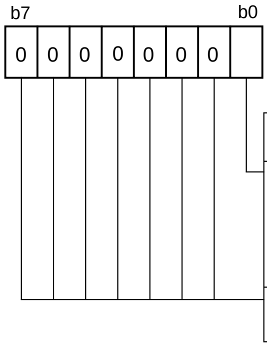


Endpoint index register (USBINDEX) [address 0018_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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
ADIDX	Address index bit	0 : USB function 1 : USB HUB	0	–	O	O
b7:b3	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.11 Structure of Endpoint index register

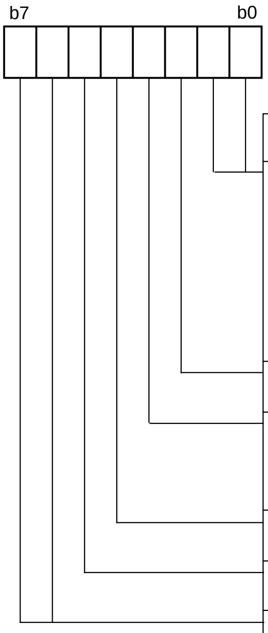


EP00 stage register (EP00STG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.12 Structure of EP00 stage register

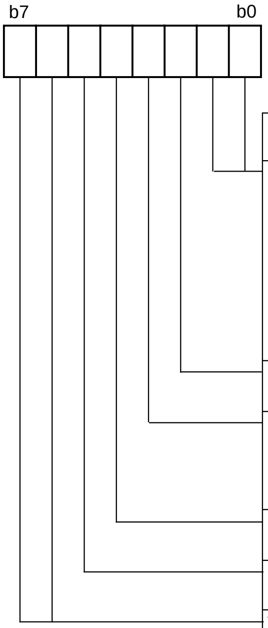


EP01 set register (EP01CFG) [address 0019$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.13 Structure of EP01 set register

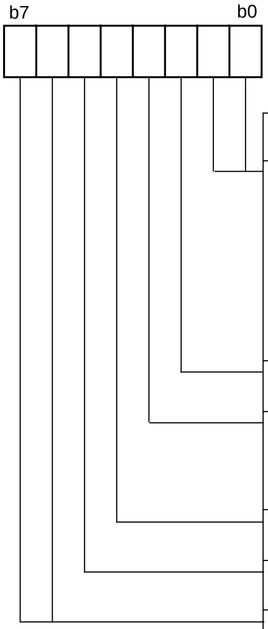


EP02 set register (EP02CFG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.14 Structure of EP02 set register

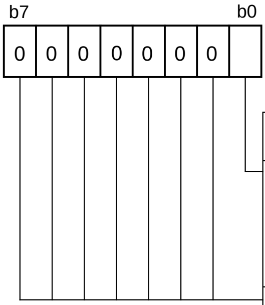


EP03 set register (EP03CFG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.15 Structure of EP03 set register

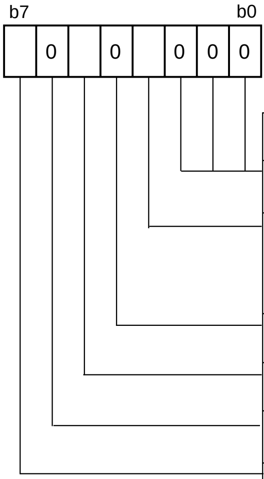


EP10 stage register (EP10STG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
SETUP10	SETUP packet detection bit	This bit is set to “1” at reception of SETUP packet. Writing “0” clears this bit if the next SETUP token does not occur. Writing “1” causes no state change of the status flags. This bit change is not for an interrupt source.	1	1	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.16 Structure of EP10 stage register

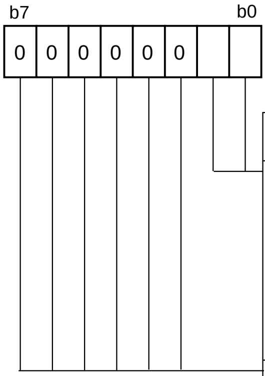


EP11 set register (EP11CFG) [address 0019_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
b2:b0	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
SQCL11	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	–		
b4	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
DIR11	Transfer direction bit	0 : IN transfer disabled 1 : IN (Data is transmitted to the host.)	0	–	O	O
b6	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
TYP11	Transfer type bite	0 : Transfer disabled 1 : Interrupt transfer	0	–	O	O

–: State remaining

Fig. 3.4.17 Structure of EP11 set register

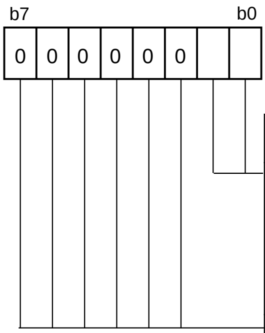


EP00 control register 1 (EP00CON1) [address $001A_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.18 Structure of EP00 control register 1

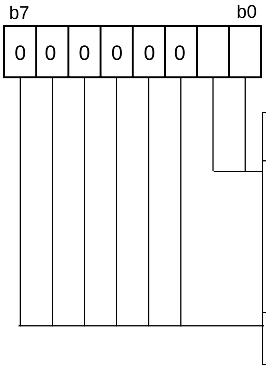


EP01 control register 1 (EP01CON1) [address 001A16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.19 Structure of EP01 control register 1

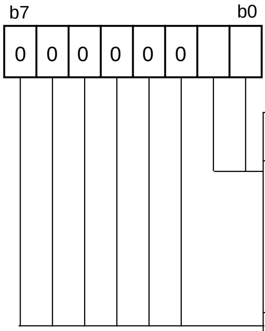


EP02 control register 1 (EP02CON1) [address 001A16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.20 Structure of EP02 control register 1

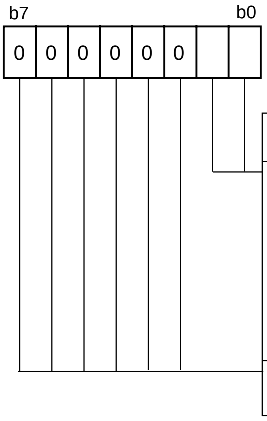


EP03 control register 1 (EP03CON1) [address 001A16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.21 Structure of EP03 control register 1

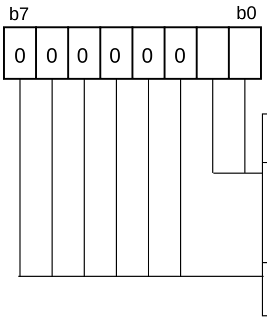


EP10 control register 1 (EP10CON1) [address $001A_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
PID10 [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. 3.4.22 Structure of EP10 control register 1

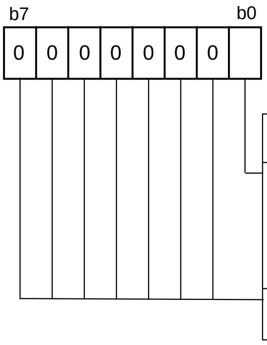


EP11 control register 1 (EP11CON1) [address $001A_{16}$]

Bit symbol	Bit name	Function	At reset H/W	At reset S/W	R	W
PID11 [1:0]	Response PID bit	b1 b0 0 0 : NAK 0 1 : Automatic response (NAK, DATA0, DATA1) 1 X : STALL	0	–	O	O
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.23 Structure of EP11 control register 1

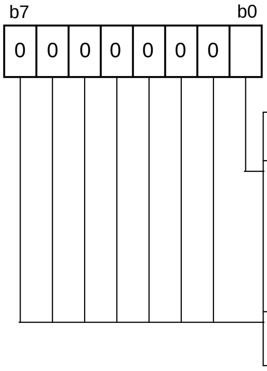


EP00 control register 2 (EP00CON2) [address $001B_{16}$]

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. 3.4.24 Structure of EP00 control register 2

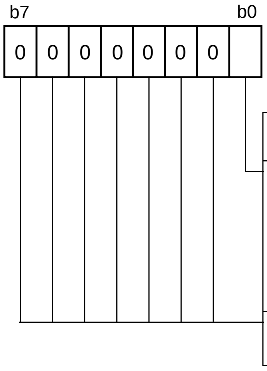


EP01 control register 2 (EP01CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.25 Structure of EP01 control register 2

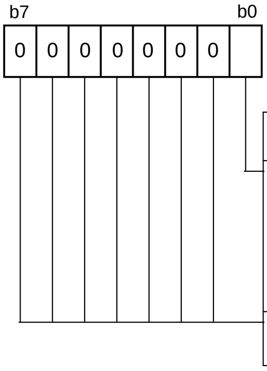


EP02 control register 2 (EP02CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.26 Structure of EP02 control register 2

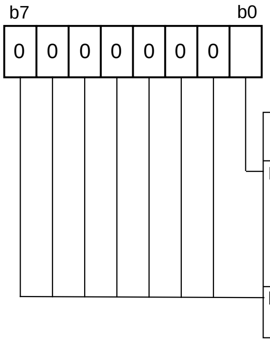


EP03 control register 2 (EP03CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.27 Structure of EP03 control register 2

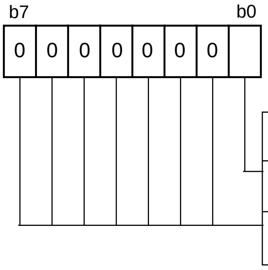


EP10 control register 2 (EP10CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BVAL10	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.) (Valid in PID10 = “01$_2$”) 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. 3.4.28 Structure of EP10 control register 2

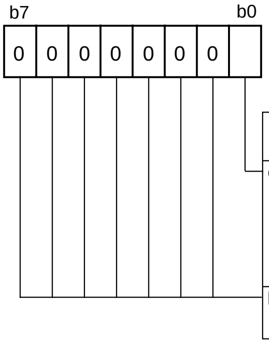


EP11 control register 2 (EP11CON2) [address $001B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
B0VAL11	Buffer 0 status bit	This bit set to “1” shows the transmitting data is in a set state (SIE is possible to read).	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.29 Structure of EP11 control register 2

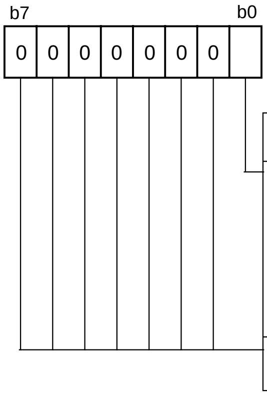


EP00 control register 3 (EP00CON3) [address $001C_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 = “01$_2$”) 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. 3.4.30 Structure of EP00 control register 3

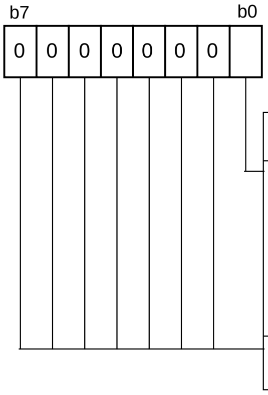


EP01 control register 3 (EP01CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.31 Structure of EP01 control register 3

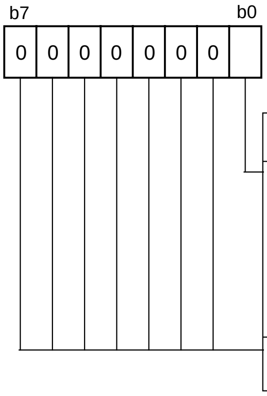


EP02 control register 3 (EP02CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.32 Structure of EP02 control register 3

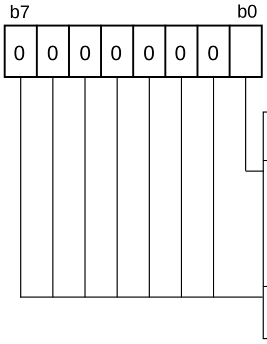


EP03 control register 3 (EP03CON3) [address 001C16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.33 Structure of EP03 control register 3

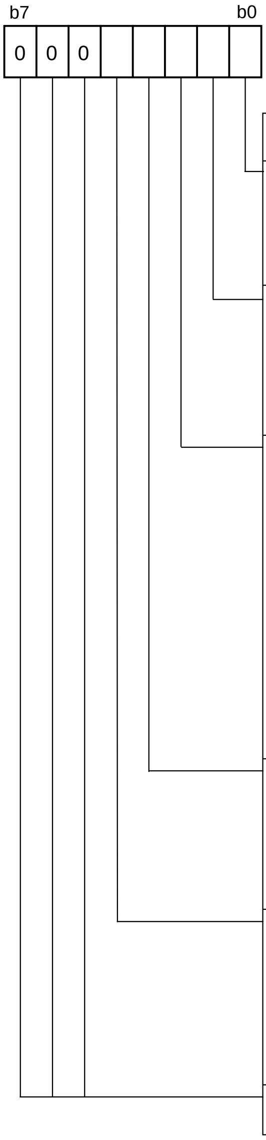


EP10 control register 3 (EP10CON3) [address 001C$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
CTENDE10	Control transfer completion enable bit	0 : NAK transmission in the status stage 1 : Control transfer completion enabled (SIE transmits NULL/ACK.) (Valid in PID10 = “01$_2$”) At reception of SETUP token: This bit is cleared to “0” by the hardware.	0	–	○	○
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	○	○

–: State remaining

Fig. 3.4.34 Structure of EP10 control register 3

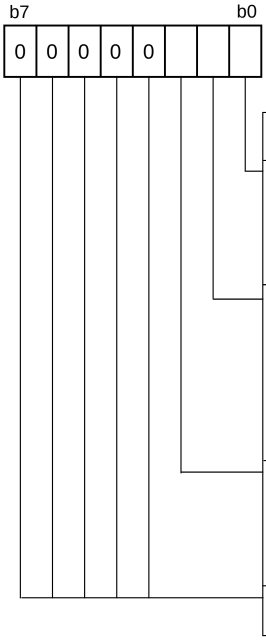


EP00 interrupt source register (EP00REQ) [address 001D$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	○	○
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	○	○
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	○	○
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	○	○
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	○	○
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	○	○

–: State remaining

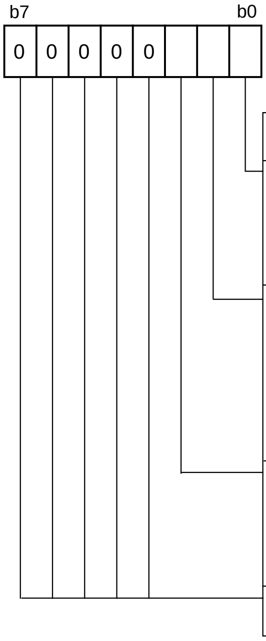
Fig. 3.4.35 Structure of EP00 interrupt source register



EP01 interrupt source register (EP01REQ) [address 001D$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

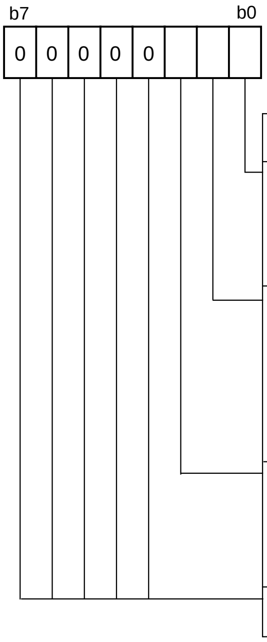
Fig. 3.4.36 Structure of EP01 interrupt source register



EP02 interrupt source register (EP02REQ) [address 001D$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.37 Structure of EP02 interrupt source register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.38 Structure of EP03 interrupt source register

b7							b0
0	0	0					

EP10 interrupt source register (EP10REQ) [address 001D16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BRDY10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. “0” can be set by software, but “1” cannot be set.	0	0	O	O
CTEND10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. “0” can be set by software, but “1” cannot be set.	0	0	O	O
CTSTS10	USB HUB/Endpoint 10 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 CTENDE10 = “0” (control transfer completion disabled) on USB HUB/Endpoint 10. “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
BSRDY10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. “0” can be set by software, but “1” cannot be set.	0	0	O	O
ERR10	USB HUB/Endpoint 10 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 HUB/Endpoint 10. 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. 3.4.39 Structure of EP10 interrupt source register

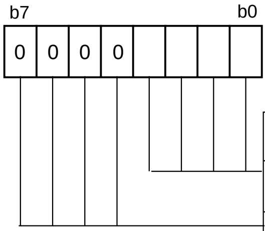
b7							b0
0	0	0	0	0	0	0	

EP11 interrupt source register (EP11REQ) [address 001D16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
B0RDY11	USB HUB/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 is ready state (enabled to be read/written) on USB HUB/Endpoint 1. “0” can be set by software, but “1” cannot be set.	0	0	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.40 Structure of EP11 interrupt source register

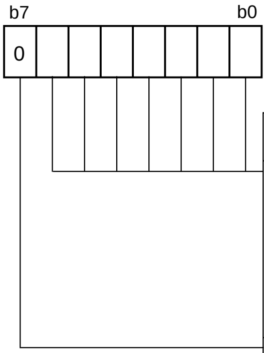


EP00 byte number register (EP00BYT) [address 001E$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.41 Structure of EP00 byte number register

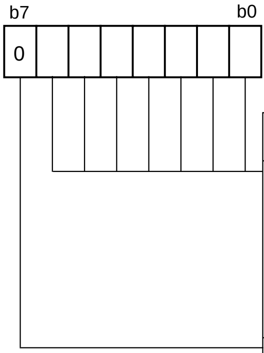


EP01 byte number register 0 (EP01BYT0) [address 001E$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.42 Structure of EP01 byte number register 0

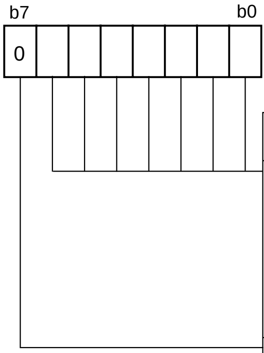


EP02 byte number register 0 (EP02BYT0) [address 001E$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.43 Structure of EP02 byte number register 0

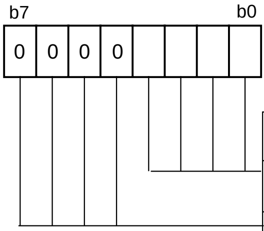


EP03 byte number register 0 (EP03BYT0) [address 001E16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.44 Structure of EP03 byte number register 0

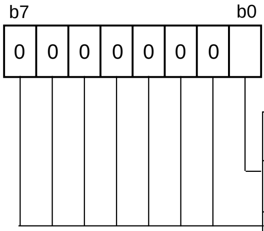


EP10 byte number register (EP10BYT) [address 001E16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BBYT10 [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. 3.4.45 Structure of EP10 byte number register

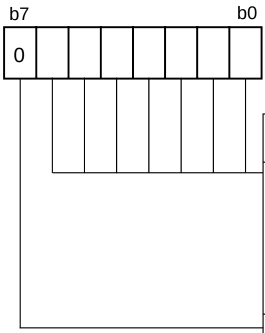


EP11 byte number register (EP11BYT0) [address 001E16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
B0BYT11	Transmit byte number bit	IN : Set the transmitting byte number.	0	—	O	O
b7:b1	Not used	Write 0 when writing. 0 is read when reading.	—	—	O	O

—: State remaining

Fig. 3.4.46 Structure of EP11 byte number register 0

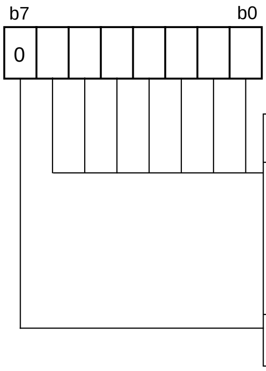


EP01 byte number register 1 (EP01BYT1) [address 001F16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.47 Structure of EP01 byte number register 1

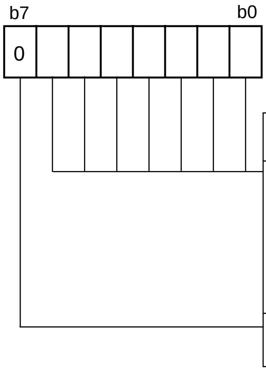


EP02 byte number register 1 (EP02BYT1) [address 001F16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.48 Structure of EP02 byte number register 1



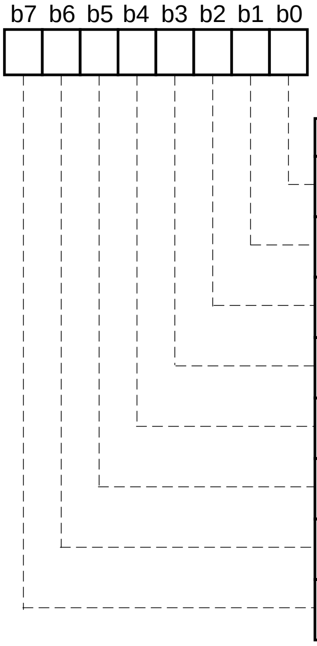
EP03 byte number register 1 (EP03BYT1) [address 001F16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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	X
b7	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.49 Structure of EP03 byte number register 1

Prescaler 12, Prescaler X

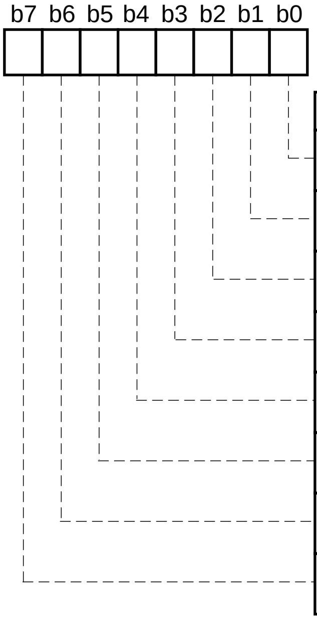


Prescaler 12 (PRE12) [Address : 20_{16}]
Prescaler X (PREX) [Address : 24_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of each prescaler. •The value set in this register is written to both each prescaler and the corresponding prescaler latch at the same time. •When this register is read out, the count value of the corresponding prescaler is read out.		1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 3.4.50 Structure of Prescaler12, Prescaler X

Timer 1



Timer 1 (T1) [Address : 21_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of timer 1. •The value set in this register is written to both timer 1 and timer 1 latch at the same time. •When this register is read out, the timer 1’s count value is read out.		1	○	○
1			0	○	○
2			0	○	○
3			0	○	○
4			0	○	○
5			0	○	○
6			0	○	○
7			0	○	○

Fig. 3.4.51 Structure of Timer 1

Timer 2, Timer X

b7 b6 b5 b4 b3 b2 b1 b0

Timer 2 (T2) [Address : 22_{16}]
Timer X (TX) [Address : 25_{16}]

B	Name	Function	At reset	R	W
0	•Set a count value of each timer. •The value set in this register is written to both each timer and each timer latch at the same time. •When this register is read out, each timer's count value is read out.		1	○	○
1			1	○	○
2			1	○	○
3			1	○	○
4			1	○	○
5			1	○	○
6			1	○	○
7			1	○	○

Fig. 3.4.52 Structure of Timer 2, Timer X

Timer X mode register

b7 b6 b5 b4 b3 b2 b1 b0

Timer X mode register (TM) [Address : 23_{16}]

B	Name	Function	At reset	R	W
0	Timer X operating mode bits	b1 b0 0 0 : Timer mode 0 1 : Pulse output mode 1 0 : Event counter mode 1 1 : Pulse width measurement mode	0	○	○
1			0	○	○
2	CNTR$_0$ active edge selection bit	The function depends on the operating mode of Timer X. (Refer to Table 2.3.1)	0	○	○
3	Timer X count stop bit	0 : Count start 1 : Count stop	0	○	○
4	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.4.53 Structure of Timer X mode register

Transmit/Receive buffer register

b7 b6 b5 b4 b3 b2 b1 b0

Transmit/Receive buffer register (TB/RB) [Address : 26$_{16}$]

B	Name	Function	At reset	R	W
0	The transmission data is written to or the receive data is read out from this buffer register. • At writing: A data is written to the transmit buffer register. • At reading: The contents of the receive buffer register are read out.		?	○	○
1			?	○	○
2			?	○	○
3			?	○	○
4			?	○	○
5			?	○	○
6			?	○	○
7			?	○	○

Note: The contents of transmit buffer register cannot be read out.
The data cannot be written to the receive buffer register.

Fig. 3.4.54 Structure of Transmit/Receive buffer register

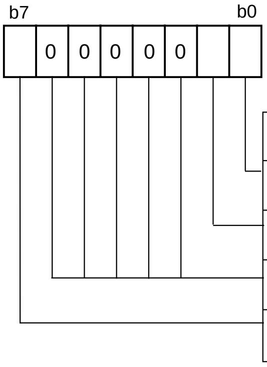
Serial I/O status register

b7 b6 b5 b4 b3 b2 b1 b0

Serial I/O status register (SIOSTS) [Address : 27$_{16}$]

B	Name	Function	At reset	R	W
0	Transmit buffer empty flag (TBE)	0 : Buffer full 1 : Buffer empty	0	○	×
1	Receive buffer full flag (RBF)	0 : Buffer empty 1 : Buffer full	0	○	×
2	Transmit shift register shift completion flag (TSC)	0 : Transmit shift in progress 1 : Transmit shift completed	0	○	×
3	Overrun error flag (OE)	0 : No error 1 : Overrun error	0	○	×
4	Parity error flag (PE)	0 : No error 1 : Parity error	0	○	×
5	Framing error flag (FE)	0 : No error 1 : Framing error	0	○	×
6	Summing error flag (SE)	0 : (OE) U (PE) U (FE) = 0 1 : (OE) U (PE) U (FE) = 1	0	○	×
7	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the contents are “1”.		1	○	×

Fig. 3.4.55 Structure of Serial I/O status register

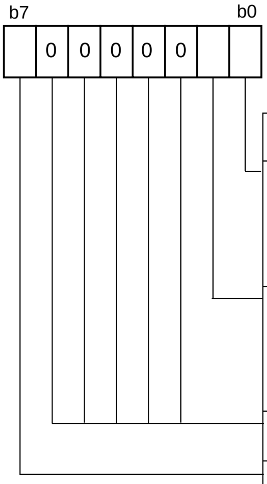


HUB interrupt source enable register (HUBICON) [address 002816]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DP1E	HUB downstream port 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	–	O	O
DP2E	HUB downstream port 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	–	O	O
b6:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
HRWUE	HUB upstream port remote-wakeup output enable bit	0 : Disabled 1 : Enabled	0	–	O	O

–: State remaining

Fig. 3.4.56 Structure of HUB interrupt source enable register

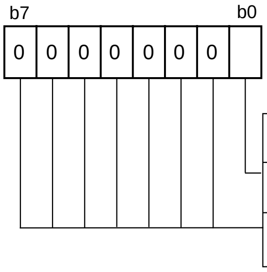


HUB interrupt source register (HUBIREQ) [address 002916]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DP1	HUB downstream port 1 interrupt bit	This bit is set to “1” when any one of DP1 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing DP1 interrupt source register to “0016”. Writing to this bit causes no state change.	0	–	O	×
DP2	HUB downstream port 1 interrupt bit	This bit is set to “1” when any one of DP2 interrupt source register’s bits at least is set to “1”. This bit is cleared to “0” by clearing DP2 interrupt source register to “0016”. Writing to this bit causes no state change.	0	–	O	×
b6:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O
HRWU	HUB upstream port remote -wakeup output enable bit	0 : Remote-wakeup being not output 1 : Remote-wakeup being output This bit change is not for a interrupt source. When detecting 2.5 µs or more of K-signal on a downstream port in Hub-suspended state, K-signal is output on from a upstream port and this bit is simultaneously set to “1”. “0” can be set by software, but “1” cannot be set.	0	–	O	O

–: State remaining

Fig. 3.4.57 Structure of HUB interrupt source register

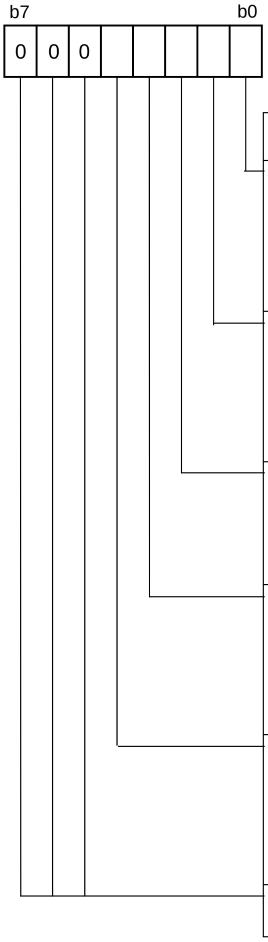


HUB downstream port index register (HUBINDEX) [address 002A16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DPIDX	HUB downstream port index bit	0 : HUB downstream port 1 1 : HUB downstream port 2	0	–	O	O
b7:b1	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.58 Structure of HUB downstream port index register

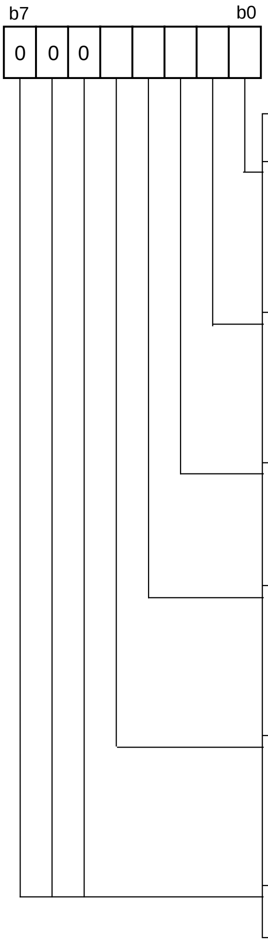


DP1 interrupt source register (DP1REQ) [address 002B$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PTDIS1	Downstream port 1 disconnect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-disconnect state (2.5 µs or more of SE0) on a downstream port 1 in DSCONN1 = “1”. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTCON1	Downstream port 1 connect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-connect state (2.5 µs or more of J- or K- state) on a downstream port 1 in DSCONN1 = “0”. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTERR1	Downstream port 1 port error interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when an error occurs on a downstream port 1. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTRSM1	Downstream port 1 resume interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a resume signal on a downstream port 1 in the condition of HUB suspended or port suspended state. “0” can be set by software, but “1” cannot be set.	0	–	O	O
PTCHG1	Downstream port 1 bus-change detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to “1” when detecting a bus-change of a downstream port 1 in the condition of HUB suspended state. It is also “1” in the internal clock halted. “0” can be set by software, but “1” cannot be set.	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.59 Structure of DP1 interrupt source register

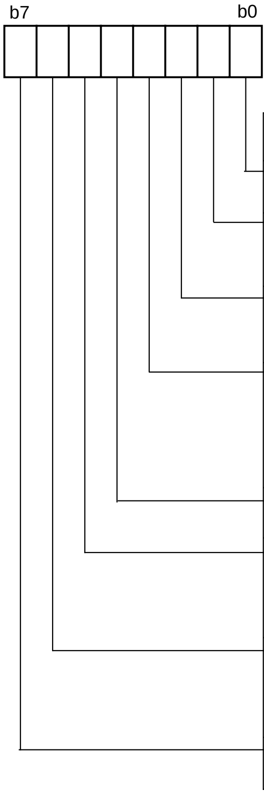


DP2 interrupt source register (DP2REQ) [address $002B_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PTDIS2	Downstream port 2 disconnect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when detecting a bus-disconnect state (2.5 µs or more of SE0) on a downstream port 2 in DSCONN2 = "1". "0" can be set by software, but "1" cannot be set.	0	–	O	O
PTCON2	Downstream port 2 connect detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when detecting a bus-connect state (2.5 µs or more of J- or K- state) on a downstream port 2 in DSCONN2 = "0". "0" can be set by software, but "1" cannot be set.	0	–	O	O
PTERR2	Downstream port 2 port error interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when an error occurs on a downstream port 2. "0" can be set by software, but "1" cannot be set.	0	–	O	O
PTRSM2	Downstream port 2 resume interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when detecting a resume signal on a downstream port 2 in the condition of HUB suspended or port suspended state. "0" can be set by software, but "1" cannot be set.	0	–	O	O
PTCHG2	Downstream port 2 bus-change detection interrupt bit	0: No interrupt request issued 1: Interrupt request issued This bit is set to "1" when detecting a bus-change of a downstream port 2 in the condition of HUB suspended state. It is also "1" in the internal clock halted. "0" can be set by software, but "1" cannot be set.	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.60 Structure of DP2 interrupt source register

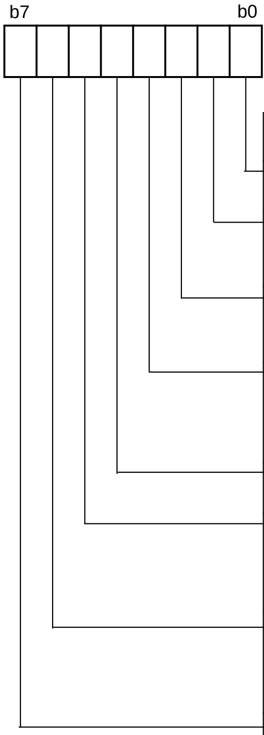


DP1 control register (DP1CON) [address 002C$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DSCONN1	Downstream port 1 connect bit	0 : Disconnect ; PTCON1 interrupt enabled 1 : Connect ; PTDIS1 interrupt enabled	0	–	O	O
DSPTEN1	Downstream port 1 enable bit	0 : Downstream port 1 disabled 1 : Downstream port 1 enabled ; This bit is cleared when an interrupt of PTDIS1 or PTERR1 is generated.	0	–	O	O
DSSUSP1	Downstream port 1 suspend bit	0 : No port suspended 1 : Port suspended; This bit is cleared when an interrupt of PTDIS1 or PTRSM1 is generated.	0	–	O	O
DSDETE1	Downstream port 1 connect-state detection enable bit	0 : Connect/disconnect-state detection disabled ; PTCON1 and PTDIS1 interrupts disabled 1 : Connect/disconnect-state detection enabled ; This bit is cleared when an interrupt of PTCON1, PTDIS1 or PTERR1 is generated.	0	–	O	O
DSRSTO1	Downstream port 1 SE0 signal transmit bit	0 : Being not output 1 : SE0 signal being output	0	–	O	O
DSRSMO1	Downstream port 1 resume signal transmit bit	0 : Being not output 1 : K-signal being output ; When writing “0”, a low-speed EOP is output and then a transition to being not output occurs.	0	–	O	O
DSRMOD1	Downstream port 1 bus-state read mode control bit	0 : Mode where a downstream port 1 bus-state is read, using RD signal 1 : Mode where a downstream port 1 bus-state is read, using EOF2 signal (internal signal)	0	–	O	O
DSLSPD1	Downstream port 1 USB transfer	0 : Full-speed mode (12MHz) 1 : Low-speed mode (1.5 MHz)	0	–	O	O

–: State remaining

Fig. 3.4.61 Structure of DP1 control register

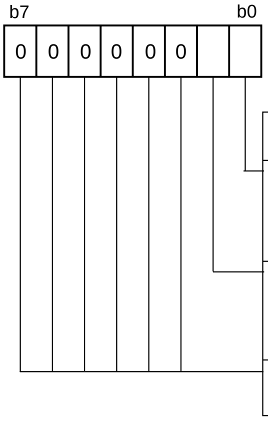


DP2 control register (DP2CON) [address 002C$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
DSCONN2	Downstream port 2 connect bit	0 : Disconnect ; PTCON2 interrupt enabled 1 : Connect ; PTDIS2 interrupt enabled	0	–	O	O
DSPTEN2	Downstream port 2 enable bit	0 : Downstream port 2 disabled 1 : Downstream port 2 enabled ; This bit is cleared when an interrupt of PTDIS2 or PTERR2 is generated.	0	–	O	O
DSSUSP2	Downstream port 2 suspend bit	0 : No port suspended 1 : Port suspended; This bit is cleared when an interrupt of PTDIS2 or PTRSM2 is generated.	0	–	O	O
DSDETE2	Downstream port 2 connect-state detection enable bit	0 : Connect-state detection disabled ; PTCON2 and PTDIS2 interrupts disabled 1 : Connect-state detection enabled ; This bit is cleared when an interrupt of PTCON2, PTDIS2 or PTERR2 is generated.	0	–	O	O
DSRSTO2	Downstream port 2 SE0 signal transmit bit	0 : Being not output 1 : SE0 signal being output	0	–	O	O
DSRSMO2	Downstream port 2 resume signal transmit bit	0 : Being not output 1 : K-signal being output ; When writing “0”, a low-speed EOP is output and then a transition to being not output occurs.	0	–	O	O
DSRMOD2	Downstream port 2 bus-state read mode control bit	0 : Mode where a downstream port 2 bus-state is read, using RD signal 1 : Mode where a downstream port 2 bus-state is read, using EOF2 signal (internal signal)	0	–	O	O
DSLSPD2	Downstream port 2 USB transfer speed select bit	0 : Full-speed mode (12MHz) 1 : Low-speed mode (1.5 MHz)	0	–	O	O

–: State remaining

Fig. 3.4.62 Structure of DP2 control register

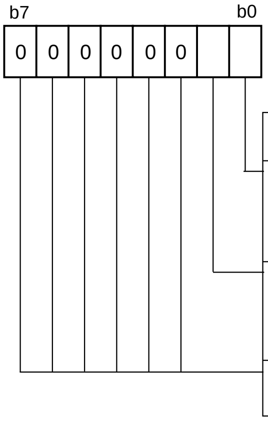


DP1 status register (DP1STS) [address $002D_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
D1MINUS	D1- signal bit	In DSRMOD1 = “0”, a downstream port 1 bus-state is read, using RD signal. In DSRMOD1 = “1”, a downstream port 1 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	○	×
D1PLUS	D1+ signal bit	In DSRMOD1 = “0”, a downstream port 1 bus-state is read, using RD signal. In DSRMOD1 = “1”, a downstream port 1 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	○	×
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	○	○

–: State remaining

Fig. 3.4.63 Structure of DP1 status register

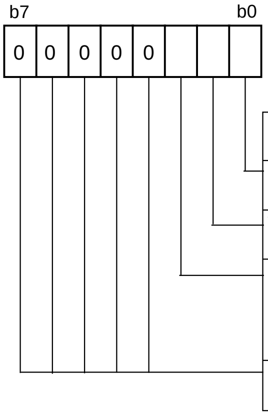


DP2 status register (DP2STS) [address $002D_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
D2MINUS	D2- signal bit	In DSRMOD2 = “0”, a downstream port 2 bus-state is read, using RD signal. In DSRMOD2 = “1”, a downstream port 2 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	○	×
D2PLUS	D2+ signal bit	In DSRMOD2 = “0”, a downstream port 2 bus-state is read, using RD signal. In DSRMOD2 = “1”, a downstream port 2 bus-state is read, using EOF2 signal (internal signal).	In-definite	In-definite	○	×
b7:b2	Not used	Write “0” when writing. “0” is read when reading.	–	–	○	○

–: State remaining

Fig. 3.4.64 Structure of DP2 status register



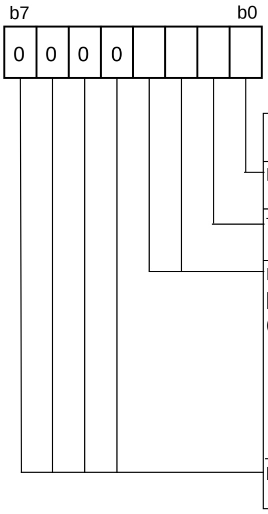
EXB interrupt source enable register (EXBICON) [address 0030_{16}]
(**Note**)

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

–: State remaining

Note: Do not set each bit simultaneously.

Fig. 3.4.65 Structure of EXB interrupt source enable register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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”.

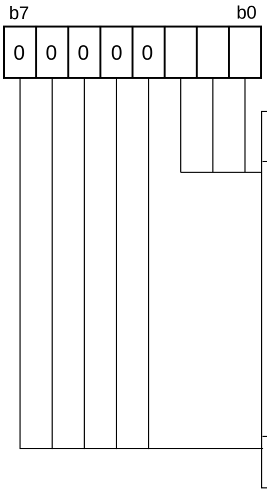
2: The memory channel status bits indicate the status of memory channel. In MC_ENB = “0” these bits are always “00$_2$”. When the memory channel operation ends, these bits are set to “11$_2$” and the memory channel operation end interrupt is generated.

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. 3.4.66 Structure of EXB interrupt source register

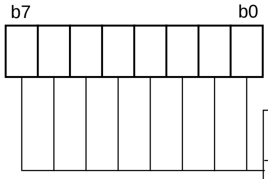


EXB index register (EXBINDEX) [address 0033$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

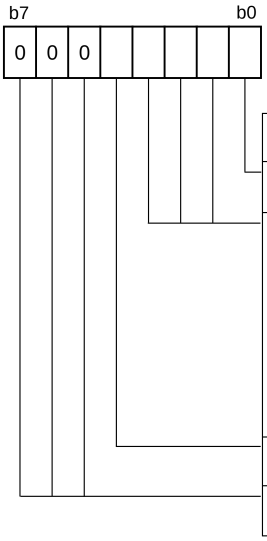
Fig. 3.4.67 Structure of EXB index register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.68 Structure of Register window 1

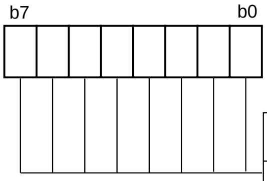


Index = 00$_{16}$: External I/O configuration register (EXBCFGL) [address 0034$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.69 Index00[low]; Structure of External I/O configuration register

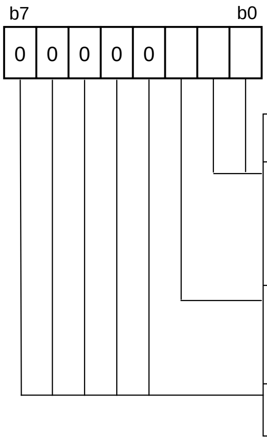


Index =01_{16} : Transmit/Receive buffer register (RXBUF/TXBUF) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.70 Index01[low]; Structure of Transmit/Receive buffer register

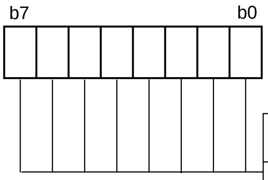


Index =02_{16} : Memory channel operation mode register (MCHMOD) [address 0034_{16}]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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

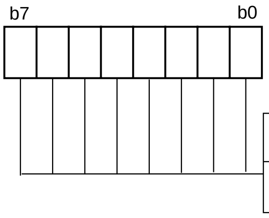
Fig. 3.4.71 Index02[low]; Structure of Memory channel operation mode register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.72 Index03[low]; Structure of Memory address counter

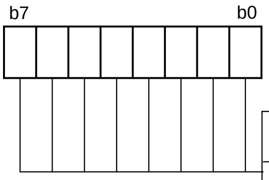


Index = 04$_{16}$: End address register (ENDADL) [address 0034$_{16}$]

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

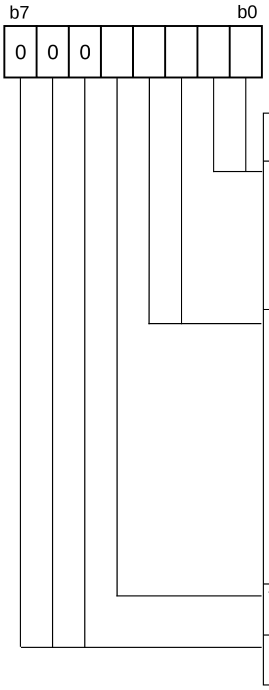
Fig. 3.4.73 Index04[low]; Structure of End address register



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

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.74 Structure of Register window 2

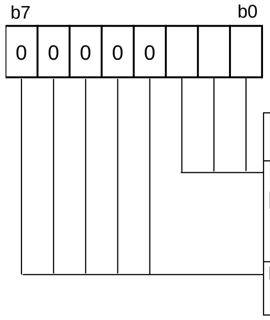


Index = 00$_{16}$: External I/O configuration register (EXBCFGH) [address 0035$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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/SCLK 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. 3.4.75 Index00[high]; Structure of External I/O configuration register

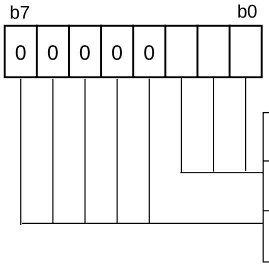


Index = 03_{16} : Memory address counter (MEMADH) [address 0035_{16}]

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. 3.4.76 Index03[high]; Structure of Memory address counter



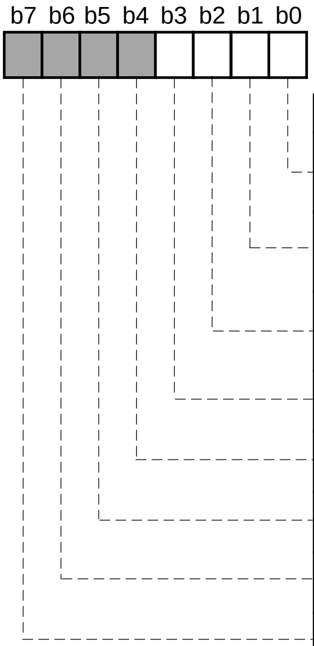
Index = 04_{16} : End address register (ENDADH) [address 0035_{16}]

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. 3.4.77 Index04[high]; Structure of End address register

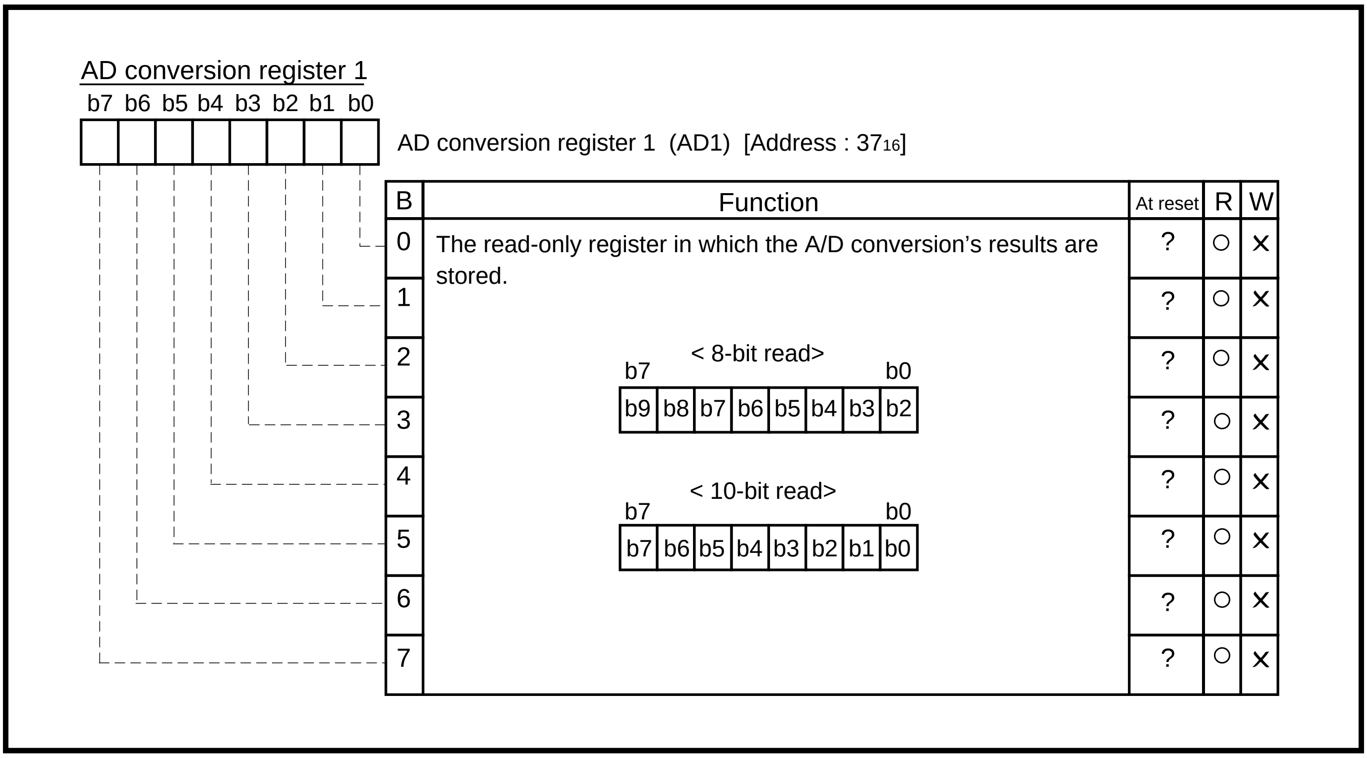
AD control register



AD control register (ADCON) [Address : 36_{16}]

B	Name	Function	At reset	R	W
0	Analog input pin selection bits	b2 b1 b0 0 0 0 : $P1_0/DQ_0/AN_0$ 0 0 1 : $P1_1/DQ_1/AN_1$ 0 1 0 : $P1_2/DQ_2/AN_2$ 0 1 1 : $P1_3/DQ_3/AN_3$ 1 0 0 : $P1_4/DQ_4/AN_4$ 1 0 1 : $P1_5/DQ_5/AN_5$ 1 1 0 : $P1_6/DQ_6/AN_6$ 1 1 1 : $P1_7/DQ_7/AN_7$	0	○	○
1					
2					
3	AD conversion completion bit	0 : Conversion in progress 1 : Conversion completed	1	○	○
4	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are indefinite.		?	○	×
5			?	○	×
6			?	○	×
7			?	○	×

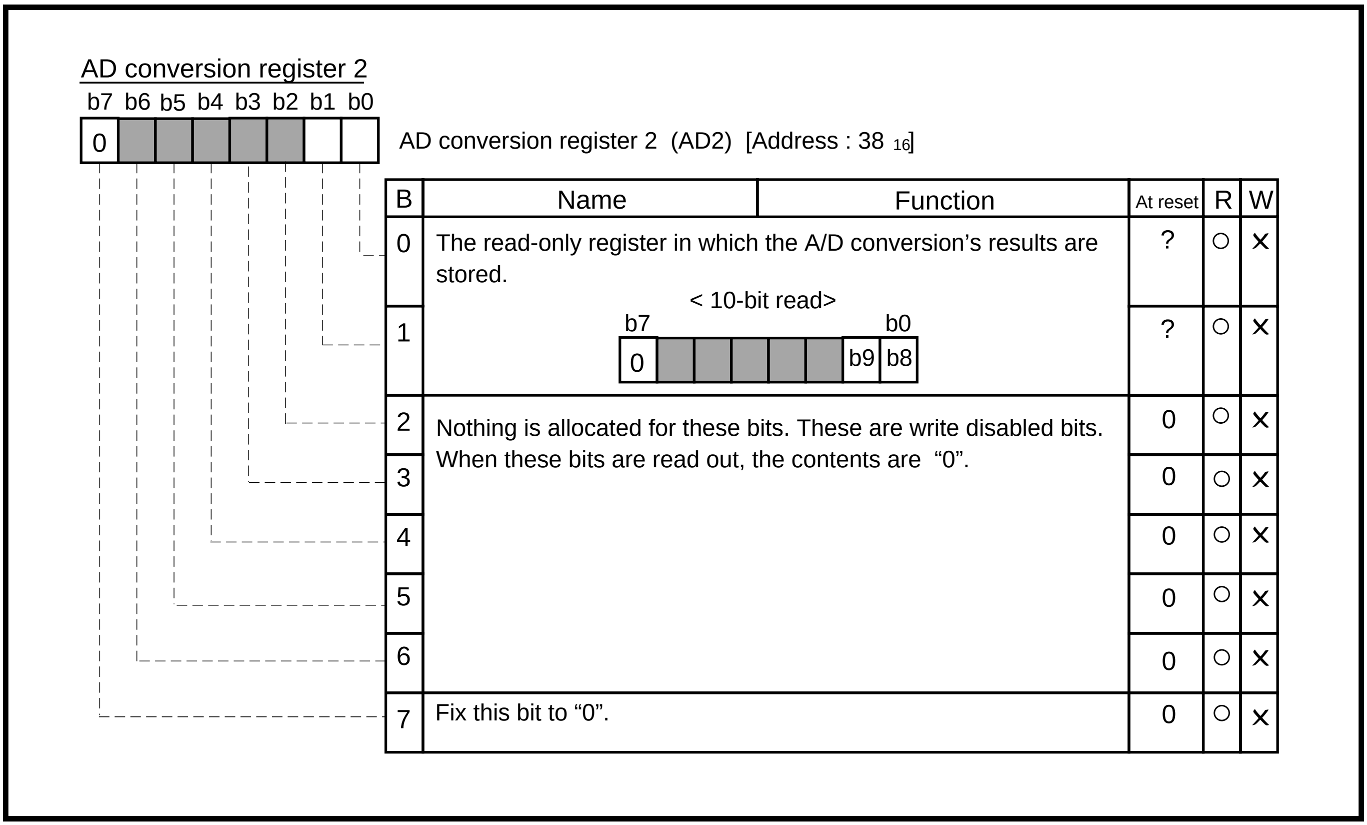
Fig. 3.4.78 Structure of AD control register



AD conversion register 1 (AD1) [Address : 37₁₆]

B	Function	At reset	R	W
0	The read-only register in which the A/D conversion’s results are stored. < 8-bit read> b7 b0 b9 b8 b7 b6 b5 b4 b3 b2 < 10-bit read> b7 b0 b7 b6 b5 b4 b3 b2 b1 b0	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

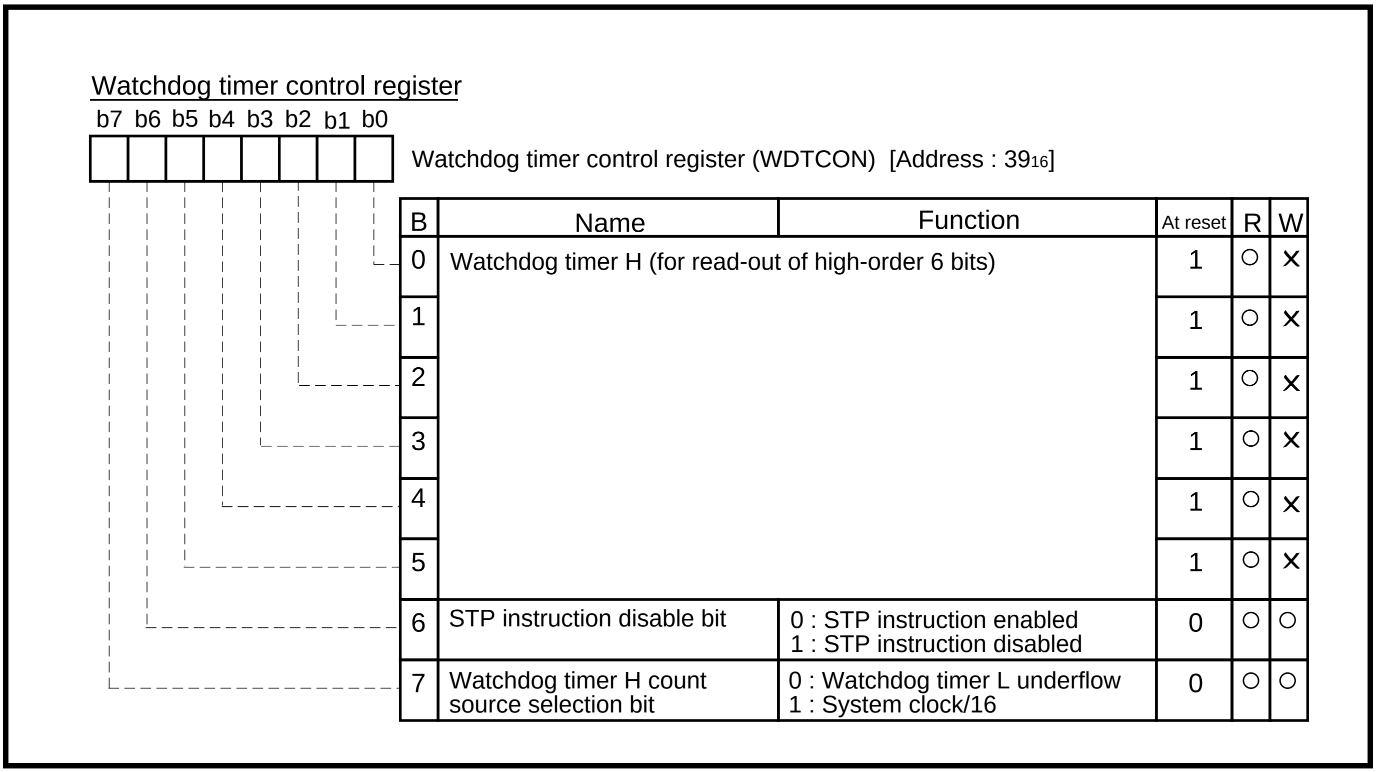
Fig. 3.4.79 Structure of AD conversion register 1



AD conversion register 2 (AD2) [Address : 38 ₁₆]

B	Name	Function	At reset	R	W
0		The read-only register in which the A/D conversion’s results are stored. < 10-bit read> b7 b0 0 ▨ ▨ ▨ ▨ ▨ b9 b8	?	○	×
1			?	○	×
2		Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.	0	○	×
3			0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7		Fix this bit to “0”.	0	○	×

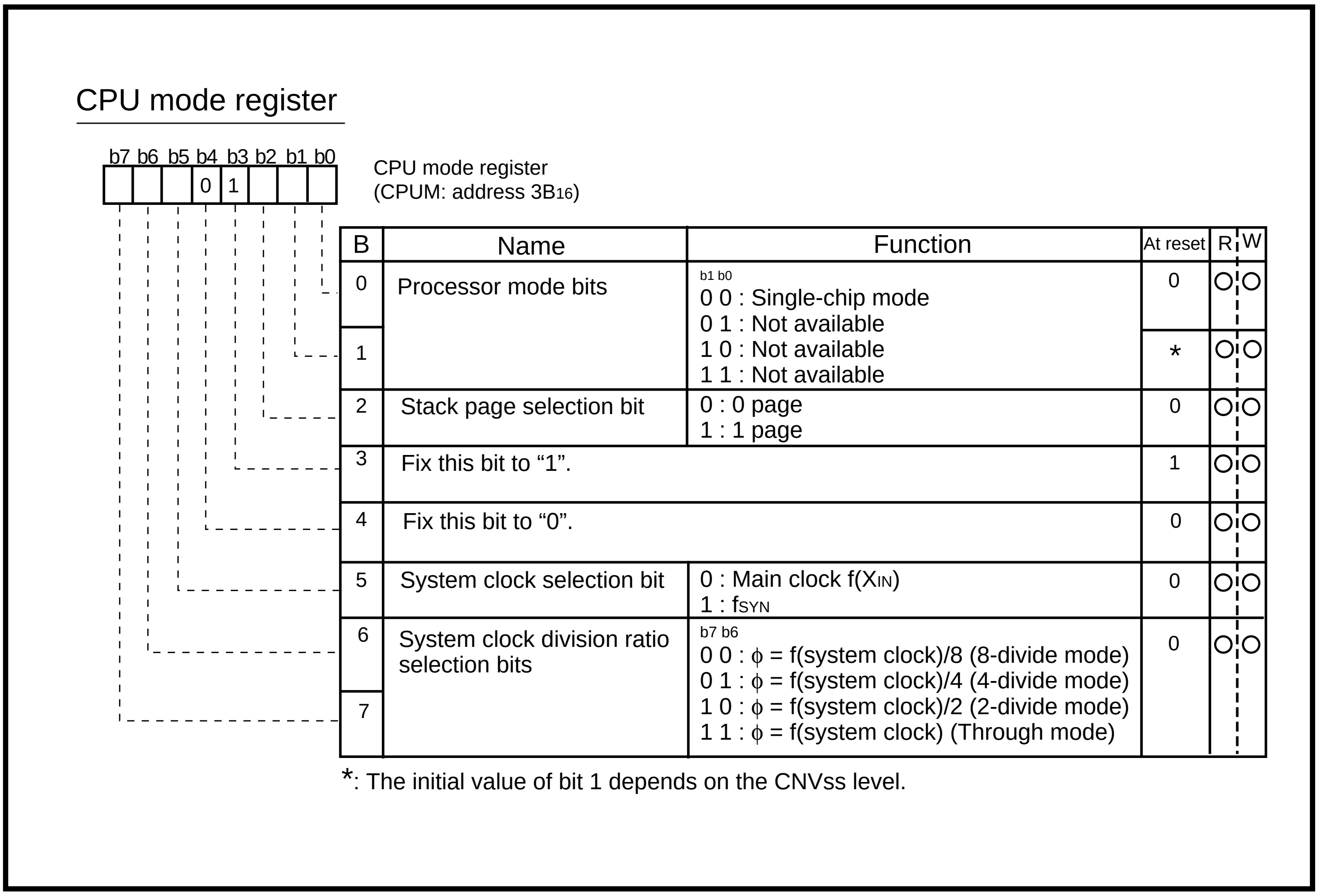
Fig. 3.4.80 Structure of AD conversion register 2



Watchdog timer control register (WDTCON) [Address : 39_{16}]

B	Name	Function	At reset	R	W
0	Watchdog timer H (for read-out of high-order 6 bits)		1	○	×
1			1	○	×
2			1	○	×
3			1	○	×
4			1	○	×
5			1	○	×
6	STP instruction disable bit	0 : STP instruction enabled 1 : STP instruction disabled	0	○	○
7	Watchdog timer H count source selection bit	0 : Watchdog timer L underflow 1 : System clock/16	0	○	○

Fig. 3.4.81 Structure of Watchdog timer control register



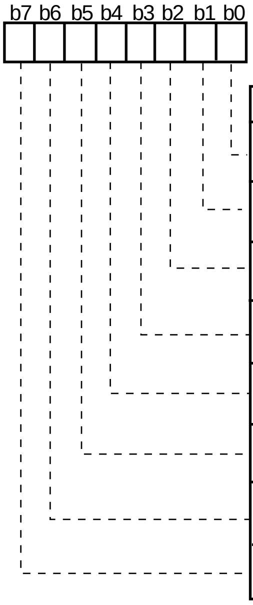
CPU mode register
(CPUM: address $3B_{16}$)

B	Name	Function	At reset	R	W
0	Processor mode bits	b1 b0 0 0 : Single-chip mode 0 1 : Not available 1 0 : Not available 1 1 : Not available	0	○	○
1			*	○	○
2	Stack page selection bit	0 : 0 page 1 : 1 page	0	○	○
3	Fix this bit to “1”.		1	○	○
4	Fix this bit to “0”.		0	○	○
5	System clock selection bit	0 : Main clock $f(X_{IN})$ 1 : f_{SYN}	0	○	○
6	System clock division ratio selection bits	b7 b6 0 0 : ϕ = f(system clock)/8 (8-divide mode) 0 1 : ϕ = f(system clock)/4 (4-divide mode) 1 0 : ϕ = f(system clock)/2 (2-divide mode) 1 1 : ϕ = f(system clock) (Through mode)	0	○	○
7					

*: The initial value of bit 1 depends on the CNVss level.

Fig. 3.4.82 Structure of CPU mode register

Interrupt request register 1



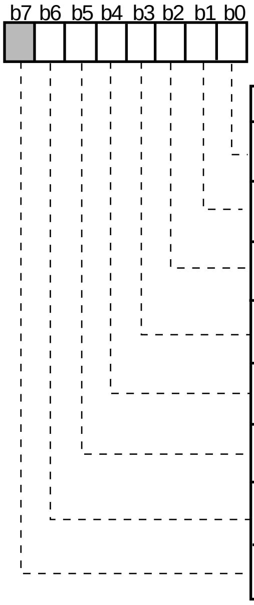
Interrupt request register 1 (IREQ1)
[Address : $3C_{16}$]

B	Name	Function	At reset	R	W
0	USB bus reset interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB SOF interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	USB device interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	EXB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	INT$_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Timer X interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	Timer 1 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Timer 2 interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻

✻ “0” can be set by software, but “1” cannot be set.

Fig. 3.4.83 Structure of Interrupt request register 1

Interrupt request register 2



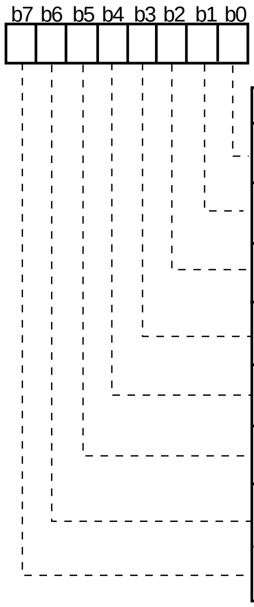
Interrupt request register 2 (IREQ2)
[Address : $3D_{16}$]

B	Name	Function	At reset	R	W
0	INT$_1$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
1	USB HUB interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
2	Serial I/O receive interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
3	Serial I/O transmit interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
4	CNTR$_0$ interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
5	Key-on wake-up interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
6	A/D conversion interrupt request bit	0 : No interrupt request issued 1 : Interrupt request issued	0	○	✻
7	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×

✻ “0” can be set by software, but “1” cannot be set.

Fig. 3.4.84 Structure of Interrupt request register 2

Interrupt control register 1

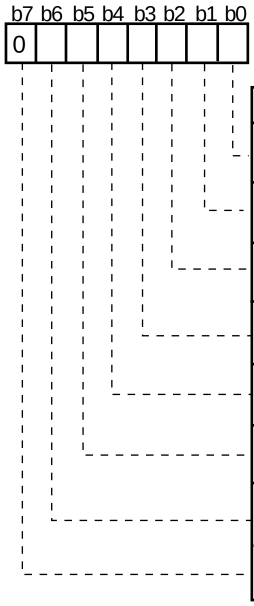


Interrupt control register 1 (ICON1)
[Address : 3E16]

B	Name	Function	At reset	R	W
0	USB bus reset interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB SOF interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	USB device interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	EXB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	INT0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Timer X interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	Timer 1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Timer 2 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○

Fig. 3.4.85 Structure of Interrupt control register 1

Interrupt control register 2

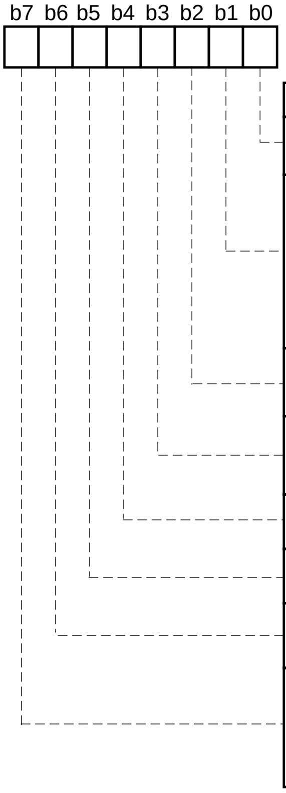


Interrupt control register 2 (ICON2)
[Address : 3F16]

B	Name	Function	At reset	R	W
0	INT1 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
1	USB HUB interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
2	Serial I/O receive interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
3	Serial I/O transmit interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
4	CNTR0 interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
5	Key-on wake-up interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
6	A/D conversion interrupt enable bit	0 : Interrupt disabled 1 : Interrupt enabled	0	○	○
7	Fix this bit to “0”.		0	○	○

Fig. 3.4.86 Structure of Interrupt control register 2

Serial I/O control register

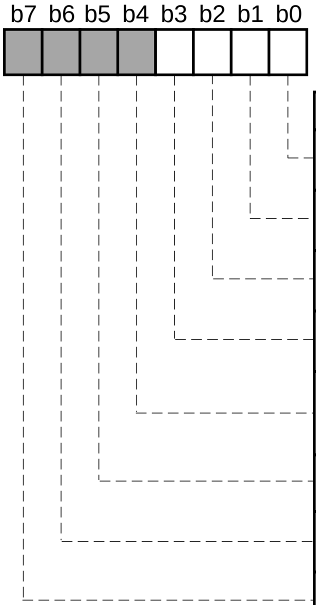


Serial I/O control register (SIOCON) [Address : $0FE0_{16}$]

B	Name	Function	At reset	R	W
0	BRG count source selection bit (CSS)	0 : System clock 1 : System clock/4	0	○	○
1	Serial I/O synchronous clock selection bit (SCS)	• In clock synchronous serial I/O 0 : BRG output divided by 4 1 : External clock input • In UART 0 : BRG output divided by 16 1 : External clock input divided by 16	0	○	○
2	$\overline{\text{SRDY}}$ output enable bit (SRDY)	0 : $P4_3$ pin operates as ordinary I/O pin 1 : $P4_3$ pin operates as $\overline{\text{SRDY}}$ output pin	0	○	○
3	Transmit interrupt source selection bit (TIC)	0 : Interrupt when transmit buffer has emptied 1 : Interrupt when transmit shift operation is completed	0	○	○
4	Transmit enable bit (TE)	0 : Transmit disabled 1 : Transmit enabled	0	○	○
5	Receive enable bit (RE)	0 : Receive disabled 1 : Receive enabled	0	○	○
6	Serial I/O mode selection bit (SIOM)	0 : Clock asynchronous(UART) serial I/O 1 : Clock synchronous serial I/O	0	○	○
7	Serial I/O enable bit (SIOE)	0 : Serial I/O disabled (pins $P4_0$ to $P4_3$ operate as ordinary I/O pins) 1 : Serial I/O enabled (pins $P4_0$ to $P4_3$ operate as serial I/O pins)	0	○	○

Fig. 3.4.87 Structure of Serial I/O control register

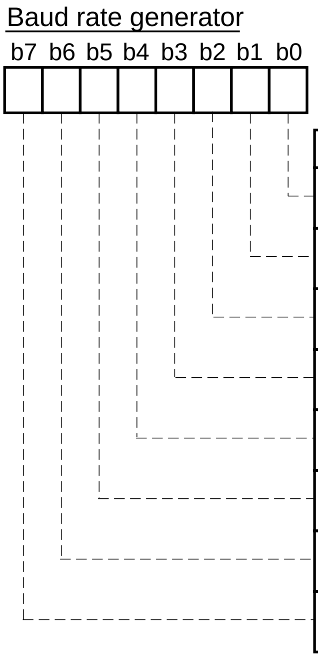
UART control register



UART control register (UARTCON) [Address : $0FE1_{16}$]

B	Name	Function	At reset	R	W
0	Character length selection bit (CHAS)	0 : 8 bits 1 : 7 bits	0	○	○
1	Parity enable bit (PARE)	0 : Parity checking disabled 1 : Parity checking enabled	0	○	○
2	Parity selection bit (PARS)	0 : Even parity 1 : Odd parity	0	○	○
3	Stop bit length selection bit (STPS)	0 : 1 stop bit 1 : 2 stop bits	0	○	○
4	Nothing is allocated for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
5	Nothing is allocated for these bits. These are write disabled bits. When these bits are read out, the contents are “1”.		1	○	×
6			1	○	×
7			1	○	×

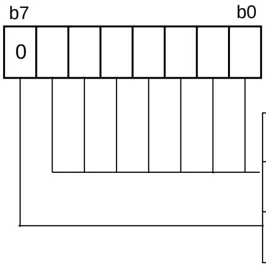
Fig. 3.4.88 Structure of UART control register



Baud rate generator (BRG) [Address : 0FE2$_{16}$]

B	Function	At reset	R	W
0	Set a count value of baud rate generator.	?	○	○
1		?	○	○
2		?	○	○
3		?	○	○
4		?	○	○
5		?	○	○
6		?	○	○
7		?	○	○

Fig. 3.4.89 Structure of Baud rate generator

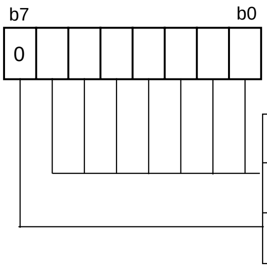


EP01 MAX. packet size register (EP01MAX) [address 0FEC$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.90 Structure of EP01 MAX. packet size register

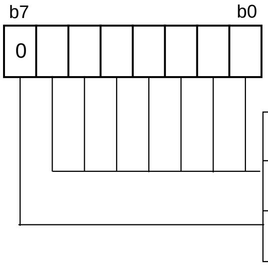


EP02 MAX. packet size register (EP02MAX) [address 0FEC$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.91 Structure of EP02 MAX. packet size register

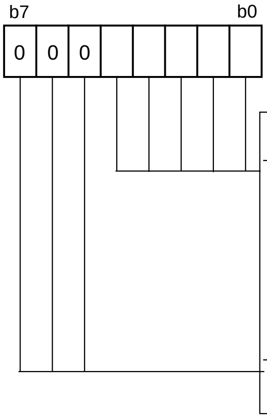


EP03 MAX. packet size register (EP03MAX) [address 0FEC16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.92 Structure of EP03 MAX. packet size register

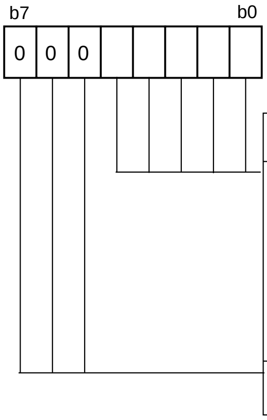


EP00 buffer area set register (EP00BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.93 Structure of EP00 buffer area set register

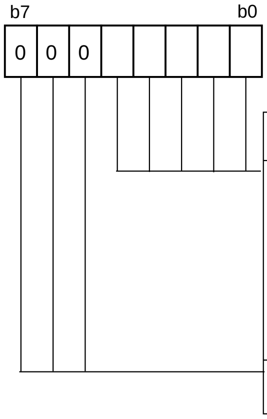


EP01 buffer area set register (EP01BUF) [address 0FED16]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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. 3.4.94 Structure of EP01 buffer area set register

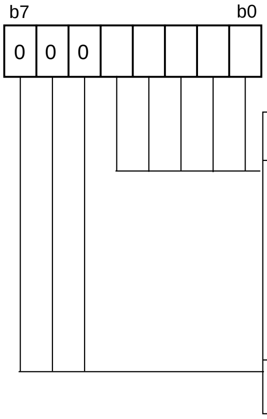


EP02 buffer area set register (EP02BUF) [address $0FED_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 : 0040_{16} 0 0 0 1 1 : 0060_{16} .... 1 1 1 1 0 : $03C0_{16}$ 1 1 1 1 1 : $03E0_{16}$	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.95 Structure of EP02 buffer area set register

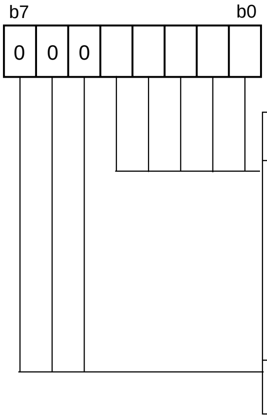


EP03 buffer area set register (EP03BUF) [address $0FED_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/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 : 0040_{16} 0 0 0 1 1 : 0060_{16} .... 1 1 1 1 0 : $03C0_{16}$ 1 1 1 1 1 : $03E0_{16}$	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.96 Structure of EP03 buffer area set register

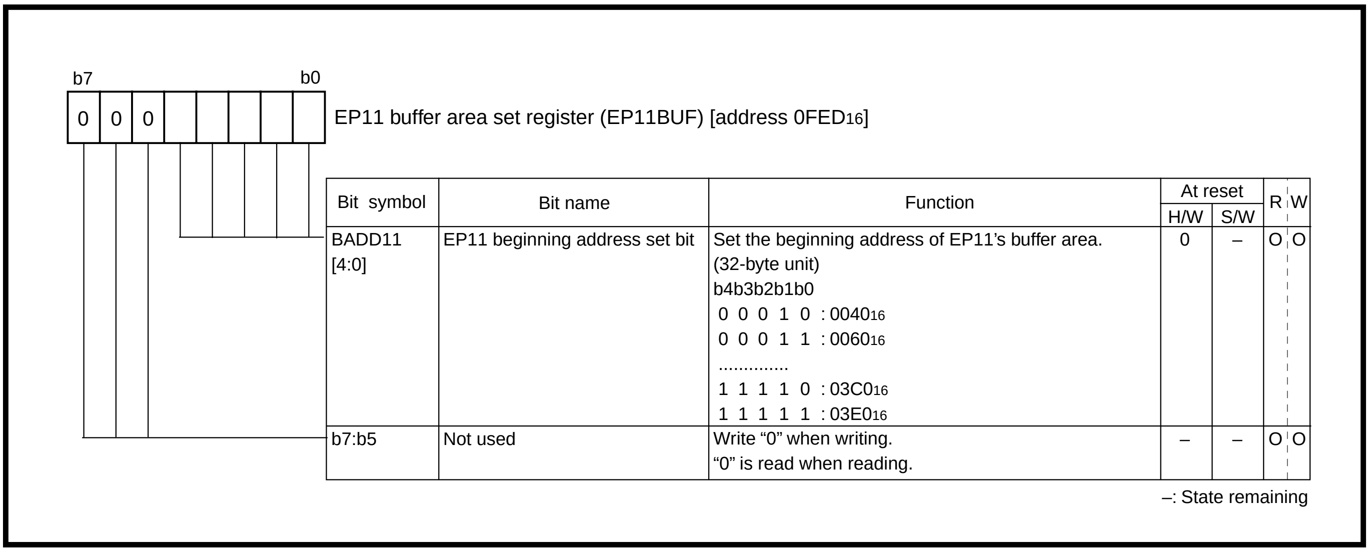


EP10 buffer area set register (EP10BUF) [address $0FED_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BADD10 [4:0]	EP10 beginning address set bit	Set the beginning address of EP10's buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040_{16} 0 0 0 1 1 : 0060_{16} .... 1 1 1 1 0 : $03C0_{16}$ 1 1 1 1 1 : $03E0_{16}$	0	–	O	O
b7:b5	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.97 Structure of EP10 buffer area set register

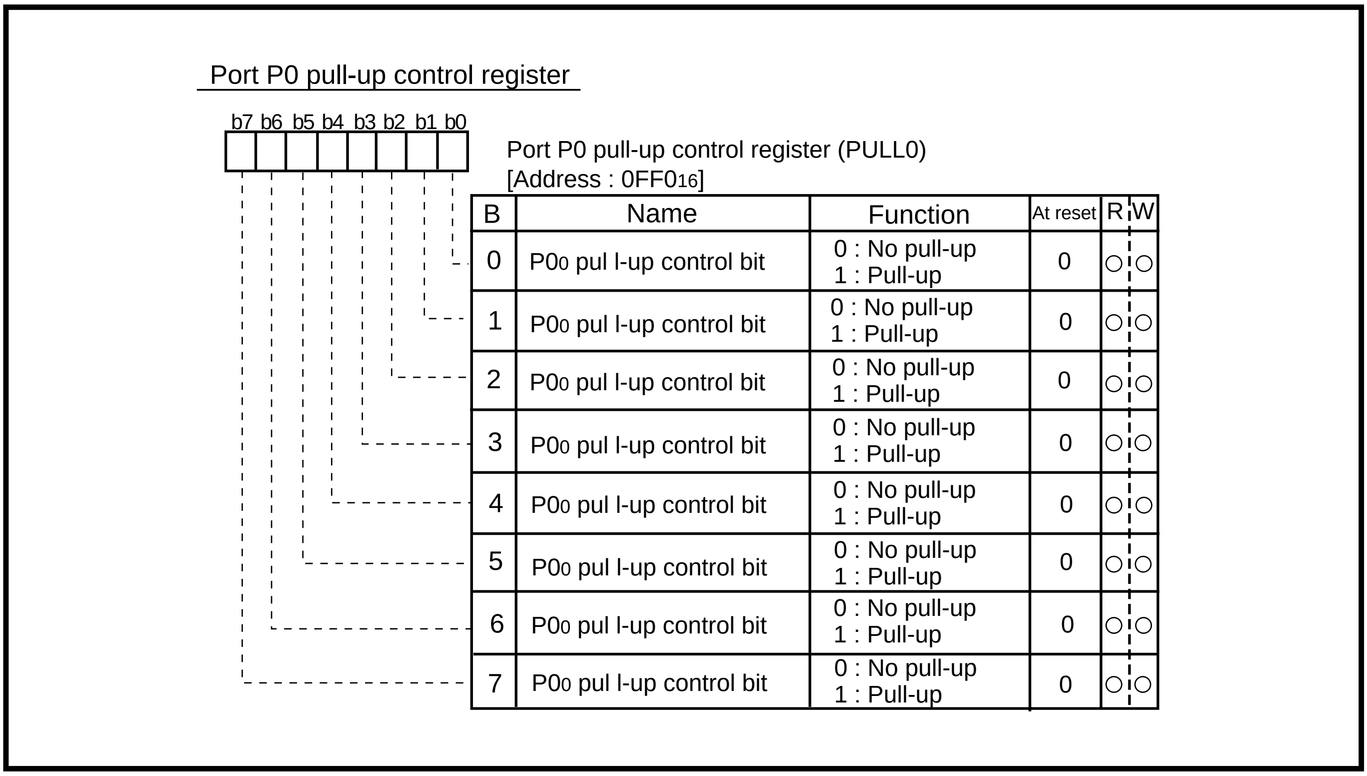


EP11 buffer area set register (EP11BUF) [address 0FED$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
BADD11 [4:0]	EP11 beginning address set bit	Set the beginning address of EP11’s buffer area. (32-byte unit) b4b3b2b1b0 0 0 0 1 0 : 0040$_{16}$ 0 0 0 1 1 : 0060$_{16}$ .... 1 1 1 1 0 : 03C0$_{16}$ 1 1 1 1 1 : 03E0$_{16}$	0	–	O	O
b7:b5	Not used	Write “0” when writing. “0” is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.98 Structure of EP11 buffer area set register

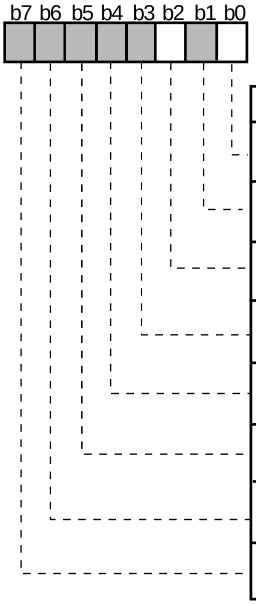


Port P0 pull-up control register (PULL0)
[Address : 0FF0$_{16}$]

B	Name	Function	At reset	R	W
0	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
2	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
3	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
4	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
5	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
6	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
7	P0$_0$ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○

Fig. 3.4.99 Structure of Port P0 pull-up control register

Port P5 pull-up control register

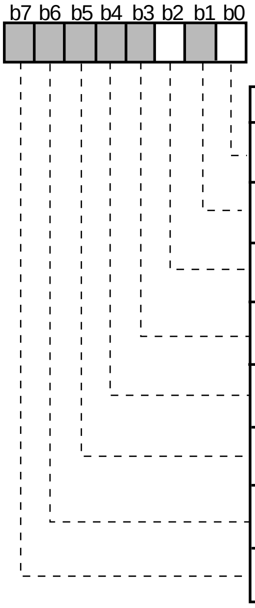


Port P5 pull-up control register (PULL5)
[Address : 0FF2₁₆]

B	Name	Function	At reset	R	W
0	P5₀ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
1	Nothing is arranged for this bit. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
2	P5₂ pul l-up control bit	0 : No pull-up 1 : Pull-up	0	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

Fig. 3.4.100 Structure of Port P5 pull-up control register

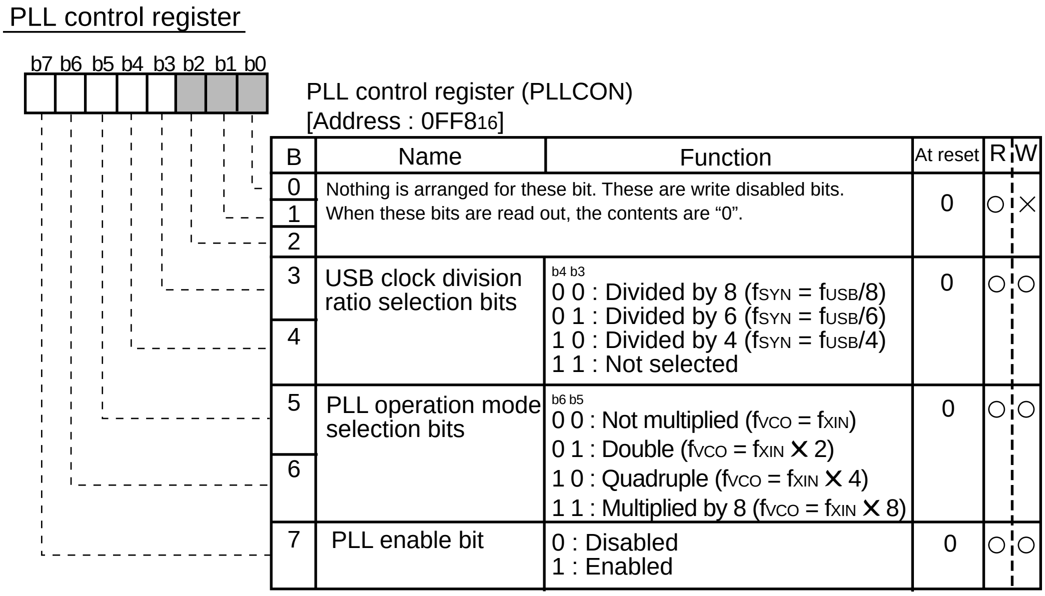
Interrupt edge selection register



Interrupt edge selection register (INTEDGE)
[Address : 0FF3₁₆]

B	Name	Function	At reset	R	W
0	INT₀ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
1	Nothing is arranged for this bits. This is a write disabled bit. When this bit is read out, the contents are “0”.		0	○	×
2	INT₁ interrupt edge selection bit	0 : Falling edge active 1 : Rising edge active	0	○	○
3	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are “0”.		0	○	×
4			0	○	×
5			0	○	×
6			0	○	×
7			0	○	×

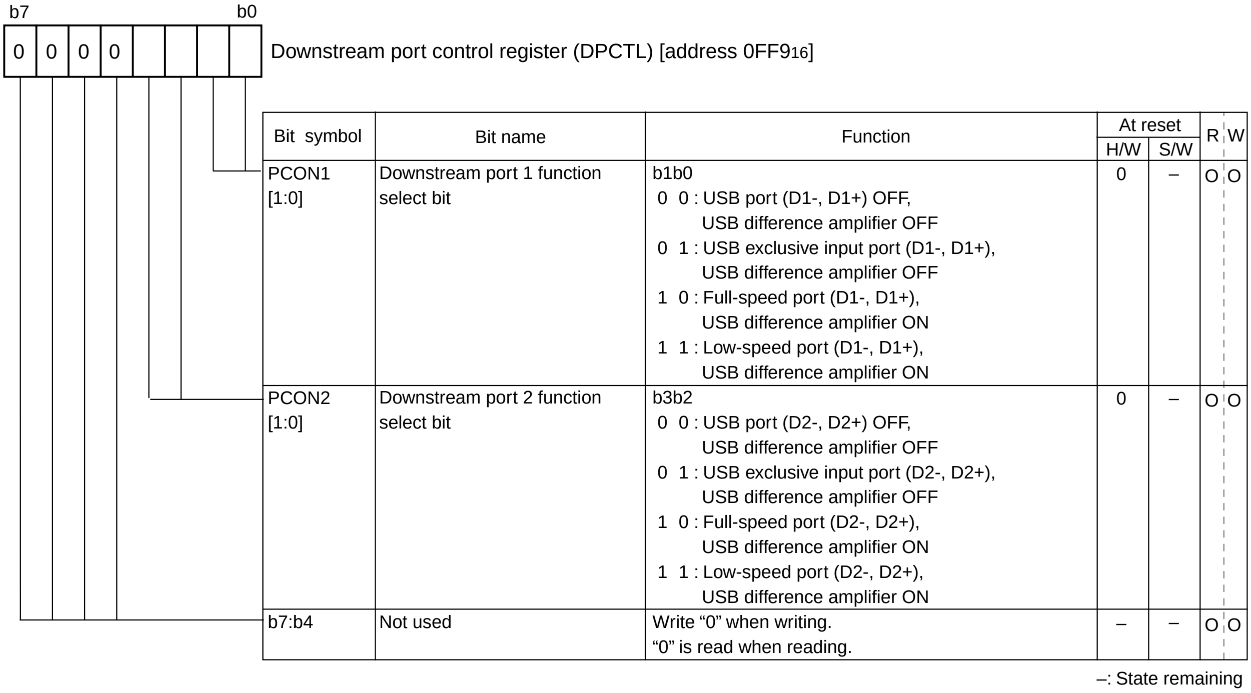
Fig. 3.4.101 Structure of Interrupt edge selection register



PLL control register (PLLCON)
[Address : 0FF8$_{16}$]

B	Name	Function	At reset	R	W
0 1 2	Nothing is arranged for these bit. These are write disabled bits. When these bits are read out, the contents are "0".		0	○	×
3 4	USB clock division ratio selection bits	b4 b3 0 0 : Divided by 8 ($f_{SYN} = f_{USB}/8$) 0 1 : Divided by 6 ($f_{SYN} = f_{USB}/6$) 1 0 : Divided by 4 ($f_{SYN} = f_{USB}/4$) 1 1 : Not selected	0	○	○
5 6	PLL operation mode selection bits	b6 b5 0 0 : Not multiplied ($f_{VCO} = f_{XIN}$) 0 1 : Double ($f_{VCO} = f_{XIN} \times 2$) 1 0 : Quadruple ($f_{VCO} = f_{XIN} \times 4$) 1 1 : Multiplied by 8 ($f_{VCO} = f_{XIN} \times 8$)	0	○	○
7	PLL enable bit	0 : Disabled 1 : Enabled	0	○	○

Fig. 3.4.102 Structure of PLL control register



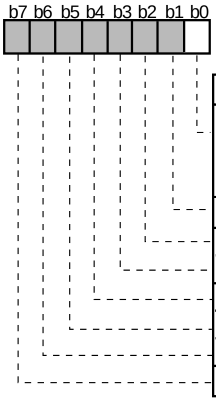
Downstream port control register (DPCTL) [address 0FF9$_{16}$]

Bit symbol	Bit name	Function	At reset		R	W
			H/W	S/W		
PCON1 [1:0]	Downstream port 1 function select bit	b1b0 0 0 : USB port (D1-, D1+) OFF, USB difference amplifier OFF 0 1 : USB exclusive input port (D1-, D1+), USB difference amplifier OFF 1 0 : Full-speed port (D1-, D1+), USB difference amplifier ON 1 1 : Low-speed port (D1-, D1+), USB difference amplifier ON	0	–	O	O
PCON2 [1:0]	Downstream port 2 function select bit	b3b2 0 0 : USB port (D2-, D2+) OFF, USB difference amplifier OFF 0 1 : USB exclusive input port (D2-, D2+), USB difference amplifier OFF 1 0 : Full-speed port (D2-, D2+), USB difference amplifier ON 1 1 : Low-speed port (D2-, D2+), USB difference amplifier ON	0	–	O	O
b7:b4	Not used	Write "0" when writing. "0" is read when reading.	–	–	O	O

–: State remaining

Fig. 3.4.103 Structure of Downstream port control register

MISRG

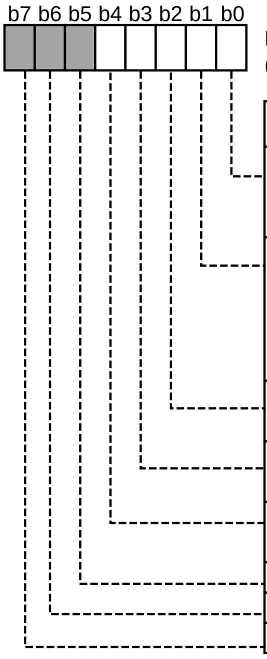


MISRG
(MISRG: address $0FFB_{16}$)

B	Name	Functions	At reset	R	W
0	Oscillation stabilizing time set after STP instruction released bit	0 : Automatically set “01_{16}” to Timer 1, “FF_{16}” to Prescaler 12 1 : Automatically set nothing	0	○	○
1	Nothing is arranged for these bits. These are write disabled bits. When these bits are read out, the contents are indefinite.		?	×	×
2					
3					
4					
5					
6					
7					

Fig. 3.4.104 Structure of MISRG

Flash memory control register



Flash memory control register
(FMCR : address $0FFE_{16}$) **(Note 1)**

b	Name	Functions	At reset	R	W
0	RY/$\overline{\text{BY}}$ status flag	0 : Busy (being written or erased) 1 : Ready	1	○	×
1	CPU rewrite mode select bit **(Note 2)**	0 : Normal mode (Software commands invalid) 1 : CPU rewrite mode (Software commands acceptable)	0	○	○
2	CPU rewrite mode entry flag	0: Normal mode 1: CPU rewrite mode	0	○	×
3	Flash memory reset bit **(Note 3)**	0: Normal operation 1: Reset	0	○	○
4	User area/Boot area selection bit **(Note 4)**	0: User ROM area 1: Boot ROM area	0	○	○
5	Nothing is arranged for these bits. If writing, set “0”. When these bits are read out, the contents are undefined.		Undefined	×	○
6			Undefined	×	○
7			Undefined	×	○

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”.

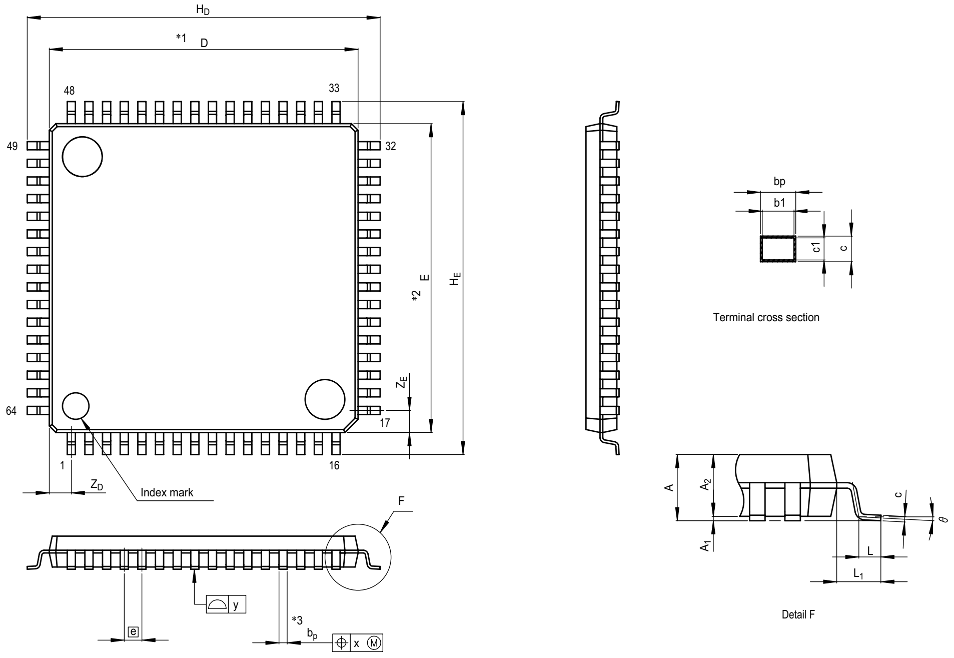
4: Use the control program in the area except the built-in flash memory for write to this bit.

Fig. 3.4.105 Structure of Flash memory control register

3.5 Package outline

PLQP0064GA-A

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

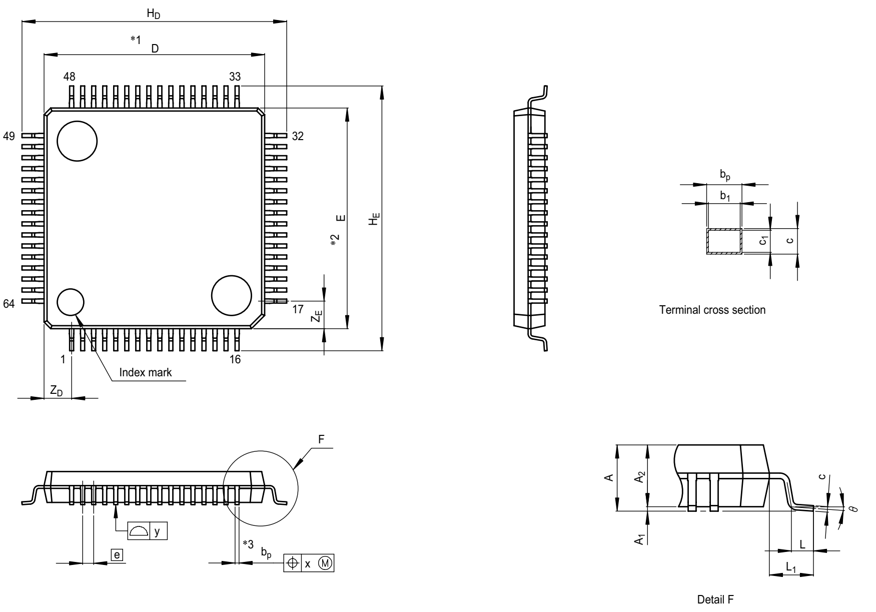


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

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	—

THIS PAGE IS BLANK FOR REASONS OF LAYOUT.

3.6 Machine instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
ADC (Note 1) (Note 5)	When T = 0 A ← A + M + C When T = 1 M(X) ← M(X) + M + C	When T = 0, this instruction adds the contents M, C, and A; and stores the results in A and C. When T = 1, this instruction adds the contents of M(X), M and C; and stores the results in M(X) and C. When T=1, the contents of A remain unchanged, but the contents of status flags are changed. M(X) represents the contents of memory where is indicated by X.				69	2	2							65	3	2			
AND (Note 1)	When T = 0 A ← A ∧ M When T = 1 M(X) ← M(X) ∧ M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise AND operation and stores the result back in A. When T = 1, this instruction transfers the contents M(X) and M to the ALU which performs a bit-wise AND operation and stores the results back in M(X). When T = 1 the contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				29	2	2							25	3	2			
ASL	7 0 C ← ▭ ← 0	This instruction shifts the content of A or M by one bit to the left, with bit 0 always being set to 0 and bit 7 of A or M always being contained in C.							0A	2	1				06	5	2			
BBC (Note 4)	Ai or Mi = 0?	This instruction tests the designated bit i of M or A and takes a branch if the bit is 0. The branch address is specified by a relative address. If the bit is 1, next instruction is executed.										13 + 20i	4	2				17 + 20i	5	3
BBS (Note 4)	Ai or Mi = 1?	This instruction tests the designated bit i of the M or A and takes a branch if the bit is 1. The branch address is specified by a relative address. If the bit is 0, next instruction is executed.										03 + 20i	4	2				07 + 20i	5	3
BCC (Note 4)	C = 0?	This instruction takes a branch to the appointed address if C is 0. The branch address is specified by a relative address. If C is 1, the next instruction is executed.																		
BCS (Note 4)	C = 1?	This instruction takes a branch to the appointed address if C is 1. The branch address is specified by a relative address. If C is 0, the next instruction is executed.																		
BEQ (Note 4)	Z = 1?	This instruction takes a branch to the appointed address when Z is 1. The branch address is specified by a relative address. If Z is 0, the next instruction is executed.																		
BIT	A ∧ M	This instruction takes a bit-wise logical AND of A and M contents; however, the contents of A and M are not modified. The contents of N, V, Z are changed, but the contents of A, M remain unchanged.													24	3	2			
BMI (Note 4)	N = 1?	This instruction takes a branch to the appointed address when N is 1. The branch address is specified by a relative address. If N is 0, the next instruction is executed.																		
BNE (Note 4)	Z = 0?	This instruction takes a branch to the appointed address if Z is 0. The branch address is specified by a relative address. If Z is 1, the next instruction is executed.																		

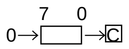
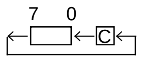
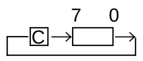
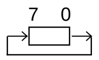
Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
ADC	75	4	2				6D	4	3	7D	5	3	79	5	3							61	6	2	71	6	2							N	V	•	•	•	•	Z	C
AND	35	4	2				2D	4	3	3D	5	3	39	5	3							21	6	2	31	6	2							N	•	•	•	•	•	Z	•
ASL	16	6	2				0E	6	3	1E	7	3																						N	•	•	•	•	•	Z	C
BBC																																		•	•	•	•	•	•	•	•
BBS																																		•	•	•	•	•	•	•	•
BCC																												90	2	2				•	•	•	•	•	•	•	•
BCS																												B0	2	2				•	•	•	•	•	•	•	•
BEQ																												F0	2	2				•	•	•	•	•	•	•	•
BIT							2C	4	3																									M_7	M_6	•	•	•	•	Z	•
BMI																												30	2	2				•	•	•	•	•	•	•	•
BNE																												D0	2	2				•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
BPL (Note 4)	N = 0?	This instruction takes a branch to the appointed address if N is 0. The branch address is specified by a relative address. If N is 1, the next instruction is executed.																		
BRA	PC ← PC ± offset	This instruction branches to the appointed address. The branch address is specified by a relative address.																		
BRK	B ← 1 (PC) ← (PC) + 2 M(S) ← PC_H S ← S – 1 M(S) ← PC_L S ← S – 1 M(S) ← PS S ← S – 1 I← 1 PC_L ← AD_L PC_H ← AD_H	When the BRK instruction is executed, the CPU pushes the current PC contents onto the stack. The BADRS designated in the interrupt vector table is stored into the PC.	00	7	1															
BVC (Note 4)	V = 0?	This instruction takes a branch to the appointed address if V is 0. The branch address is specified by a relative address. If V is 1, the next instruction is executed.																		
BVS (Note 4)	V = 1?	This instruction takes a branch to the appointed address when V is 1. The branch address is specified by a relative address. When V is 0, the next instruction is executed.																		
CLB	Ai or Mi ← 0	This instruction clears the designated bit i of A or M.										1B + 20i	2	1				1F + 20i	5	2
CLC	C ← 0	This instruction clears C.	18	2	1															
CLD	D ← 0	This instruction clears D.	D8	2	1															
CLI	I ← 0	This instruction clears I.	58	2	1															
CLT	T ← 0	This instruction clears T.	12	2	1															
CLV	V ← 0	This instruction clears V.	B8	2	1															
CMP (Note 3)	When T = 0 A – M When T = 1 M(X) – M	When T = 0, this instruction subtracts the contents of M from the contents of A. The result is not stored and the contents of A or M are not modified. When T = 1, the CMP subtracts the contents of M from the contents of M(X). The result is not stored and the contents of X, M, and A are not modified. M(X) represents the contents of memory where is indicated by X.				C9	2	2							C5	3	2			
COM	M ← $\overline{M}$	This instruction takes the one's complement of the contents of M and stores the result in M.													44	5	2			
CPX	X – M	This instruction subtracts the contents of M from the contents of X. The result is not stored and the contents of X and M are not modified.				E0	2	2							E4	3	2			
CPY	Y – M	This instruction subtracts the contents of M from the contents of Y. The result is not stored and the contents of Y and M are not modified.				C0	2	2							C4	3	2			
DEC	A ← A – 1 or M ← M – 1	This instruction subtracts 1 from the contents of A or M.							1A	2	1				C6	5	2			

Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
BPL																												10	2	2				•	•	•	•	•	•	•	•
BRA																												80	4	2				•	•	•	•	•	•	•	•
BRK																																		•	•	•	1	•	1	•	•
BVC																												50	2	2				•	•	•	•	•	•	•	•
BVS																												70	2	2				•	•	•	•	•	•	•	•
CLB																																		•	•	•	•	•	•	•	•
CLC																																		•	•	•	•	•	•	•	0
CLD																																		•	•	•	•	0	•	•	•
CLI																																		•	•	•	•	•	0	•	•
CLT																																		•	•	0	•	•	•	•	•
CLV																																		•	0	•	•	•	•	•	•
CMP	D5	4	2				CD	4	3	DD	5	3	D9	5	3							C1	6	2	D1	6	2							N	•	•	•	•	•	Z	C
COM																																		N	•	•	•	•	•	Z	•
CPX							EC	4	3																									N	•	•	•	•	•	Z	C
CPY							CC	4	3																									N	•	•	•	•	•	Z	C
DEC	D6	6	2				CE	6	3	DE	7	3																						N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
DEX	X ← X – 1	This instruction subtracts one from the current contents of X.	CA	2	1															
DEY	Y ← Y – 1	This instruction subtracts one from the current contents of Y.	88	2	1															
DIV	A ← (M(zz + X + 1), M(zz + X)) / A M(S) ← one's complement of Remainder S ← S – 1	Divides the 16-bit data in M(zz+(X)) (low-order byte) and M(zz+(X)+1) (high-order byte) by the contents of A. The quotient is stored in A and the one's complement of the remainder is pushed onto the stack.																		
EOR (Note 1)	When T = 0 A ← A ∀ M When T = 1 M(X) ← M(X) ∀ M	When T = 0, this instruction transfers the contents of the M and A to the ALU which performs a bit-wise Exclusive OR, and stores the result in A. When T = 1, the contents of M(X) and M are transferred to the ALU, which performs a bit-wise Exclusive OR and stores the results in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				49	2	2							45	3	2			
INC	A ← A + 1 or M ← M + 1	This instruction adds one to the contents of A or M.							3A	2	1				E6	5	2			
INX	X ← X + 1	This instruction adds one to the contents of X.	E8	2	1															
INY	Y ← Y + 1	This instruction adds one to the contents of Y.	C8	2	1															
JMP	If addressing mode is ABS $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ If addressing mode is IND $PC_L \leftarrow M (AD_H, AD_L)$ $PC_H \leftarrow M (AD_H, AD_L + 1)$ If addressing mode is ZP, IND $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	This instruction jumps to the address designated by the following three addressing modes: Absolute Indirect Absolute Zero Page Indirect Absolute																		
JSR	$M(S) \leftarrow PC_H$ S ← S – 1 $M(S) \leftarrow PC_L$ S ← S – 1 After executing the above, if addressing mode is ABS, $PC_L \leftarrow AD_L$ $PC_H \leftarrow AD_H$ if addressing mode is SP, $PC_L \leftarrow AD_L$ $PC_H \leftarrow FF$ If addressing mode is ZP, IND, $PC_L \leftarrow M(00, AD_L)$ $PC_H \leftarrow M(00, AD_L + 1)$	This instruction stores the contents of the PC in the stack, then jumps to the address designated by the following addressing modes: Absolute Special Page Zero Page Indirect Absolute																		
LDA (Note 2)	When T = 0 A ← M When T = 1 M(X) ← M	When T = 0, this instruction transfers the contents of M to A. When T = 1, this instruction transfers the contents of M to (M(X)). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				A9	2	2							A5	3	2			
LDM	M ← nn	This instruction loads the immediate value in M.													3C	4	3			
LDX	X ← M	This instruction loads the contents of M in X.				A2	2	2							A6	3	2			
LDY	Y ← M	This instruction loads the contents of M in Y.				A0	2	2							A4	3	2			

Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
DEX																																		N	•	•	•	•	•	Z	•
DEY																																		N	•	•	•	•	•	Z	•
DIV	E2	16	2																															•	•	•	•	•	•	•	•
EOR	55	4	2				4D	4	3	5D	5	3	59	5	3							41	6	2	51	6	2							N	•	•	•	•	•	Z	•
INC	F6	6	2				EE	6	3	FE	7	3																						N	•	•	•	•	•	Z	•
INX																																		N	•	•	•	•	•	Z	•
INY																																		N	•	•	•	•	•	Z	•
JMP							4C	3	3							6C	5	3	B2	4	2													•	•	•	•	•	•	•	•
JSR							20	6	3										02	7	2										22	5	2	•	•	•	•	•	•	•	•
LDA	B5	4	2				AD	4	3	BD	5	3	B9	5	3							A1	6	2	B1	6	2							N	•	•	•	•	•	Z	•
LDM																																		•	•	•	•	•	•	•	•
LDX				B6	4	2	AE	4	3				BE	5	3																			N	•	•	•	•	•	Z	•
LDY	B4	4	2				AC	4	3	BC	5	3																						N	•	•	•	•	•	Z	•

Symbol	Function	Details	Addressing mode																																																		Processor status register							
			IMP			IMM			A			BIT, A			ZP			BIT, ZP			ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
LSR		This instruction shifts either A or M one bit to the right such that bit 7 of the result always is set to 0, and the bit 0 is stored in C.							4A	2	1				46	5	2				56	6	2				4E	6	3	5E	7	3																						0	•	•	•	•	•	Z	C
MUL	M(S) • A ← A ✻ M(zz + X) S ← S – 1	Multiplies Accumulator with the memory specified by the Zero Page X address mode and stores the high-order byte of the result on the Stack and the low-order byte in A.																			62	15	2																															•	•	•	•	•	•	•	•
NOP	PC ← PC + 1	This instruction adds one to the PC but does no otheroperation.	EA	2	1																																																	•	•	•	•	•	•	•	•
ORA (Note 1)	When T = 0 A ← A V M When T = 1 M(X) ← M(X) V M	When T = 0, this instruction transfers the contents of A and M to the ALU which performs a bit-wise “OR”, and stores the result in A. When T = 1, this instruction transfers the contents of M(X) and the M to the ALU which performs a bit-wise OR, and stores the result in M(X). The contents of A remain unchanged, but status flags are changed. M(X) represents the contents of memory where is indicated by X.				09	2	2							05	3	2				15	4	2				0D	4	3	1D	5	3	19	5	3							01	6	2	11	6	2							N	•	•	•	•	•	Z	•
PHA	S ← S – 1	This instruction pushes the contents of A to the memory location designated by S, and decrements the contents of S by one.	48	3	1																																																	•	•	•	•	•	•	•	•
PHP	M(S) ← PS S ← S – 1	This instruction pushes the contents of PS to the memory location designated by S and decrements the contents of S by one.	08	3	1																																																	•	•	•	•	•	•	•	•
PLA	S ← S + 1 A ← M(S)	This instruction increments S by one and stores the contents of the memory designated by S in A.	68	4	1																																																	N	•	•	•	•	•	Z	•
PLP	S ← S + 1 PS ← M(S)	This instruction increments S by one and stores the contents of the memory location designated by S in PS.	28	4	1																																																	(Value saved in stack)							
ROL		This instruction shifts either A or M one bit left through C. C is stored in bit 0 and bit 7 is stored in C.							2A	2	1				26	5	2				36	6	2				2E	6	3	3E	7	3																						N	•	•	•	•	•	Z	C
ROR		This instruction shifts either A or M one bit right through C. C is stored in bit 7 and bit 0 is stored in C.							6A	2	1				66	5	2				76	6	2				6E	6	3	7E	7	3																						N	•	•	•	•	•	Z	C
RRF		This instruction rotates 4 bits of the M content to the right.													82	8	2																																					•	•	•	•	•	•	•	•
RTI	S ← S + 1 PS ← M(S) S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S)	This instruction increments S by one, and stores the contents of the memory location designated by S in PS. S is again incremented by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and stores the contents of memory location designated by S in PCH.	40	6	1																																																	(Value saved in stack)							
RTS	S ← S + 1 PCL ← M(S) S ← S + 1 PCH ← M(S) (PC) ← (PC) + 1	This instruction increments S by one and stores the contents of the memory location designated by S in PCL. S is again incremented by one and the contents of the memory location is stored in PCH. PC is incremented by 1.	60	6	1																																																	•	•	•	•	•	•	•	•

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A			ZP			BIT, ZP		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
SBC (Note 1) (Note 5)	When T = 0 A ← A – M – C When T = 1 M(X) ← M(X) – M – C	When T = 0, this instruction subtracts the value of M and the complement of C from A, and stores the results in A and C. When T = 1, the instruction subtracts the contents of M and the complement of C from the contents of M(X), and stores the results in M(X) and C. A remain unchanged, but status flag are changed. M(X) represents the contents of memory where is indicated by X.				E9	2	2							E5	3	2			
SEB	Ai or Mi ← 1	This instruction sets the designated bit i of A or M.										0B + 20i	2	1				0F + 20i	5	2
SEC	C ← 1	This instruction sets C.	38	2	1															
SED	D ← 1	This instruction set D.	F8	2	1															
SEI	I ← 1	This instruction set I.	78	2	1															
SET	T ← 1	This instruction set T.	32	2	1															
STA	M ← A	This instruction stores the contents of A in M. The contents of A does not change.													85	4	2			
STP		This instruction resets the oscillation control F/F and the oscillation stops. Reset or interrupt input is needed to wake up from this mode.	42	2	1															
STX	M ← X	This instruction stores the contents of X in M. The contents of X does not change.													86	4	2			
STY	M ← Y	This instruction stores the contents of Y in M. The contents of Y does not change.													84	4	2			
TAX	X ← A	This instruction stores the contents of A in X. The contents of A does not change.	AA	2	1															
TAY	Y ← A	This instruction stores the contents of A in Y. The contents of A does not change.	A8	2	1															
TST	M = 0?	This instruction tests whether the contents of M are “0” or not and modifies the N and Z.													64	3	2			
TSX	X ← S	This instruction transfers the contents of S in X.	BA	2	1															
TXA	A ← X	This instruction stores the contents of X in A.	8A	2	1															
TXS	S ← X	This instruction stores the contents of X in S.	9A	2	1															
TYA	A ← Y	This instruction stores the contents of Y in A.	98	2	1															
WIT		The WIT instruction stops the internal clock but not the oscillation of the oscillation circuit is not stopped. CPU starts its function after the Timer X over flows (comes to the terminal count). All registers or internal memory contents except Timer X will not change during this mode. (Of course needs VDD).	C2	2	1															

Notes 1 : The number of cycles “n” is increased by 3 when T is 1.
2 : The number of cycles “n” is increased by 2 when T is 1.
3 : The number of cycles “n” is increased by 1 when T is 1.
4 : The number of cycles “n” is increased by 2 when branching has occurred.
5 : N, V, and Z flags are invalid in decimal operation mode.

Symbol	Addressing mode																																	Processor status register							
	ZP, X			ZP, Y			ABS			ABS, X			ABS, Y			IND			ZP, IND			IND, X			IND, Y			REL			SP			7	6	5	4	3	2	1	0
	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	N	V	T	B	D	I	Z	C
SBC	F5	4	2				ED	4	3	FD	5	3	F9	5	3							E1	6	2	F1	6	2							N	V	•	•	•	•	Z	C
SEB																																		•	•	•	•	•	•	•	•
SEC																																		•	•	•	•	•	•	•	1
SED																																		•	•	•	•	1	•	•	•
SEI																																		•	•	•	•	•	1	•	•
SET																																		•	•	1	•	•	•	•	•
STA	95	5	2				8D	5	3	9D	6	3	99	6	3							81	7	2	91	7	2							•	•	•	•	•	•	•	•
STP																																		•	•	•	•	•	•	•	•
STX				96	5	2	8E	5	3																									•	•	•	•	•	•	•	•
STY	94	5	2				8C	5	3																									•	•	•	•	•	•	•	•
TAX																																		N	•	•	•	•	•	Z	•
TAY																																		N	•	•	•	•	•	Z	•
TST																																		N	•	•	•	•	•	Z	•
TSX																																		N	•	•	•	•	•	Z	•
TXA																																		N	•	•	•	•	•	Z	•
TXS																																		•	•	•	•	•	•	•	•
TYA																																		N	•	•	•	•	•	Z	•
WIT																																		•	•	•	•	•	•	•	•

Symbol	Contents	Symbol	Contents
IMP	Implied addressing mode	+	Addition
IMM	Immediate addressing mode	–	Subtraction
A	Accumulator or Accumulator addressing mode	✻	Multiplication
BIT, A	Accumulator bit addressing mode	/	Division
BIT, A, R	Accumulator bit relative addressing mode	∧	Logical OR
ZP	Zero page addressing mode	V	Logical AND
BIT, ZP	Zero page bit addressing mode	∀	Logical exclusive OR
BIT, ZP, R	Zero page bit relative addressing mode	—	Negation
ZP, X	Zero page X addressing mode	←	Shows direction of data flow
ZP, Y	Zero page Y addressing mode	X	Index register X
ABS	Absolute addressing mode	Y	Index register Y
ABS, X	Absolute X addressing mode	S	Stack pointer
ABS, Y	Absolute Y addressing mode	PC	Program counter
IND	Indirect absolute addressing mode	PS	Processor status register
		PCH	8 high-order bits of program counter
ZP, IND	Zero page indirect absolute addressing mode	PCL	8 low-order bits of program counter
		ADH	8 high-order bits of address
IND, X	Indirect X addressing mode	ADL	8 low-order bits of address
IND, Y	Indirect Y addressing mode	FF	FF in Hexadecimal notation
REL	Relative addressing mode	nn	Immediate value
SP	Special page addressing mode	zz	Zero page address
C	Carry flag	M	Memory specified by address designation of any addressing mode
Z	Zero flag		
I	Interrupt disable flag	M(X)	Memory of address indicated by contents of index register X
D	Decimal mode flag		
B	Break flag	M(S)	Memory of address indicated by contents of stack pointer
T	X-modified arithmetic mode flag		
V	Overflow flag	M(ADH, ADL)	Contents of memory at address indicated by ADH and ADL, in ADH is 8 high-order bits and ADL is 8 low-order bits.
N	Negative flag		
		M(00, ADL)	Contents of address indicated by zero page ADL
		Ai	Bit i (i = 0 to 7) of accumulator
		Mi	Bit i (i = 0 to 7) of memory
		OP	Opcode
		n	Number of cycles
		#	Number of bytes

3.7 List of instruction code

D_3 – D_0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
D_7 – D_4	Hexadecimal notation	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

▨ : 3-byte instruction

▒ : 2-byte instruction

□ : 1-byte instruction

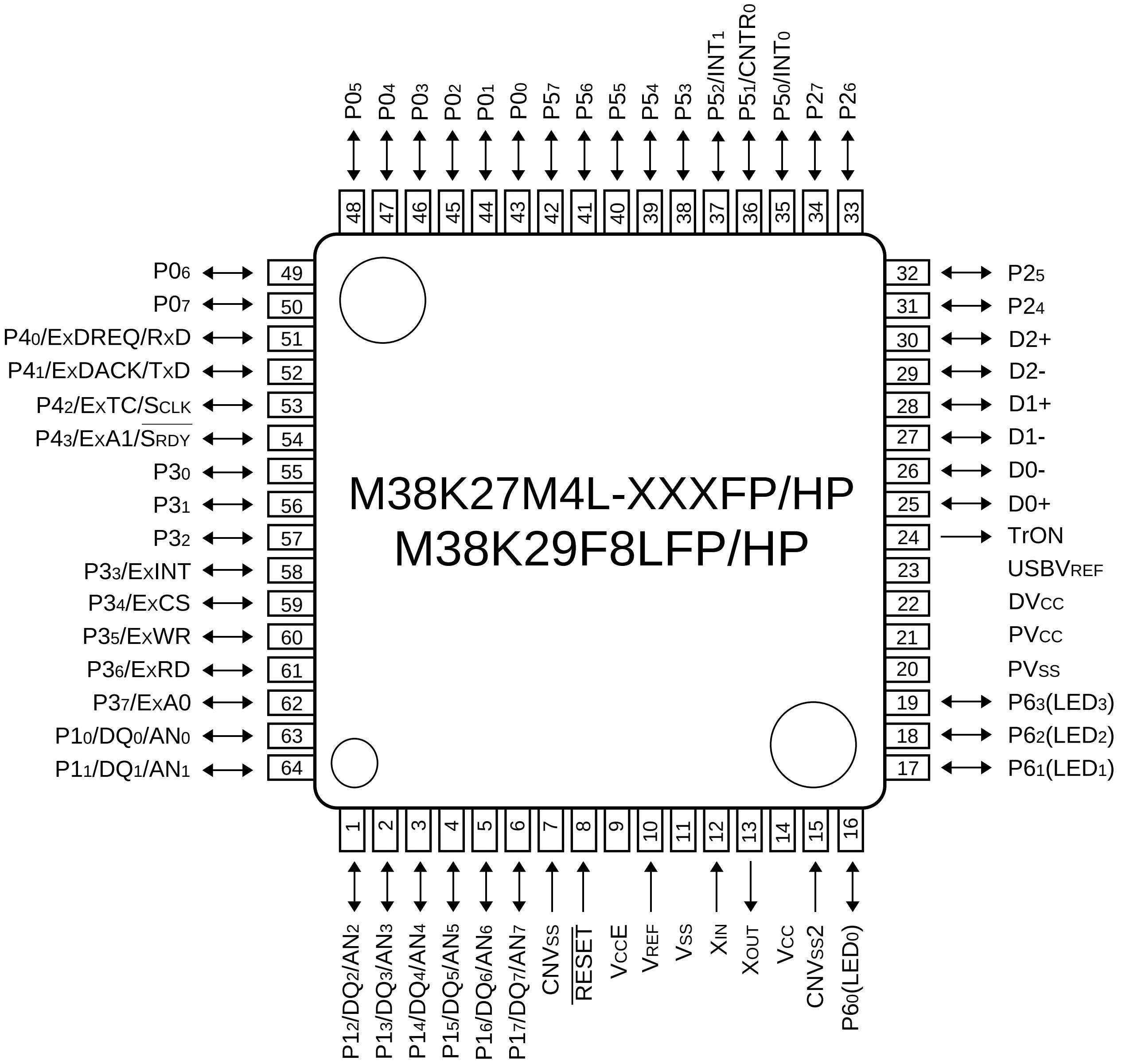
3.8 SFR memory map

0000_{16}	Port P0 (P0)	0020_{16}	Prescaler 12 (PRE12)
0001_{16}	Port P0 direction register (P0D)	0021_{16}	Timer 1 (T1)
0002_{16}	Port P1 (P1)	0022_{16}	Timer 2 (T2)
0003_{16}	Port P1 direction register (P1D)	0023_{16}	Timer X mode register (TM)
0004_{16}	Port P2 (P2)	0024_{16}	Prescaler X (PREX)
0005_{16}	Port P2 direction register (P2D)	0025_{16}	Timer X (TX)
0006_{16}	Port P3 (P3)	0026_{16}	Transmit/Receive buffer register (TB/RB)
0007_{16}	Port P3 direction register (P3D)	0027_{16}	Serial I/O status register (SIOSTS)
0008_{16}	Port P4 (P4)	0028_{16}	HUB interrupt source enable register (HUBICON)
0009_{16}	Port P4 direction register (P4D)	0029_{16}	HUB interrupt source register (HUBIREQ)
$000A_{16}$	Port P5 (P5)	$002A_{16}$	HUB down stream port index register (HUBINDEX)
$000B_{16}$	Port P5 direction register (P5D)	$002B_{16}$	HUB port field register 1 (DPXREG1)
$000C_{16}$	Port P6 (P6)	$002C_{16}$	HUB port field register 2 (DPXREG2)
$000D_{16}$	Port P6 direction register (P6D)	$002D_{16}$	HUB port field register 3 (DPXREG3)
$000E_{16}$	Reserved **(Note)**	$002E_{16}$	Reserved **(Note)**
$000F_{16}$	Reserved **(Note)**	$002F_{16}$	Reserved **(Note)**
0010_{16}	USB control register (USBCON)	0030_{16}	EXB interrupt source enable register (EXBICON)
0011_{16}	USB function/Hub enable register (USBAE)	0031_{16}	EXB interrupt source register (EXBIREQ)
0012_{16}	USB function address register (USBA0)	0032_{16}	Reserved **(Note)**
0013_{16}	USB HUB address register (USBA1)	0033_{16}	EXB index register (EXBINDEX)
0014_{16}	Frame number register Low (FNUML)	0034_{16}	Register window 1 (EXBREG1)
0015_{16}	Frame number register High (FNUMH)	0035_{16}	Register window 2 (EXBREG2)
0016_{16}	USB interrupt source enable register (USBICON)	0036_{16}	AD control register (ADCON)
0017_{16}	USB interrupt source register (USBIREQ)	0037_{16}	AD conversion register 1 (AD1)
0018_{16}	Endpoint index register (USBINDEX)	0038_{16}	AD conversion register 2 (AD2)
0019_{16}	Endpoint field register 1 (EPXXREG1)	0039_{16}	Watchdog timer control register (WDTCON)
$001A_{16}$	Endpoint field register 2 (EPXXREG2)	$003A_{16}$	Reserved **(Note)**
$001B_{16}$	Endpoint field register 3 (EPXXREG3)	$003B_{16}$	CPU mode register (CPUM)
$001C_{16}$	Endpoint field register 4 (EPXXREG4)	$003C_{16}$	Interrupt request register 1(IREQ1)
$001D_{16}$	Endpoint field register 5 (EPXXREG5)	$003D_{16}$	Interrupt request register 2(IREQ2)
$001E_{16}$	Endpoint field register 6 (EPXXREG6)	$003E_{16}$	Interrupt control register 1(ICON1)
$001F_{16}$	Endpoint field register 7 (EPXXREG7)	$003F_{16}$	Interrupt control register 2(ICON2)

$0FE0_{16}$	Serial I/O control register (SIOCON)	$0FF0_{16}$	Port P0 pull-up control register (PULL0)
$0FE1_{16}$	UART control register (UARTCON)	$0FF1_{16}$	Reserved **(Note)**
$0FE2_{16}$	Baud rate generator (BRG)	$0FF2_{16}$	Port P5 pull-up control register (PULL5)
$0FE3_{16}$	Reserved **(Note)**	$0FF3_{16}$	Interrupt edge selection register (INTEDGE)
$0FE4_{16}$	Reserved **(Note)**	$0FF4_{16}$	Reserved **(Note)**
$0FE5_{16}$	Reserved **(Note)**	$0FF5_{16}$	Reserved **(Note)**
$0FE6_{16}$	Reserved **(Note)**	$0FF6_{16}$	Reserved **(Note)**
$0FE7_{16}$	Reserved **(Note)**	$0FF7_{16}$	Reserved **(Note)**
$0FE8_{16}$	Reserved **(Note)**	$0FF8_{16}$	PLL control register (PLLCON)
$0FE9_{16}$	Reserved **(Note)**	$0FF9_{16}$	Downstream port control register (DPCTL)
$0FEA_{16}$	Reserved **(Note)**	$0FFA_{16}$	Reserved **(Note)**
$0FEB_{16}$	Reserved **(Note)**	$0FFB_{16}$	MISRG
$0FEC_{16}$	Endpoint field register 8 (EPXXREG8)	$0FFC_{16}$	Reserved **(Note)**
$0FED_{16}$	Endpoint field register 9 (EPXXREG9)	$0FFD_{16}$	Reserved **(Note)**
$0FEE_{16}$	Reserved **(Note)**	$0FFE_{16}$	Flash memory control register (FMCR)
$0FEF_{16}$	Reserved **(Note)**	$0FFF_{16}$	Reserved **(Note)**

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

3.9 Pin configurations



REVISION HISTORY	38K2 GROUP USER’S MANUAL

Rev.	Date	Description	
		Page	Summary
1.0	2/13/03		First Edition
2.0	10/15/06	All pages	Package names “64P6U-A” → “PLQP0064GA-A” revised Package names “64P6Q-A” → “PLQP0064KB-A” revised 38K2 group (Standard) deleted
		Chapter 1	
		94	Fig. 137 revised
		97	CLOCK GENERATING CIRCUIT; “No external resistor is needed .... resistor exists on-chip.” → “No external resistor is needed .... depending on conditions.)
		98	Fig. 141; Pulled up added, NOTE added Fig. 144 revised
		128	NOTES ON USAGE; Power Source Voltage, USB Communication added
		Chapter 2	
		3	Fig. 2.1.3; “Do not set bits of .... If writing to these bits, write “0”.” added
		Chapter 3	
		20	3.2 deleted
		35	3.3.6 (3) USB Communication added
		48	Fig. 3.5.2; “Do not set bits of .... If writing to these bits, write “0”.” added
		92, 93	3.6 Package outline revised

RENESAS 8-BIT SINGLE-CHIP MICROCOMPUTER
USER'S MANUAL
38K2 Group

Publication Data :	**Rev.1.00**	**Feb 13, 2003**
	Rev.2.00	**Oct 15, 2006**
Published by :	**Sales Strategic Planning Div.**	
	Renesas Technology Corp.	

38K2 Group
User's Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan