To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Description

The M16C/6K7 (144-pin version) group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and are packaged in a 144-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. To communicate with host CPU, the LPC bus interface is built in. In this way, this MCU can work as slave controller in the personal computer system.
The specification is target oriented specification for the development of M16C/6K7 (144-pin version).

Features

- Memory capacity ..... ROM (Refer to the figure of ROM Expansion)
  RAM 3K bytes
- The Min. time of instruction execution ... 125.0ns (f(X_{IN})=8MHz, with 0 wait, Vcc=3V)
- Supply voltage ..... 3.0 to 3.6V (f(X_{IN})=8MHz with 0 wait)
- Low power consumption ..... Target 41.3mW (f(X_{IN})=8MHz, with 0 wait, V_{CC} = 3.3V)
- Interrupts ..... 36 internal and 15 external interrupt sources, 4 software interrupt sources; 7 levels (including key input interrupt)
- Multifunction 16-bit timer ..... 5 output timers + 6 input timers
- Serial I/O ..... 5 channels (3 for UART or clock synchronous, 2 for clock synchronous)
- DMAC ..... 2 channels
- Host interface ..... LPC bus interface X 4
- A-D converter ..... 10 bits X 8 channels (Expandable up to 10 channels)
- D-A converter ..... 8 bits X 2 channels
- Comparator circuit ..... 8 channels
- PWM ..... 14 bits X 4 channels
- Watchdog timer 1
- I^2C bus interface ..... 2 channels
- Serial interrupt output ..... 6 factors (2 fixed factors, 4 programmable factors)
- PS/2 interface ..... 3 channels
- Programmable I/O 129
- Input port ..... 1 ($P8_5$ shared with $\overline{NMI}$ pin)
- Clock generating circuit ..... 2 built-in clock generation circuits
  (built-in feedback resistor, and external ceramic or quartz oscillator)

Applications

Notebook PC, others

Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error. Specifications in this data sheet may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

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

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

Type name	M306K5F8LRP	M306K7F8LRP
Pin numbers	144-pin	144-pin
ROM	NEW DINOR Flash memory	NEW DINOR Flash memory
Built-in ROM area	User ROM area Address $0E8000_{16}$ - $0F5FFF_{16}$ Address $0FE000_{16}$ - $0FFFFF_{16}$ Boot ROM area In parallel I/O mode Address $0FE000_{16}$ - $0FFFFF_{16}$ In CPU reprogram mode & standard serial I/O mode Address $0DE000_{16}$ - $0DFFFF_{16}$	User ROM area Address $0EF000_{16}$ - $0FFFFF_{16}$ Boot ROM area Address $0FF000_{16}$ - $0FFFFF_{16}$
Address $03B4_{16}$	Flash memory control register After reset $XXXX0001_2$	Flash memory recognition register After reset 00000000_2
Address $03B7_{16}$ After reset 00000000_2	Flash memory recognition register After reset 00000000_2	Flash memory control register After reset $XX000001_2$
The power supply for program/erase	Vcc 3.0 - 3.6V M2 pin 4.5 - 5.25V	Vcc 3.0 - 3.6V
M2 pin	The input pin of power supply for program/erase	Not exist
Programmable I/O	128 Port 0 - Port 16 without $P8_4$	129 Port 0 - Port 16
The I/O voltage in P2, P3 Programmable I/O ports	3.3V/5V	3.3V
ISA bus interface	Exist	Not exist
Serial interrupt output & LPC bus interface	Exist	Exist
Timer B TB2IN	Not exist	Exist

Note 1: Timer B TB2IN refers to the count source input and pulse period measurement in event count mode/ pulse input in pulse width measurement mode.

Pin Configuration

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

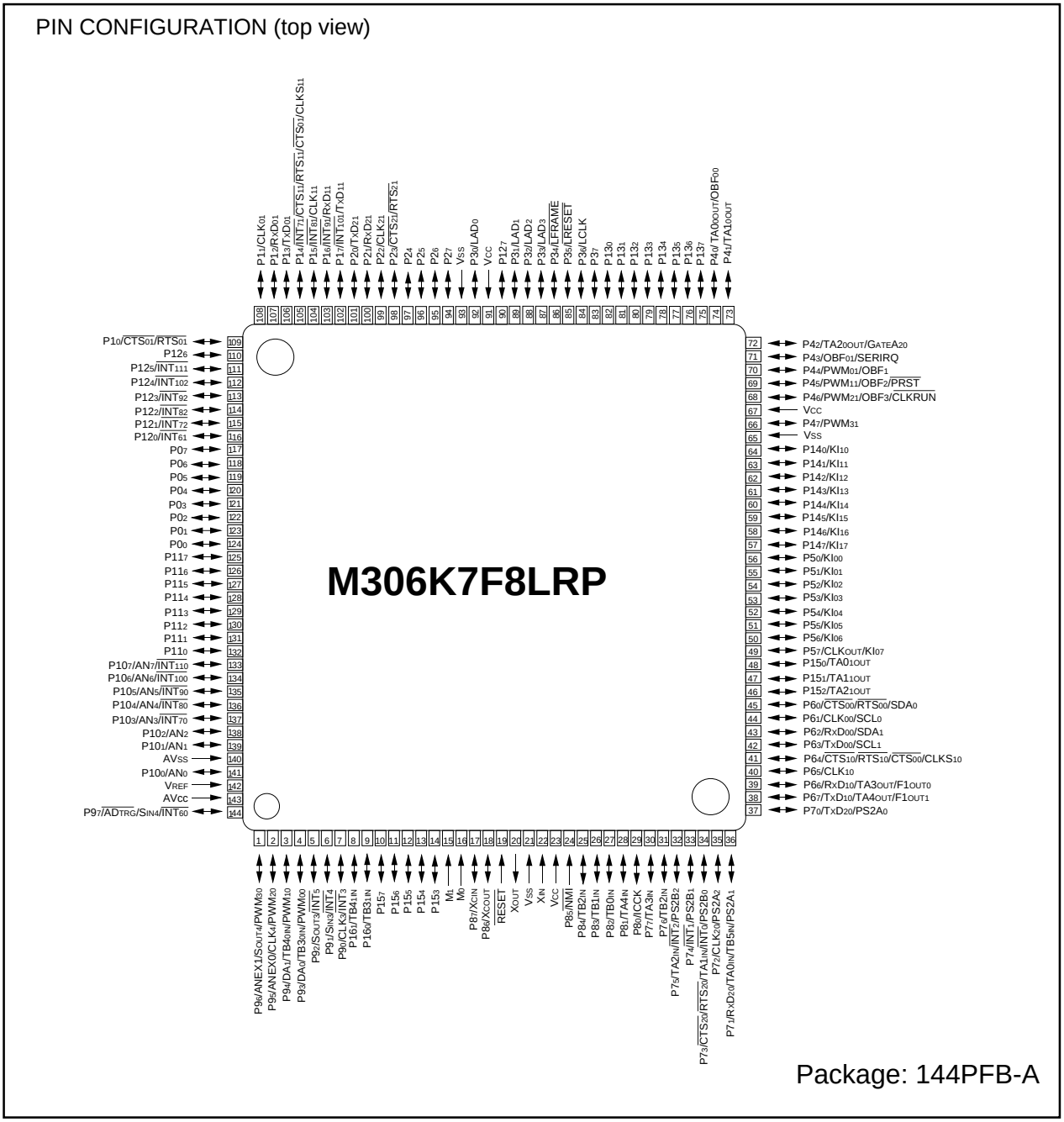
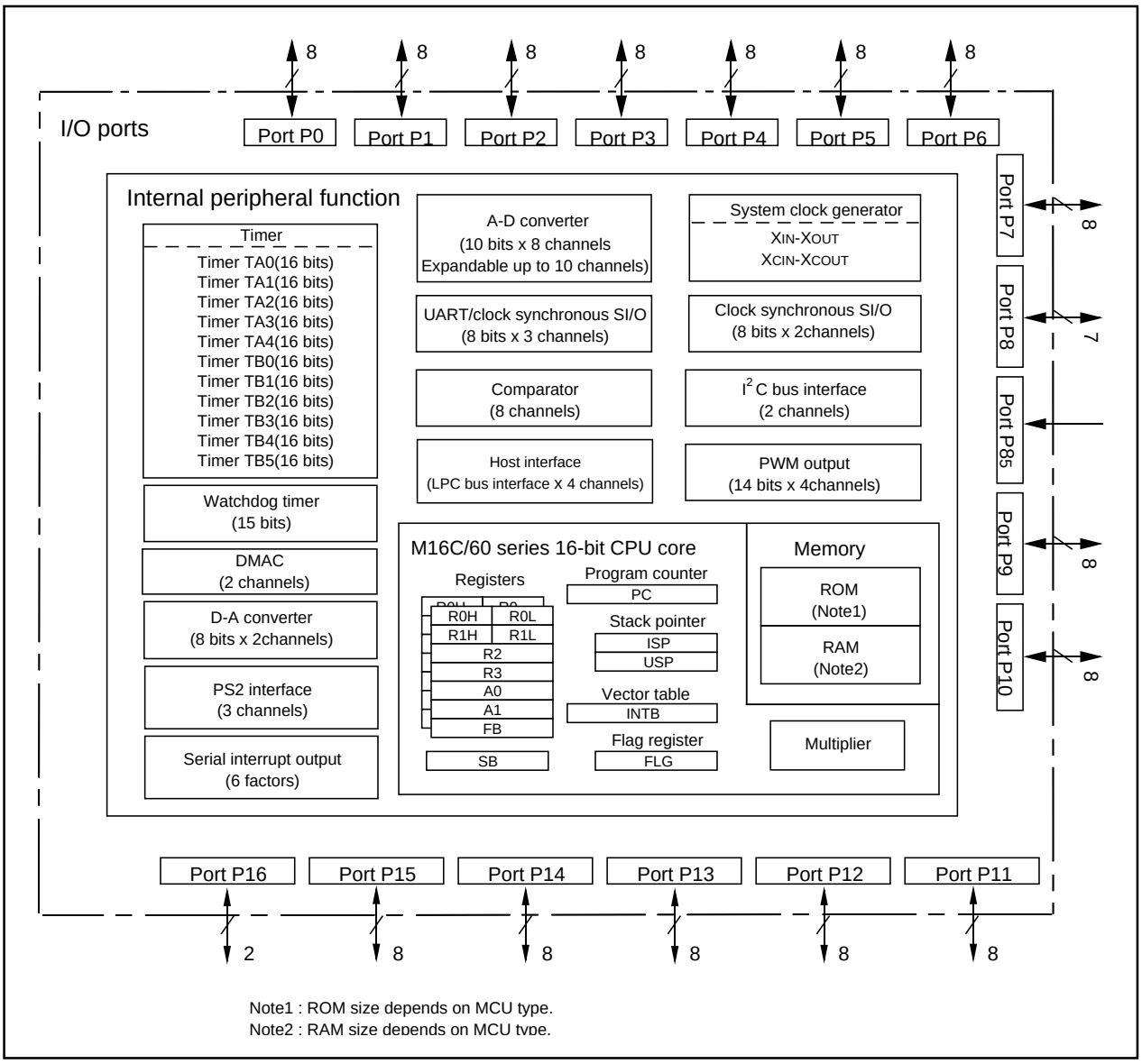


Fig. AA-1 Pin configuration (top view)

Block Diagram

Fig.AA-2 is a block diagram of the M16C/6K7 (144-pin version) group.



Note1 : ROM size depends on MCU type.
Note2 : RAM size depends on MCU type.

Fig.AA-2 Block diagram of M16C/6K7 (144-pin version) group

Performance Outline

Table AA-1 is a performance outline of M16C/6K7 (144-pin version) group.

Table AA-1 Performance outline of M16C/6K7 (144-pin version) group

Item		Performance
Number of basic instructions		91 instructions
The Min. time of instruction execution		125ns (f(XIN)=8MHz, with 0 wait, Vcc=3V)
Memory capacity	ROM	(See the figure of ROM Expansion)
	RAM	3K bytes
I/O port	P0 to P10 (except P85)	8 bits x 10, 7 bits x 1
	P11 to P16	8 bitsx5, 2 bitsx1
Input port	P85	1 bit x 1
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits x 5
	TB0, TB1, TB2, TB3, TB4, TB5	16 bits x 6
Serial I/O	UART0, UART1, UART2	(UART or clock synchronous) x 3
	SI/O3, SI/O4	(Clock synchronous) x 2
A-D converter		10 bits x (8 + 2) channels
D-A converter		8 bits x 2
DMAC		2 channels (trigger: 24 sources)
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		36 internal and 15 external sources, 4 software sources, 7 levels
Host interface		4 channels (LPC bus interface)
Comparator circuit		8 channels
PWM		14 bits x 4
I^2C bus interface		2 channels
PS2 interface		3 channels
Serial interrupt output		6 factors (2 fixed factors, 4 programmable factors)
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)
Supply voltage		3.0 to 3.6V (f(XIN)=8MHz with 0 wait)
Power consumption		41.3mW (3.3V, f(XIN)=8MHz, with 0 wait)
I/O characteristics	I/O withstand voltage	3.3V
	Output current	5mA
Device configuration		CMOS high performance silicon gate
Package		144-pin plastic mold QFP

Mitsubishi plans to release the following products in the M16C/6K7 (144-pin version) group:

(1) Support for flash memory version

(2) ROM capacity

(3) Package

144PFB-A : Plastic molded QFP(flash memory version)

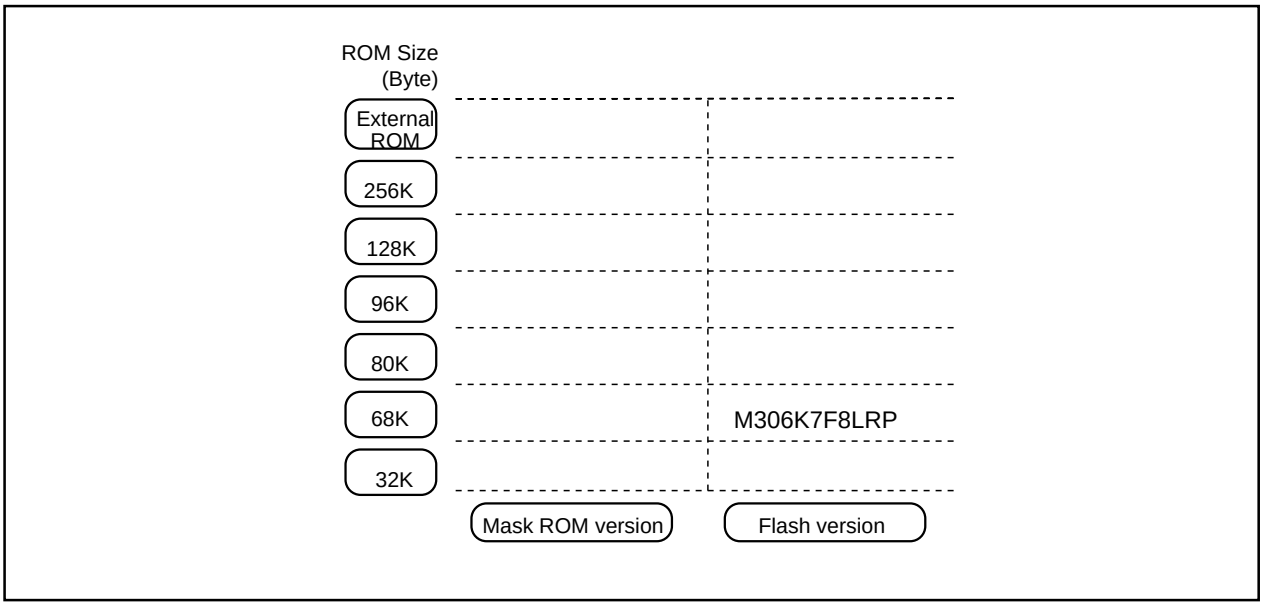


Fig.AA-3 ROM expansion

Table AA-2 Product list

Type No.	ROM size	RAM size	Package type	Host Interface	Remarks
M306K7F8LRP	68K bytes	3K bytes	144PFB-A	LPC	Flash memory (NEW DINOR) version

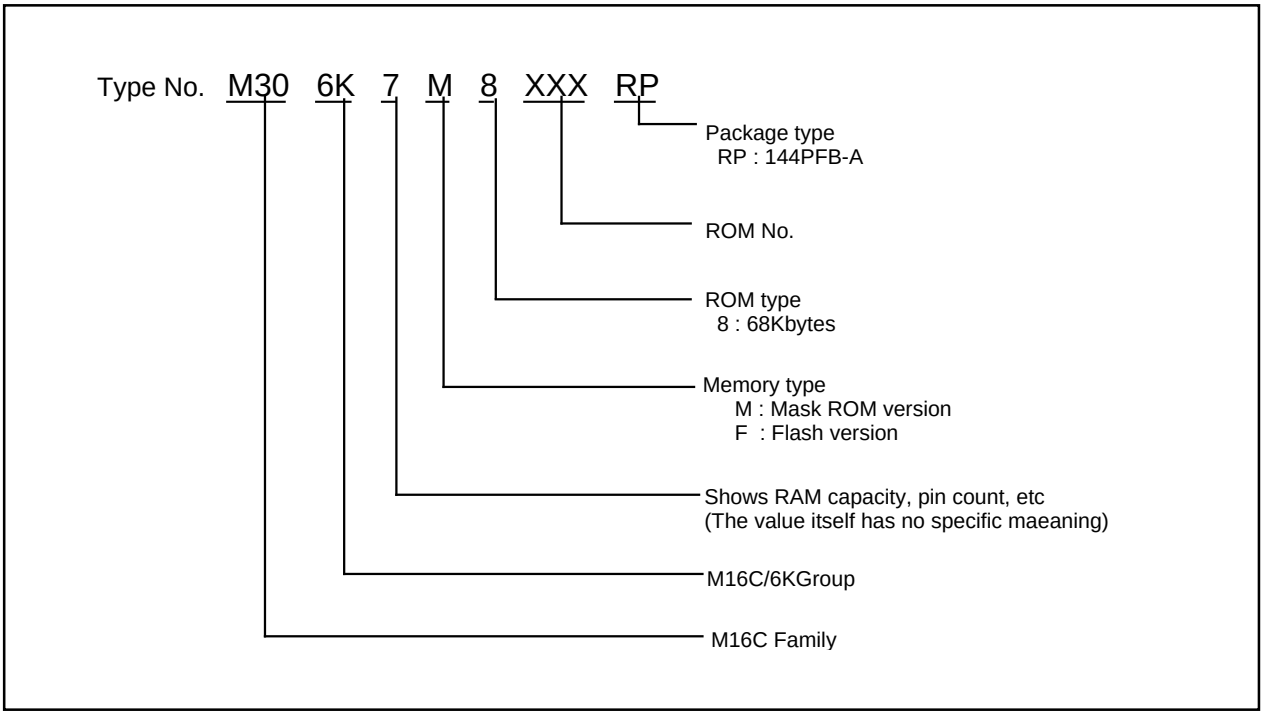


Fig.AA-4 Type No., memory size, and package

Pin Description

Pin name	Signal name	I/O type	Function
Vcc, Vss	Power supply input		Apply 3.0 to 3.6 V to Vcc . Apply 0V to Vss
$\overline{\text{RESET}}$	Reset input	Input	A “L” on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
M0,M1	Chip mode setting	Input	Connect to Vss
AVcc	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vcc.
AVss	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vss.
VREF	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
P00 to P07	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor. This port supports CMOS input level. And output type supports CMOS 3 state or N channel open drain selectable.
P10 to P17	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as external interrupt pins as selected by software.
P20 to P27	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type just supports CMOS 3 state only). The 4 bits P24-P27 are available for directly driving LED's.
P30 to P37	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type just supports CMOS 3 state only). The port can be used for LPC bus interface I/O pins by software selection.
P40 to P47	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type just supports CMOS 3 state only). By software selecting, the port can also be used for LPC bus interface I/O pins, Timer A0 to A2 output pins PWM output pins or serial interrupt output I/O pins. P40 to P46 pins' level can be read regardless the setting of input port or output port. If P40 or P43 are used for output ports, the function that clears P40 or P43 to "0" after the read of output data buffer from host CPU is available.
P50 to P57	I/O port P5	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only). Key on wake interrupt 0 and comparator input function support. P57 in this port outputs a divide-by-8 or divide-by-32 clock of XIN or a clock of the same frequency as XCIN as selected by software.

Pin Description

Pin name	Signal name	I/O type	Function
P60 to P67	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that P60 to P63's output type is N channel open drain only; P64 to P67's output type is CMOS 3 state only; P60 to P63 no internal pull-up registor support.) By software selecting, this port can be used for I2C-BUS interface, UART0/UART1 input/output pin, timerA3,A4 output pin or same frequency with XIN clock output pin. When P60 to P63 used as I2C-BUS interface SDA,SCL, the input level of these pins are CMOS/SMBUS selectable.
P70 to P77	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that P70 to P77 output type is N channel open drain only; no internal pull-up registor support.) By software selecting, this port can be used for external interrupt input pin, timerA0 to A3 and timerB5 input pin, PS2 interface input/output pin, or UART2 input/output pin. P70 to P75 pins' level can be read regardless of the setting of input port or output port.
P80 to P84, P86, P87, P85	I/O port P8 I/O port P85	Input/output Input/output Input/output Input	P80 to P84, P86, and P87 are I/O ports with the same functions as P0. (Except that P86 to P87's output type is CMOS 3 state only; P80 to P84's output type is N channel open drain only; P85 is input port only; the P80 to P84 and P85 are no internal pull-up registor support.) By software selecting, this port can be used for timer A4, B0 to B2, I2C-BUS interface I/O pins. P86 and P87 can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between P86 (XCOUT pin) and P87 (XCIN pin). P85 is an input-only port that also functions for $\overline{\text{NMI}}$. The $\overline{\text{NMI}}$ interrupt is generated when the input at this pin changes from “H” to “L”. The $\overline{\text{NMI}}$ function cannot be cancelled using software.
P90 to P97	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, the port can be used for external interrupt, timer B3 to B4, A-D converter extended input pins, A-D trigger, SI/O3, SI/O4 I/O pins, PWM, D-A converter output pins.
P100 to P107	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, the port can be used for A-D converter, external interrupt input pins.

Pin Description

Pin name	Signal name	I/O type	Function
$P11_0$ to $P11_7$	I/O port P11	Input/output	This is an 8-bit I/O port equivalent to P0.
$P12_0$ to $P12_7$	I/O port P12	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, this port can be used for external interrupt input pin.
$P13_0$ to $P13_7$	I/O port P13	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is N channel open drain only; no internal pull-up registor support.)
$P14_0$ to $P14_7$	I/O port P14	Input/output	This is an 8-bit I/O port equivalent to P0. The port can be used for key on wake-up interrupt 1 input pins. $P14_0$ to $P14_3$ are available for directly driving LED's.
$P15_0$ to $P15_7$	I/O port P15	Input/output	This is an 8-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, this port can be used for timer A0 to A2's output pin.
$P16_0$, $P16_1$	I/O port P16	Input/output	This is an 2-bit I/O port equivalent to P0. (Except that output type is CMOS 3 state only.) By software selecting, this port can be used for timer B3 and B4 input pin.

Operation of Functional Blocks

The M16C/6K7 (144-pin version) group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also peripheral units such as timers, serial I/O, D-A converter, DMAC, A-D converter, host bus interface, comparator, PWM output , I^2C BUS interface, PS2 interface and I/O ports are included.
The following explains each unit.

Memory

Fig.CA-1 is the memory map. The address space extends up to 1M bytes from address 00000_{16} to $FFFFF_{16}$. From $FFFFF_{16}$ to the address decreasing direction ROM is allocated. For example, in the M306K7F8LRP, there is 68K bytes of internal ROM from $EF000_{16}$ to $FFFFF_{16}$. The vector table for fixed interrupts such as the reset and $\overline{NMI}$ are mapped from $FFFDC_{16}$ to $FFFFF_{16}$. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.
From 00400_{16} to the address increasing direction RAM is allocated. For example, in the M306K7F8LRP, 3K bytes of internal RAM is mapped to the space from 00400_{16} to $00FFF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.
The SFR area is mapped from 00000_{16} to $003FF_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Fig.CA-2 to CA-5 are location of peripheral unit control registers. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.
The special page vector table is mapped from $FFE00_{16}$ to $FFFDB_{16}$. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

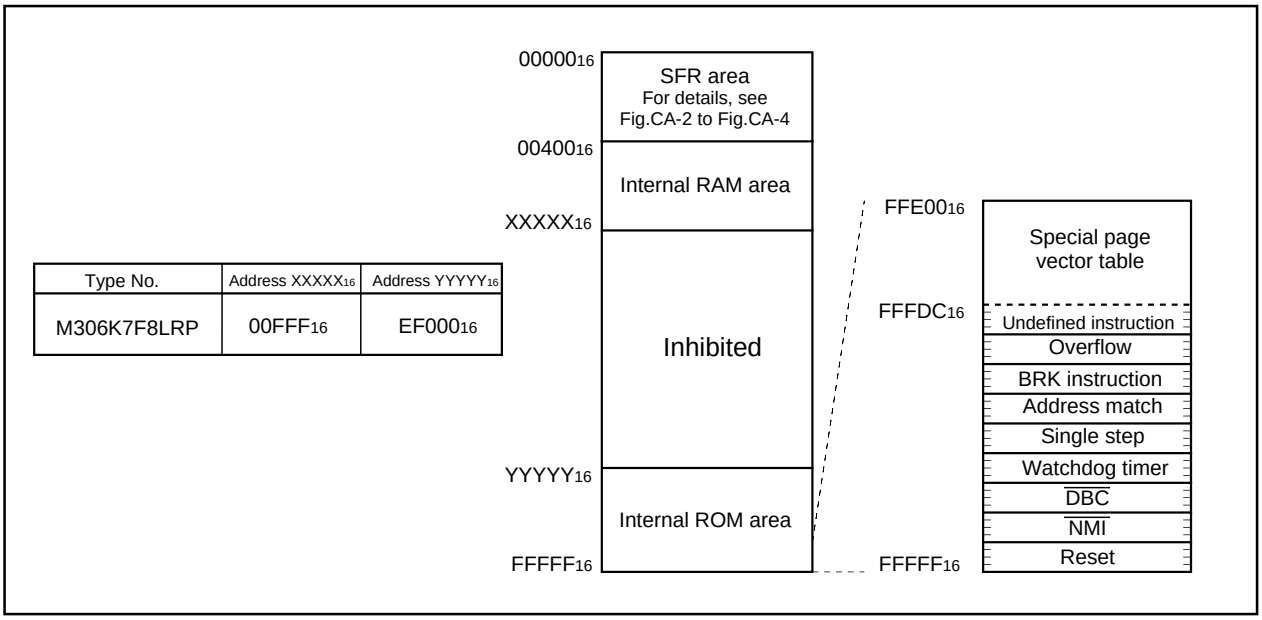


Fig.CA-1 Memory map

Central Processing Unit (CPU)

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

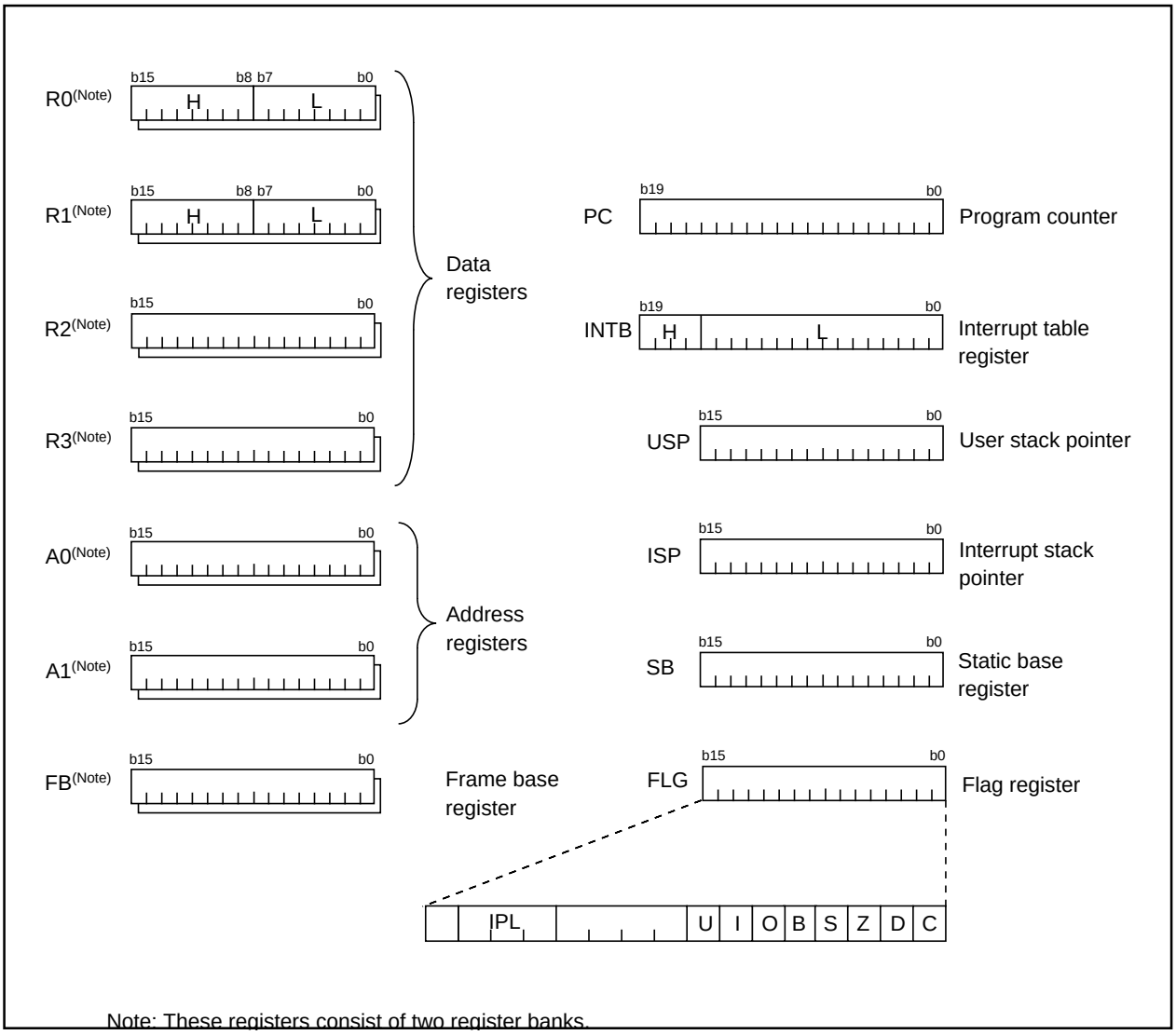


Fig.BA-1 Central processing unit register

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

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

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

(2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be u2sed for address register indirect addressing and address register relative addressing.

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

(3) Frame base register (FB)

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

(4) Program counter (PC)

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

(5) Interrupt table register (INTB)

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

(6) Stack pointer (USP/ISP)

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

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

(7) Static base register (SB)

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

(8) Flag register (FLG)

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

• Bit 0: Carry flag (C flag)

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

• Bit 1: Debug flag (D flag)

This flag enables a single-step interrupt.

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

• Bit 2: Zero flag (Z flag)

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

• Bit 3: Sign flag (S flag)

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

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

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

• Bit 5: Overflow flag (O flag)

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

• Bit 6: Interrupt enable flag (I flag)

This flag enables a maskable interrupt.

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

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

Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.
This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt No. 0 to 31 is executed.
• Bits 8 to 11: Reserved area
• Bits 12 to 14: Processor interrupt priority level (IPL)
Processor interrupt priority level (IPL) is configured with the three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.
If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.
• Bit 15: Reserved area

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

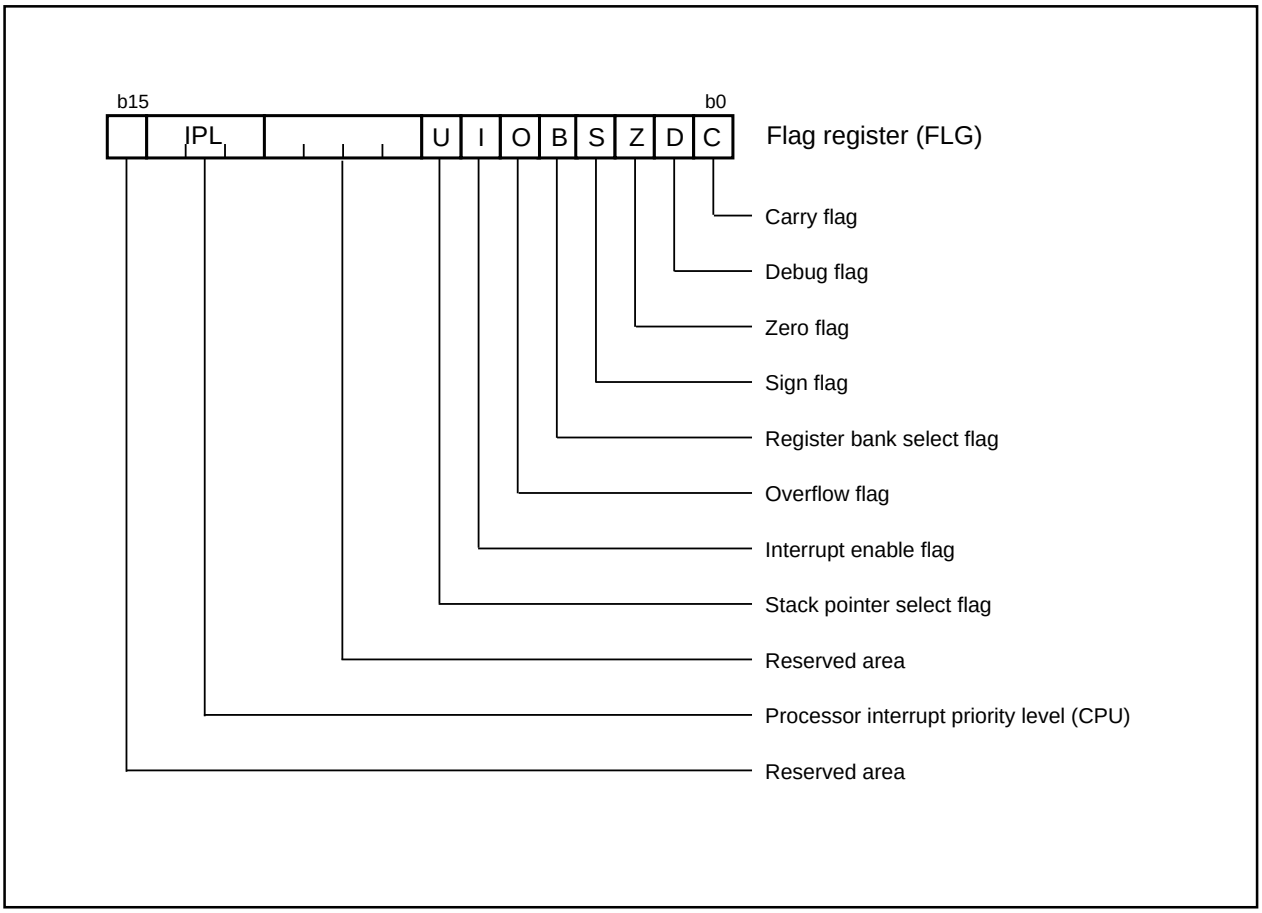


Fig.BA-2 Flag register (FLG)

Reset

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

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

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

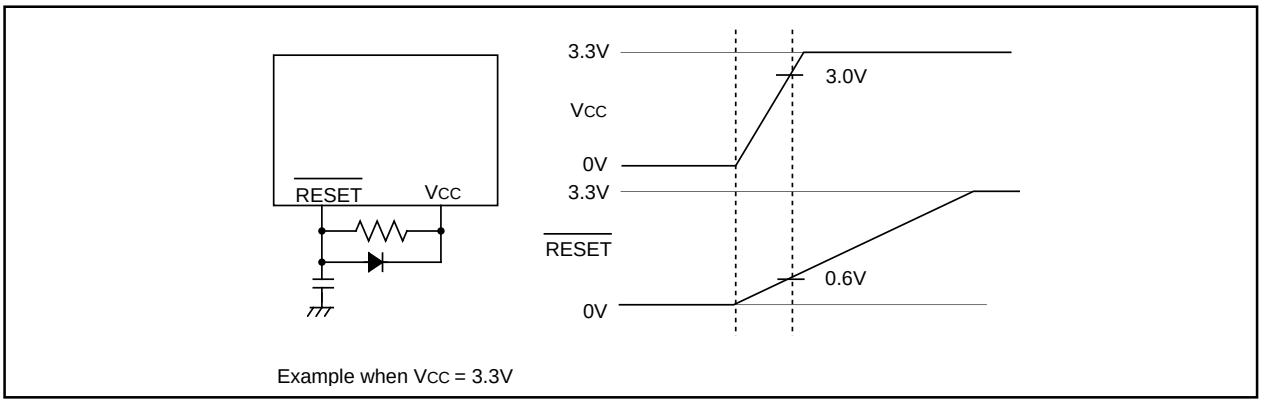


Fig.VB-1 Example reset circuit

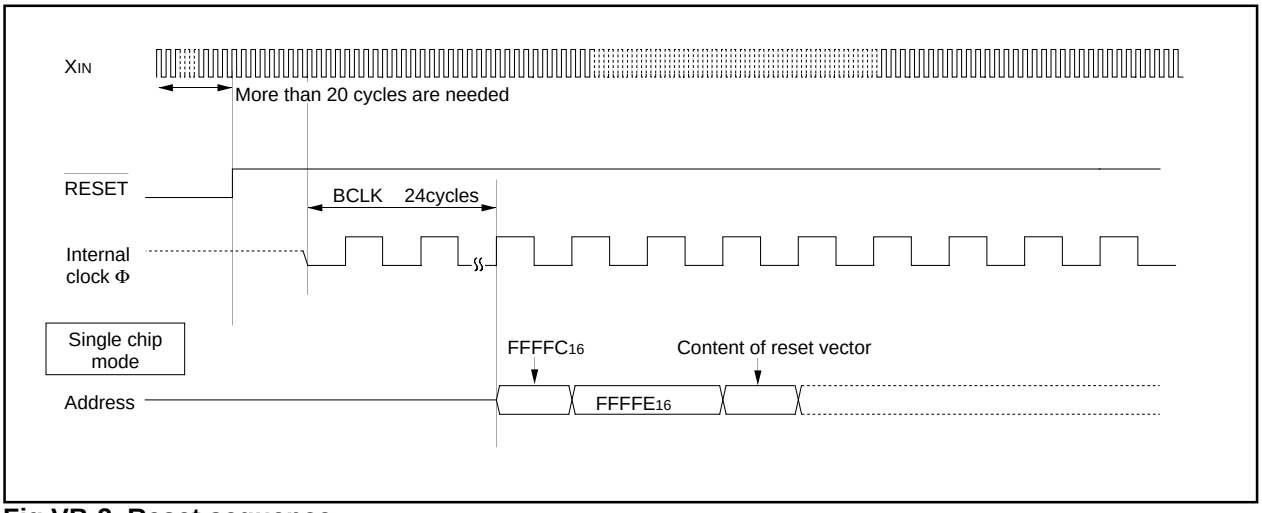
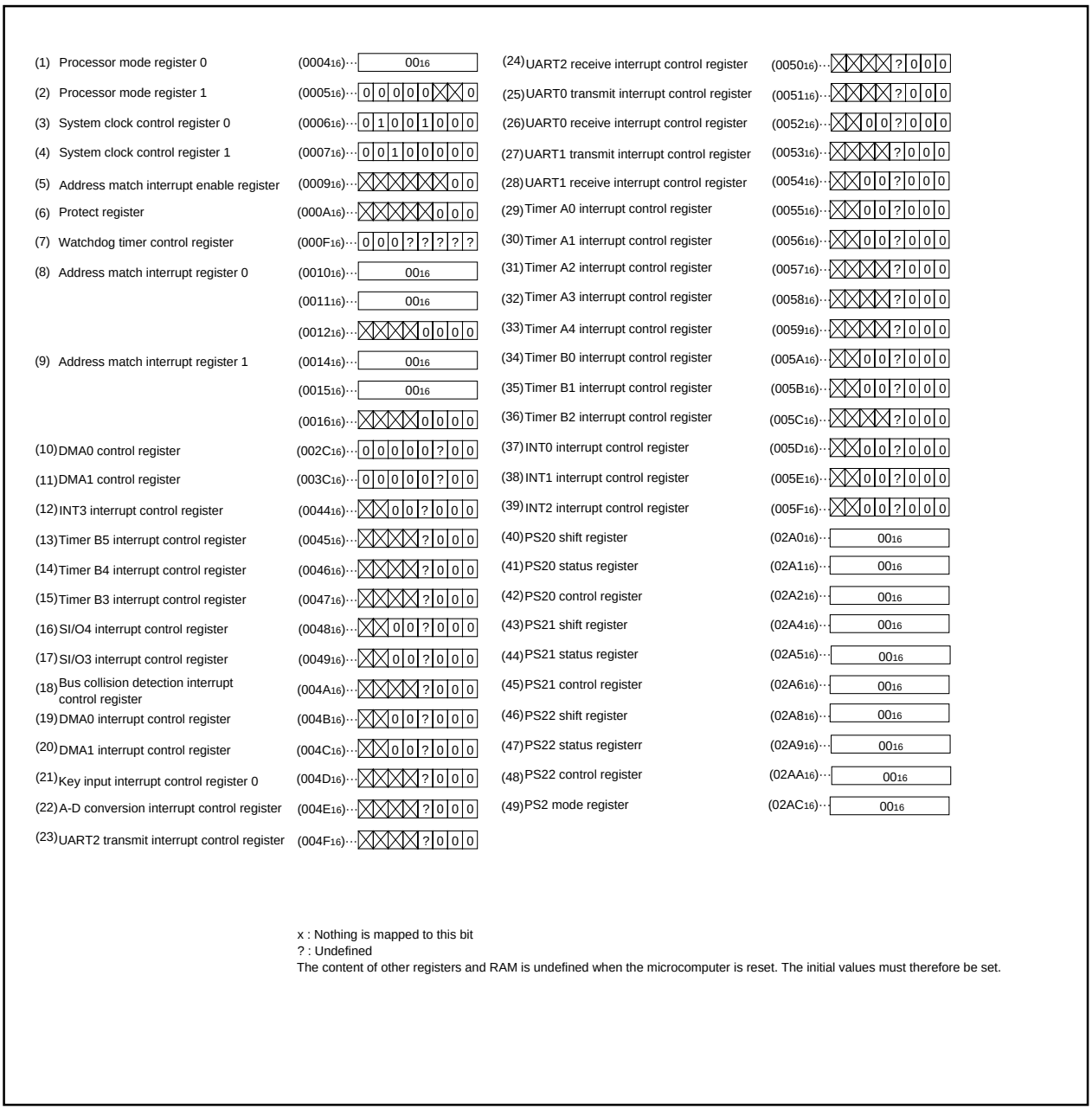


Fig.VB-2 Reset sequence

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

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

Pin name	Status
	CNV_{SS} = V_{SS} (M_0)
P0	I/O port (floating)
P1	I/O port (floating)
P2, P3, $P4_0$ to $P4_3$	I/O port (floating)
$P4_4$	I/O port (floating)
$P4_5$ to $P4_7$	I/O port (floating)
$P5_0$	I/O port (floating)
$P5_1$	I/O port (floating)
$P5_2$	I/O port (floating)
$P5_3$	I/O port (floating)
$P5_4$	I/O port (floating)
$P5_5$	I/O port (floating)
$P5_6$	I/O port (floating)
$P5_7$	I/O port (floating)
P6, P7, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, P9, P10	I/O port (floating)
P11, P12, P13, P14	I/O port (floating)
P15, P16	I/O port (floating)



x : Nothing is mapped to this bit
? : Undefined
The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

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

(50)	Data bus buffer status register 0	(02C1$_{16}$)···	(76)	PWM control register 1	(0309$_{16}$)···
(51)	Data bus buffer status register 1	(02C3$_{16}$)···	(77)	I^{2}C0 address register	(0322$_{16}$)···
(52)	Data bus buffer status register 2	(02C5$_{16}$)···	(78)	I^{2}C0 control register	(0323$_{16}$)···
(53)	Data bus buffer status register 3	(02C7$_{16}$)···	(79)	I^{2}C0 clock control register	(0324$_{16}$)···
(54)	ISA bus control register 0	(02C8$_{16}$)···	(80)	I^{2}C0 start/stop condiction control register	(0325$_{16}$)···
(55)	ISA bus control register 1	(02C9$_{16}$)···	(81)	I^{2}C0 control register 1	(0326$_{16}$)···
(56)	GateA20 control register	(02CA$_{16}$)···	(82)	I^{2}C0 control register 2	(0327$_{16}$)···
(57)	Port P11 direction register	(02E2$_{16}$)···	(83)	I^{2}C0 status register	(0328$_{16}$)···
(58)	Port P12 direction register	(02E3$_{16}$)···	(84)	I^{2}C1 address register	(0332$_{16}$)···
(59)	Port P13 direction register	(02E6$_{16}$)···	(85)	I^{2}C1 control register	(0333$_{16}$)···
(60)	Port P14 direction register	(02E7$_{16}$)···	(86)	I^{2}C1 clock control register	(0334$_{16}$)···
(61)	Port P15 direction register	(02EA$_{16}$)···	(87)	I^{2}C1 start/stop condiction control register	(0335$_{16}$)···
(62)	Port P16 direction register	(02EB$_{16}$)···	(88)	I^{2}C1 control register 1	(0336$_{16}$)···
(63)	Port function selection register 0	(02F8$_{16}$)···	(89)	I^{2}C1 control register 2	(0337$_{16}$)···
(64)	Port function selection register 1	(02F9$_{16}$)···	(90)	I^{2}C1 status register	(0338$_{16}$)···
(65)	Port P4 input register	(02FA$_{16}$)···	(91)	TimerB3,4,5 count start flag	(0340$_{16}$)···
(66)	Port P7 input register	(02FB$_{16}$)···	(92)	Interrupt factor selection register 1	(0356$_{16}$)···
(67)	Pull-up control register 3	(02FC$_{16}$)···	(93)	Interrupt factor selection register 2	(0357$_{16}$)···
(68)	Pull-up control register 4	(02FD$_{16}$)···	(94)	Interrupt factor selection register 3	(0358$_{16}$)···
(69)	Port control register 1	(02FE$_{16}$)···	(95)	Interrupt factor selection register 4	(0359$_{16}$)···
(70)	Port control register 2	(02FF$_{16}$)···	(96)	TimerB3 mode register	(035B$_{16}$)···
(71)	PWM0L register	(0301$_{16}$)···	(97)	TimerB4 mode register	(035C$_{16}$)···
(72)	PWM1L register	(0303$_{16}$)···	(98)	TimerB5 mode register	(035D$_{16}$)···
(73)	PWM2L register	(0305$_{16}$)···	(99)	Interrupt factor selection register 0	(035F$_{16}$)···
(74)	PWM3L register	(0307$_{16}$)···	(100)	SI/O3 control register	(0362$_{16}$)···
(75)	PWM control register 0	(0308$_{16}$)···	(101)	SI/O4 control register	(0366$_{16}$)···

x : Nothing is mapped to this bit
? : Undefined

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

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

(102)	UART2 special mode register	(0377$_{16}$)···	00$_{16}$		
(103)	UART2 transmit/receive mode register	(0378$_{16}$)···	00$_{16}$		
(104)	UART2 transmit/receive control register 0	(037C$_{16}$)···	0 0 0 0 1 0 0 0		
(105)	UART2 transmit/receive control register 1	(037D$_{16}$)···	0 0 0 0 0 0 1 0		
(106)	Count start flag	(0380$_{16}$)···	00$_{16}$		
(107)	Clock prescaler reset flag	(0381$_{16}$)···	0 x x x x x x x		
(108)	One-shot start flag	(0382$_{16}$)···	0 0 x 0 0 0 0 0		
(109)	Trigger select flag	(0383$_{16}$)···	00$_{16}$		
(110)	Up-down flag	(0384$_{16}$)···	00$_{16}$		
(111)	Timer A0 mode register	(0396$_{16}$)···	00$_{16}$		
(112)	Timer A1 mode register	(0397$_{16}$)···	00$_{16}$		
(113)	Timer A2 mode register	(0398$_{16}$)···	00$_{16}$		
(114)	Timer A3 mode register	(0399$_{16}$)···	00$_{16}$		
(115)	Timer A4 mode register	(039A$_{16}$)···	00$_{16}$		
(116)	Timer B0 mode register	(039B$_{16}$)···	0 0 ? x 0 0 0 0		
(117)	Timer B1 mode register	(039C$_{16}$)···	0 0 ? x 0 0 0 0		
(118)	Timer B2 mode register	(039D$_{16}$)···	0 0 ? x 0 0 0 0		
(119)	UART0 transmit/receive mode register	(03A0$_{16}$)···	00$_{16}$		
(120)	UART0 transmit/receive control register 0	(03A4$_{16}$)···	0 0 0 0 1 0 0 0		
(121)	UART0 transmit/receive control register 1	(03A5$_{16}$)···	0 0 0 0 0 0 1 0		
(122)	UART1 transmit/receive mode register	(03A8$_{16}$)···	00$_{16}$		
(123)	UART1 transmit/receive control register 0	(03AC$_{16}$)···	0 0 0 0 1 0 0 0		
(124)	UART1 transmit/receive control register 1	(03AD$_{16}$)···	0 0 0 0 0 0 1 0		
(125)	UART transmit/receive control register 2	(03B0$_{16}$)···	x 0 0 0 0 0 0 0		
(126)	Flash memory recognition register	(03B4$_{16}$)···	0 0 0 0 0 0 0 0		
(127)	Flash memory control register	(03B7$_{16}$)···	x x x x 0 0 0 1		
(128)	DMA0 cause select register	(03B8$_{16}$)···	00$_{16}$		
(129)	DMA1 cause select register	(03BA$_{16}$)···	00$_{16}$		
(130)	A-D control register 0	(03D6$_{16}$)···	0 0 0 0 0 ? ? ?		
(131)	A-D control register 1	(03D7$_{16}$)···	00$_{16}$		
(132)	D-A control register	(03DC$_{16}$)···	00$_{16}$		
(133)	Comparator control register	(03DE$_{16}$)···	00$_{16}$		
(134)	Port P0 direction register	(03E2$_{16}$)···	00$_{16}$		
(135)	Port P1 direction register	(03E3$_{16}$)···	00$_{16}$		
(136)	Port P2 direction register	(03E6$_{16}$)···	00$_{16}$		
(137)	Port P3 direction register	(03E7$_{16}$)···	00$_{16}$		
(138)	Port P4 direction register	(03EA$_{16}$)···	00$_{16}$		
(139)	Port P5 direction register	(03EB$_{16}$)···	00$_{16}$		
(140)	Port P6 direction register	(03EE$_{16}$)···	00$_{16}$		
(141)	Port P7 direction register	(03EF$_{16}$)···	00$_{16}$		
(142)	Port P8 direction register	(03F2$_{16}$)···	0 0 x 0 0 0 0 0		
(143)	Port P9 direction register	(03F3$_{16}$)···	00$_{16}$		
(144)	Port P10 direction register	(03F6$_{16}$)···	00$_{16}$		
(145)	Pull-up control register 0	(03FC$_{16}$)···	00$_{16}$		
(146)	Pull-up control register 1	(03FD$_{16}$)···	00$_{16}$		
(147)	Pull-up control register 2	(03FE$_{16}$)···	00$_{16}$		
(148)	Port control register 0	(03FF$_{16}$)···	00$_{16}$		
(149)	Data registers (R0/R1/R2/R3)		0000$_{16}$		
(150)	Address registers (A0/A1)		0000$_{16}$		
(151)	Frame base register (FB)		0000$_{16}$		
(152)	Interrupt table register (INTB)		00000$_{16}$		
(153)	User stack pointer (USP)		0000$_{16}$		
(154)	Interrupt stack pointer (ISP)		0000$_{16}$		
(155)	Static base register (SB)		0000$_{16}$		
(156)	Flag register (FLG)		0000$_{16}$		

x : Nothing is mapped to this bit
? : Undefined

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

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

(157) Serial interrupt control register 0	($02B0_{16}$)···	00_{16}
(158) Serial interrupt control register 1	($02B1_{16}$)···	00_{16}
(159) IRQ request register 0	($02B2_{16}$)···	00_{16}
(160) IRQ request register 1	($02B3_{16}$)···	00_{16}
(161) IRQ request register 2	($02B4_{16}$)···	00_{16}
(162) IRQ request register 3	($02B5_{16}$)···	00_{16}
(163) IRQ request register 4	($02B6_{16}$)···	00_{16}
(164) LPC1 address register L	($02D0_{16}$)···	00_{16}
(165) LPC1 address register H	($02D1_{16}$)···	00_{16}
(166) LPC2 address register L	($02D2_{16}$)···	00_{16}
(167) LPC2 address register H	($02D3_{16}$)···	00_{16}
(168) LPC3 address register L	($02D4_{16}$)···	00_{16}
(169) LPC3 address register H	($02D5_{16}$)···	00_{16}
(170) LPC control register	($02D6_{16}$)···	00_{16}

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

Fig.VB-6 Device's internal status after a reset is cleared (2)

0000₁₆	
0001₁₆	
0002₁₆	
0003₁₆	
0004₁₆	Processor mode register 0 (PM0)
0005₁₆	Processor mode register 1(PM1)
0006₁₆	System clock control register 0 (CM0)
0007₁₆	System clock control register 1 (CM1)
0008₁₆	
0009₁₆	Address match interrupt enable register (AIER)
000A₁₆	Protect register (PRCR)
000B₁₆	
000C₁₆	
000D₁₆	
000E₁₆	Watchdog timer start register (WDTS)
000F₁₆	Watchdog timer control register (WDC)
0010₁₆ 0011₁₆ 0012₁₆	Address match interrupt register 0 (RMAD0)
0013₁₆	
0014₁₆ 0015₁₆ 0016₁₆	Address match interrupt register 1 (RMAD1)
0017₁₆	
0018₁₆	
0019₁₆	
001A₁₆	
001B₁₆	
001C₁₆	
001D₁₆	
001E₁₆	
001F₁₆	
0020₁₆ 0021₁₆ 0022₁₆	DMA0 source pointer (SAR0)
0023₁₆	
0024₁₆ 0025₁₆ 0026₁₆	DMA0 destination pointer (DAR0)
0027₁₆	
0028₁₆ 0029₁₆	DMA0 transfer counter (TCR0)
002A₁₆	
002B₁₆	
002C₁₆	DMA0 control register (DM0CON)
002D₁₆	
002E₁₆	
002F₁₆	
0030₁₆ 0031₁₆ 0032₁₆	DMA1 source pointer (SAR1)
0033₁₆	
0034₁₆ 0035₁₆ 0036₁₆	DMA1 destination pointer (DAR1)
0037₁₆	
0038₁₆ 0039₁₆	DMA1 transfer counter (TCR1)
003A₁₆	
003B₁₆	
003C₁₆	DMA1 control register (DM1CON)
003D₁₆	
003E₁₆	
003F₁₆	

0040₁₆	
0041₁₆	
0042₁₆	
0043₁₆	
0044₁₆	INT3 interrupt control register (INT3IC)
0045₁₆	Timer B5 interrupt control register (TB5IC) INT9 interrupt control register (INT9IC)
0046₁₆	Timer B4 interrupt control register (TB4IC)
0047₁₆	Timer B3 interrupt control register (TB3IC)
0048₁₆	SI/O4 interrupt control register (S4IC) INT6 interrupt control register (INT6IC)
0049₁₆	SI/O3 interrupt control register (S3IC) INT5 interrupt control register (INT5IC)
004A₁₆	Bus collision detection interrupt control register (BCNIC) INT4 interrupt control register (INT4IC)
004B₁₆	DMA0 interrupt control register (DM0IC) INT8 interrupt control register (INT8IC)
004C₁₆	DMA1 interrupt control register (DM1IC) INT7 interrupt control register (INT7IC)
004D₁₆	Key input interrupt 0 control register (KUP0IC)
004E₁₆	A-D conversion interrupt control register (ADIC)
004F₁₆	UART2 transmit interrupt control register (S2TIC) IBF0 interrupt control register (IBF0IC)
0050₁₆	UART2 receive interrupt control register (S2RIC) IBF1 interrupt control register (IBF1IC)
0051₁₆	UART0 transmit interrupt control register (S0TIC) I^2C0 interrupt control register (IIC0IC)
0052₁₆	UART0 receive interrupt control register (S0RIC) SCL0,SDA0 interrupt control register (SCLDA0IC) INT11 interrupt control register (INT11IC)
0053₁₆	UART1 transmit interrupt control register (S1TIC) I^2C1 interrupt control register (IIC1IC)
0054₁₆	UART1 receive interrupt control register (S1RIC) SCL1,SDA1 interrupt control register (SCLDA1IC) INT10 interrupt control register (INT10IC)
0055₁₆	Timer A0 interrupt control register (TA0IC) INT8 interrupt control register (INT8IC)
0056₁₆	Timer A1 interrupt control register (TA1IC) INT7 interrupt control register (INT7IC)
0057₁₆	Timer A2 interrupt control register (TA2IC)
0058₁₆	Timer A3 interrupt control register (TA3IC) IBF2 interrupt control register (IBF2IC)
0059₁₆	Timer A4 interrupt control register (TA4IC) IBF3 interrupt control register (IBF3IC)
005A₁₆	Timer B0 interrupt control register (TB0IC) SCL0,SDA0 interrupt control register (SCLDA0IC) INT11 interrupt control register (INT11IC)
005B₁₆	Timer B1 interrupt control register (TB1IC) SCL1,SDA1 interrupt control register (SCLDA1IC) INT10 interrupt control register (INT10IC)
005C₁₆	Timer B2 interrupt control register (TB2IC) Key input interrupt 1 control register (KUP1IC)
005D₁₆	INT0 interrupt control register (INT0IC) PS20 interrupt control register (PS20IC)
005E₁₆	INT1 interrupt control register (INT1IC) PS21 interrupt control register (PS21IC)
005F₁₆	INT2 interrupt control register (INT2IC) PS22 interrupt control register (PS22IC)
0060₁₆	
0061₁₆	
≈	≈
027D₁₆	
027E₁₆	
027F₁₆	

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

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

0280₁₆	
0281₁₆	
0282₁₆	
0283₁₆	
0284₁₆	
0285₁₆	
0286₁₆	
0287₁₆	
0288₁₆	
0289₁₆	
028A₁₆	
028B₁₆	
028C₁₆	
028D₁₆	
028E₁₆	
028F₁₆	
0290₁₆	
0291₁₆	
0292₁₆	
0293₁₆	
0294₁₆	
0295₁₆	
0296₁₆	
0297₁₆	
0298₁₆	
0299₁₆	
029A₁₆	
029B₁₆	
029C₁₆	
029D₁₆	
029E₁₆	
029F₁₆	
02A0₁₆	PS20 shift register (PS20SR)
02A1₁₆	PS20 status register (PS20STS)
02A2₁₆	PS20 control register (PS20CON)
02A3₁₆	
02A4₁₆	PS21 shift register (PS21SR)
02A5₁₆	PS21 status register (PS21STS)
02A6₁₆	PS21 control register (PS21CON)
02A7₁₆	
02A8₁₆	PS22 shift register (PS22SR)
02A9₁₆	PS22 status register (PS22STS)
02AA₁₆	PS22 control register (PS22CON)
02AB₁₆	
02AC₁₆	PS2 mode register (PS2MOD)
02AD₁₆	
02AE₁₆	
02AF₁₆	
02B0₁₆	Serial Interrupt control register 0 (SERCON0)
02B1₁₆	Serial Interrupt control register 1 (SERCON1)
02B2₁₆	IRQ request register 0 (IRQ0)
02B3₁₆	IRQ request register 1 (IRQ1)
02B4₁₆	IRQ request register 2 (IRQ2)
02B5₁₆	IRQ request register 3 (IRQ3)
02B6₁₆	IRQ request register 4 (IRQ4)
02B7₁₆	
02B8₁₆	
02B9₁₆	
02BA₁₆	
02BB₁₆	
02BC₁₆	
02BD₁₆	
02BE₁₆	
02BF₁₆	

02C0₁₆	Data bus buffer register0 (DBB0)
02C1₁₆	Data bus buffer status register0 (DBBSTS0)
02C2₁₆	Data bus buffer register1 (DBB1)
02C3₁₆	Data bus buffer status register1 (DBBSTS1)
02C4₁₆	Data bus buffer register2 (DBB2)
02C5₁₆	Data bus buffer status register2 (DBBSTS2)
02C6₁₆	Data bus buffer register3 (DBB3)
02C7₁₆	Data bus buffer status register3 (DBBSTS3)
02C8₁₆	ISA control register0 (DBBCON0)
02C9₁₆	ISA control register1 (DBBCON1)
02CA₁₆	Gate A20 control register (GA20CON)
02CB₁₆	
02CC₁₆	
02CD₁₆	
02CE₁₆	
02CF₁₆	
02D0₁₆	LPC1 address registerL (LPC1ADL)
02D1₁₆	LPC1 address registerH (LPC1ADH)
02D2₁₆	LPC2 address registerL (LPC2ADL)
02D3₁₆	LPC2 address registerH (LPC2ADH)
02D4₁₆	LPC3 address registerL (LPC3ADL)
02D5₁₆	LPC3 address registerH (LPC3ADH)
02D6₁₆	LPC control register (LPCCON)
02D7₁₆	
02D8₁₆	
02D9₁₆	
02DA₁₆	
02DB₁₆	
02DC₁₆	
02DD₁₆	
02DE₁₆	
02DF₁₆	
02E0₁₆	Port P11 (P11)
02E1₁₆	Port P12 (P12)
02E2₁₆	Port P11 direction register (PD11)
02E3₁₆	Port P12 direction register (PD12)
02E4₁₆	Port P13 (P13)
02E5₁₆	Port P14 (P14)
02E6₁₆	Port P13 direction register (PD13)
02E7₁₆	Port P14direction register (PD14)
02E8₁₆	Port P15 (P15)
02E9₁₆	Port P16 (P16)
02EA₁₆	Port P15 direction register (PD15)
02EB₁₆	Port P16 direction register (PD16)
02EC₁₆	
02ED₁₆	
02EE₁₆	
02EF₁₆	
02F0₁₆	
02F1₁₆	
02F2₁₆	
02F3₁₆	
02F4₁₆	
02F5₁₆	
02F6₁₆	P14 event register (P14EV)
02F7₁₆	Port control register3 (PCR3)
02F8₁₆	Port function selection register0 (PSL0)
02F9₁₆	Port function selection register1 (PSL1)
02FA₁₆	Port P4 input register (P4PIN)
02FB₁₆	Port P7 input register (P7PIN)
02FC₁₆	Pull-up control register3 (PUR3)
02FD₁₆	Pull-up control register4 (PUR4)
02FE₁₆	Port control register1 (PCR1)
02FF₁₆	Port control register2 (PCR2)

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

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

0300₁₆	PWM0H register (PWM0H)
0301₁₆	PWM0L register (PWM0L)
0302₁₆	PWM1H register (PWM1H)
0303₁₆	PWM1L register (PWM1L)
0304₁₆	PWM2H register (PWM2H)
0305₁₆	PWM2L register (PWM2L)
0306₁₆	PWM3H register (PWM3H)
0307₁₆	PWM3L register (PWM3L)
0308₁₆	PWM control register0 (PWMCON0)
0309₁₆	PWM control register1 (PWMCON1)
030A₁₆	
030B₁₆	
030C₁₆	
030D₁₆	
030E₁₆	
030F₁₆	
0310₁₆	
0311₁₆	
0312₁₆	
0313₁₆	
0314₁₆	
0315₁₆	
0316₁₆	
0317₁₆	
0318₁₆	
0319₁₆	
031A₁₆	
031B₁₆	
031C₁₆	
031D₁₆	
031E₁₆	
031F₁₆	
0320₁₆	I^2C0 data shift register (S00)
0321₁₆	
0322₁₆	I^2C0 address register (S0D0)
0323₁₆	I^2C0 control register0 (S1D0)
0324₁₆	I^2C0 clock control register (S20)
0325₁₆	I^2C0 start/stop condition control register (S2D0)
0326₁₆	I^2C0 control register1 (S3D0)
0327₁₆	I^2C0 control register2 (S4D0)
0328₁₆	I^2C0 status register (S10)
0329₁₆	
032A₁₆	
032B₁₆	
032C₁₆	
032D₁₆	
032E₁₆	
032F₁₆	
0330₁₆	I^2C1 data shift register (S01)
0331₁₆	
0332₁₆	I^2C1 address register (S0D1)
0333₁₆	I^2C1 control register0 (S1D1)
0334₁₆	I^2C1 clock control register (S21)
0335₁₆	I^2C1 start/stop condition control register (S2D1)
0336₁₆	I^2C1 control register1 (S3D1)
0337₁₆	I^2C1 control register2 (S4D1)
0338₁₆	I^2C1 status register (S11)
0339₁₆	
033A₁₆	
033B₁₆	
033C₁₆	
033D₁₆	
033E₁₆	
033F₁₆	

0340₁₆	TimerB3,4,5 count start flag (TBSR)
0341₁₆	
0342₁₆	
0343₁₆	
0344₁₆	
0345₁₆	
0346₁₆	
0347₁₆	
0348₁₆	
0349₁₆	
034A₁₆	
034B₁₆	
034C₁₆	
034D₁₆	
034E₁₆	
034F₁₆	
0350₁₆ / 0351₁₆	TimerB3 register (TB3)
0352₁₆ / 0353₁₆	TimerB4 register (TB4)
0354₁₆ / 0355₁₆	TimerB5 register (TB5)
0356₁₆	Interrupt event select register1 (IFSR1)
0357₁₆	Interrupt event select register2 (IFSR2)
0358₁₆	Interrupt event select register3 (IFSR3)
0359₁₆	Interrupt event select register4 (IFSR4)
035A₁₆	
035B₁₆	TimerB3 mode register (TB3MR)
035C₁₆	TimerB4 mode register (TB4MR)
035D₁₆	TimerB5 mode register (TB5MR)
035E₁₆	
035F₁₆	Interrupt event select register0 (IFSR0)
0360₁₆	SI/O3 transmit/receive register (S3TRR)
0361₁₆	
0362₁₆	SI/O3 control register (S3C)
0363₁₆	SI/O3 communication speed register (S3BRG)
0364₁₆	SI/O4 transmit/receive register (S4TRR)
0365₁₆	
0366₁₆	SI/O4 control register (S4C)
0367₁₆	SI/O4 communication speed register (S4BRG)
0368₁₆	
0369₁₆	
036A₁₆	
036B₁₆	
036C₁₆	
036D₁₆	
036E₁₆	
036F₁₆	
0370₁₆	
0371₁₆	
0372₁₆	
0373₁₆	
0374₁₆	
0375₁₆	
0376₁₆	
0377₁₆	UART2 special mode register (U2SMR)
0378₁₆	UART2 transmit/receive mode register (U2MR)
0379₁₆	UART2 communication speed register (U2BRG)
037A₁₆ / 037B₁₆	UART2 tranmit buffer register (U2TB)
037C₁₆	UART2 transmit/receive control register0 (U2C0)
037D₁₆	UART2 transmit/receive control register1 (U2C1)
037E₁₆ / 037F₁₆	UART2 receive buffer register (U2RB)

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

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

0380₁₆	Count start flag (TABSR)
0381₁₆	Clock prescaler reset flag (CPSRF)
0382₁₆	One-shot start flag (ONSF)
0383₁₆	Trigger select register (TRGSR)
0384₁₆	Up-down flag (UDF)
0385₁₆	
0386₁₆ 0387₁₆	TimerA0 (TA0)
0388₁₆ 0389₁₆	TimerA1 (TA1)
038A₁₆ 038B₁₆	TimerA2 (TA2)
038C₁₆ 038D₁₆	TimerA3 (TA3)
038E₁₆ 038F₁₆	TimerA4 (TA4)
0390₁₆ 0391₁₆	TimerB0 (TB0)
0392₁₆ 0393₁₆	TimerB1 (TB1)
0394₁₆ 0395₁₆	TimerB2 (TB2)
0396₁₆	TimerA0 mode register (TA0MR)
0397₁₆	TimerA1 mode register (TA1MR)
0398₁₆	TimerA2 mode register (TA2MR)
0399₁₆	TimerA3 mode register (TA3MR)
039A₁₆	TimerA4 mode register (TA4MR)
039B₁₆	TimerB0 mode register (TB0MR)
039C₁₆	TimerB1 mode register (TB1MR)
039D₁₆	TimerB2 mode register (TB2MR)
039E₁₆	
039F₁₆	
03A0₁₆	UART0 transmit/receive mode register (U0MR)
03A1₁₆	UART0 communication speed register (U0BRG)
03A2₁₆ 03A3₁₆	UART0 tranmit buffer register (U0TB)
03A4₁₆	UART0 transmit/receive control register0 (U0C0)
03A5₁₆	UART0 transmit/receive control register1 (U0C1)
03A6₁₆ 03A7₁₆	UART0 receive buffer register (U0RB)
03A8₁₆	UART1 transmit/receive mode register (U1MR)
03A9₁₆	UART1 communication speed register (U1BRG)
03AA₁₆ 03AB₁₆	UART1 tranmit buffer register (U1TB)
03AC₁₆	UART1 transmit/receive control register0 (U1C0)
03AD₁₆	UART1 transmit/receive control register1 (U1C1)
03AE₁₆ 03AF₁₆	UART1 receive buffer register (U1RB)
03B0₁₆	UART transmit/receive control register2 (UCON)
03B1₁₆	
03B2₁₆	
03B3₁₆	
03B4₁₆	Flash memory recognition register (FMRR)
03B5₁₆	
03B6₁₆	
03B7₁₆	Flash memory control register (FMCR)
03B8₁₆	DMA0 request cacse select register (DM0SL)
03B9₁₆	
03BA₁₆	DMA1 request cacse select register (DM1SL)
03BB₁₆	
03BC₁₆	
03BD₁₆	
03BE₁₆	
03BF₁₆	

03C0₁₆ 03C1₁₆	A-D register0 (AD0)
03C2₁₆ 03C3₁₆	A-D register1 (AD1)
03C4₁₆ 03C5₁₆	A-D register2 (AD2)
03C6₁₆ 03C7₁₆	A-D register3 (AD3)
03C8₁₆ 03C9₁₆	A-D register4 (AD4)
03CA₁₆ 03CB₁₆	A-D register5 (AD5)
03CC₁₆ 03CD₁₆	A-D register6 (AD6)
03CE₁₆ 03CF₁₆	A-D register7 (AD7)
03D0₁₆	
03D1₁₆	
03D2₁₆	
03D3₁₆	
03D4₁₆	
03D5₁₆	
03D6₁₆	A-D control register0 (ADCON0)
03D7₁₆	A-D control register1 (ADCON1)
03D8₁₆	D-A register0 (DA0)
03D9₁₆	
03DA₁₆	D-A register1 (DA1)
03DB₁₆	
03DC₁₆	D-A control register (DACON)
03DD₁₆	
03DE₁₆	Comparator control register (CMPCON)
03DF₁₆	Comparator data register (CMPD)
03E0₁₆	Port P0 (P0)
03E1₁₆	Port P1 (P1)
03E2₁₆	Port P0 direction register (P0D)
03E3₁₆	Port P1 direction register (P1D)
03E4₁₆	Port P2 (P2)
03E5₁₆	Port P3 (P3)
03E6₁₆	Port P2 direction register (P2D)
03E7₁₆	Port P3 direction register (P3D)
03E8₁₆	Port P4 (P4)
03E9₁₆	Port P5 (P5)
03EA₁₆	Port P4 direction register (P4D)
03EB₁₆	Port P5 direction register (P5D)
03EC₁₆	Port P6 (P6)
03ED₁₆	Port P7 (P7)
03EE₁₆	Port P6 direction register (P6D)
03EF₁₆	Port P7 direction register (P7D)
03F0₁₆	Port P8 (P8)
03F1₁₆	Port P9 (P9)
03F2₁₆	Port P8 direction register (P8D)
03F3₁₆	Port P9 direction register (P9D)
03F4₁₆	Port P10 (P10)
03F5₁₆	
03F6₁₆	Port P10 direction register (P10D)
03F7₁₆	
03F8₁₆	
03F9₁₆	
03FA₁₆	
03FB₁₆	
03FC₁₆	Pull-up control register0 (PUR0)
03FD₁₆	Pull-up control register1 (PUR1)
03FE₁₆	Pull-up control register2 (PUR2)
03FF₁₆	Port control register0 (PCR0)

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

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

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004_{16}) applies a (software) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are retained.

Processor Mode

(1) Types of Processor Mode

The single-chip mode is supported in processor mode.

• Single-chip mode

In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. Ports P0 to P16 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions. Fig. BG-1 shows the structure of processor mode register 0 and processor mode register 1.

Processor mode register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0		0		

Symbol	Address	When reset
PM0	0004_{16}	00_{16} (Note 2)

Bit symbol	Bit name	Function	R	W
PM00	Processor mode bit	b1 b0 0 0: Single-chip mode 0 1: Inhibited 1 0: Inhibited 1 1: Inhibited	○	○
PM01			○	○
Reserved bit		Must always be set to “0”	—	○
PM03	Software reset bit	The device is reset when this bit is set to “1”. The value of this bit is “0” when read.	○	○
Reserved bit		Must always be set to “0”	—	○

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Processor mode register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0	0	X	X	0

Symbol	Address	When reset
PM1	0005_{16}	$00000XX0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
Reserved bit		Must always be set to “0”	—	○
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Fig.BG-1 Processor mode register 0 and 1

Bus control

(1) Software wait

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 0005_{16}) (Note) .

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

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

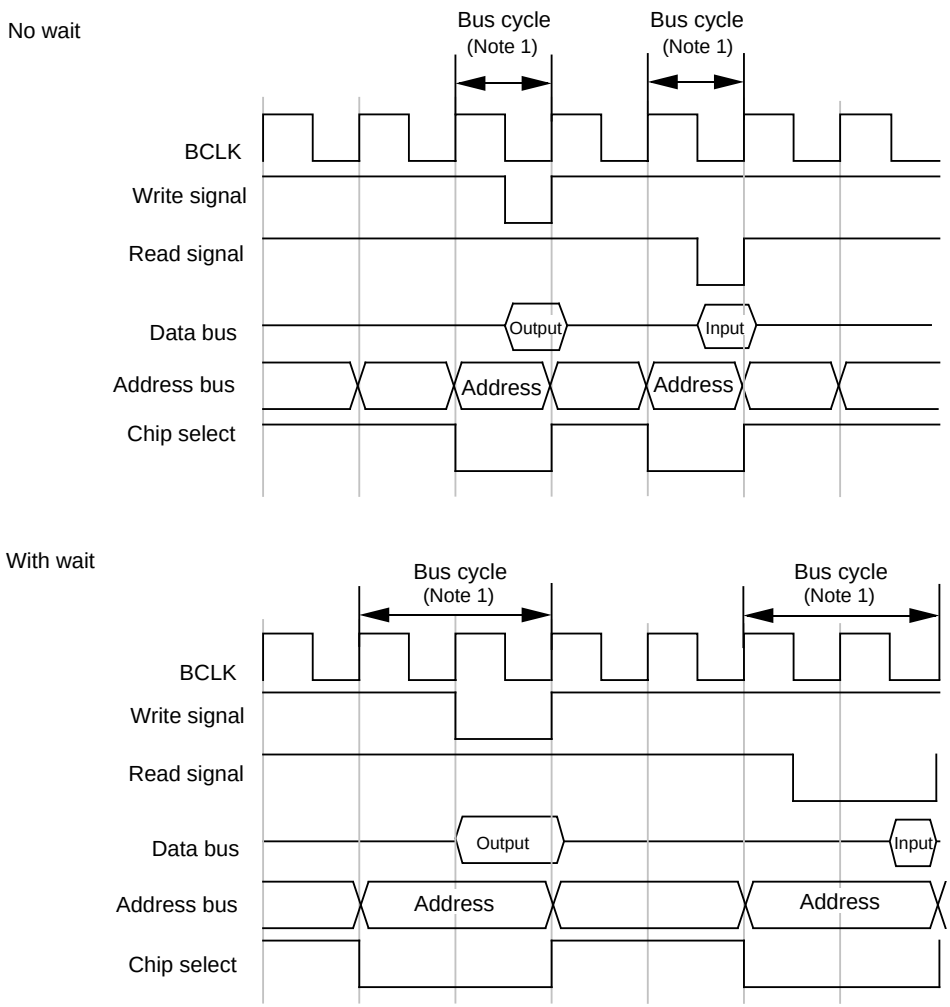
The SFR area is always accessed in two BCLK cycles regardless of the setting of these control bits.

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

Note: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address $000A_{16}$) to “1”.

Table.EF-1 Software waits and bus cycles

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



Note 1: This timing sample shows the lenth of bus cycle. It is possible that the read cyles, write cycle comes after this cycle in succession.

Fig.EF-1 Typical bus timings using software wait

Clock Generating Circuit

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

Table.WA-1 Main clock and sub clock generating circuits

	Main clock generating circuit	Sub clock generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer A/B's count clock source
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

Example of oscillator circuit

Fig.WA-1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure WA-2 shows some examples of sub clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Fig.WA-1 and WA-2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.

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.
When the oscillation drive capacity is set to low, check that oscillation is stable.
Apply the feedback register between XIN and XOUT if required by oscillator maker.

Fig.WA-1 Examples of main clock

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.
When the oscillation drive capacity is set to low, check that oscillation is stable.
Apply the feedback register between XCIN and XCOUT if required by oscillator maker.

Fig.WA-2 Examples of sub clock

Clock Control

Fig.WA-3 shows the block diagram of the clock generating circuit.

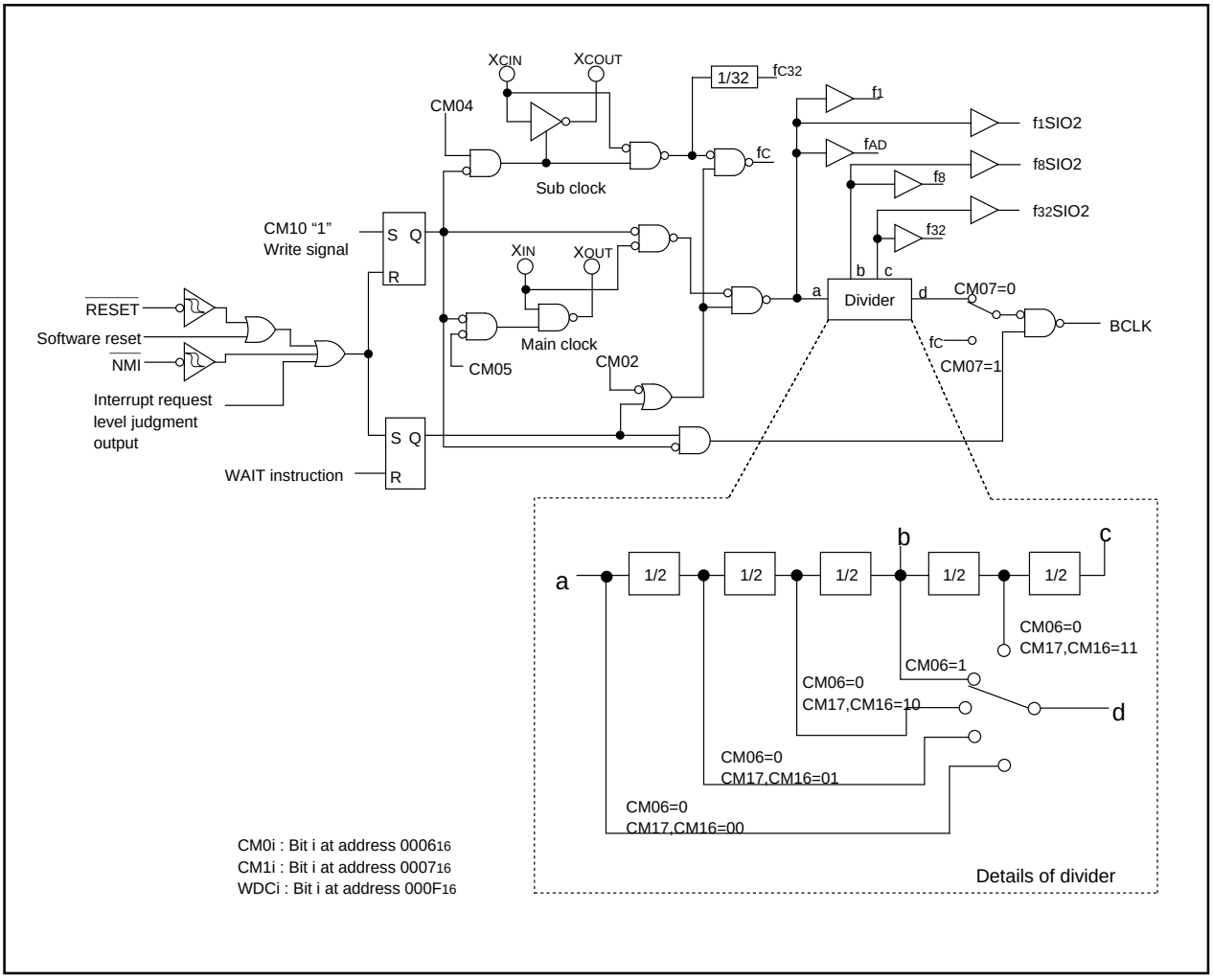


Fig.WA-3 Clock generating circuit

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

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). After switching the CPU operation clock to sub clock stopping the clock reduces the power dissipation.

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit defaults to “1” when shifting from high speed mode or mid-speed mode to stop mode and after a reset.

(2) Sub clock

The sub clock is generated by the sub clock oscillation circuit. No sub clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006_{16}), the sub clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006_{16}). However, be sure that the sub clock oscillation has fully stabilized before switching.

After the oscillation of the sub clock oscillation circuit has stabilized, the drive capacity of the sub clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub clock oscillation circuit reduces the power dissipation. This bit changes to “1” when shifting to stop mode and at a reset.

(3) BCLK

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

When shifting from high speed mode or mid-speed mode to stop mode, the main clock division select bit (bit 6 at 0006_{16}) is set to “1”. The bit maintains in low speed mode and low power save mode.

(4) Peripheral function clock

f_1, f_8, f_{32}, f_{1SIO2}, f_{8SIO2}, f_{32SIO2}, f_{AD}

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006_{16}) to “1” and then executing a WAIT instruction.

(5) f_{C32}

This clock is derived by dividing the sub clock by 32. It is used for the timer A and timer B counts.

(6) f_C

This clock has the same frequency as the sub clock. It is used for the BCLK and for the watchdog timer.

Fig.WA-4 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

Symbol CM0 | Address 000616 | When reset 4816

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port P57 0 1 : fc output 1 0 : f8 output 1 1 : f32 output	O	O
CM01			O	O
CM02	WAIT peripheral function clock stop bit	0 :Do not stop peripheral clock in wait mode 1 :Stop peripheral clock in wait mode (Note8)	O	O
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation	O	O
CM05	Main clock (XIN-XOUT) stop bit (Note 3) (Note 4) (Note 5)	0 : On 1 : Off	O	O
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	O	O
CM07	System clock select bit (Note 6)	0 : XIN, XOUT 1 : XCIN, XCOUT	O	O

Note 1: Set bit 0 of the protect register (address 000A16) to "1" before writing to this register.
Note 2: Changes to "1" when shiffing to stop mode.
Note 3: When entering power saving mode, main clock stops using this bit. When returning from stop mode and operating with XIN, set this bit to "0". When main clock oscillation is operating by itself, set system clock select bit (CM07) to "1" before setting this bit to "1".
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to "1", XOUT turns "H". The built-in feedback resistor remains ON, so XIN turns pulled up to XOUT ("H") via the feedback resistor.
Note 6: In the case of setting the bit from "0" to "1", set port XC select bit (CM04) to "1" and wait for the subclock being stable before wrting the bit. Don't write in the same time.
In the case of setting the bit from "1" to "0", set main clock stop bit (CM05) to "0" and wait for the main clock being stable before write the bit.
Note 7: The bit is set to "1" when shifting from high speed mode or mid speed mode to stop mode and after reset. The bit maintains in low speed mode and power save mode.
Note 8: fc32 is not included. Do not set to "1" when using low-speed or low power dissipation mode.

System clock control register 1 (Note 1)

Symbol CM1 | Address 000716 | When reset 2016

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	O	O
CM17				

Note 1: Set bit 0 of the protect register (address 000A16) to "1" before writing to this register.
Note 2: Changes to "1" when shiffing to stop mode.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 000616) is "0". If "1", division mode is fixed at 8.
Note 4: If this bit is set to "1", XOUT turns "H", and the built-in feedback resistor turns null.

Fig.WA-4 System clock control registers 0 and 1

Clock Output

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006_{16}) enable f_8, f_{32}, or fc to be output from the P57/CLKOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) is set to “1”, the output of f_8 and f_{32} stops when a WAIT instruction is executed.

Stop Mode

Writing “1” to the all-clock stop control bit (bit 0 at address 0007_{16}) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that VCC remains above 2V.

The oscillation , BCLK, f_1 to f_{32}, f_{1SIO2} to f_{32SIO2}, fC, f_{C32}, and f_{AD} stop in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UARTi(i = 0 to 2) ,SIO3,4 functions provided an external clock is selected. Table.WA-2 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled.After the restoration by interrupt, the corresponding interrupt routine will be processed.When shifting from high speed mode or mid-speed mode to stop mode, the main clock division select bit 0 (bit 6 at 0006_{16}) is set to “1”.

Table.WA-2 Port status during stop mode

Pin		Single-chip mode
Port		Retains status before stop mode
CLKOUT	When fc selected	“H”
	When f_8, f_{32} selected	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. However, because the peripheral function clock (fc32) that is generated by sub clock does not stop, there is no reducing of power dissipation. Do not set the bit to "1" then enter wait mode in low speed mode and low power dissipation mode.Table.WA-3 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts from interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

Table.WA-3 Port status during wait mode

Pin		Single-chip mode
CLK_{OUT}	When f_C selected	Does not stop
	When f_8, f_{32} selected	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”, the status immediately prior to entering wait mode is maintained.
Port		maintained the status immediately prior to enterig wait mode

Status Transition Of BCLK

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

After a reset, operation defaults to division by 8 mode. When shifting from high speed mode or mid-speed mode to stop mode, and after a reset main clock division select bit 0 (bit 6 at address 0006_{16}) is set to "1". It is matained in low speed mode and low power dissipation mode.

(1) Division by 2 mode

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

(2) Division by 4 mode

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

(3) Division by 8 mode

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

(4) Division by 16 mode

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

(5) No-division mode

The main clock is used as the BCLK.

(6) Low-speed mode

fC is used as the BCLK. Note that oscillation of both the main and sub clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

(7) Low power dissipation mode

fC is the BCLK and the main clock is stopped.

Precaution

In the case of switching the BCLK count source from XIN to XCIN, or from XCIN to XIN, it is necessary that the destination clock count source be stable. The transition should be waited by software after the oscillation being stable.

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

CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	1	0	0	0	Invalid	Division by 2 mode
1	0	0	0	0	Invalid	Division by 4 mode
Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
1	1	0	0	0	Invalid	Division by 16 mode
0	0	0	0	0	Invalid	No-division mode
Invalid	Invalid	1	Invalid	0	1	Low-speed mode
Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode

Power control

The following is a description of the power control modes:

Modes

Power control is available in three modes.

(1) Normal operation mode

• High-speed mode
Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the internal clock selected. Each peripheral function operates according to its assigned clock.

• Medium-speed mode
Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates according to the internal clock selected. Each peripheral function operates according to its assigned clock.

• Low-speed mode
fC becomes the BCLK. The CPU operates according to the fc clock. The fc clock is supplied by the sub clock. Each peripheral function operates according to its assigned clock.

• Low power consumption mode
The main clock operating in low-speed mode is stopped. The CPU operates according to the fC clock. The fc clock is supplied by the sub clock. The only peripheral functions that operate are those with the sub-clock selected as the count source.

(2) Wait mode
The CPU operation is stopped. The oscillators do not stop.

(3) Stop mode
All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Fig.WA-5 is the state transition diagram of (1) to (3).

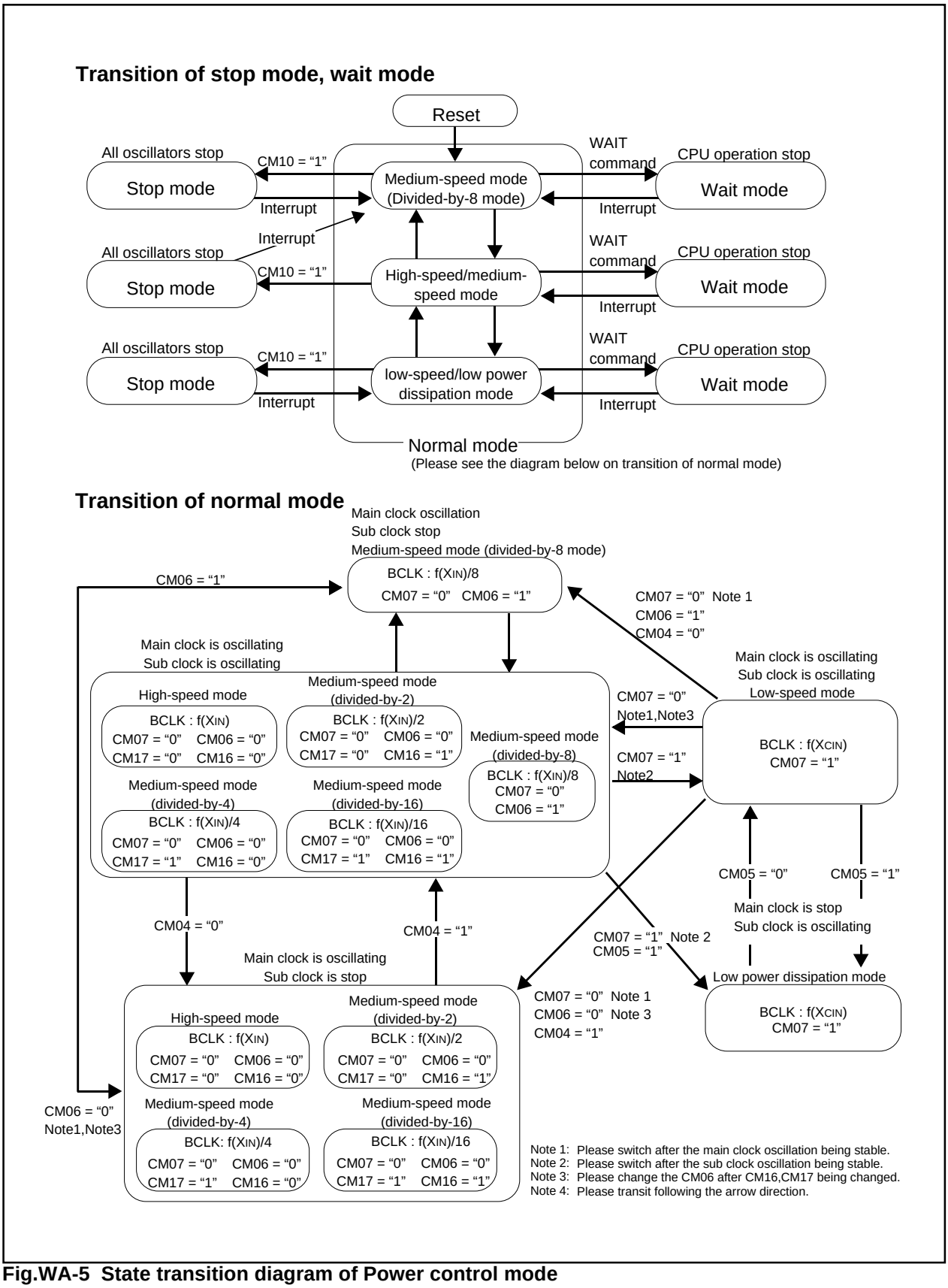


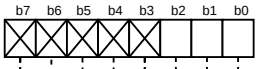
Fig.WA-5 State transition diagram of Power control mode

Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Fig.WA-6 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), system clock control register 1 (address 0007_{16}), port P9 direction register (address $03F3_{16}$) , SI/O3 control register (address 0362_{16}) and SI/O4 control register (address 0366_{16}) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P9.

If, after “1” (write-enabled) has been written to the port P9 direction register and SI/Oi control register (i=3,4) write-enable bit (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to “0” (write-inhibited). However, the system clock control registers 0 and 1 write-enable bit (bit 0 at $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



Symbol	Address	When reset
PRCR	$000A_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P9 direction register (address $03F3_{16}$) and SI/Oi control register (i=3,4) (address 0362_{16} and 0366_{16}) (Note)	0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

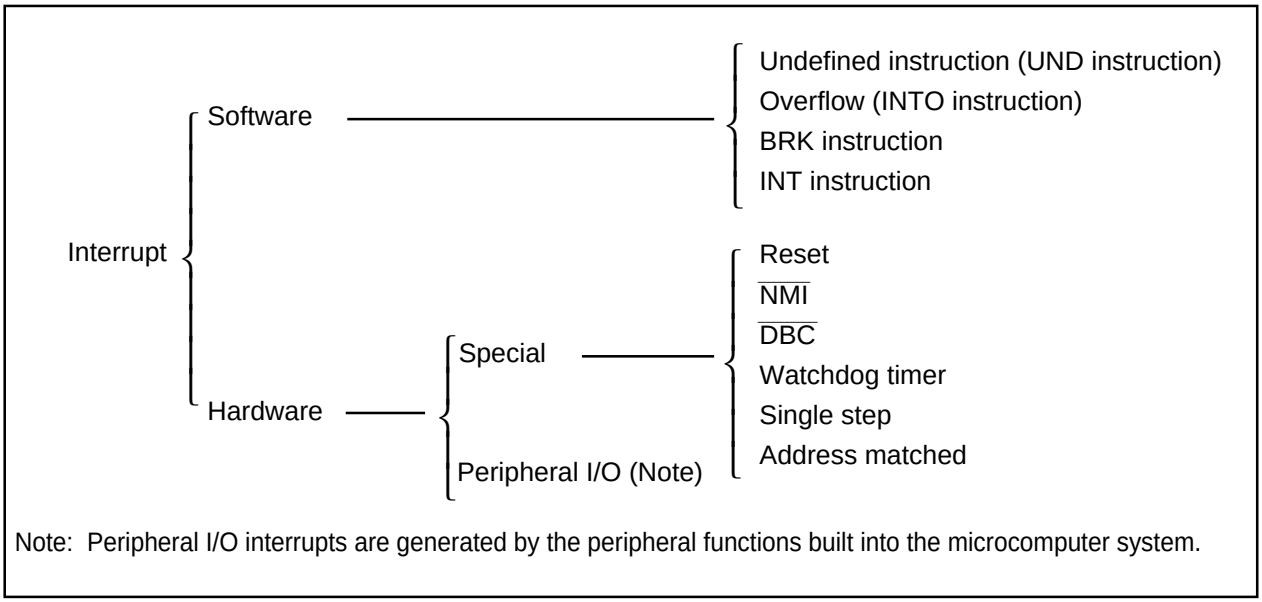
Note: Writing a value to an address after “1” is written to this bit returns the bit to “0”. Other bits do not automatically return to “0” and they must therefore be reset by the program.

Fig.WA-6 Protect register

Overview of Interrupt

Type of Interrupts

Fig.DD-1 lists the types of interrupts.



Note: Peripheral I/O interrupts are generated by the peripheral functions built into the microcomputer system.

Fig.DD-1 Classification of interrupts

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

Software Interrupts

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

• Undefined instruction interrupt
An undefined instruction interrupt occurs when executing the UND instruction.
• Overflow interrupt
An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to “1”. The following are instructions whose O flag changes by arithmetic:
ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB
• BRK interrupt
A BRK interrupt occurs when executing the BRK instruction.
• INT interrupt
An INT interrupt occurs when assigning one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does. The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.
So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to “0” and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

Hardware Interrupts

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

(1) Special interrupts

Special interrupts are non-maskable interrupts.

• Reset

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

• $\overline{\text{NMI}}$ interrupt

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

• $\overline{\text{DBC}}$ interrupt

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

• Watchdog timer interrupt

Generated by the watchdog timer.

• Single-step interrupt

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

• Address match interrupt

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

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

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors are also dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

1)Bus collision detection interrupt

This is an interrupt that the serial I/O bus collision detection generates.

2)DMA0 interrupt, DMA1 interrupt

These are interrupts that DMA generates.

3)Key-input interrupt 0 / Key-input interrupt 1

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

4)A-D conversion interrupt

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

5)UART0, UART1, UART2/NACK, SI/O3 and SI/O4 transmission interrupt

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

6)UART0, UART1, UART2/ACK, SI/O3 and SI/O4 reception interrupt

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

7)Timer A0 interrupt through timer A4 interrupt

These are interrupts that timer A generates

8)Timer B0 interrupt through timer B5 interrupt

These are interrupts that timer B generates.

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

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

10)IBF0 to IBF3 interrupt

These are interrupts that host bus interface generates.

11)I^2C0,I^2C1,SCL0,SDA0,SCL1,SDA1 interrupt

These are interrupts that I^2C bus interface generates.

12)PS20 to PS22 interrupt

These are interrupt that PS2 interface generates.

Interrupts and Interrupt Vector Tables

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

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

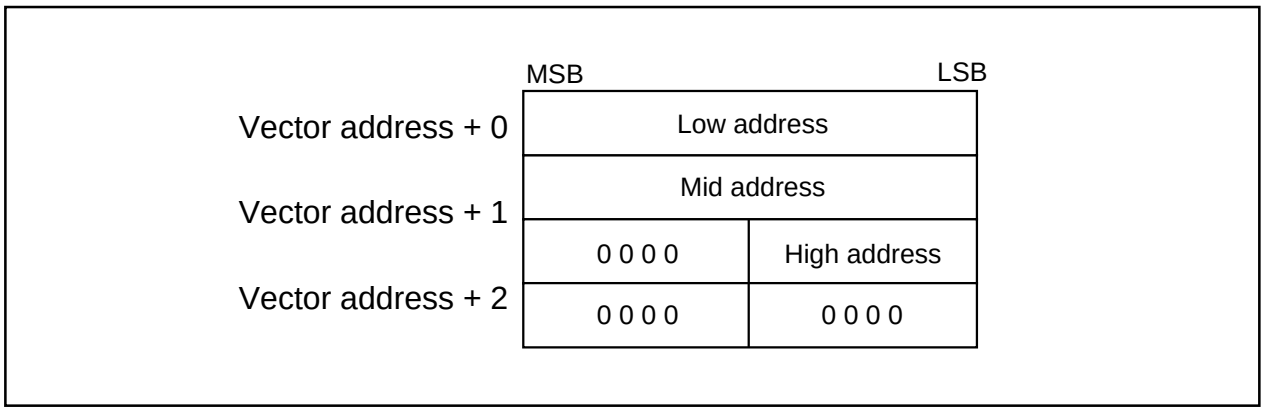


Fig.DD-2 Format for specifying interrupt vector addresses

• Fixed vector tables

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

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

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction
BRK instruction	FFFE416 to FFFE716	If the vector contains FF16, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE816 to FFFEB16	There is an address-matching interrupt enable bit
Single step (Note)	FFFEC16 to FFFEF16	Do not use
Watchdog timer	FFFF016 to FFFF316	
$\overline{\text{DBC}}$ (Note)	FFFF416 to FFFF716	Do not use
$\overline{\text{NMI}}$	FFFF816 to FFFFB16	External interrupt by input to $\overline{\text{NMI}}$ pin
Reset	FFFFC16 to FFFFF16	

Note: Interrupts used for debugging purposes only.

• Variable vector tables

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

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

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked by I flag
Software interrupt number 4	+16 to +19 (Note 1)	$\overline{\text{INT3}}$	
Software interrupt number 5	+20 to +23 (Note 1)	Timer B5/$\overline{\text{INT9}}$ (Note 2)	
Software interrupt number 6	+24 to +27 (Note 1)	Timer B4	
Software interrupt number 7	+28 to +31 (Note 1)	Timer B3	
Software interrupt number 8	+32 to +35 (Note 1)	SI/O4/$\overline{\text{INT6}}$ (Note 2)	
Software interrupt number 9	+36 to +39 (Note 1)	SI/O3/$\overline{\text{INT5}}$ (Note 2)	
Software interrupt number 10	+40 to +43 (Note 1)	Bus collision detection/$\overline{\text{INT4}}$ (Note 2)	
Software interrupt number 11	+44 to +47 (Note 1)	DMA0/$\overline{\text{INT8}}$ (Note 3)	
Software interrupt number 12	+48 to +51 (Note 1)	DMA1/$\overline{\text{INT7}}$ (Note 3)	
Software interrupt number 13	+52 to +55 (Note 1)	Key input interrupt 0	
Software interrupt number 14	+56 to +59 (Note 1)	A-D	
Software interrupt number 15	+60 to +63 (Note 1)	UART2 transmit/IBF0 (Note 2)	
Software interrupt number 16	+64 to +67 (Note 1)	UART2 receive/IBF1 (Note 2)	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit/I^2C0 (Note 2)	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive/SCL0,SDA0/$\overline{\text{INT11}}$ (Note 3)	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmit/I^2C1 (Note 2)	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 receive/SCL1,SDA1/$\overline{\text{INT10}}$ (Note 3)	
Software interrupt number 21	+84 to +87 (Note 1)	Timer A0/$\overline{\text{INT8}}$ (Note 3)	
Software interrupt number 22	+88 to +91 (Note 1)	Timer A1/$\overline{\text{INT7}}$ (Note 3)	
Software interrupt number 23	+92 to +95 (Note 1)	Timer A2	
Software interrupt number 24	+96 to +99 (Note 1)	Timer A3/IBF2	
Software interrupt number 25	+100 to +103 (Note 1)	Timer A4/IBF3	
Software interrupt number 26	+104 to +107 (Note 1)	Timer B0/$\overline{\text{INT11}}$/SCL0,SDA0 (Note 3)	
Software interrupt number 27	+108 to +111 (Note 1)	Timer B1/$\overline{\text{INT10}}$/SCL1,SDA1 (Note 3)	
Software interrupt number 28	+112 to +115 (Note 1)	Timer B2/Key input interrupt 1 (Note 2)	
Software interrupt number 29	+116 to +119 (Note 1)	$\overline{\text{INT0}}$/PS20 (Note 2)	
Software interrupt number 30	+120 to +123 (Note 1)	$\overline{\text{INT1}}$/PS21 (Note 2)	
Software interrupt number 31	+124 to +127 (Note 1)	$\overline{\text{INT2}}$/PS22 (Note 2)	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked by I flag

Note 1: Address relative to address in interrupt table register (INTB).
Note 2: It is selected by interrupt request cause bit.
Note 3: Depend on interrupt event selection bit setting.Please do not set same interrupt event at the same time.

Interrupt Control

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

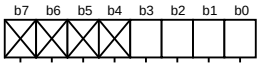
Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bits and processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt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

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

The interrupt factors in the same vector share the same interrupt control register. Which factor to be used depends on interrupt factor selection bit of interrupt event selection register i(address:035F16,035616, to 035816, i = 0 to 3) setting. After setting the interrupt factor, the corresponding interrupt request bit must be set to "0" before changing the interrupt.

Interrupt control register

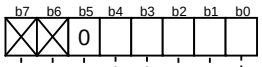
Symbol	Address	When reset
TBiIC(i=3,4)	0047_{16} ,0046_{16}	$XXXXX000_2$
KUP0IC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
S2TIC/IBF0IC	$004F_{16}$	$XXXXX000_2$
S2RIC/IBF1IC	0050_{16}	$XXXXX000_2$
SiTIC/IICjIC(i=0,1)	0051_{16} ,0053_{16}	$XXXXX000_2$
(j=0,1)	0051_{16} ,0053_{16}	$XXXXX000_2$
TA2IC	0057_{16}	$XXXXX000_2$
TAiIC/IBFjIC(i=3,4)	0058_{16} ,0059_{16}	$XXXXX000_2$
(j=2,3)	0058_{16} ,0059_{16}	$XXXXX000_2$
TB2IC/KUP1IC	$005C_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 1)
Nothing is assigned.			–	–

Note 1: Can only be writing by "0" (Please do not write "1" to this bit)

Fig.DD-3 Interrupt control registers(1)



Symbol	Address	When reset
INT3IC	0044₁₆	XX00X000₂
TB5IC/INT9IC	0045₁₆	XX00X000₂
SiIC/INTjIC (i=4, 3)	0048₁₆, 0049₁₆	XX00X000₂
(j=6, 5)	0048₁₆, 0049₁₆	XX00X000₂
BCNIC/INT4IC	004A₁₆	XX00X000₂
DMiIC/INTjIC(i=0, 1)	004B₁₆, 004C₁₆	XX00X000₂
(j=8, 7)	004B₁₆, 004C₁₆	XX00X000₂
SiRIC/SCLDAjIC/INTkIC(i=0, 1)	0052₁₆, 0054₁₆	XX00X000₂
(j=0, 1)	0052₁₆, 0054₁₆	XX00X000₂
(k=11,10)	0052₁₆, 0054₁₆	XX00X000₂
TAiIC/INTjIC(i=0, 1)	0055₁₆, 0056₁₆	XX00X000₂
(j=8, 7)	0055₁₆, 0056₁₆	XX00X000₂
TBiIC/INTjIC/SCLDkIC(i=0, 1)	005A₁₆, 005B₁₆	XX00X000₂
(j=11, 10)	005A₁₆, 005B₁₆	XX00X000₂
(k=0,1)	005A₁₆, 005B₁₆	XX00X000₂
INTiIC/PS2jIC(i=0 to 2)	005D₁₆ to 005F₁₆	XX00X000₂
(j=0 to 2)	005D₁₆ to 005F₁₆	XX00X000₂

Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note 1)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned.			–	–

Note 1: Can only be written by "0" (Please do not write "1" to this bit)

Fig.DD-4 Interrupt control registers (2)

Interrupt Enable Flag (I flag)

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

Interrupt Request Bit

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

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

Set the interrupt priority level using the interrupt priority level select bits in the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.
Table.DD-3 shows the settings of interrupt priority levels and Table.DD-4 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:
· interrupt enable flag (I flag) = 1
· interrupt request bit = 1
· interrupt priority level > processor interrupt priority level (IPL)

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

Table.DD-3 Settings of interrupt priority levels

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

Table.DD-4 Interrupt levels enabled according to the contents of the IPL

IPL (IPL_2 IPL_1 IPL_0)	Enabled interrupt priority levels
0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

Rewrite the interrupt control register

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

Example 1:

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

Example 2:

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

Example 3:

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

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

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

Interrupt Sequence

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

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

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

(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 0000_{16}.

(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.

(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0" (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)

(4) Saves the content of the temporary register (Note) within the CPU in the stack area.

(5) Saves the content of the program counter (PC) in the stack area.

(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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

Note: This register cannot be utilized by the user.

Interrupt Response Time

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

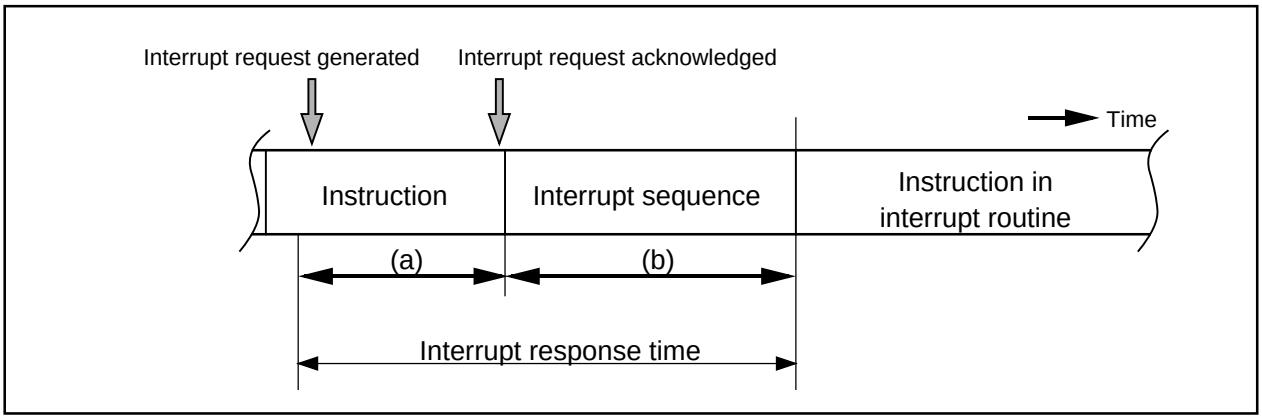


Fig.DD-5 Interrupt response time

Time (a) is dependent on the instruction under execution. 30 cycles is the maximum required for the DIVX instruction (without wait).
Time (b) is as shown in Table.DD-5

Table.DD-5 Time required for executing the interrupt sequence

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

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

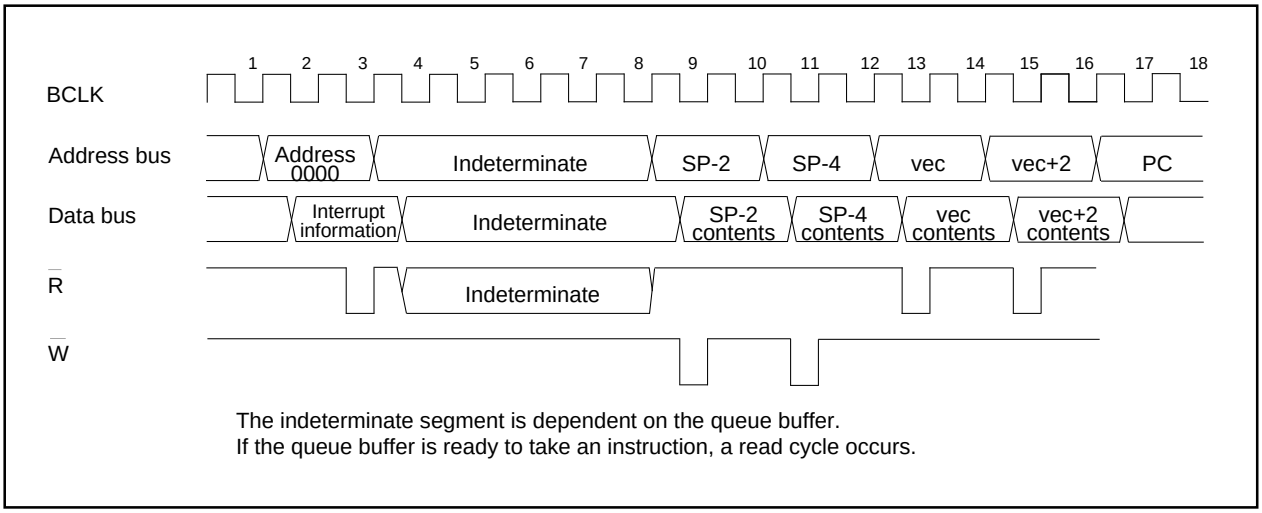


Fig.DD-6 Time required for executing the interrupt sequence

Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.
If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table.DD-6 is set in the IPL.

Table.DD-6 Relationship between interrupts without interrupt priority levels and IPL

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

Saving Registers

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

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

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

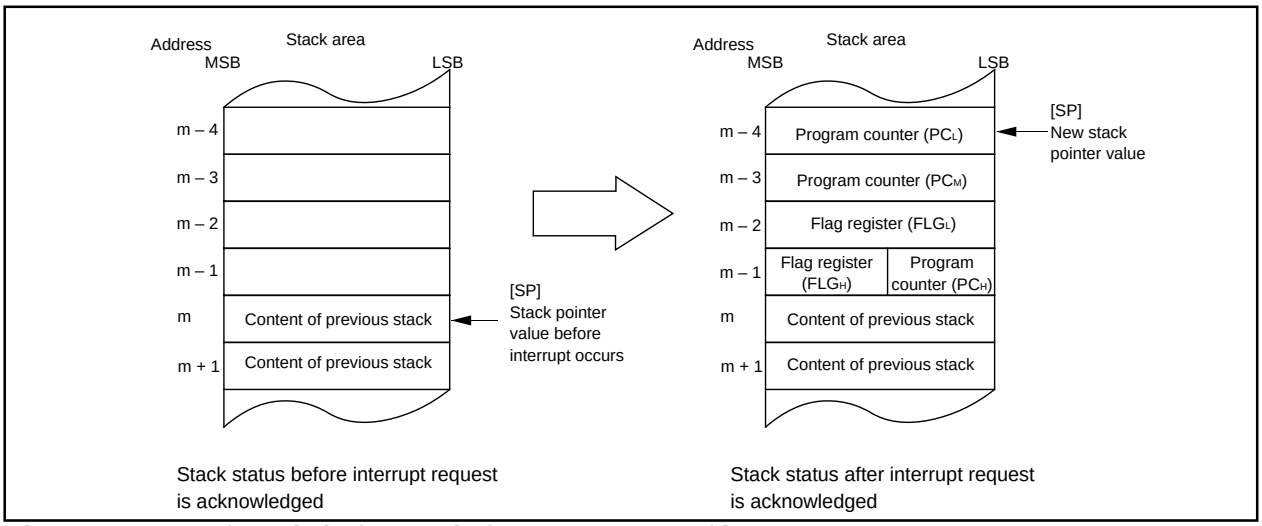


Fig.DD-7 State of stack before and after acceptance of interrupt request

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

Note: Stack pointer is indicated by U flag when software number 32 - 63 INT command is executed, otherwise is indicated by ISP.

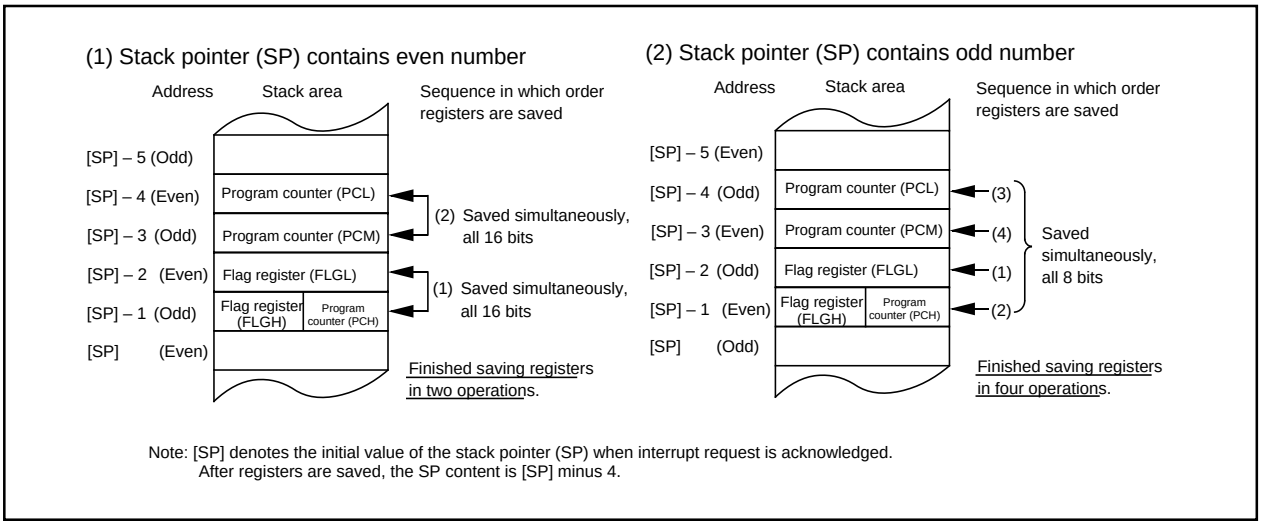


Fig.DD-8 Operation of saving registers

Returning from an Interrupt Routine

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

Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.
Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bits. If the same interrupt priority level is assigned, however, the interrupt with higher hardware priority is accepted.
Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.
Fig.DD-9 shows the priorities of hardware interrupts.
Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine

Interrupt priority level judgement circuit

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

Reset > $\overline{\text{NMI}}$ > $\overline{\text{DBC}}$ > Watchdog timer > Peripheral I/O > Single step > Address match

Fig.DD-9 Hardware interrupts priorities

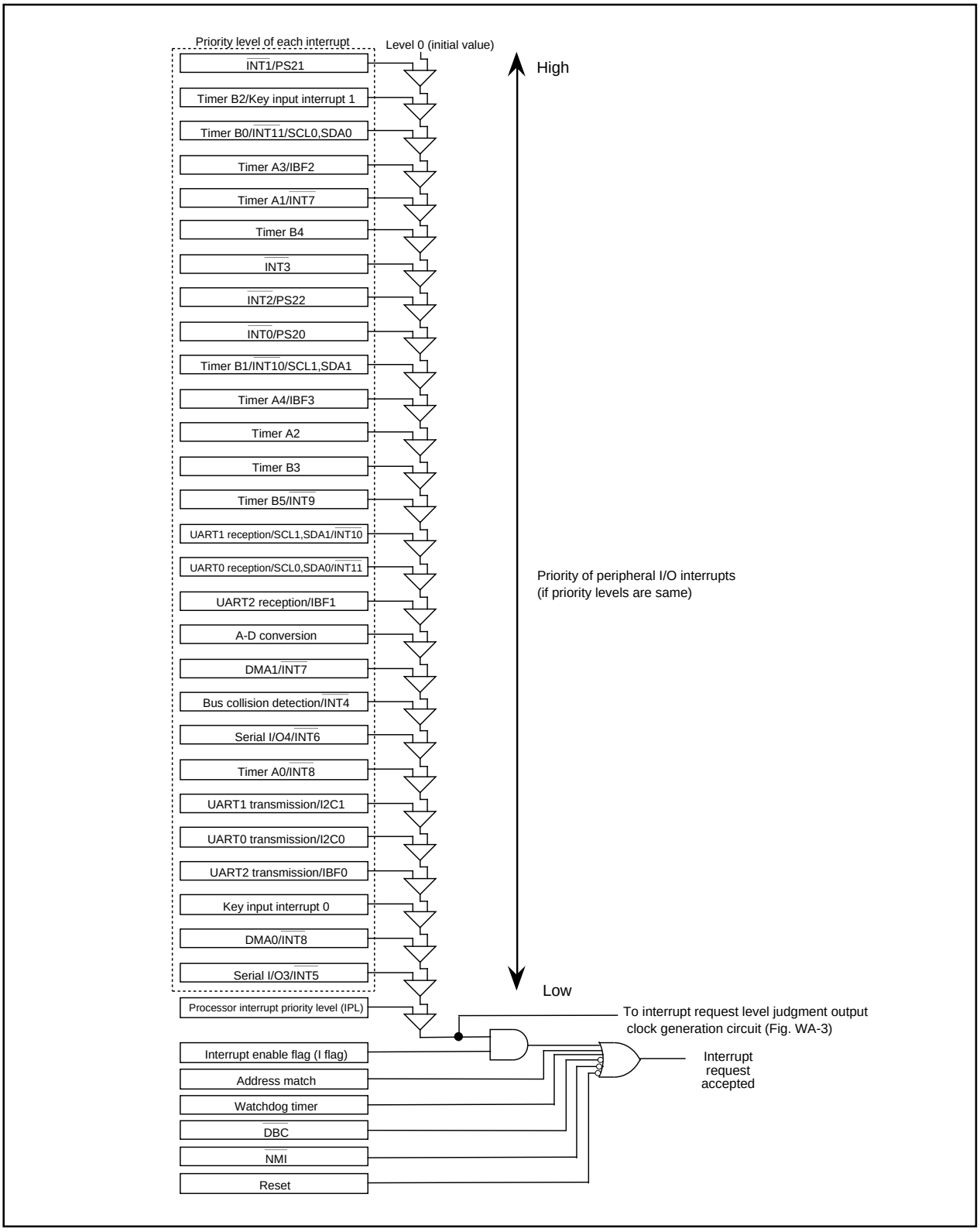


Fig.DD-10 Interrupt priority judgement circuit

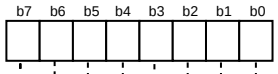
Factor selection

Numbers of interrupt factors share the same interrupt registers in the addresses of 0045$_{16}$, 0048$_{16}$ - 004C$_{16}$, 004F$_{16}$ - 0056$_{16}$, 0058$_{16}$ -005F$_{16}$. The setting of interrupt factor selection bits of interrupt factor selection registers 0 - 3 (addresses of 035F$_{16}$, 0356$_{16}$ - 0358$_{16}$) select the interrupt factor. After the selection of interrupt factor, the corresponding interrupt request bit must be "0" before enabling the interrupt.
Fig.DD-11 - Fig.DD-13 show the structure of interrupt factor selection register 0 - 3.

INT Interrupt

INT0 to INT11 are triggered by the edges of external inputs. The edge polarity can be selected using the polarity select bit.
INT0 to INT2 and INT4 to INT11 have polarity switching bit in the interrupt event select register. The polarity switching bit has to set to "0" when INT interrupt event is not selected.
As for external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting “1” in the INTi interrupt polarity switching bit of the interrupt factor selection register0,4 (035F$_{16}$,0359$_{16}$). To select both edges, set the polarity switching bit of the corresponding interrupt control register to ‘falling edge’ (“0”).
Fig.DD-11, Fig.DD-13 show the Interrupt factor selection register 0, 4.

Interrupt factor selection register 0

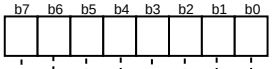


Symbol	Address	When reset
IFSR0	035F$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
IFSR00	INT0 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR01	INT1 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR02	INT2 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR03	INT3 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR04	INT4 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR05	INT5 interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR06	Interrupt factor selection bit (Selecting interrupt factor in the address of 0049$_{16}$)	0 : SIO3 1 : INT5	○	○
IFSR07	Interrupt factor selection bit (Selecting interrupt factor in the address of 0048$_{16}$)	0 : SIO4 1 : INT6	○	○

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

Interrupt factor selection register 1

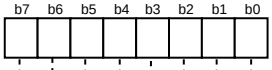


Symbol	Address	When reset
IFSR1	0356_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
IFSR10	Interrupt factor selection bit (Selecting interrupt factor in the address of 0045_{16})	0 : TimerB5 1 : INT9	○	○
IFSR11	Interrupt factor selection bit (Selecting interrupt factor in the address of $004A_{16}$)	0 : Bus collision detection 1 : INT4	○	○
IFSR12	Interrupt factor selection bit (Selecting interrupt factor in the address of $004B_{16}$)	0 : DMA0 1 : INT8	○	○ (Note1)
IFSR13	Interrupt factor selection bit (Selecting interrupt factor in the address of $004C_{16}$)	0 : DMA1 1 : INT7	○	○ (Note1)
IFSR14	Interrupt factor selection bit (Selecting interrupt factor in the address of $004F_{16}$)	0 : UART2 transmission 1 : IBF0	○	○
IFSR15	Interrupt factor selection bit (Selecting interrupt factor in the address of 0050_{16})	0 : UART2 reception 1 : IBF1	○	○
IFSR16	Interrupt factor selection bit (Selecting interrupt factor in the address of 0051_{16})	0 : UART0 transmission 1 : I^2C0	○	○
IFSR17	Interrupt factor selection bit (Selecting interrupt factor in the address of 0053_{16})	0 : UART1 transmission 1 : I^2C1	○	○

Note 1: Do not select the bit if $\overline{INT7}$, $\overline{INT8}$ are selected by interrupt factor selection register 3.

Interrupt factor selection register 2



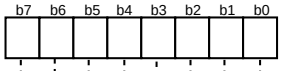
Symbol	Address	When reset
IFSR2	0357_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
IFSR20	Interrupt factor selection bit (Note1) (Selecting interrupt factor in the address of 0052_{16})	b1 b0 0 0 : UART0 reception 0 1 : SCL0,SDA0 1 0 : INT11 1 1 : Inhibited	○	○
IFSR21			○	○
IFSR22	Interrupt factor selection bit (Note1) (Selecting interrupt factor in the address of 0054_{16})	b3 b2 0 0 : UART1 reception 0 1 : SCL1,SDA1 1 0 : INT10 1 1 : Inhibited	○	○
IFSR23			○	○
IFSR24	Interrupt factor selection bit (Note1) (Selecting interrupt factor in the address of $005A_{16}$)	b5 b4 0 0 : TimerB0 0 1 : INT11 1 0 : SCL0,SDA0 1 1 : Inhibited	○	○
IFSR25			○	○
IFSR26	Interrupt factor selection bit (Note1) (Selecting interrupt factor in the address of $005B_{16}$)	b7 b6 0 0 : TimerB1 0 1 : INT10 1 0 : SCL1,SDA1 1 1 : Inhibited	○	○
IFSR27			○	○

Note1 : Do not select $\overline{INT10}$, $\overline{INT11}$, SCL0, SDA0, SCL1, SDA1 simultaneously in the interrupt control registers.

Fig.DD-12 Interrupt factor selection register(2)

Interrupt factor selection register 3

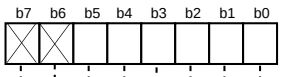


Symbol	Address	When reset
IFSR3	0358₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
IFSR30	Interrupt factor selection bit (Selecting interrupt factor in the address of 0055₁₆)	0 : TimerA0 1 : INT8	○	○ (Note1)
IFSR31	Interrupt factor selection bit (Selecting interrupt factor in the address of 0056₁₆)	0 : TimerA1 1 : INT7	○	○ (Note1)
IFSR32	Interrupt factor selection bit (Selecting interrupt factor in the address of 0058₁₆)	0 : TimerA3 1 : IBF2	○	○
IFSR33	Interrupt factor selection bit (Selecting interrupt factor in the address of 0059₁₆)	0 : TimerA4 1 : IBF3	○	○
IFSR34	Interrupt factor selection bit (Selecting interrupt factor in the address of 005C₁₆)	0 : TimerB2 1 : Key input interrupt 1	○	○
IFSR35	Interrupt factor selection bit (Selecting interrupt factor in the address of 005D₁₆)	0 : $\overline{\text{INT0}}$ 1 : PS20	○	○
IFSR36	Interrupt factor selection bit (Selecting interrupt factor in the address of 005E₁₆)	0 : $\overline{\text{INT1}}$ 1 : PS21	○	○
IFSR37	Interrupt factor selection bit (Selecting interrupt factor in the address of 005F₁₆)	0 : $\overline{\text{INT2}}$ 1 : PS22	○	○

Note1: Do not select the bit if $\overline{\text{INT7}}$, $\overline{\text{INT8}}$ are selected by interrupt factor selection register 1.

Interrupt factor selection register 4



Symbol	Address	When reset
IFSR4	0359₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
IFSR40	$\overline{\text{INT6}}$ Interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR41	$\overline{\text{INT7}}$ Interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR42	$\overline{\text{INT8}}$ Interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR43	$\overline{\text{INT9}}$ Interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR44	$\overline{\text{INT10}}$ Interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
IFSR45	$\overline{\text{INT11}}$ Interrupt polarity switching bit	0 : One edge 1 : Two edges	○	○
Nothing is assigned			—	—

Fig.DD-13 Interrupt factor selection register(3)

NMI Interrupt

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

This pin cannot be used as a normal port input.

Key Input Interrupt 0

If the direction register of any of P50 to P57 is set for input and a falling edge is input to that port, a key input interrupt 0 is generated. A key input interrupt 0 can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. Fig.DD-14 shows the block diagram of the key input interrupt 0. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

Key Input Interrupt 1

If the direction register of any of P140 to P147 is set for input and a falling edge is input to that port, a key input interrupt 1 is generated. The key input interrupt 1's function is equity to key input interrupt 0 except for the valid falling edge input will be latched. When there is a valid falling edge input in P140 to P147 the corresponding bit of P14 event register (Address : 02F616) will be set to "1". After interrupt request is generated, the interrupt generated by "L" input can be conformed by reading the the register even the pin has been returned to "H". A dummy write to the P14 event register clears the register to "0".

The block diagram of key input interrupt 1 is shown on Fig.DD-15 and the timing diagram of key input interrupt 1 is shown on Fig.DD-16.

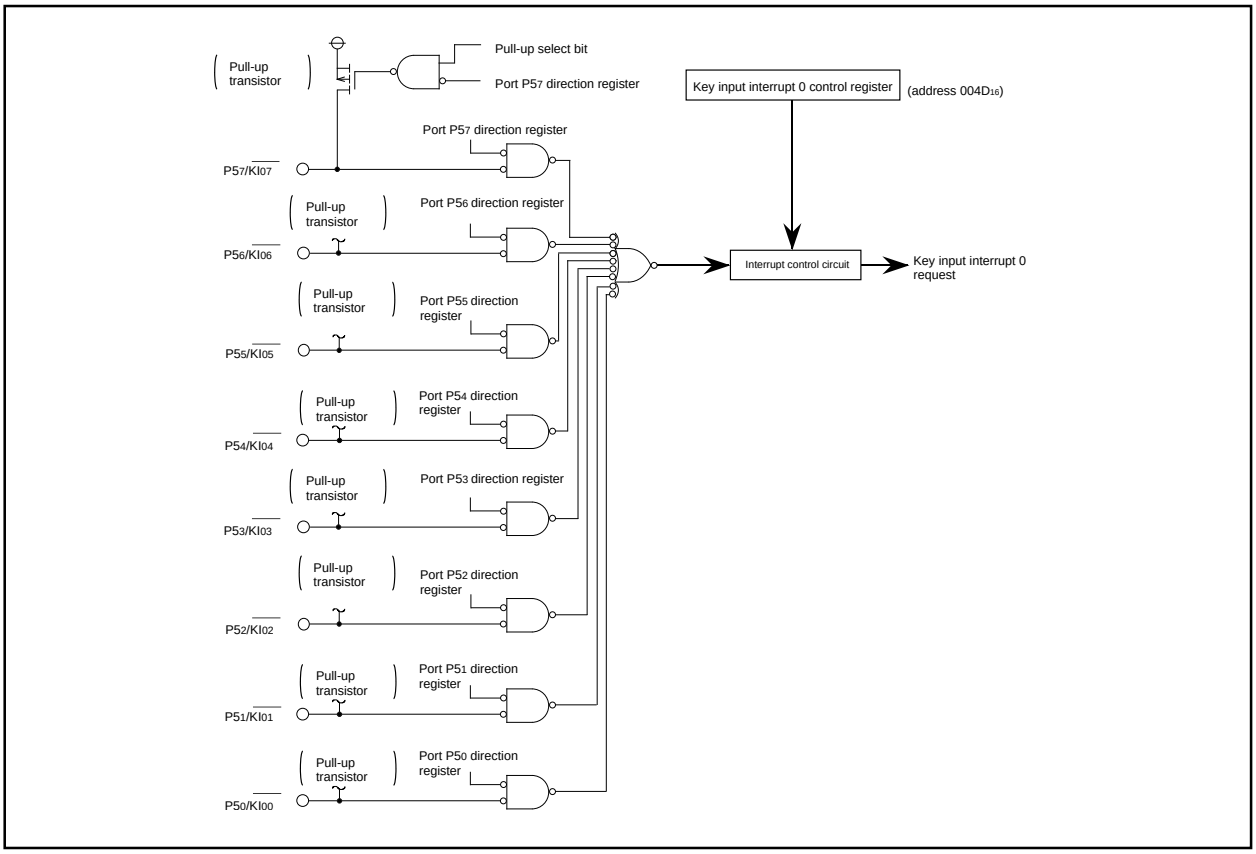


Fig.DD-14 Block diagram of key input interrupt 0

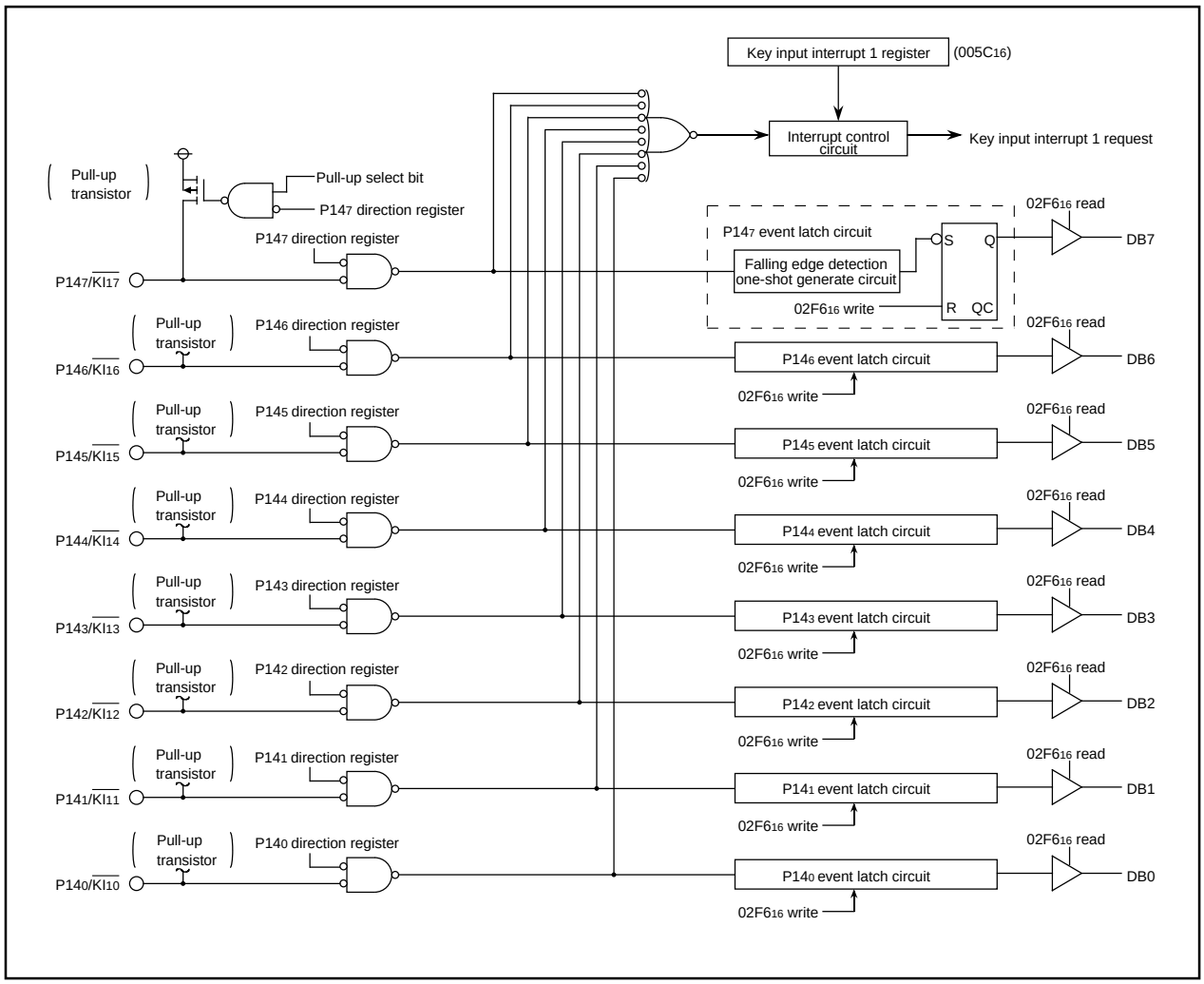
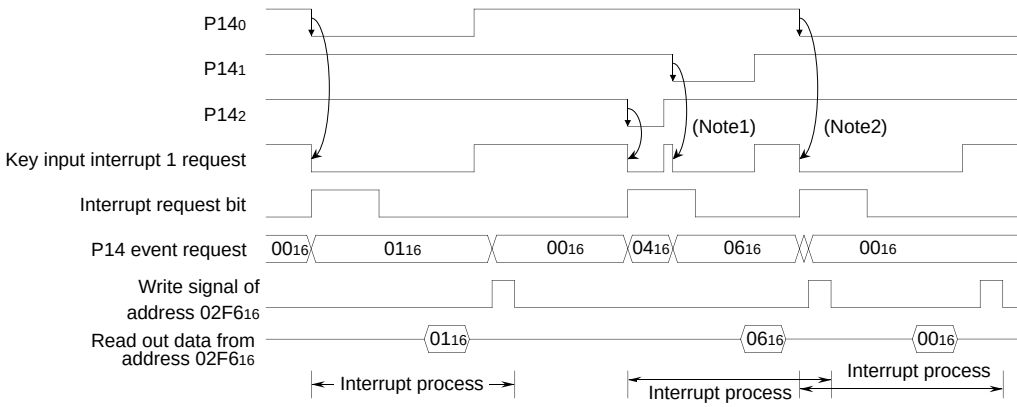


Fig.DD-15 Block diagram of key input interrupt 1



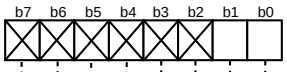
Note 1: The input sequence order can not be conformed if there are numbers of valid falling edge inputs,
Note2 : If the valid falling edge input and the write to the address of 02F616 occur simultaneously, the event may not be latched.

Fig.DD-16 Block diagram of key input interrupt 1

Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The stacked value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed.
Fig.DD-17 shows the address match interrupt-related registers.

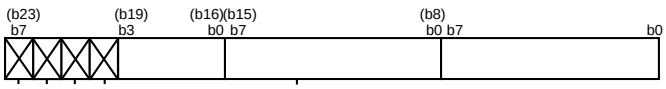
Address match interrupt enable register



Symbol	Address	When reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminated.			—	—

Address match interrupt register i (i = 0, 1)



Symbol	Address	When reset
RMAD0	0012_{16} to 0010_{16}	$X00000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X00000_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminated.		—	—

Fig.DD-17 Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 0000_{16}

• When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
The interrupt request bit of the certain interrupt written in address 0000_{16} will then be set to “0”.
Reading address 0000_{16} by software sets the request bit, which the interrupt source is enabled with the highest priority, to “0”.
Though the interrupt is generated, the interrupt routine may not be executed.
Hence do not read address 0000_{16} by software.

(2) Setting the stack pointer

• The value of the stack pointer is initialized to 0000_{16} right after the reset. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. When using the $\overline{\text{NMI}}$ interrupt, initialize the stack point at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.

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

• As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistor (pull-up) if unused. Be sure to work on it.
• The $\overline{\text{NMI}}$ pin also serves as $P8_5$, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the $\overline{\text{NMI}}$ interrupt is input.
• Do not reset the CPU with the input to the $\overline{\text{NMI}}$ pin being in the “L” state.
• Do not attempt to go into stop mode with the input to the $\overline{\text{NMI}}$ pin being in the “L” state. With the input to the $\overline{\text{NMI}}$ being in the “L” state, the CM10 is fixed to “0”, so attempting to go into stop mode is turned down.
• Do not attempt to go into wait mode with the input to the $\overline{\text{NMI}}$ pin being in the “L” state. With the input to the $\overline{\text{NMI}}$ pin being in the “L” state, the CPU stops but the oscillation does not stop, so no power is saved. In this instance, the CPU is returned to the normal state by a later interrupt.
• Signals input to the $\overline{\text{NMI}}$ pin require an "L" level of 1 clock or more, from the operation clock of the CPU.

(4) External interrupt

• Either an “L” level or an “H” level of at least 380 ns width is necessary for the signal input to pins $\overline{\text{INT}_0}$ through $\overline{\text{INT}_{11}}$ regardless of the CPU operation clock.
• When the polarity of the $\overline{\text{INT}_0}$ to $\overline{\text{INT}_{11}}$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Fig.DD-18 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

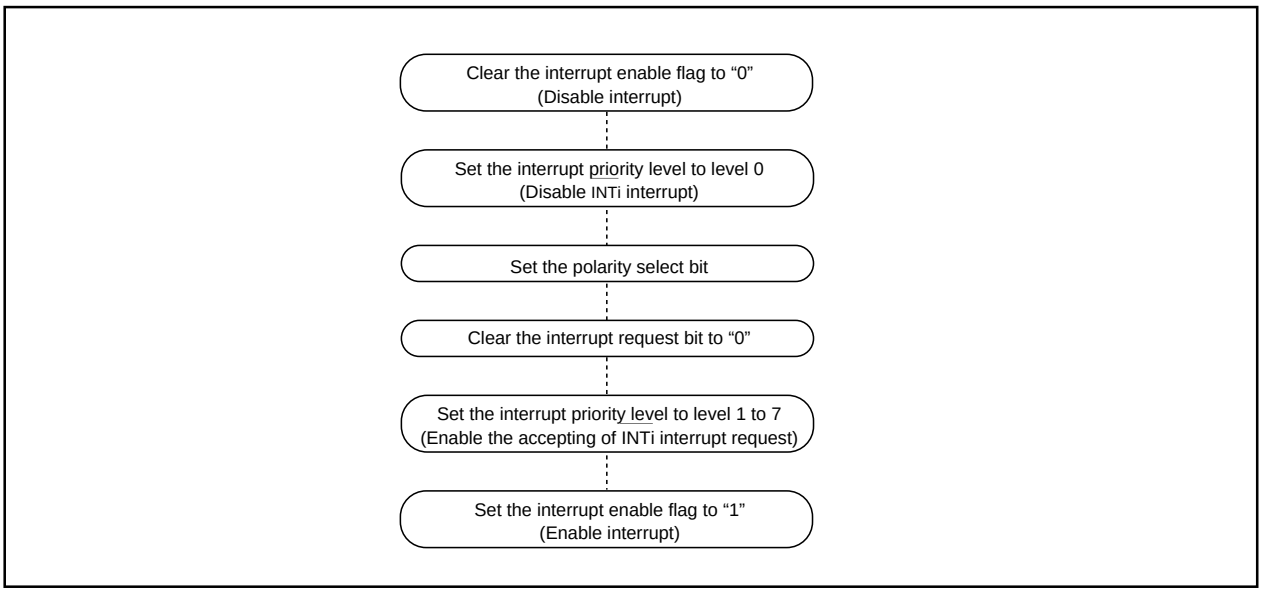


Fig.DD-18 Switching condition of $\overline{\text{INT}}$ interrupt request

(5) Rewrite the interrupt control register

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

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

Example 1:

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

Example 2:

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

Example 3:

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

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

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

Instructions : AND, OR, BCLR, BSET

Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. When XIN is selected for the BCLK, bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F16). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the pre-scaler.

With XIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{pre-scaler dividing ratio (16 or 128) X watchdog timer count (32768)}}{\text{BCLK}}$$

With XCIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{pre-scaler dividing ratio (2) X watchdog timer count (32768)}}{\text{BCLK}}$$

For example, suppose that BCLK runs at 10 MHz and that 16 has been chosen for the dividing ratio of the pre-scaler, then the watchdog timer's period becomes approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address 000E16).
Fig.DG-1 shows the block diagram of the watchdog timer. Fig.DG-2 shows the watchdog timer-related registers.

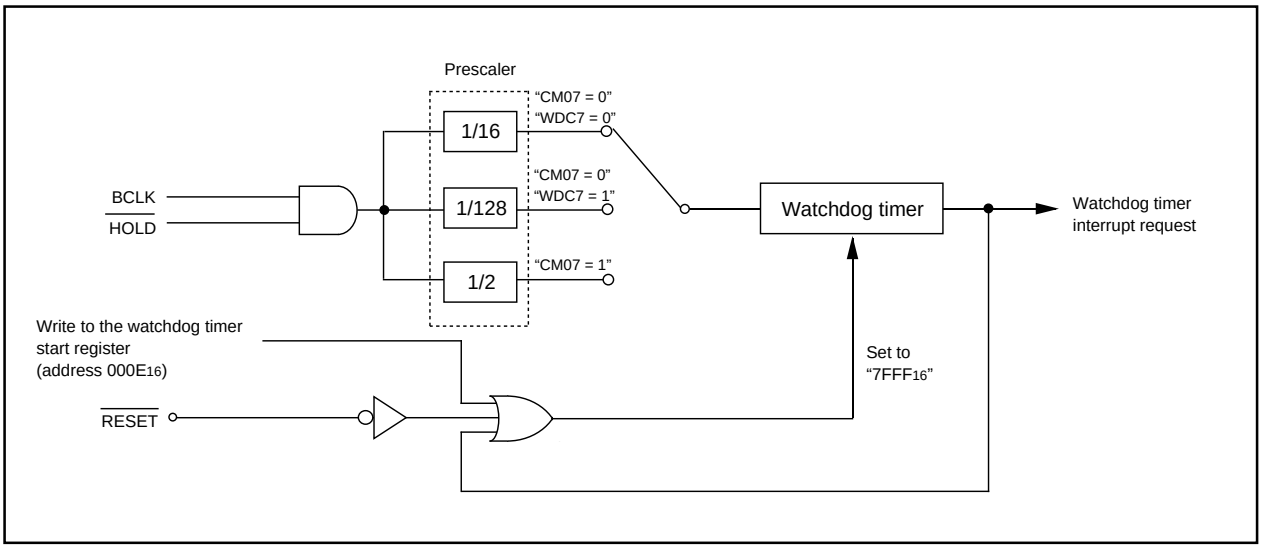


Fig.DG-1 Block diagram of watchdog timer

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	When reset
WDC	$000F_{16}$	$000XXXXX_2$

Bit symbol	Bit name	Function	R	W
High-order bit of watchdog timer			O	X
Reserved bit		Must always be set to “0”	O	O
Reserved bit		Must always be set to “0”	O	O
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	O	O

Watchdog timer start register

b7	b0

Symbol	Address	When reset
WDTS	$000E_{16}$	Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “$7FFF_{16}$” regardless of whatever value is written.	X	O

Fig.DG-2 Watchdog timer control and start registers

DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Fig.EC-1 shows the block diagram of the DMAC. Table.EC-1 shows the DMAC specifications. Fig.EC-2 to EC-4 show the registers used by the DMAC.

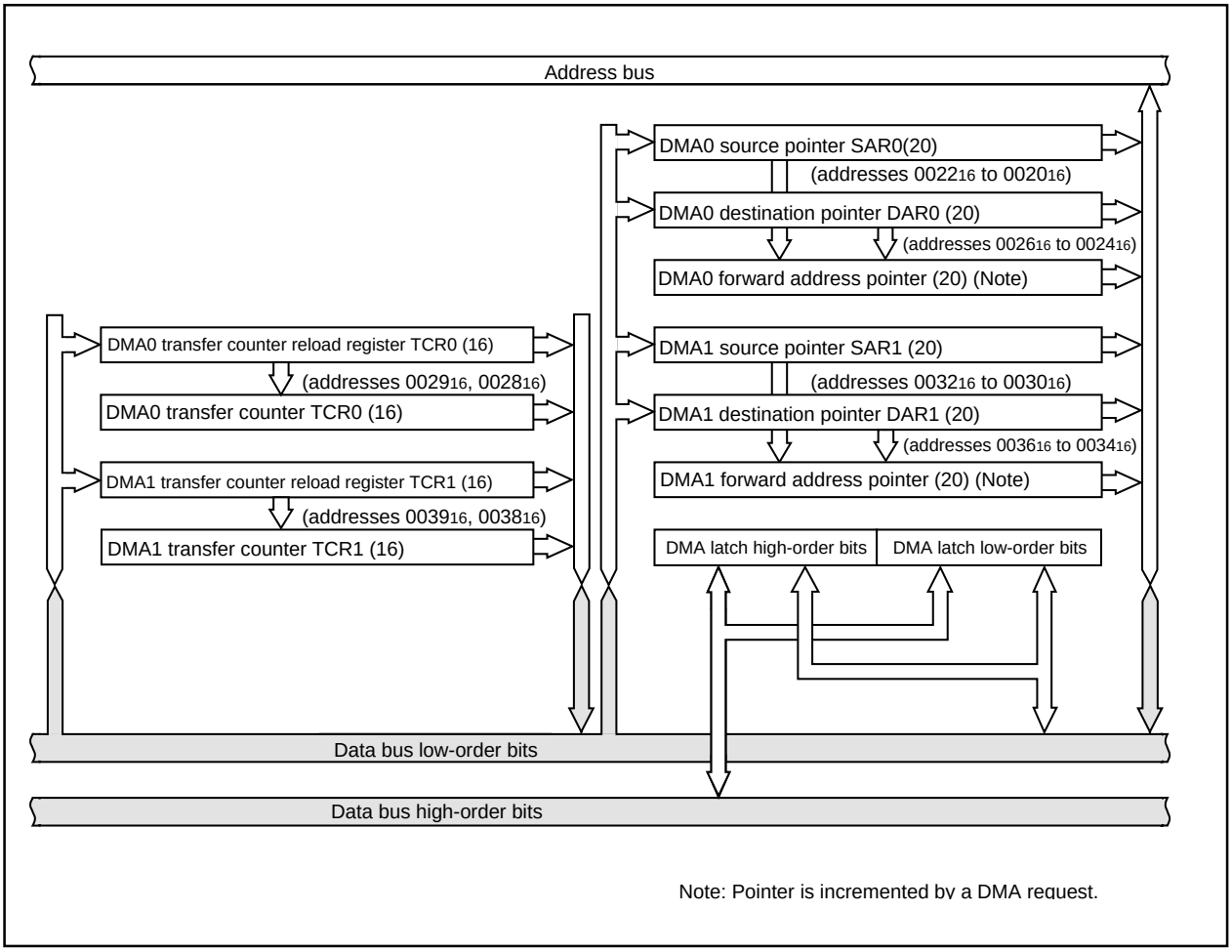


Fig.EC-1 Block diagram of DMAC

Either a write signal to the software DMA request bit or an interrupt request signal is used as a DMA transfer request signal. But the DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level. The DMA transfer doesn't affect any interrupts neither.

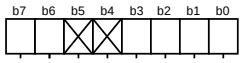
If the DMAC is active (the DMA enable bit is set to 1), data transfer starts every time a DMA transfer request signal occurs. If the cycle of the occurrences of DMA transfer request signals is higher than the DMA transfer cycle, there can be instances in which the number of transfer requests doesn't agree with the number of transfers. For details, see the description of the DMA request bit.

Table.EC-1 DMAC specifications

Item	Specification
No. of channels	2 (cycle steal method)
Transfer memory space	• From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address (Note that DMA-related registers [0020$_{16}$ to 003F$_{16}$] cannot be accessed)
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of $\overline{INT0}$ or $\overline{INT1}$ or both edge Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 transfer and reception interrupt requests UART1 transfer and reception interrupt requests UART2 transfer and reception interrupt requests Serial I/O3, 4 interrpt requests A-D conversion interrupt requests IBF0 to IBF3 interrupt requests Software triggers
Channel priority	DMA0 takes precedence if DMA0 and DMA1 requests are generated simultaneously
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to "0", and the DMAC turns inactive • Repeat transfer mode After the transfer counter underflows, the value of the transfer counter reload register is reloaded to the transfer counter. The DMAC remains active unless a "0" is written to the DMA enable bit.
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
Active	When the DMA enable bit is set to "1", the DMAC is active. When the DMAC is active, data transfer starts every time a DMA transfer request signal occurs.
Inactive	• When the DMA enable bit is set to "0", the DMAC is inactive. • After the transfer counter underflows in single transfer mode
Forward address pointer and reload timing for transfer counter	At the time of starting data transfer immediately after turning the DMAC active, the value of one of source pointer and destination pointer - the one specified for the forward direction - is reloaded to the forward direction address pointer,and the value of the transfer counter reload register is reloaded to the transfer counter.
Writing to register	Registers specified for forward direction transfer are always write enabled. Registers specified for fixed address transfer are write-enabled when the DMA enable bit is "0".
Reading the register	Can be read at any time. However, when the DMA enable bit is "1", reading the register set up as the forward register is the same as reading the value of the forward address pointer.

Note: DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level.

DMA0 request factor selection register

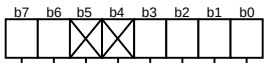


Symbol: DM0SL | Address: $03B8_{16}$ | When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request factor selection bits	b3 b2 b1 b0 0 0 0 0 : Falling edge of $\overline{\text{INT0}}$ pin 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3 0 1 1 0 : Timer A4 (DMS=0) /two edges of $\overline{\text{INT0}}$ pin (DMS=1) 0 1 1 1 : Timer B0 (DMS=0) Timer B3 (DMS=1) 1 0 0 0 : Timer B1 (DMS=0) Timer B4 (DMS=1) 1 0 0 1 : Timer B2 (DMS=0) Timer B5 (DMS=1) 1 0 1 0 : UART0 transmit (DMS=0) /IBF0(DMS=1) 1 0 1 1 : UART0 receive (DMS=0) /IBF1(DMS=1) 1 1 0 0 : UART2 transmit (DMS=0) /IBF2(DMS=1) 1 1 0 1 : UART2 receive(DMS=0) /IBF3(DMS=1) 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 transmit	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–
DMS	DMA request factor expansion bit	0 : Normal 1 : Expanded factor	○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to “1” (When read, the value of this bit is always “0”)	○	○

Fig.EC-2 DMAC register (1)

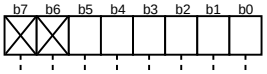
DMA1 request factor selection register



Symbol: DM1SL | Address: 03BA16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request factor selection bits	b3 b2 b1 b0 0 0 0 0 : Falling edge of INT1 pin 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3(DMS=0) /serial I/O3 (DMS=1) 0 1 1 0 : Timer A4 (DMS=0) /serial I/O4 (DMS=1) 0 1 1 1 : Timer B0 (DMS=0) /two edges of INT1 (DMS=1) 1 0 0 0 : Timer B1 1 0 0 1 : Timer B2 1 0 1 0 : UART0 transmit (DMS=0) /IBF0(DMS=1) 1 0 1 1 : UART0 receive (DMS=0) /IBF1(DMS=1) 1 1 0 0 : UART2 transmit(DMS=0) /IBF2(DMS=1) 1 1 0 1 : UART2 receive(DMS=0) /IBF3(DMS=1) 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 receive	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–
DMS	DMA request factor expansion bit	0 : Normal 1 : Expanded factor	○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to “1” (When read, the value of this bit is always “0”)	○	○

DMAi control register



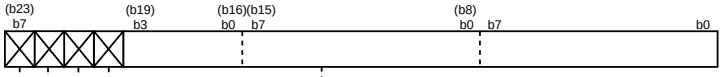
Symbol: DMiCON(i=0,1) | Address: 002C16, 003C16 | When reset: 00000X002

Bit symbol	Bit name	Function	R	W
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	○	○
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	○	○
DMAS	DMA request bit (Note 1)	0 : DMA not requested 1 : DMA requested	○	○ (Note 2)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	○	○
DSD	Source address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
DAD	Destination address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

Note 1: DMA request can be cleared by resetting the bit to "0".
Note 2: This bit can only be set to “0”.
Note 3: Source address direction select bit and destination address direction select bit cannot be set to “1” simultaneously.

Fig.EC-3 DMAC register (2)

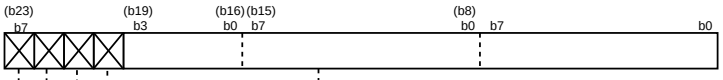
DMAi source pointer (i = 0, 1)



Symbol	Address	When reset
SAR0	0022_{16} to 0020_{16}	Indeterminate
SAR1	0032_{16} to 0030_{16}	Indeterminate

Function	Transfer count specification	R	W
• Source pointer Stores the source address	00000_{16} to FFFFF$_{16}$	O	O
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—	—

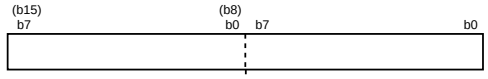
DMAi destination pointer (i = 0, 1)



Symbol	Address	When reset
DAR0	0026_{16} to 0024_{16}	Indeterminate
DAR1	0036_{16} to 0034_{16}	Indeterminate

Function	Transfer count specification	R	W
• Destination pointer Stores the destination address	00000_{16} to FFFFF$_{16}$	O	O
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.		—	—

DMAi transfer counter (i = 0, 1)



Symbol	Address	When reset
TCR0	0029_{16}, 0028_{16}	Indeterminate
TCR1	0039_{16}, 0038_{16}	Indeterminate

Function	Transfer count specification	R	W
• Transfer counter Set a value one less than the transfer count	0000_{16} to FFFF$_{16}$	O	O

Fig.EC-4 DMAC register (3)

(1) Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses.

* Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

Fig.EC-5 shows the example of the transfer cycles for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle. For example (2) in Fig.EC-5, if data are being transferred in 16-bit units on an 8-bit bus, two bus cycles are required for both the source read cycle and the destination write cycle.

(2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table.EC-2 shows the number of DMAC transfer cycles.

The number of DMAC transfer cycles can be calculated as follows:

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x k

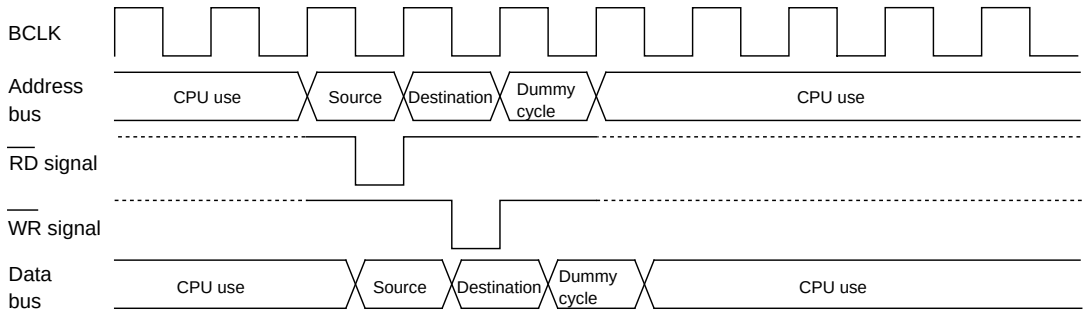
Table.EC-2 No. of DMAC transfer cycles

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion mode Microprocessor mode	
			No. of read cycles	No. of write cycles	No. of read cycles	No. of write cycles
8-bit transfers (DMBIT= “1”)	16-bit (BYTE= “L”)	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit (BYTE = “H”)	Even	—	—	1	1
		Odd	—	—	1	1
16-bit transfers (DMBIT= “0”)	16-bit (BYTE = “L”)	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit (BYTE = “H”)	Even	—	—	2	2
		Odd	—	—	2	2

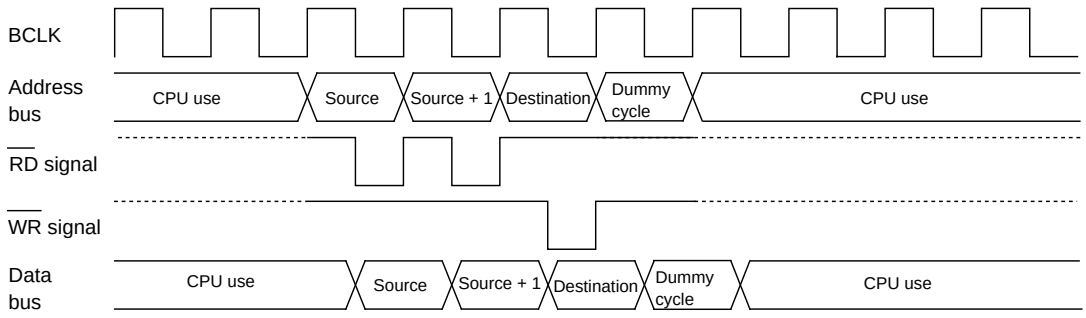
Coefficient j, k

Internal memory			External memory		
Internal ROM/RAM No wait	Internal ROM/RAM With wait	SFR area	Separate bus No wait	Separate bus With wait	Multiplex bus
1	2	2	1	2	3

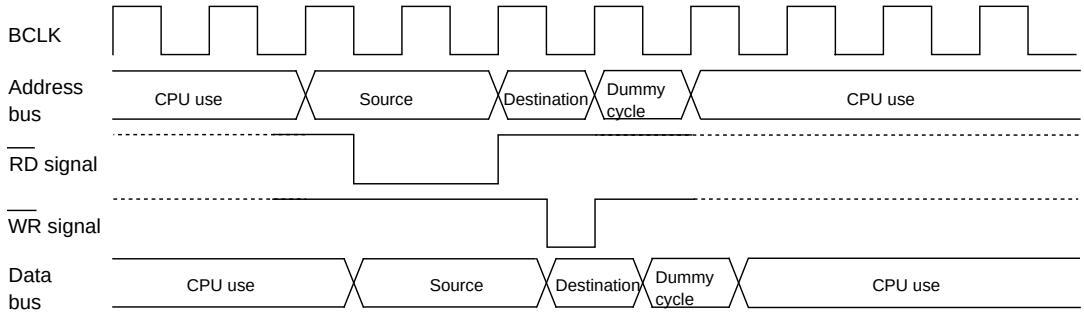
(1) 8-bit transfers
16-bit transfers from even address and the source address is even.



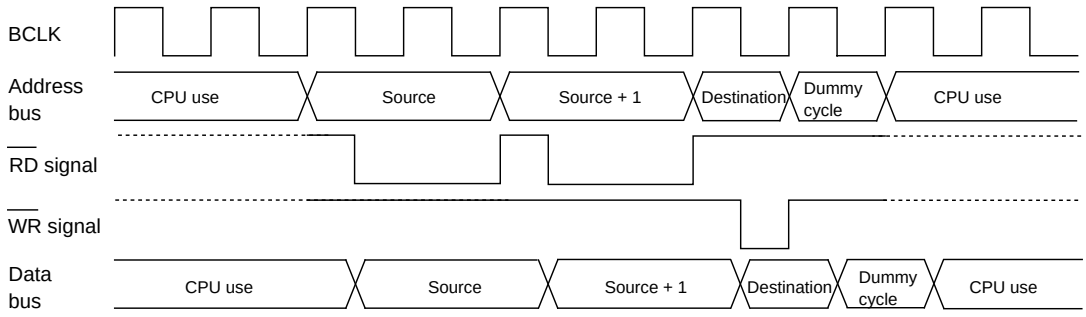
(2) 16-bit transfers and the source address is odd
Transferring 16-bit data on an 8-bit data bus (In this case, there are also two destination write cycles).



(3) One wait is inserted into the source read under the conditions in (1)



(4) One wait is inserted into the source read under the conditions in (2)
(When 16-bit data is transferred on an 8-bit data bus, there are two destination write cycles).



Note: The same timing changes occur with the respective conditions at the destination as at the source.

Fig.EC-5 Example of the transfer cycles for a source read

DMA enable bit

Setting the DMA enable bit to "1" makes the DMAC active. The DMAC carries out the following operations at the time data transfer starts immediately after DMAC is turned active.

(1) Reloads the value of one of the source pointer and the destination pointer - the one specified for the forward direction - to the forward direction address pointer.

(2) Reloads the value of the transfer counter reload register to the transfer counter.

Thus overwriting "1" to the DMA enable bit with the DMAC being active carries out the operations given above, so the DMAC operates again from the initial state at the instant "1" is overwritten to the DMA enable bit.

DMA request bit

The DMAC can generate a DMA transfer request signal triggered by a factor chosen in advance out of DMA request factors for each channel.

DMA request factors include the following.

* Factors effected by using the interrupt request signals from the built-in peripheral functions and software DMA factors (internal factors) effected by a program.
* External factors effected by utilizing the input from external interrupt signals.

For the selection of DMA request factors, see the descriptions of the DMAi factor selection register.

The DMA request bit turns to "1" if the DMA transfer request signal occurs regardless of the DMAC's state (regardless of whether the DMA enable bit is set "1" or to "0"). It turns to "0" immediately before data transfer starts.

In addition, it can be set to "0" by use of a program, but cannot be set to "1".

There can be instances in which a change in DMA request factor selection bit causes the DMA request bit to turn to "1". So be sure to set the DMA request bit to "0" after the DMA request factor selection bit is changed.

The DMA request bit turns to "1" if a DMA transfer request signal occurs, and turns to "0" immediately just before data transfer starts. If the DMAC is active, data transfer starts immediately, so the value of the DMA request bit, if read by use of a program, turns out to be "0" in most cases. To examine whether the DMAC is active, read the DMA enable bit.

Here follows the timing of changes in the DMA request bit.

(1) Internal factors

Except the DMA request factors triggered by software, the timing for the DMA request bit to turn to "1" due to an internal factor is the same as the timing for the interrupt request bit of the interrupt control register to turn to "1" due to several factors.

Turning the DMA request bit to "0" due to an internal factor is timed to be effected immediately just before the transfer starts.

(2) External factors

An external factor is a factor caused to occur by the edge of input from the INTi pin (i depends on which DMAC channel is used).

Selecting the INTi pins as external factors using the DMA request factor selection bit causes input from these pins to become the DMA transfer request signals.

The timing for the DMA request bit to turn to "1" when an external factor is selected synchronizes with the signal's edge applicable to the function specified by the DMA request factor selection bit (synchronizes with the falling edge of the input signal to each INTi pin, for example).

With an external factor selected, the DMA request bit is timed to turn to "0" immediately just before data transfer starts similarly to the state in which an internal factor is selected.

(3) The priorities of channels and DMA transfer timing

If a DMA transfer request signal falls on tne same sampling cycle (a sampling cycle means one period from the rising edge to the falling edge of BCLK), the DMA request bits of applicable channels concurrently turn to "1". If the channels are active at that moment, DMA0 is given a high priority to start data transfer. When DMA0 finishes data transfer, it gives the bus right to the CPU. When the CPU finishes single bus access, then DMA1 starts data transfer and gives the bus right to the CPU.

An example in which DMA transfer is carried out in minimum cycles at the time when DMA transfer request signals due to external factors concurrently occur.

Fig.EC-6 An example of DMA transfer effected by external factors.

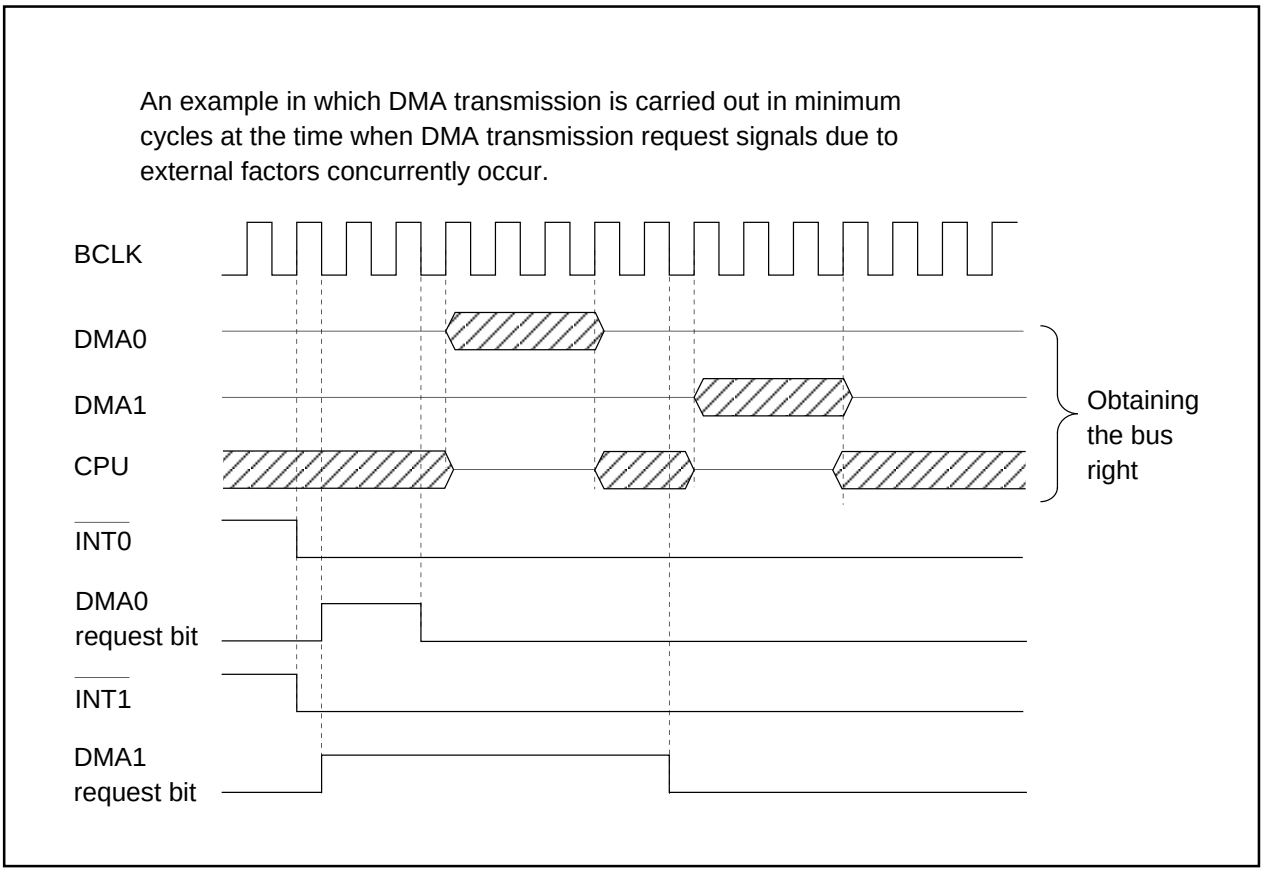
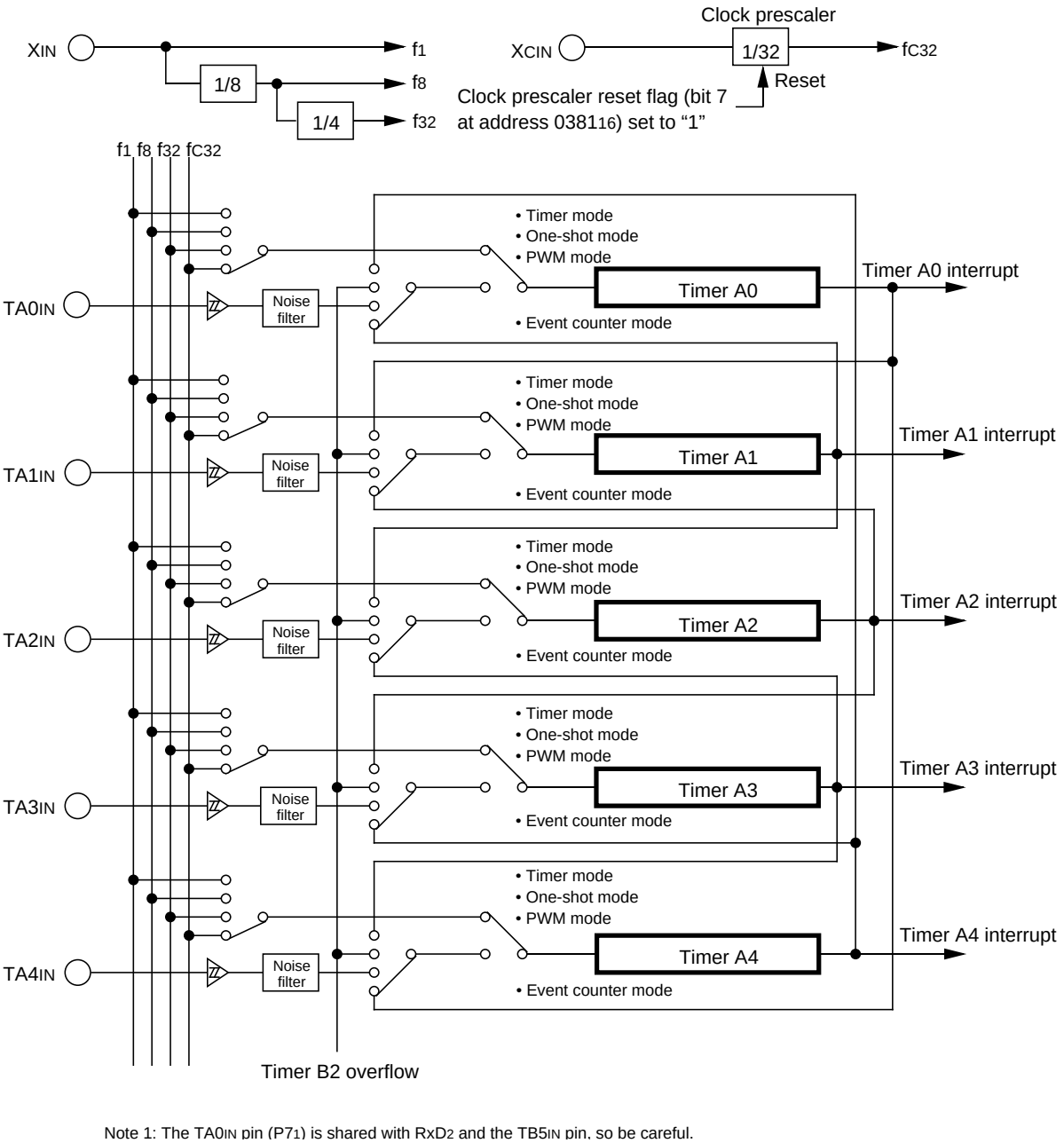


Fig.EC-6 An example of DMA transfer effected by external factors

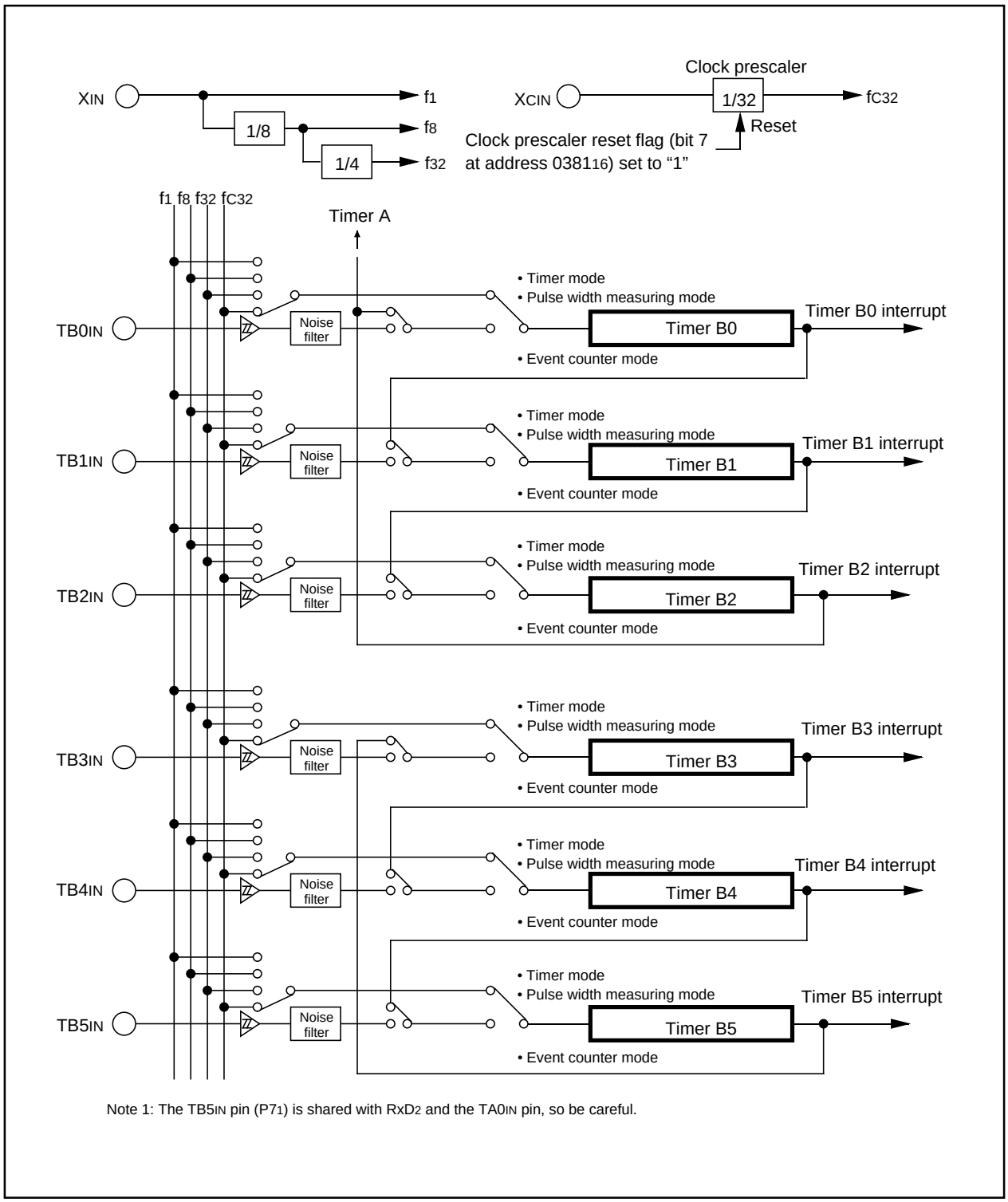
Timer

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



Note 1: The TA0IN pin (P71) is shared with RxD2 and the TB5IN pin, so be careful.

Fig.FB-1 Timer A block diagram



Note 1: The TB5IN pin (P71) is shared with RxD2 and the TA0IN pin, so be careful.

Fig.FB-2 Timer B block diagram

Timer A

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

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

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches "0000_{16}".
- Pulse width modulation (PWM) mode: The timer continually outputs pulse with arbitrary width.

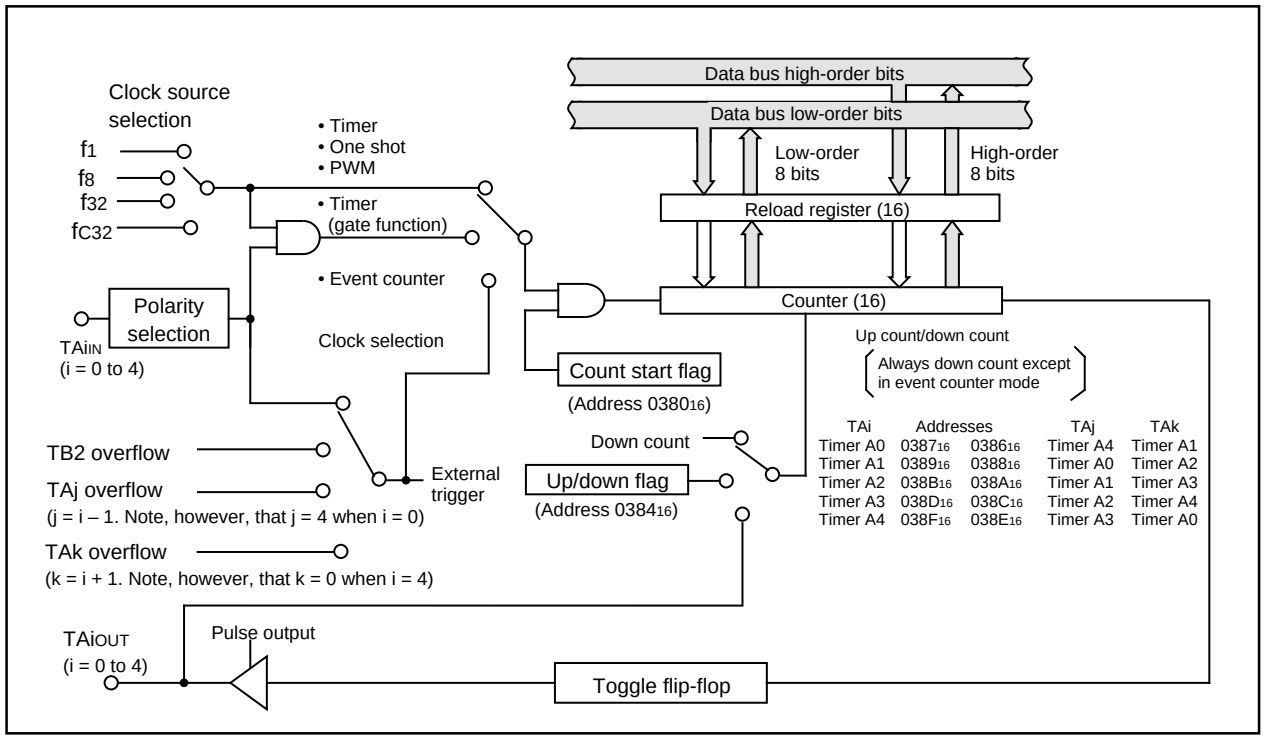


Fig.FB-3 Block diagram of timer A

Timer Ai mode register

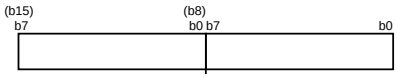
Symbol	Address	When reset
TAiMR(i=0 to 4)	0396_{16} to $039A_{16}$	00_{16}

b7	b6	b5	b4	b3	b2	b1	b0

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source selection bits (Function varies with each operation mode)		○	○
TCK1			○	○

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

Timer Ai register (Note)

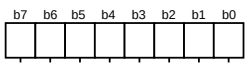


Symbol	Address	When reset
TA0	0387$_{16}$,0386$_{16}$	Indeterminate
TA1	0389$_{16}$,0388$_{16}$	Indeterminate
TA2	038B$_{16}$,038A$_{16}$	Indeterminate
TA3	038D$_{16}$,038C$_{16}$	Indeterminate
TA4	038F$_{16}$,038E$_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Count the internal count source	0000$_{16}$ to FFFF$_{16}$	O	O
• Event counter mode Count pulses from external input or the overflow of timer	0000$_{16}$ to FFFF$_{16}$	O	O
• One-shot timer mode Count one shot width	0000$_{16}$ to FFFF$_{16}$	X	O
• Pulse width modulation mode (16-bit PWM) Function as a 16-bit pulse width modulator	0000$_{16}$ to FFFE$_{16}$	X	O
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00$_{16}$ to FE$_{16}$ (Both high-order and low-order addresses)	X	O

Note: Read and write data in 16-bit units.

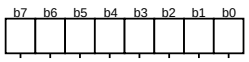
Count start flag



Symbol	Address	When reset
TABSR	0380$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stop count 1 : Start count	O	O
TA1S	Timer A1 count start flag		O	O
TA2S	Timer A2 count start flag		O	O
TA3S	Timer A3 count start flag		O	O
TA4S	Timer A4 count start flag		O	O
TB0S	Timer B0 count start flag		O	O
TB1S	Timer B1 count start flag		O	O
TB2S	Timer B2 count start flag		O	O

Up/down flag

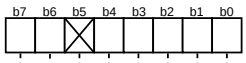


Symbol	Address	When reset
UDF	0384$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching factor	O	O
TA1UD	Timer A1 up/down flag		O	O
TA2UD	Timer A2 up/down flag		O	O
TA3UD	Timer A3 up/down flag		O	O
TA4UD	Timer A4 up/down flag		O	O
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	X	O
TA3P	Timer A3 two-phase pulse signal processing select bit		X	O
TA4P	Timer A4 two-phase pulse signal processing select bit		X	O

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

One-shot start flag

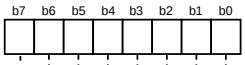


Symbol	Address	When reset
ONSF	0382_{16}	$00X00000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is "0"	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
TA3OS	Timer A3 one-shot start flag		○	○
TA4OS	Timer A4 one-shot start flag		○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			—	—
TA0TGL	Timer A0 event/trigger selection bits	b7 b6 0 0 : Input on $TA0_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

Note: Set the corresponding port direction register to "0".

Trigger selection register

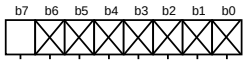


Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger selection bits	b1 b0 0 0 : Input on $TA1_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger selection bits	b3 b2 0 0 : Input on $TA2_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	○	○
TA2TGH			○	○
TA3TGL	Timer A3 event/trigger selection bits	b5 b4 0 0 : Input on $TA3_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	○	○
TA3TGH			○	○
TA4TGL	Timer A4 event/trigger selection bits	b7 b6 0 0 : Input on $TA4_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	○	○
TA4TGH			○	○

Note: Set the corresponding port direction register to "0".

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is "0")	○	○

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

(1) Timer mode

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

Table.FB-1 Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents and then continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TAiIN pin's input signal • Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TAiMR(i=0 to 4) Address: 039_{16} to $039A_{16}$ When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function selection bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	○	○
MR1	Gate function selection bits	b4 b3 0 X (Note 2): Gate function not available (TAiIN pin is a normal port pin) 1 0 : Timer counts only when TAiIN pin is held "L" (Note 3) 1 1 : Timer counts only when TAiIN pin is held "H" (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source selection bits	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: The bit can be "0" or "1".
Note 3: Set the corresponding port direction register to "0".

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

(2) Event counter mode

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

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

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

Item	Specification
Count source	• External signal input to TAiIN pin (effective edge can be selected by software) • TB2 overflow, TAj overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register con tents and then continuing counting (Note)
Divide ratio	1/ ($FFFF_{16}$ - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TAiOUT pin's polarity is reversed

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

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

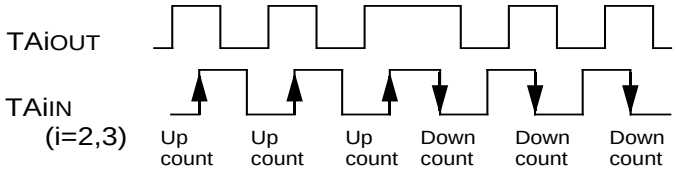
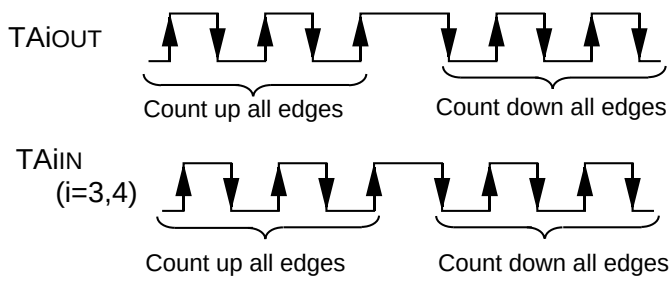
Symbol: TAiMR(i = 0, 1) — Address: 0396_{16}, 0397_{16} — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function selection bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TAiOUT pin is a pulse output pin)	○	○
MR1	Count polarity selection bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Up/down switching factor selection bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 4)	○	○
MR3	0 (Must always be fixed to "0" in event counter mode)		○	○
TCK0	Count operation type selection bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid in event counter mode Can be "0" or "1"		○	○

Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 0382_{16} and 0383_{16}).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an "L" signal is input to the TAiOUT pin, the downcount is activated. When "H", the upcount is activated. Set the corresponding port direction register to "0".

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

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

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN and TAiOUT pin
Count operation	• Up count or down count can be selected by two-phase pulse signals • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	1/ ($FFFF_{16}$ - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	• When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function	• Normal processing operation The timer up-counts by the rising edge of TAiIN pin and down-counts by the falling edge fo TAiIN pin during the "H" level period of input signal in TAiOUT pin.  • Multiply-by-4 processing operation If the phase relationship is such that the TAiIN pin goes “H” when the input signal on the TAiOUT pin is “H”, the timer counts up rising and falling edges on the TAiOUT and TAiIN pins. If the phase relationship is such that the TAiIN pin goes “L” when the input signal on the TAiOUT pin is “H”, the timer counts down rising and falling edges on the TAiOUT and TAiIN pins. 

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

Timer Ai mode register
(When not using two-phase pulse signals' processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol	Address	When reset
TAiMR(i = 2 to 4)	0398$_{16}$ to 039A$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Pulse output function selection bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	○	○
MR1	Count polarity selection bit (Note 2)	0 : Counts external signal's falling edges 1 : Counts external signal's rising edges	○	○
MR2	Up/down switching factor selection bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 3)	○	○
MR3	0 : (Must always be “0” in event counter mode)		○	○
TCK0	Count operation type selection bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse signals' processing operation selection bit (Note 4)(Note 5)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: This bit is valid when only counting an external signal.
Note 3: Set the corresponding port direction register to “0”.
Note 4: This bit is valid for the timer A3 mode register.
For timer A2 and A4 mode registers, this bit can be “0 ”or “1”.
Note 5: When performing two-phase signal processing, make sure the two-phase pulse signals' processing operation selection bit (address 0384$_{16}$) is set to “1”. Also, always be sure to set the event/trigger selection bit (addresses 0382$_{16}$ and 0383$_{16}$) to “00”.

Timer Ai mode register
(When using two-phase pulse signals' processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	0	1

Symbol	Address	When reset
TAiMR(i = 2 to 4)	0398$_{16}$ to 039A$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
MR1	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
MR2	1 (Must always be “1” when using two-phase pulse signal processing)		○	○
MR3	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
TCK0	Count operation type selection bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation selection bit (Note 1)(Note 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: This bit is valid for timer A3 mode register.
For timer A2 and A4 mode registers, this bit can be “0” or “1”.
Note 2: When performing two-phase pulse signals' processing, make sure the two-phase pulse signals' processing operation selection bit (address 0384$_{16}$) is set to “1”. Also, always be sure to set the event/trigger selection bit (addresses 0382$_{16}$ and 0383$_{16}$) to “00”.

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

(3) One-shot timer mode

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

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

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol	Address	When reset
TAiMR(i = 0 to 4)	0396_{16} to $039A_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	Pulse output function selection bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	○	○
MR1	External trigger selection bit (Note 2)	0 : Falling edge of TAiIN pin's input signal (Note 3) 1 : Rising edge of TAiIN pin's input signal (Note 3)	○	○
MR2	Trigger selection bit	0 : One-shot start flag is valid 1 : Selected by event/trigger selection register	○	○
MR3	0 (Must always be "0" in one-shot timer mode)		○	○
TCK0	Count source selection bits	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: Valid only when the TAiIN pin is selected by the event/trigger selection bit (addresses 0382_{16} and 0383_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 3: Set the corresponding port direction register to "0".

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

(4) Pulse width modulation (PWM) mode

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

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

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM	• High level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM	• High level width n×(m+1) / fi n : values set to timer Ai register's high-order address • Cycle time $(2^{8}-1)$× (m+1) / fi m : values set to timer Ai register's low-order address
Count start condition	• External trigger is input • The timer overflows • The count start flag is set (= 1)
Count stop condition	• The count start flag is reset (= 0)
Interrupt request generation timing	The falling edge of PWM pulse
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol: TAiMR(i=0 to 4) | Address: 039616 to 039A16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be "1" in PWM mode)		○	○
MR1	External trigger selection bit (Note 1)	0: Falling edge of TAiIN pin's input signal (Note 2) 1: Rising edge of TAiIN pin's input signal (Note 2)	○	○
MR2	Trigger selection bit	0: Count start flag is valid 1: Selected by event/trigger selection register	○	○
MR3	16/8-bit PWM mode selection bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source selection bits	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Valid only when the TAiIN pin is selected by the event/trigger selection bit (addresses 038216 and 038316). If timer overflow is selected, this bit can be "1" or "0".
Note 2: Set the corresponding port direction register to "0".

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

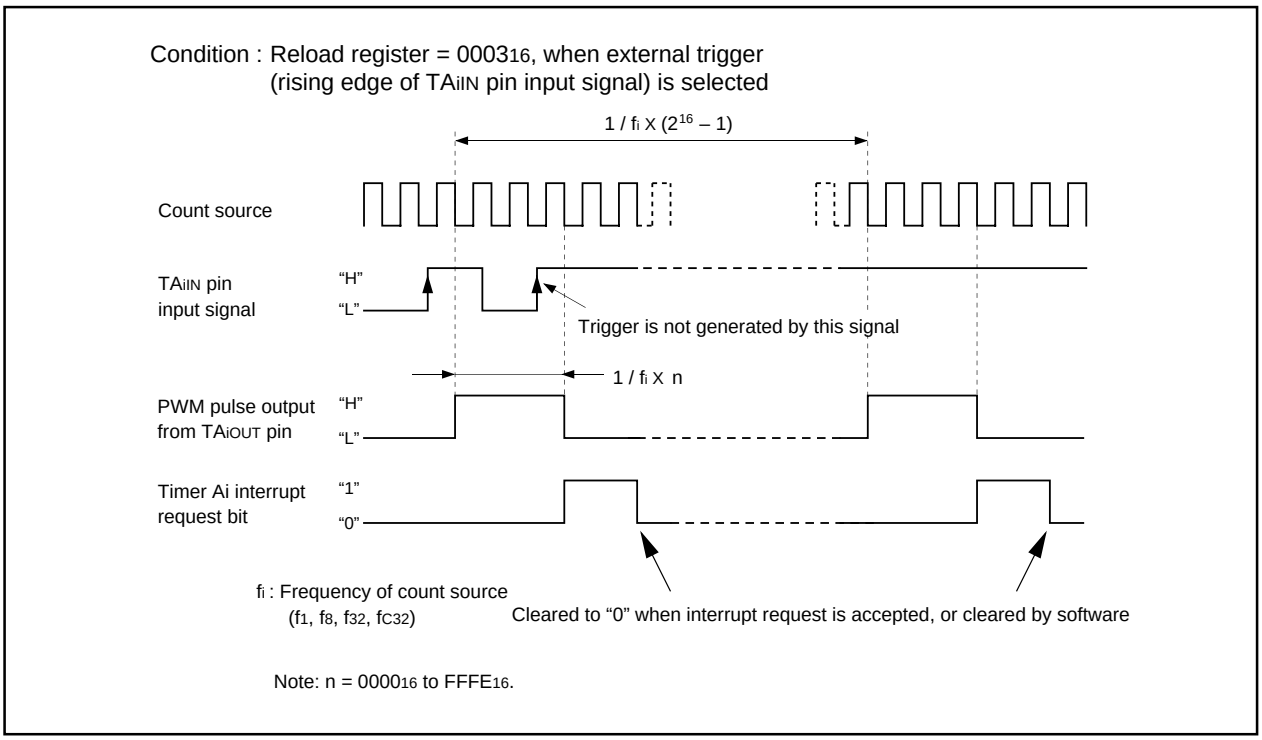


Fig.FB-12 Example of how a 16-bit pulse width modulator operates

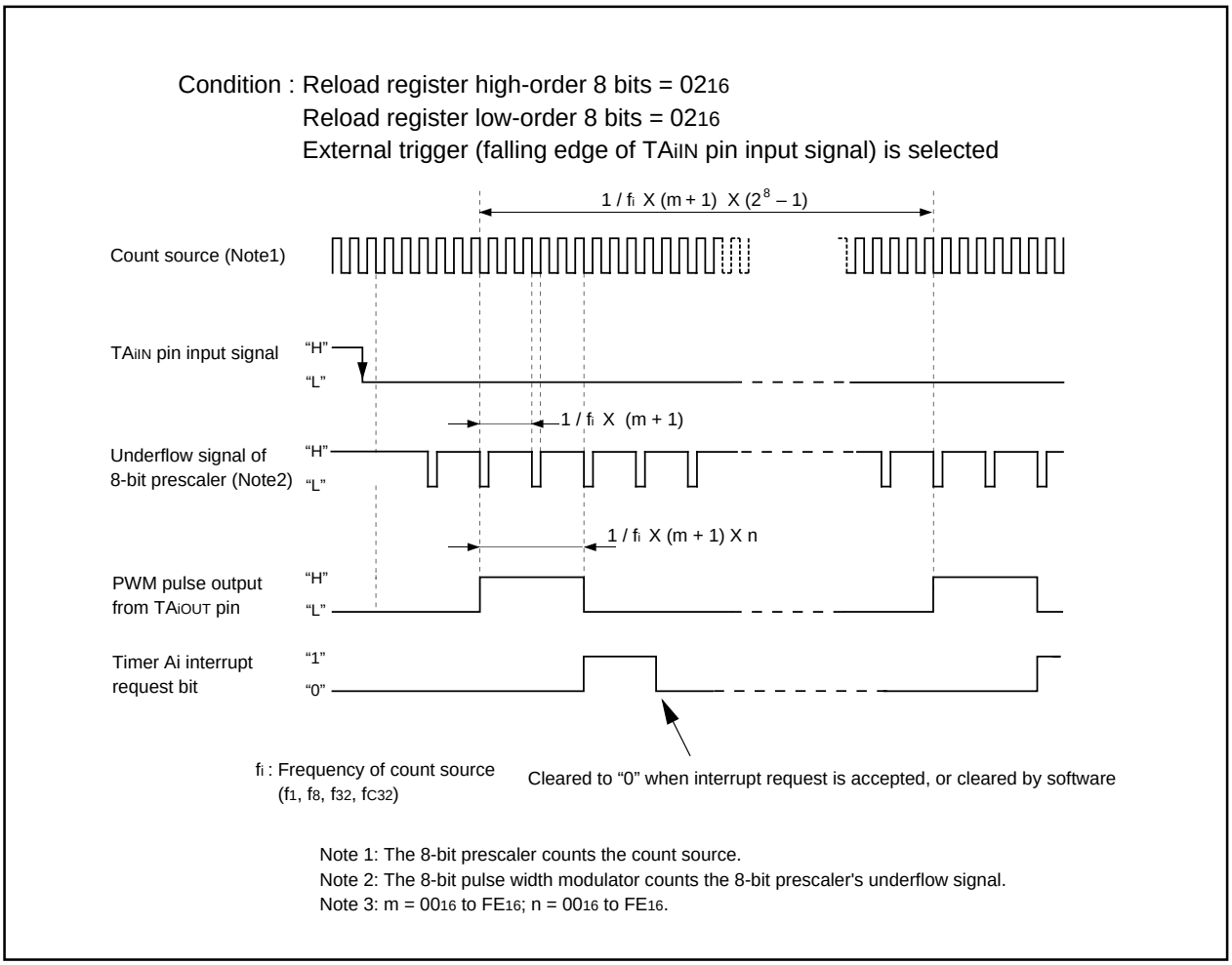


Fig.FB-13 Example of how an 8-bit pulse width modulator operates

Timer B

Fig.FB-14 shows the block diagram of timer B. Fig.FB-15 and FB-16 show the timer B-related registers.

Use the timer Bi mode register (i = 0 to 5) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

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

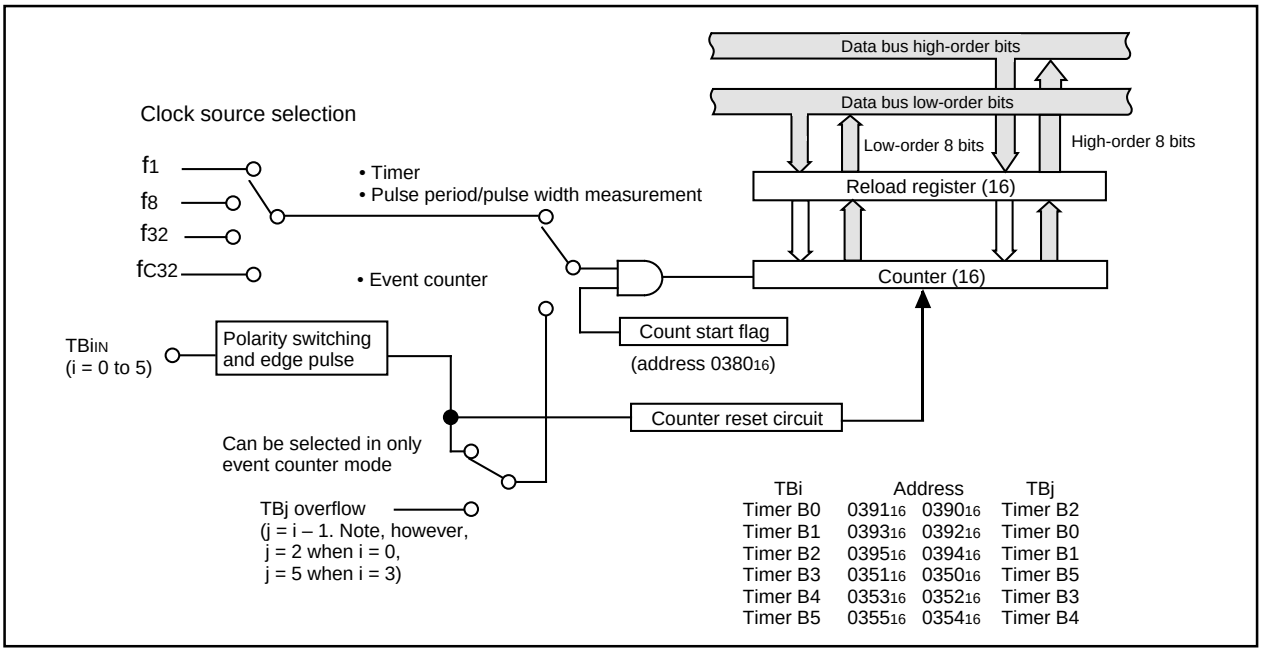


Fig.FB-14 Block diagram of timer B

Timer Bi mode register

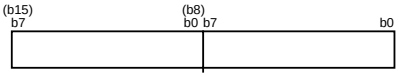
Symbol	Address	When reset
TBiMR(i = 0 to 5)	039B16 to 039D16	00XX00002
	035B16 to 035D16	00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1) × (Note 2)	○ ×
MR3			○	×
TCK0	Count source selection bits (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

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

Timer Bi register (Note)

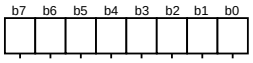


Symbol	Address	When reset
TB0	0391₁₆, 0390₁₆	Indeterminate
TB1	0393₁₆, 0392₁₆	Indeterminate
TB2	0395₁₆, 0394₁₆	Indeterminate
TB3	0351₁₆, 0350₁₆	Indeterminate
TB4	0353₁₆, 0352₁₆	Indeterminate
TB5	0355₁₆, 0354₁₆	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000₁₆ to FFFF₁₆	O	O
• Event counter mode Counts external pulses input or a timer overflow	0000₁₆ to FFFF₁₆	O	O
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	O	×

Note: Read and write data in 16-bit units.

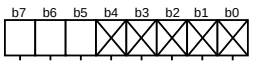
Count start flag



Symbol	Address	When reset
TABSR	0380₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	O	O
TA1S	Timer A1 count start flag		O	O
TA2S	Timer A2 count start flag		O	O
TA3S	Timer A3 count start flag		O	O
TA4S	Timer A4 count start flag		O	O
TB0S	Timer B0 count start flag		O	O
TB1S	Timer B1 count start flag		O	O
TB2S	Timer B2 count start flag		O	O

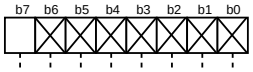
Timer B3, 4, 5 count start flag



Symbol	Address	When reset
TBSR	0340₁₆	000XXXXX₂

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	O	O
TB4S	Timer B4 count start flag		O	O
TB5S	Timer B5 count start flag		O	O

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	0381₁₆	0XXXXXXX₂

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	O	O

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

(1) Timer mode

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

Table.FB-6 Timer specifications in timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	•Counts down •When the timer underflows, it reloads the reload register contents and then continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBi_{IN} pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	•When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol	Address	When reset
TBiMR(i=0 to 5)	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be “0” or “1”		○	○
MR1			○	○
MR2	0 (Fixed to “0” in timer mode ; i = 0, 3)		○ (Note 1)	○
	Nothing is assigned (i = 1, 2, 4, 5). In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. In an attempt to write to this bit, write “0”. The value, if read in timer mode, turns out to be indeterminate.		○	×
TCK0	Count source selection bits	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

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

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table.FB-7) Fig.FB-18 shows the timer Bi mode register in event counter mode.

Table.FB-7 Timer specifications in event counter mode

Item	Specification
Count source	•External signals input to TBiIN pin •Effective edge of count source can be a rising edge, a falling edge, or both edges as selected by software
Count operation	•Counts down •When the timer underflows, it reloads the reload register contents and then continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	•When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter •When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol	Address	When reset
TBiMR(i=0 to 5)	039B16 to 039D16	00XX00002
	035B16 to 035D16	00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity selection bits (Note 1)	b3 b2 0 0 : Counts external signal's falling edges	○	○
MR1		0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Inhibited	○	○
MR2	0 (Fixed to "0" in event counter mode; i = 0, 3)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2, 4, 5). In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		× (Note 3)	×
MR3	Invalid in event counter mode. In an attempt to write to this bit, write "0". The value, if read in event counter mode, turns out to be indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be "0" or "1".		○	○
TCK1	Event clock selection	0 : Input from TBiIN pin (Note 4) 1 : TBj overflow (j = i – 1; however, j = 2 when i = 0, j = 5 when i = 3)	○	○

Note 1: Valid only when input from the TBiIN pin is selected as the event clock. If timer's overflow is selected, this bit can be "0" or "1".
Note 2: Timer B0, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.
Note 4: Set the corresponding port direction register to "0".

Fig.FB-18 Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

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

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

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	•Up count •At measurement pulse's effective edge, after the count value is transferred to reload register, it is cleared to "0000_{16}" and then continues counting.
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	•When measurement pulse's effective edge is input (Note 1) •When an overflow occurs. (Simultaneously, the timer Bi overflow flag becomes "1". The timer Bi overflow flag becomes "0" when the count start flag is "1" and a value is written to the timer Bi mode register.)
TBi_{IN} pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.
Note 2: After count starts, the value read out from the timer Bi register is indeterminate until the second effective edge is input .

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol	Address	When reset
TBiMR(i=0 to 5)	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode selection bits	b1 b0 1 0 : Pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode selection bits	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
MR2	0 (Fixed to "0" in pulse period/pulse width measurement mode; i = 0, 3)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2, 4, 5). In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		× (Note 3)	×
MR3	Timer Bi overflow flag (Note 1)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
TCK0	Count source selection bits	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: The timer Bi overflow flag becomes "0" when the count start flag is "1" and a value is written to the timer Bi mode register. This flag cannot be set to "1" by software.
Note 2: Timer B0, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.

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

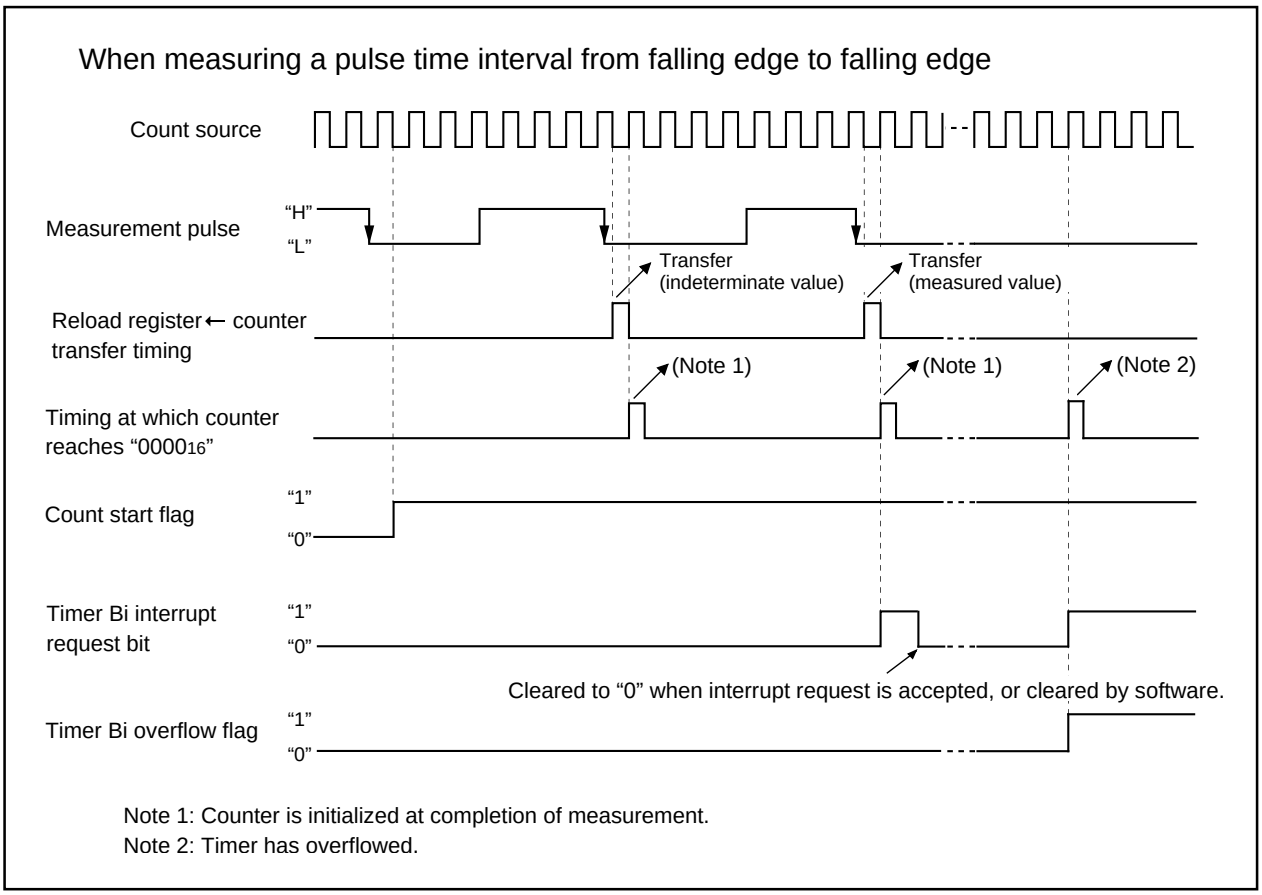


Fig.FB-20 Operation timing when measuring a pulse period

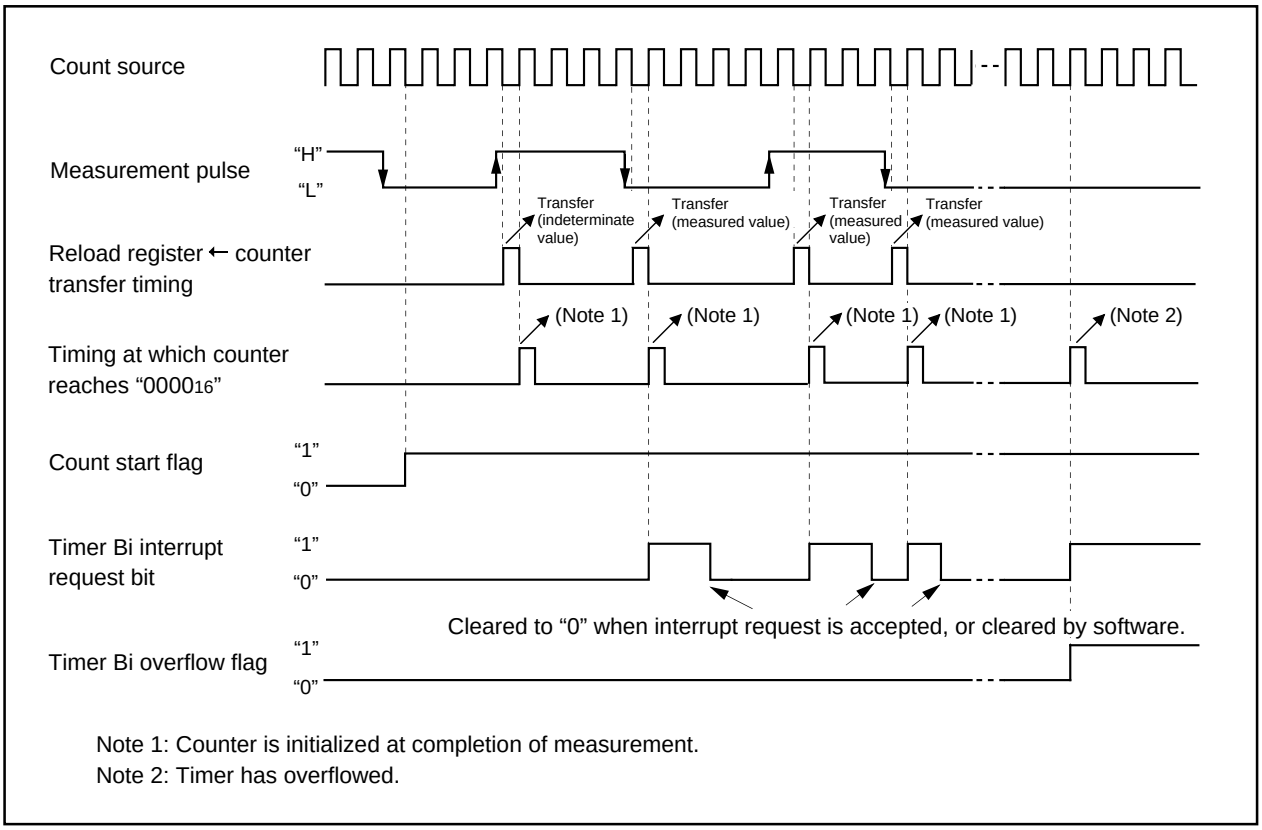


Fig.FB-21 Operation timing when measuring a pulse width

Serial I/O

Serial I/O is configured as five channels: UART0, UART1, UART2, S I/O3 and S I/O4.

UART0 to 2

Each of UART0 - UART2 has an exclusive timer to generate a transfer clock, operating independently from each other.

Fig.GA-1 shows the block diagram of UART0, UART1 and UART2. Fig.GA-2 and GA-3 show the block diagram of the transmit/receive unit.

UARTi (i = 0 to 2) has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode selection bits (bits 0 to 2 at addresses $03A0_{16}$, $03A8_{16}$ and 0378_{16}) determine whether UARTi is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0, UART1 and UART2 have almost the same functions. UART0 through UART2 are almost equal in their functions with minor exceptions. UART2, in particular, is compliant with the SIM interface with some extra settings added in clock-asynchronous serial I/O mode (Note). It also has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

Table.GA-1 shows the comparison of functions of UART0 through UART2, and Fig.GA-4 to GA-8 show the registers related to UARTi.

Note: SIM : Subscriber Identity Module

Table.GA-1 Comparison of functions of UART0 through UART2

Function	UART0	UART1	UART2
CLK polarity selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 2)
Continuous receive mode selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
Transfer clock output from multiple pins selection	Impossible	Possible (Note 1)	Impossible
Separate $\overline{CTS}/\overline{RTS}$ pins	Possible	Impossible	Impossible
Serial data logic switch	Impossible	Impossible	Possible (Note 4)
Sleep mode selection	Possible (Note 3)	Possible (Note 3)	Impossible
TxD, RxD I/O polarity switch	Impossible	Impossible	Possible
TxD, RxD port output format	CMOS output	CMOS output	N-channel open-drain output
Parity error signal output	Impossible	Impossible	Possible (Note 4)
Bus collision detection	Impossible	Impossible	Possible

Note 1: Only in clock synchronous serial I/O mode.
Note 2: Only in clock synchronous serial I/O mode and 8-bit UART mode.
Note 3: Only in UART mode.
Note 4: Can be used for SIM interface.

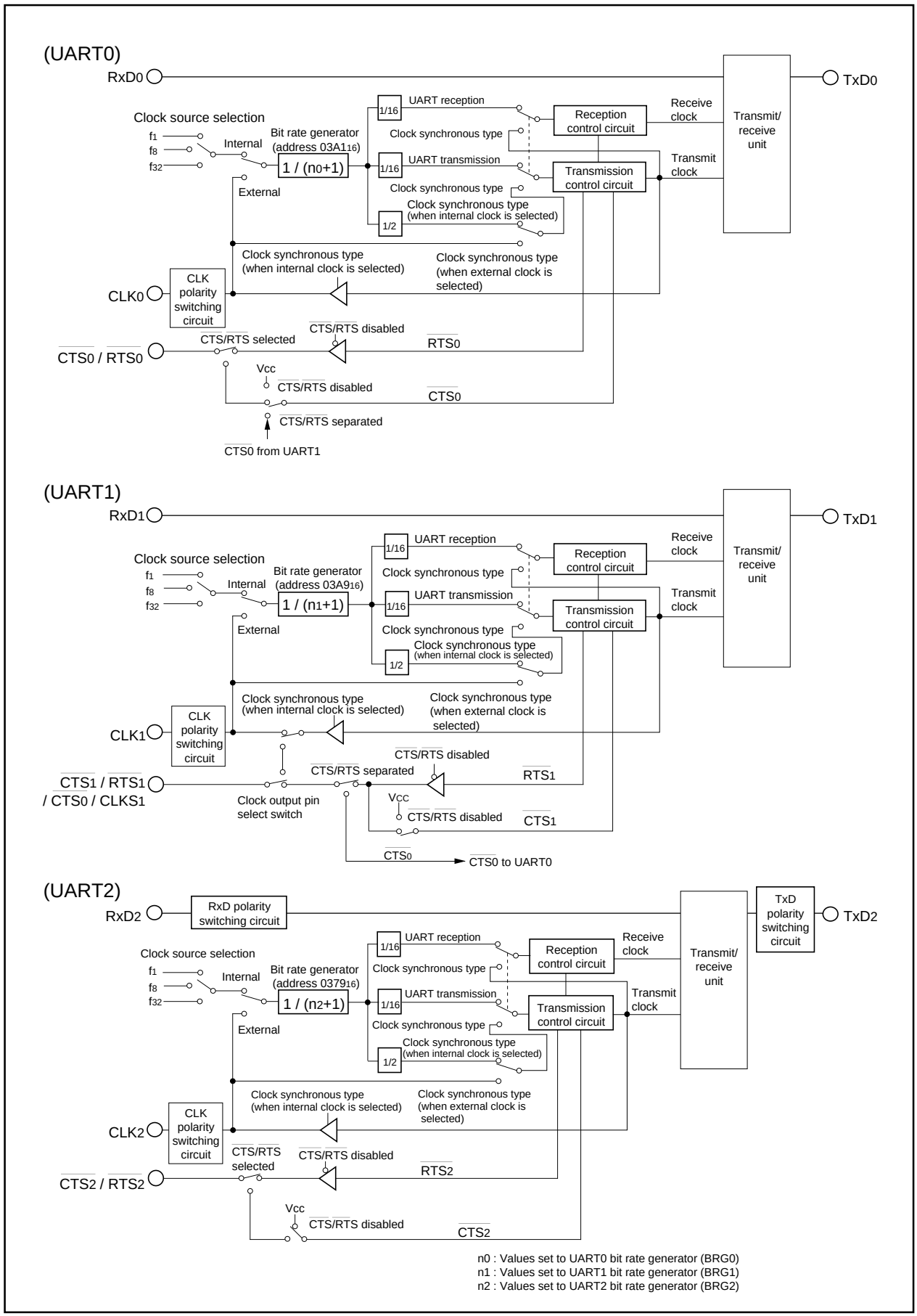


Fig.GA-1 Block diagram of UARTi (i = 0 to 2)

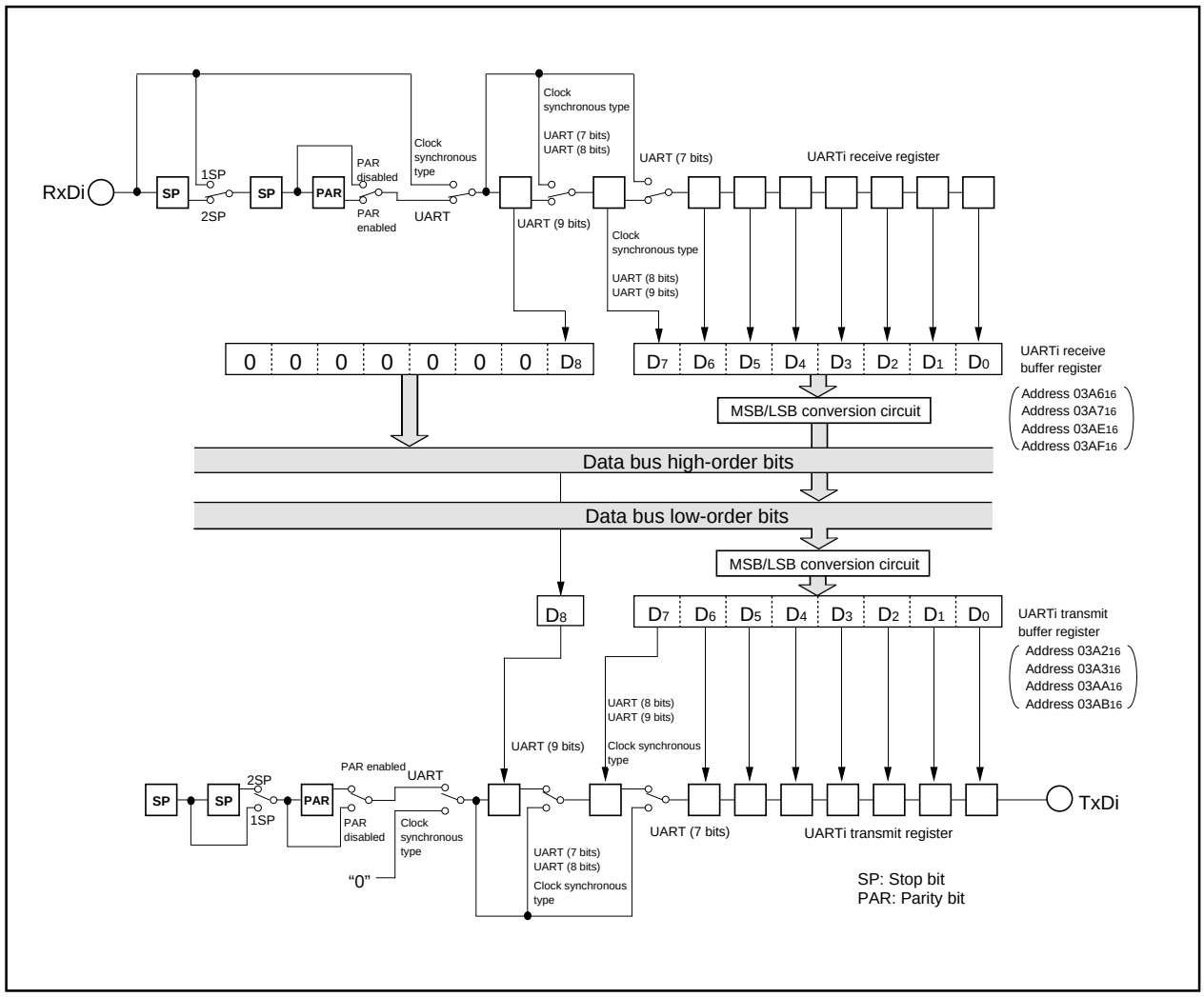


Fig.GA-2 Block diagram of UARTi (i = 0, 1) transmit/receive unit

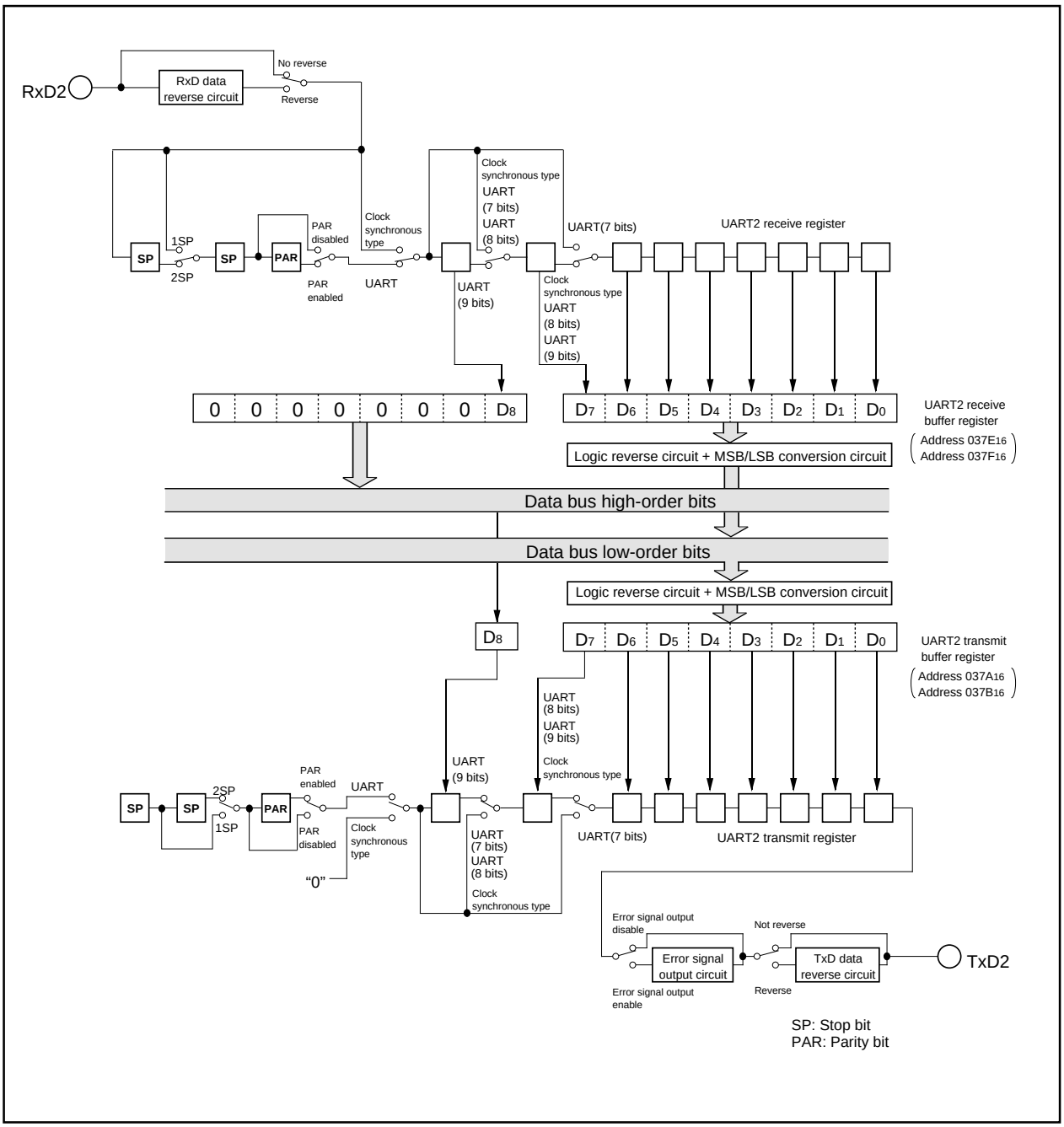
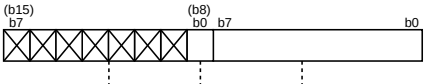


Fig.GA-3 Block diagram of UART2 transmit/receive unit

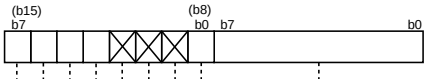
UARTi transmit buffer register



Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U2TB	$037B_{16}$, $037A_{16}$	Indeterminate

Function	R	W
Transmit data	×	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turn out to be indeterminate.	—	—

UARTi receive buffer register

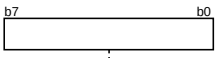


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U2RB	$037F_{16}$, $037E_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	———	Receive data	Receive data	○	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.				—	—
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note 1)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note 1)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note 1)	Invalid	0 : No error 1 : Error found	○	×

Note 1: Bits 15 through 12 are set to “0” when the serial I/O mode selection bits (bits 2 to 0 at addresses $03A0_{16}$, $03A8_{16}$ and 0378_{16}) are set to “000_2” or the receive enable bit is set to “0”.
(Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$, $03AE_{16}$ and $037E_{16}$) is read out.

UARTi bit rate generator

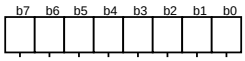


Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate
U2BRG	0379_{16}	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	×	○

Fig.GA-4 Serial I/O-related registers (1)

UARTi transmit/receive mode register

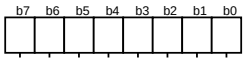


Symbol: UiMR(i=0,1) | Address: $03A0_{16}$, $03A8_{16}$ | When reset: 00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode selection bits	Must be fixed to 001	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1		b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited		○	○
SMD2				○	○
CKDIR	Internal/external clock selection bit	0 : Internal clock 1 : External clock (Note 1)	0 : Internal clock 1 : External clock (Note 1)	○	○
STPS	Stop bit length selection bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity selection bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep selection bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note 1: Set the corresponding port direction register to "0".

UART2 transmit/receive mode register



Symbol: U2MR | Address: 0378_{16} | When reset: 00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode selection bits	Must be fixed to 001	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1		b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited		○	○
SMD2				○	○
CKDIR	Internal/external clock selection bit	0 : Internal clock 1 : External clock (Note 1)	Must always be "0"	○	○
STPS	Stop bit length selection bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity selection bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
IOPOL	TxD, RxD I/O polarity switching bit	0 : Not reverse 1 : Reverse Usually set to “0”	0 : Not reverse 1 : Reverse Usually set to “0”	○	○

Note 1: Set the corresponding port direction register to "0".

Fig.GA-5 Serial I/O-related registers (2)

UARTi transmit/receive control register 0

Symbol: UiC0(i=0,1) — Address: 03A4$_{16}$, 03AC$_{16}$ — When reset: 08$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source selection bits	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{CTS}$/$\overline{RTS}$ function selection bit	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}$/$\overline{RTS}$ disable bit	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (Pins function as programmable I/O port)	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (Pins function as programmable I/O port)	○	○
NCH	Data output selection bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity selection bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be "0"	○	○
UFORM	Transfer format selection bit	0 : LSB first 1 : MSB first	Must always be "0"	○	○

Note 1: Set the corresponding port direction register to "0".
Note 2: The settings of the corresponding port register and port direction register are invalid.

UART2 transmit/receive control register 0

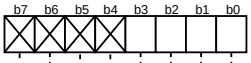
Symbol: U2C0 — Address: 037C$_{16}$ — When reset: 08$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source selection bits	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{CTS}$/$\overline{RTS}$ function selection bit	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	Valid when bit 4 = "0" 0 : $\overline{CTS}$ function is selected (Note 1) 1 : RTS function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}$/$\overline{RTS}$ disable bit	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (Pins function programmable I/O port)	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : CTS/RTS function disabled (Pins function programmable I/O port)	○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".				—	—
CKPOL	CLK polarity selection bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be "0"	○	○
UFORM	Transfer format selection bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

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

Fig.GA-6 Serial I/O-related registers (3)

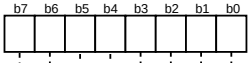
UARTi transmit/receive control register 1



Symbol: UiC1(i=0,1) | Address: 03A5$_{16}$,03AD$_{16}$ | When reset: 02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	O	O
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	O	X
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	O	O
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	O	X
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.				–	–

UART2 transmit/receive control register 1

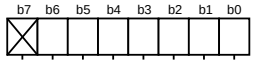


Symbol: U2C1 | Address: 037D$_{16}$ | When reset: 02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	O	O
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	O	X
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	O	O
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	O	X
U2IRS	UART2 transmit interrupt factor selection bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	O	O
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Invalid	O	O
U2LCH	Data logic selection bit	0 : Not reverse 1 : Reverse	0 : Not reverse 1 : Reverse	O	O
U2ERE	Error signal output enable bit	Must be fixed to “0”	0 : Output disabled 1 : Output enabled	O	O

Fig.GA-7 Serial I/O-related registers (4)

UART transmit/receive control register 2

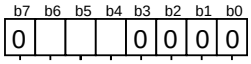


Symbol	Address	When reset
UCON	03B0$_{16}$	X0000000$_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt factor selection bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt factor selection bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Invalid	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Invalid	○	○
CLKMD0	CLK/CLKS selection bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS selection bit 1 (Note)	0 : Normal mode (CLK output is CLK1 only) 1 : Transfer clock output from multiple pins function selected	Must always be “0”	○	○
RCSP	Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ shared pin 1 : CTS/RTS separated	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ shared pin 1 : CTS/RTS separated	○	○
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.				–	–

Note: When using multiple pins to output the transfer clock, the following requirements must be met:
• UART1 internal/external clock selection bit (bit 3 at address 03A8$_{16}$) = “0”.

UART2 special mode register



Symbol	Address	When reset
U2SMR	0377$_{16}$	00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
Reserved bits		Must always be “0”		–	–
ABSCS	Bus collision detect sampling clock selection bit	Must always be “0”	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	○	○
ACSE	Auto clear function selection bit of transmit enable bit	Must always be “0”	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition selection bit	Must always be “0”	0 : Ordinary 1 : Falling edge of RxD2	○	○
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be “0”.				–	–

Fig.GA-8 Serial I/O-related registers (5)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables.GA-2 and GA-3 list the specifications of the clock synchronous serial I/O mode. Fig.GA-9 shows the UARTi transmit/ receive mode register.

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

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = “0”) : fi/ 2(n+1) (Note 1) fi = f_1, f_8, f_{32} • When external clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 0378$_{16}$ = “1”) : Input from CLKi pin
Transmission/reception control	• Selecting from $\overline{\text{CTS}}$ function/$\overline{\text{RTS}}$ function/Disable $\overline{\text{CTS}}$, $\overline{\text{RTS}}$ function
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = “1” – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = “0” – When $\overline{\text{CTS}}$ function selected, $\overline{\text{CTS}}$ input level = “L” • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0”: CLKi input level = “H” – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”: CLKi input level = “L”
Reception start condition	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = “1” – Transmit enable bit (bit 0 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = “1” – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 037D$_{16}$) = “0” • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0”: CLKi input level = “H” – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “1”: CLKi input level = “L”
Interrupt request generation timing	• When transmitting – Transmit interrupt factor selection bits (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 037D$_{16}$) = “0”: At the completion of data transmission from UARTi transfer buffer register to UARTi transmit register – Transmit interrupt factor selection bits (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 037D$_{16}$) = “1”: At the completion of data transmission from UARTi transfer register is completed • When receiving – At the completion of data transferring from UARTi receive register to UARTi receive buffer register
Error detection	• Overrun error (Note 2) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out

Note 1: “n” denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UART bit rate generator.

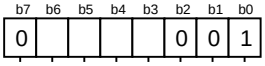
Note 2: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to “1”.

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

Item	Specification
Function selection	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection (UART1) (Note) UART1 transfer clock can be chosen by software to be output from one of the two pins set • Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) (Note) Each of UART0 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins can be assigned to separate pins • Switching serial data logic (UART2) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected. • TxD, RxD I/O polarity switching (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

Note: The transfer clock output from multiple pins and the separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins functions cannot be selected simultaneously.

UARTi transmit/receive mode registers

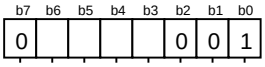


Symbol: UiMR(i=0,1) — Address: 03A0$_{16}$, 03A8$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode selection bits	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock selection bit	0 : Internal clock 1 : External clock (Note 1)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be “0” in clock synchronous serial I/O mode)		○	○

Note 1: The corresponding port direction register should be "0".

UART2 transmit/receive mode register



Symbol: U2MR — Address: 0378$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode selection bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock selection bit	0 : Internal clock 1 : External clock (Note 2)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note 1)	0 : No reverse 1 : Reverse	○	○

Note 1: Usually sent to "0".
Note 2: The corresponding port direction register should be "0".

Fig.GA-9 UARTi transmit/receive mode register in clock synchronous serial I/O mode

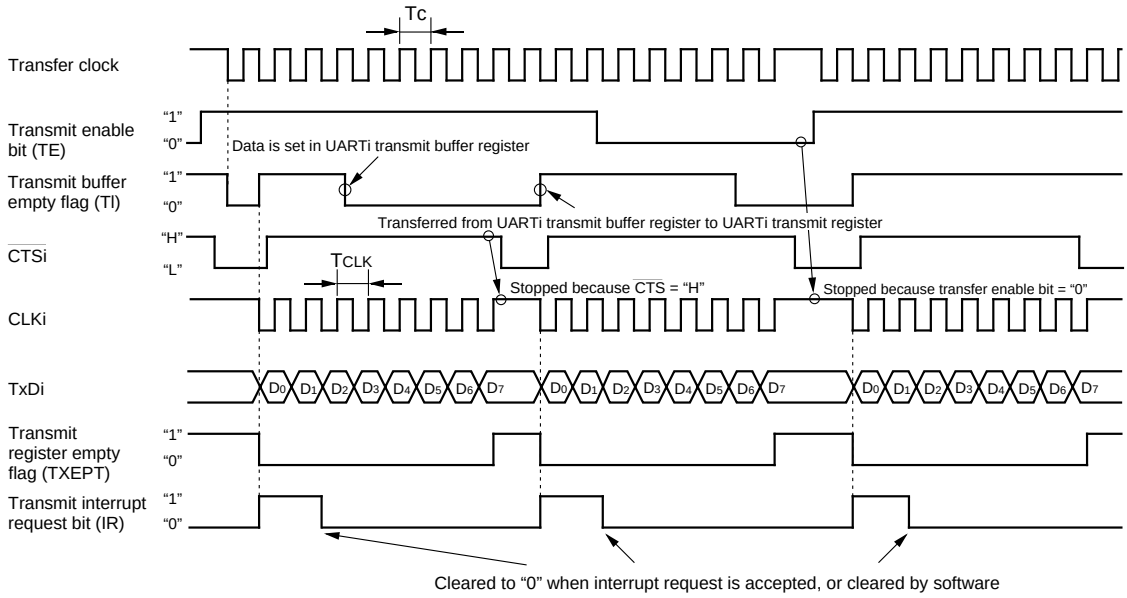
Table.GA-4 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions that the transfer clock output from multiple pins and the separation of $\overline{CTS}/\overline{RTS}$ pins are not selected. Note that for a period from when the UARTi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

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

(The function that the transfer clock output from multiple pin is not selected. The function that separates CTS/RTS pins is not selected.)

Pin name	Function	Method of selection
TxDi ($P6_3$,$P6_7$,$P7_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi ($P6_2$,$P6_6$,$P7_1$)	Serial data input	Port $P6_2$, $P6_6$ and $P7_1$ direction register (bits 2 and 6 at address $03EE_{16}$, bit 1 at address $03EF_{16}$)= "0" (Can be used as an input port when performing transmission only)
CLKi ($P6_1$,$P6_5$,$P7_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = "1" Port $P6_1$, $P6_5$ and $P7_2$ direction register (bits 1 and 5 at address $03EE_{16}$, bit 2 at address $03EF_{16}$) = "0"
$\overline{CTS}i/\overline{RTS}i$ ($P6_0$,$P6_4$,$P7_3$)	$\overline{CTS}$ input	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) ="0" $\overline{CTS}/\overline{RTS}$ function selection bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "0" The corresponding port direction bit = "0"
	$\overline{RTS}$ output	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "0" $\overline{CTS}/\overline{RTS}$ function selection bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "1"
	Programmable I/O port	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "1"

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

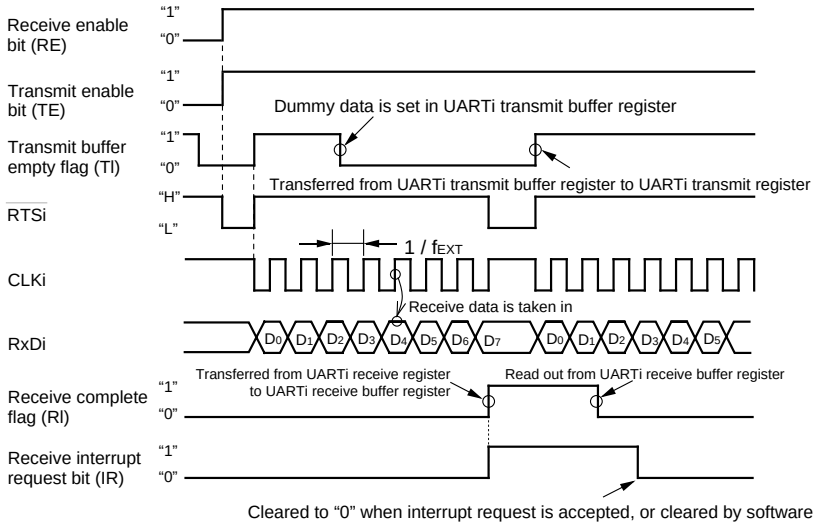


Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- Internal clock is selected.
- CTS function is selected.
- CLK polarity selection bit = "0".
- Transmit interrupt factor selection bit = "0".

Tc = TCLK = 2(n + 1) / fi
fi: frequency of BRGi count source (f1, f8, f32)
n: value set to BRGi

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



Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- RTS function is selected.
- CLK polarity selection bit = "0".

fEXT: frequency of external clock

The following conditions should be matched when the input level of CLKi pin is "H" before the data reception.
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Fig.GA-10 Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity selection function

As shown in Fig.GA-11, the CLK polarity selection bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) allows to select the polarity of the transfer clock.

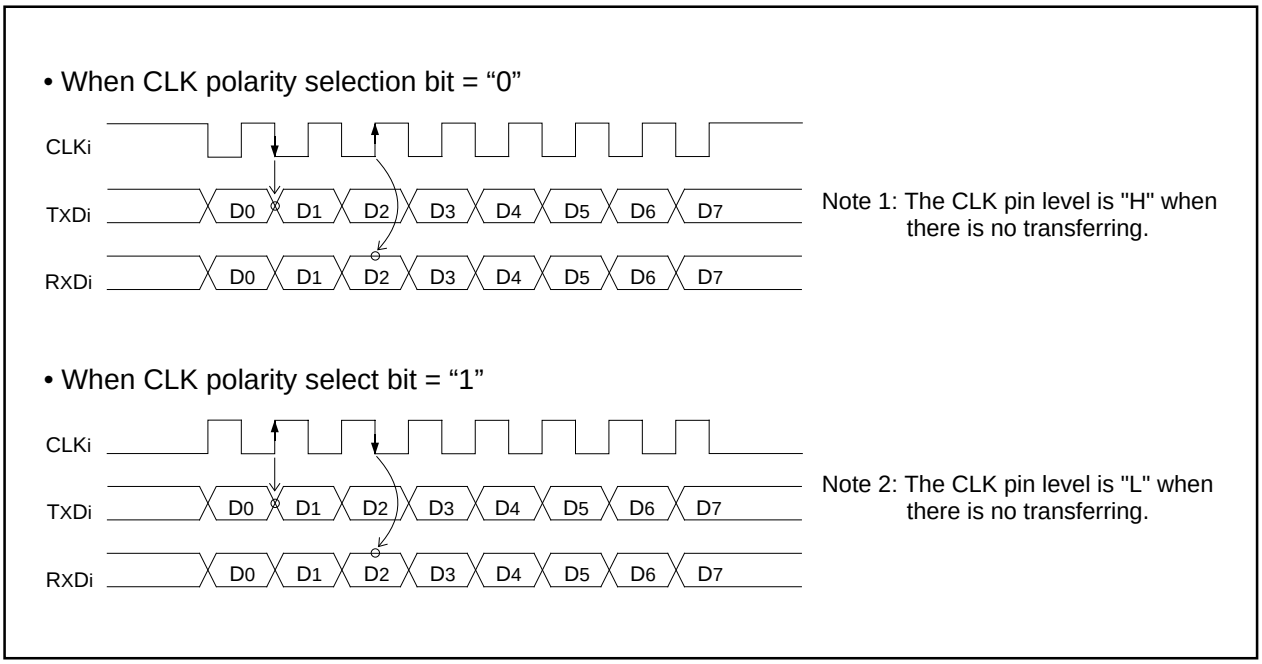


Fig.GA-11 Polarity of transfer clock

(b) LSB first/MSB first selection function

As shown in Fig.GA-12, when the transfer format selection bit (bit 7 at addresses 03A4$_{16}$, 03AC$_{16}$, 037C$_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.

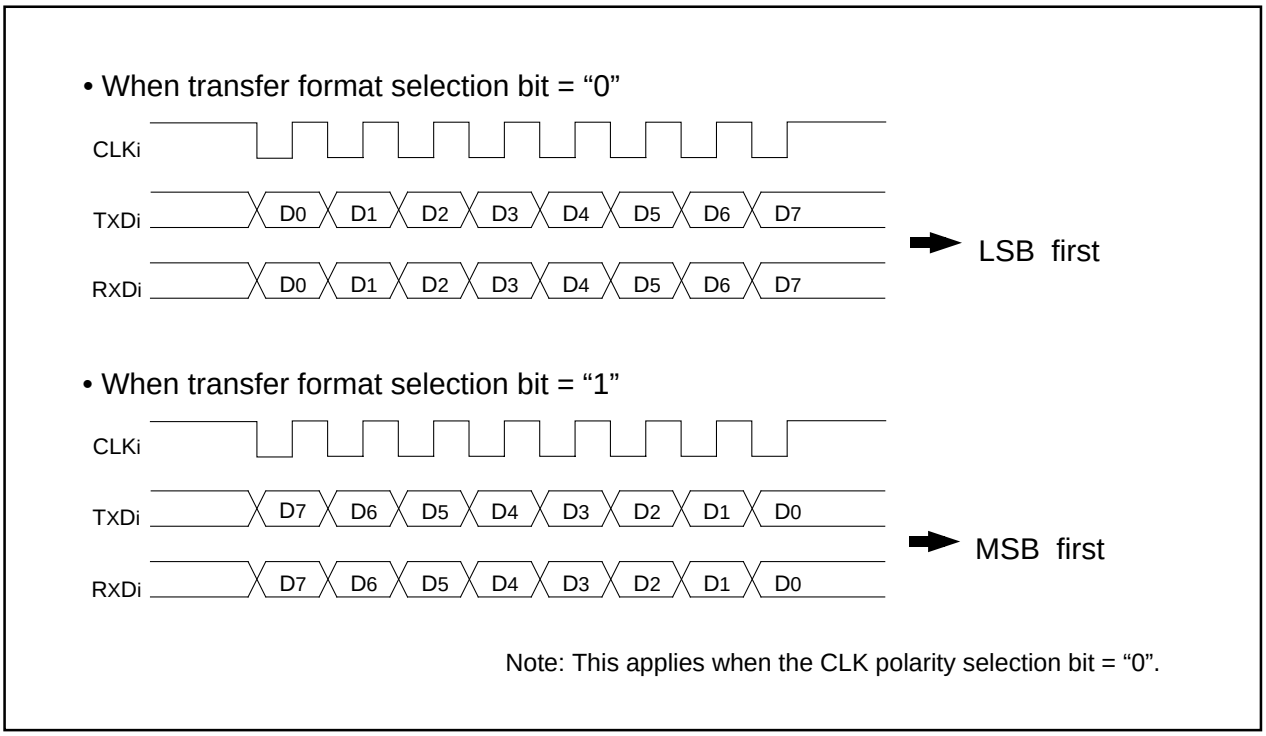


Fig.GA-12 Transfer format

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

This function allows to set two transfer clock output pins and chooses one to output a clock by the setting of CLK and CLKS selection bits (bits 4 and 5 at address $03B0_{16}$). (See Fig.GA-13) The function is valid only when the UART1 internal clock is selected. Note that when this function is selected, UART1 $\overline{CTS}/\overline{RTS}$ function cannot be used.

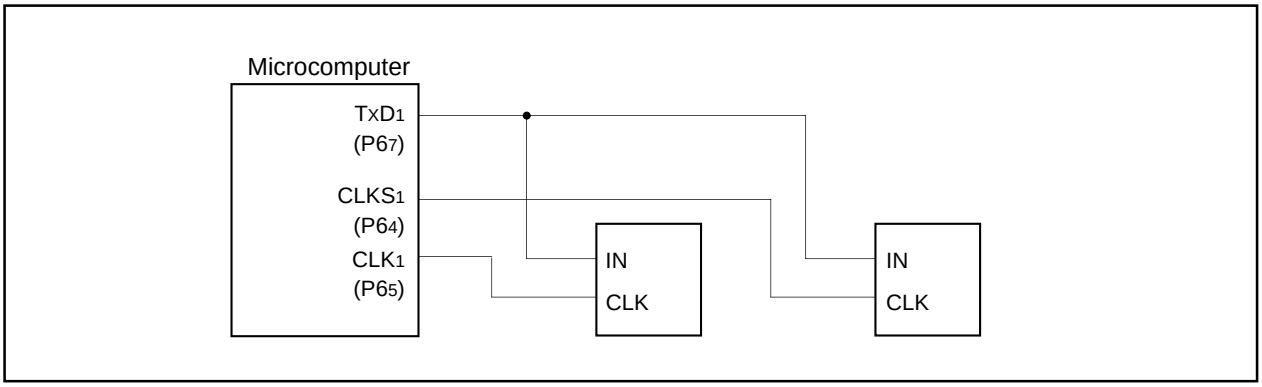


Fig.GA-13 The sample of transfer clock output from the multiple pins function

(d) Continuous receive mode

If the continuous receive mode enable bits (bits 2 and 3 at address $03B0_{16}$, bit 5 at address $037D_{16}$) are set to "1", the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

(e) Separate $\overline{CTS}/\overline{RTS}$ pins function (UART0)

This function works the same way as in the clock asynchronous serial I/O (UART) mode. The method of setting and the input/output pin functions are both the same, so refer to the selection function in the next section, "(2) Clock asynchronous serial I/O (UART) mode." Note that this function is invalid if the transfer clock output from the multiple pins function is selected.

(f) Serial data logic switch function (UART2)

When the data logic selection bit (bit6 at address $037D_{16}$) = "1", the data writing to transmit buffer register or reading from receive buffer register, are reversed. Fig.GA-14 shows the example of serial data logic switch timing.

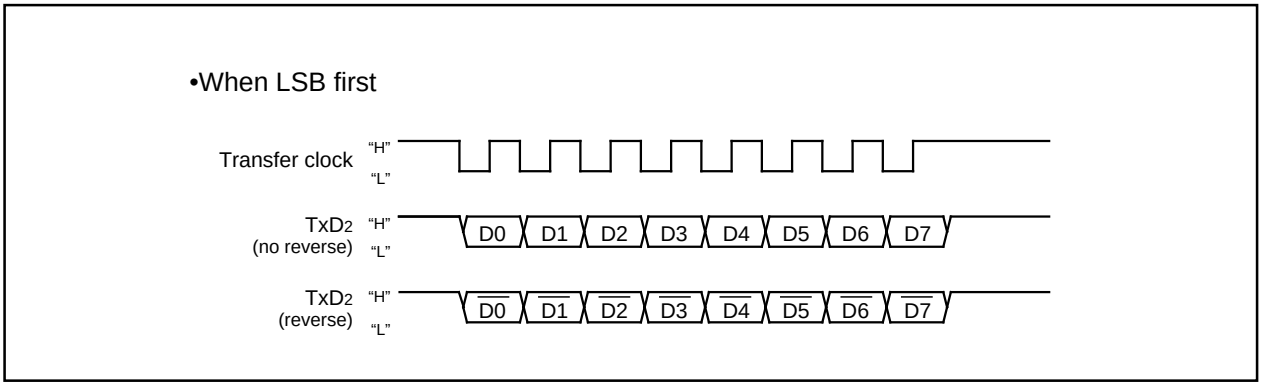


Fig.GA-14 Serial data logic switch timing

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

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables.GA-5 and GA-6 list the specifications of the UART mode. Fig.GA-15 shows the UARTi transmit/receive mode register.

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

Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$, 0378_{16} = “0”) : fi/16(n+1) (Note 1) fi = f_1, f_8, f_{32} • When external clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$ =“1”) : f_{EXT}/16(n+1)(Note 1) (Note 2) • Do not select the external clock in UART2.
Transmission/reception control	• Selecting from $\overline{CTS}$ function/ $\overline{RTS}$ function/ Disable $\overline{CTS}$, $\overline{RTS}$ function
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = “1” - Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = “0” - When $\overline{CTS}$ function is selected $\overline{CTS}$ input level = “L”
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $03A5_{16}$, $03AD_{16}$, $037D_{16}$) = “1” - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt factor selection bits (bits 0,1 at address $03B0_{16}$, bit4 at address $037D_{16}$) = “0”: At the completion of data transferring from UARTi transfer buffer register to UARTi transmit register - Transmit interrupt factor selection bits (bits 0, 1 at address $03B0_{16}$, bit4 at address $037D_{16}$) = “1”: At the completion of data transmission from UARTi transfer register • When receiving - At the completion of data transferring from UARTi receive register to UARTi receive buffer register
Error detection	• Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out • Framing error This error occurs when the number set for stop bits is not detected • Parity error This error occurs in the case that parity is enabled and the number of "1" in parity bit and character bits does not match the number of "1" in parity odd/ even setting. • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered

Note 1: ‘n’ denotes the value 00_{16} to FF_{16} that is set to the UARTi bit rate register.

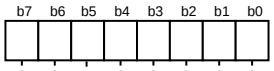
Note 2: f_{EXT} is input from the CLKi pin.

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

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

Item	Specification
Function selection	• Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) Each of UART0 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins can be assigned to separate pins • Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave microcomputers • Serial data logic switch (UART2) This function is reversing logic value of transferring data. Start bit,and stop bit are not reversed. • TxD, RxD I/O polarity switch (UART2) This function is reversing TxD port output and RxD port input. All I/O data level are reversed.

UARTi transmit / receive mode registers

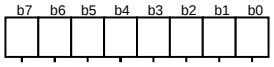


Symbol: UiMR(i=0,1) | Address: $03A0_{16}$, $03A8_{16}$ | When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode selection bits	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	O	O
SMD1			O	O
SMD2			O	O
CKDIR	Internal / external clock selection bit	0 : Internal clock 1 : External clock (Note 1)	O	O
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	O	O
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	O	O
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	O	O
SLEP	Sleep selection bit	0 : Sleep mode deselected 1 : Sleep mode selected	O	O

Note 1: The corresponding port direction register should be "0".

UART2 transmit / receive mode register



Symbol: U2MR | Address: 0378_{16} | When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode selection bits	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	O	O
SMD1			O	O
SMD2			O	O
CKDIR	Internal / external clock selection bit	Must always be "0"	O	O
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	O	O
PRY	Odd / even parity selection bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	O	O
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	O	O
IOPOL	TxD, RxD I/O polarity switching bit (Note)	0 : No reverse 1 : Reverse	O	O

Note: Usually set to “0”.

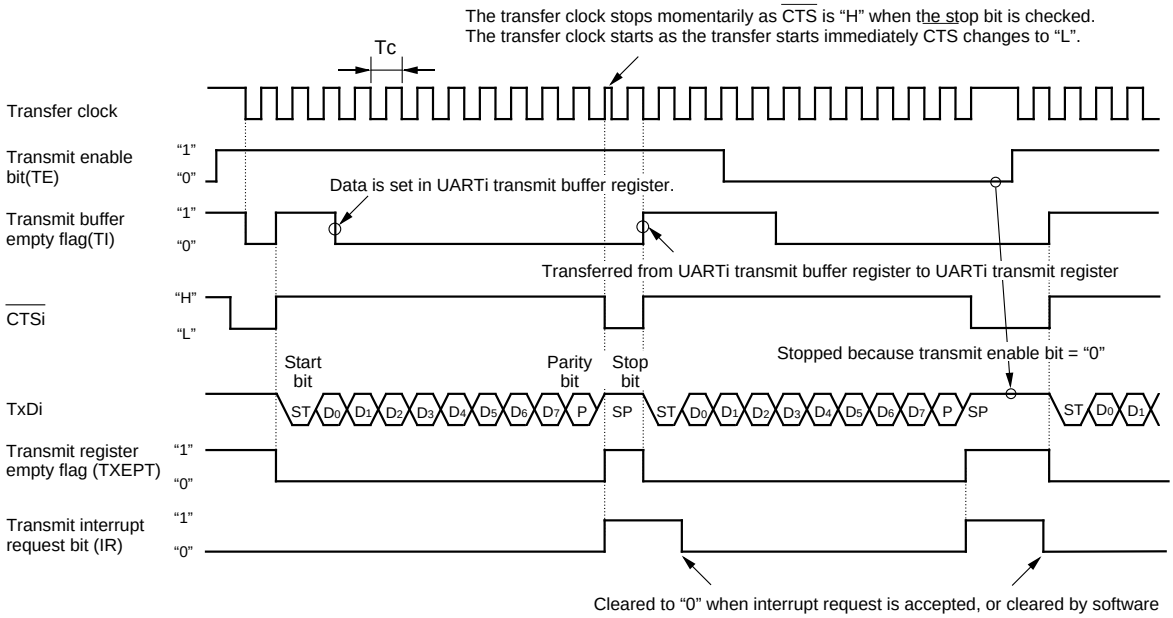
Fig.GA-15 UARTi transmit/receive mode register in UART mode

Table.GA-7 lists the functions of the input/output pins during UART mode. This table shows the pin functions when the separate $\overline{CTS}/\overline{RTS}$ pins function is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table.GA-7 Input/output pin functions in UART mode (when $\overline{CTS/RTS}$ separate function is not selected)

Pin name	Function	Method of selection
TxDi ($P6_3$,$P6_7$,$P7_0$)	Serial data output	
RxDi ($P6_2$,$P6_6$,$P7_1$)	Serial data input	Corresponding port direction register bit = "0".(Can be used as an input port when performing transmission only)
CLKi ($P6_1$,$P6_5$,$P7_2$)	Programmable I/O port	Internal/external clock selection bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, 0378_{16}) = "0"
	Transfer clock input	Internal/external clock selection bit (bit 3 at address $03A0_{16}$, $03A8_{16}$) = "1" Corresponding port direction register bit = "0" (Don't select the external clock for UART2)
$\overline{CTS}$i/$\overline{RTS}$i ($P6_0$,$P6_4$,$P7_3$)	$\overline{CTS}$ input	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) ="0" CTS/RTS function selection bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "0" Corresponding port direction register bit = "0"
	$\overline{RTS}$ output	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "0" CTS/RTS function selection bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "1"
	Programmable I/O port	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $037C_{16}$) = "1"

• Example of transmit timing when transfer data are 8 bits long (parity enabled, one stop bit)

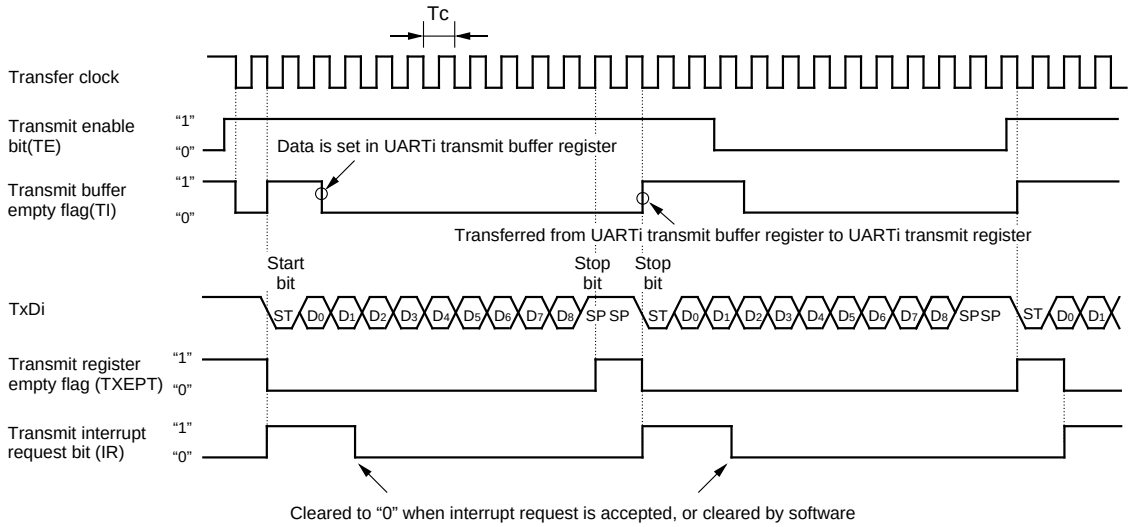


Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- CTS function is selected.
- Transmit interrupt factor selection bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

• Example of transmit timing when transfer data are 9 bits long (parity disabled, two stop bits)



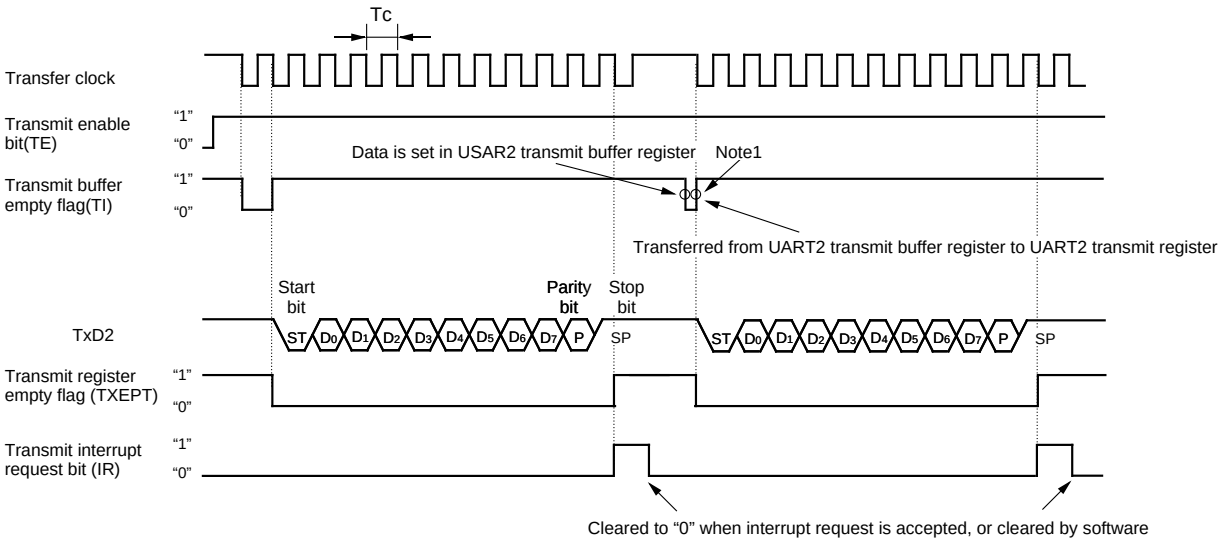
Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is disabled.
- Two stop bits.
- CTS function is disabled.
- Transmit interrupt factor selection bit = "0".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Fig.GA-16 Typical transmit timings in UART mode

• Example of transmit timing when transfer data are 8 bits long (parity enabled, one stop bit)



Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt factor selection bit = "1".

$Tc = 16 (n + 1) / fi$
fi : frequency of BRGi count source (f_1, f_8, f_{32})
n : value set to BRGi

Note1. According to the above timing, the transmission is started by the timing of BRG overflow after writing to the transmit buffer.

Fig.GA-17 Typical transmit timings in UART mode (UART2)

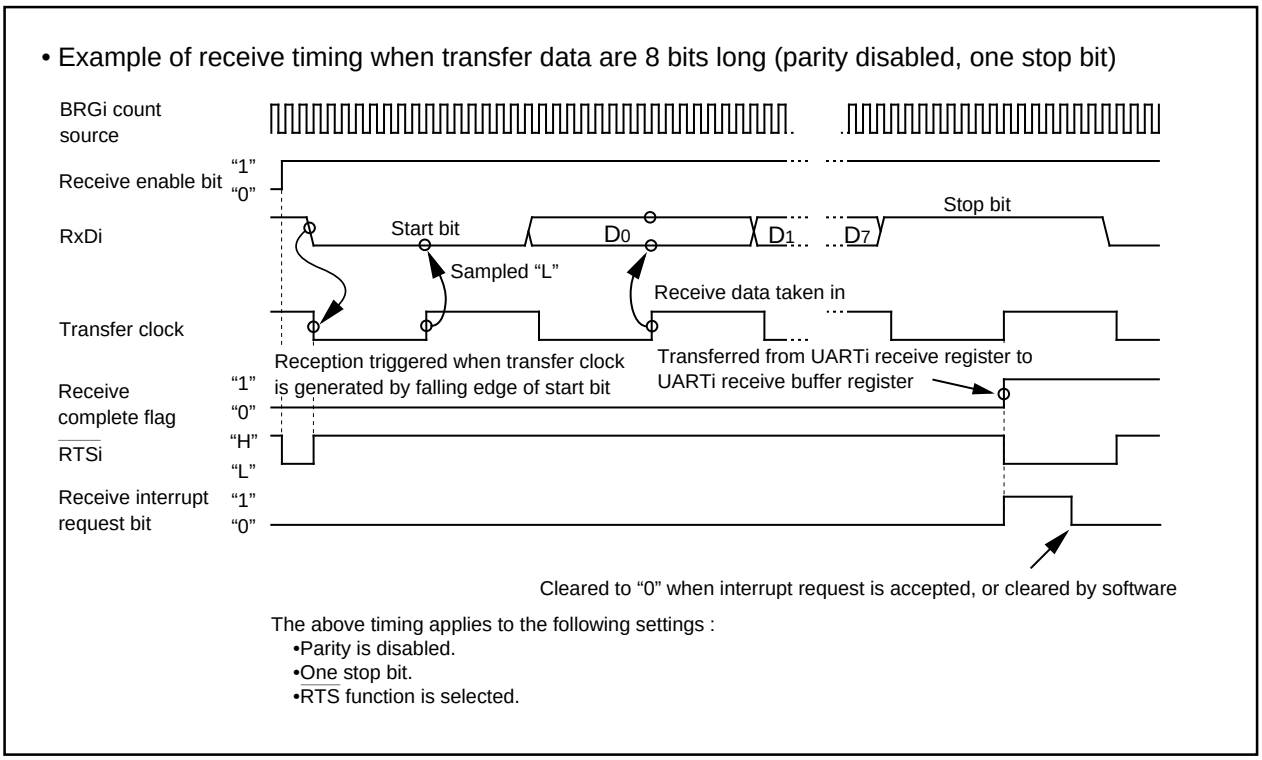


Fig.GA-18 Typical receive timing in UART mode

(a) Separate CTS/RTS pins function (UART0)

Setting the CTS/RTS separate bit (bit 6 of address 03B0$_{16}$) to "1" separates the RTS and CTS signals to different input/out pins.(Fig GA-19). Choosing one from CTS or RTS, by using of the CTS/RTS function selection bit (bit 2 of address 03A4$_{16}$). This function is effective in UART0 only. If the function is used, the user cannot use the CTS/RTS function of UART1. Set "0" both to the CTS/RTS function selection bit (bit 2 of address 03AC$_{16}$) and to the CTS/RTS disable bit (bit 4 of address 03AC$_{16}$).

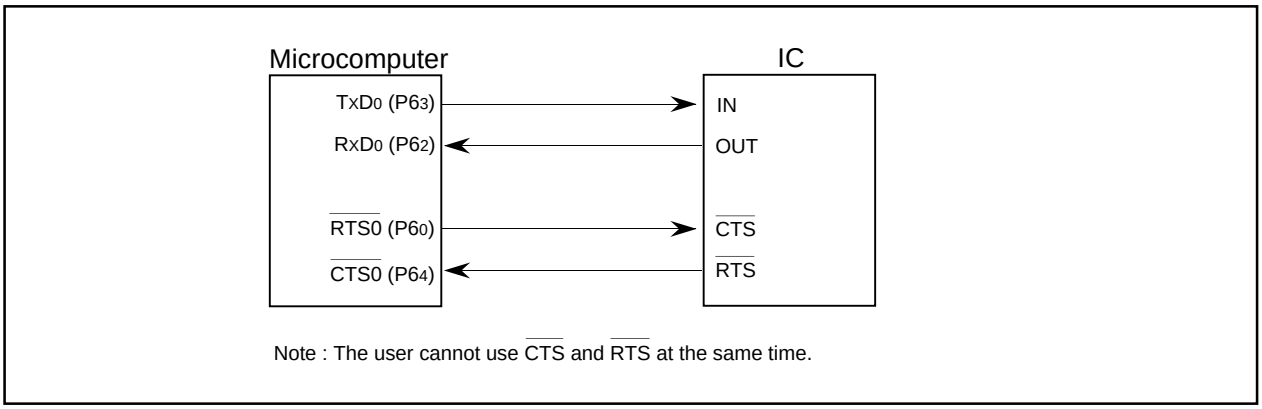


Fig.GA-19 The separate CTS/RTS pins function usage

(b) Sleep mode (UART0, UART1)

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected with UARTi. The sleep mode is selected when the sleep selection bit (bit 7 at addresses 03A0$_{16}$, 03A8$_{16}$) is set to “1” during reception. In this mode, the unit performs receive operation when the MSB of the received data = “1” and does not perform receive operation when the MSB = “0”.

(c) Function for switching serial data logic (UART2)

When the data logic selection bit (bit 6 of address 037D16) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Fig.GA-20 shows the example of timing for switching serial data logic.

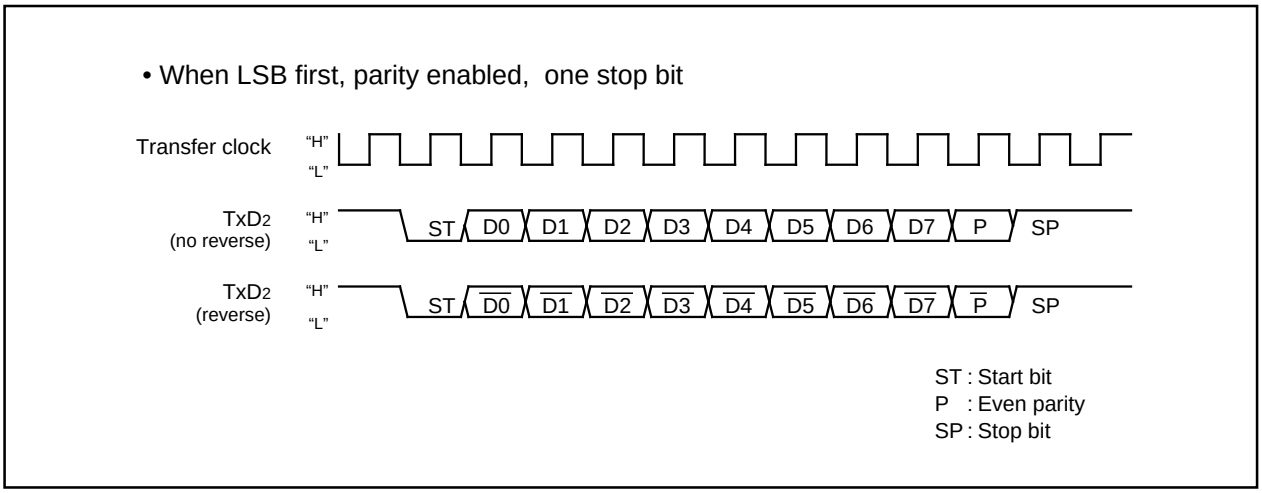


Fig.GA-20 Timing for switching serial data logic

(d) TxD, RxD I/O polarity switching function (UART2)

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

(e) Bus collision detection function (UART2)

This function is to sample the output level of the TXD pin and the input level of the RXD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Fig.GA-21 shows the example of detection timing of a bus collision (in UART mode).

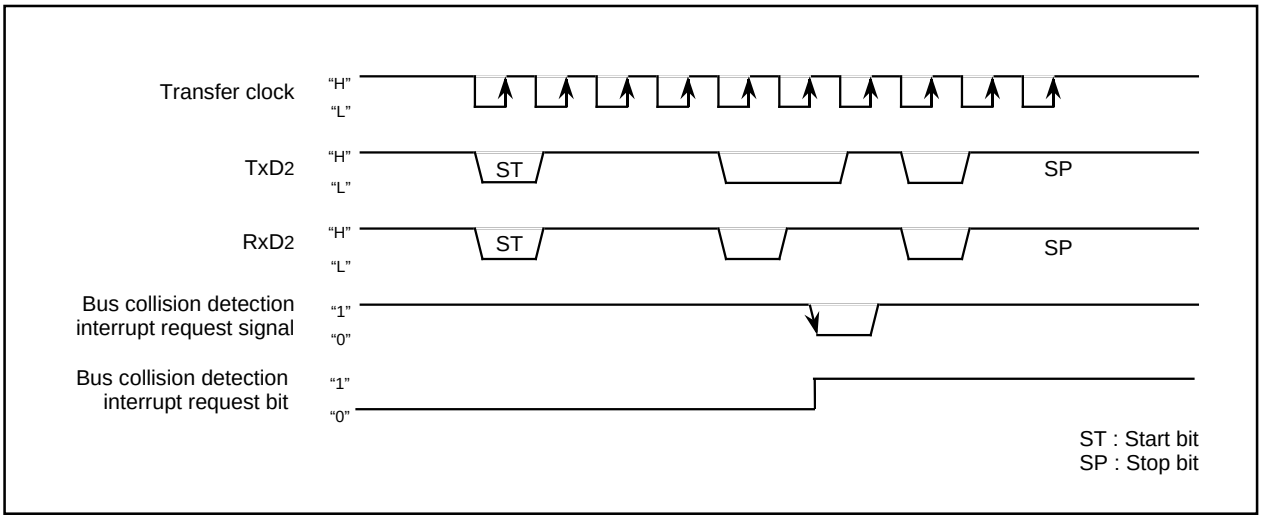


Fig.GA-21 Detection timing of a bus collision (in UART mode)

(3) Clock-asynchronous serial I/O mode (compliant with the SIM interface)

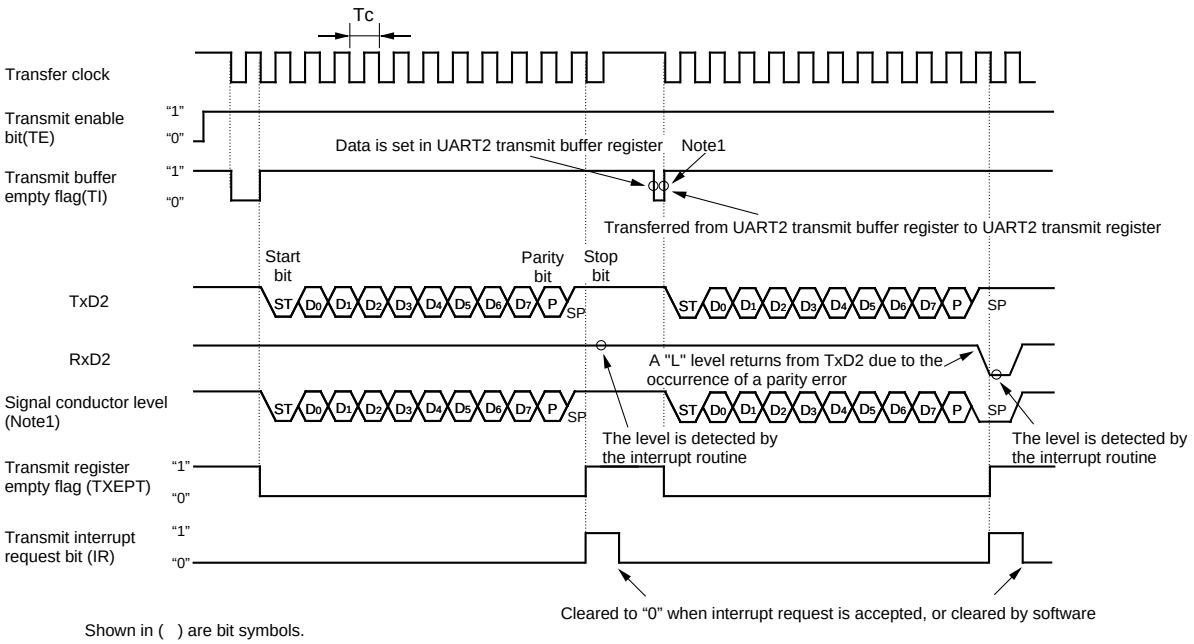
The SIM interface is used for connecting the microcomputer with a memory card IC or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to effect this function. Table.GA-8 shows the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface).

Table.GA-8 Specifications of clock-asynchronous serial I/O mode (compliant with SIM I/F)

Item	Specification
Transfer data format	• Transfer data 8-bit UART mode (bit 2 through bit 0 of address 0378_{16} = “101_2”) • One stop bit (bit 4 of address 0378_{16} = “0”) • With the direct format chosen Set parity to “even” (bit 5 and bit 6 of address 0378_{16} = “1” and “1” respectively) Set data logic to “direct” (bit 6 of address $037D_{16}$ = “0”). Set transfer format to LSB (bit 7 of address $037C_{16}$ = “0”). • With the inverse format chosen Set parity to “odd” (bit 5 and bit 6 of address 0378_{16} = “0” and “1” respectively) Set data logic to “inverse” (bit 6 of address $037D_{16}$ = “1”) Set transfer format to MSB (bit 7 of address $037C_{16}$ = “1”)
Transfer clock	• With the internal clock chosen (bit 3 of address 0378_{16} = “0”) : fi / 16 (n + 1) (Note 1) : fi=f_1, f_8, f_{32} • Don't chose external clock.
Transmission / reception control	• Disable the $\overline{CTS}$ and $\overline{RTS}$ function (bit 4 of address $037C_{16}$ = “1”)
Other settings	• The sleep mode selection function is not available for UART2 • Set transmission interrupt factor to “transmission completed” (bit 4 of address $037D_{16}$ = “1”)
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 of address $037D_{16}$) = “1” - Transmit buffer empty flag (bit 1 of address $037D_{16}$) = “0”
Reception start condition	• To start reception, the following requirements must be met: - Reception enable bit (bit 2 of address $037D_{16}$) = “1” - Detection of a start bit
Interrupt request generation timing	• When transmitting When data transmission from the UART2 transfer register is completed (bit 4 of address $037D_{16}$ = “1”) • When receiving When data transfer from the UART2 receive register to the UART2 receive buffer register is completed
Error detection	• Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 2) • Framing error (see the specifications of clock-asynchronous serial I/O) • Parity error (see the specifications of clock-asynchronous serial I/O) - On the reception side, an “L” level is output from the TxD_2 pin by use of the parity error signal output function (bit 7 of address $037D_{16}$ = “1”) when a parity error is detected - On the transmission side, a parity error is detected by the level of input to the RxD_2 pin when a transmission interrupt occurs • The error sum flag (see the specifications of clock-asynchronous serial I/O)

Note 1: ‘n’ denotes the value 00_{16} to FF_{16} that is set to the UARTi bit rate generator.

Note 2: If an overrun error occurs, the UART2 receive buffer will have the next data written in. Also Note that the UARTi receive interrupt request bit is not set to “1”.

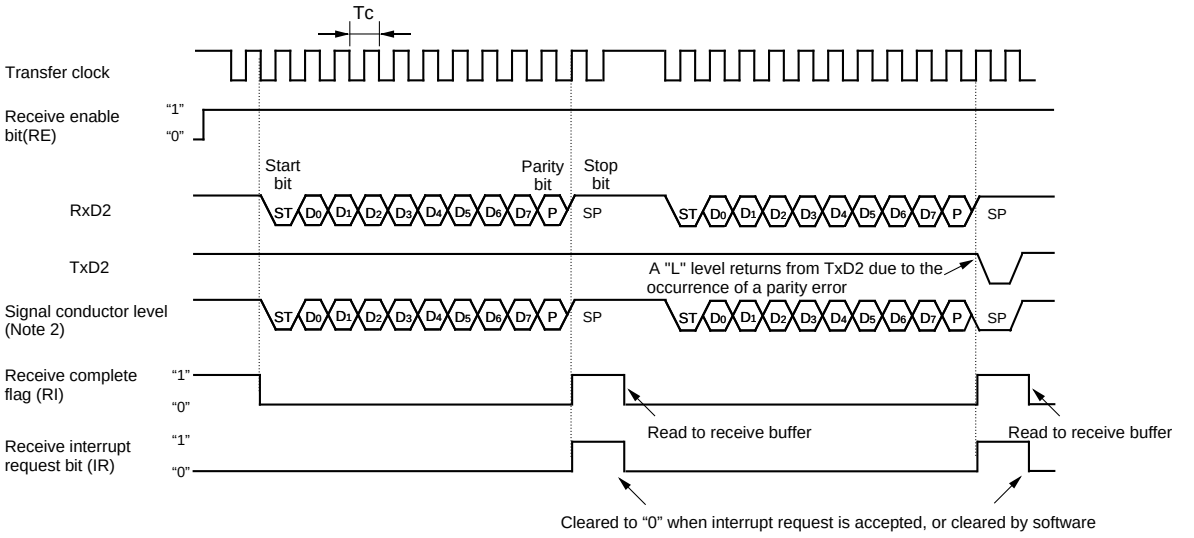


Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi
fi : frequency of BRG2 count source (f1, f8, f32)
n : value set to BRG2

Note 1. According to the above timing, the transmission is started by the timing of BRG overflow after writing to the transmit buffer.



Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi
fi : frequency of BRG2 count source (f1, f8, f32)
n : value set to BRG2

Note 2. The waveforms are the same because TxD2 and RxD2 are connected.

Fig.GA-22 Typical transmit/receive timing in UART mode (compliant with the SIM interface)

(a) Function for outputting a parity error signal

With the error signal output enable bit (bit 7 of address 037D16) assigned “1”, an “L” level from the TxD2 pin will be output when a parity error is detected. Link with this function, the timing to generate a transmission completion interrupt varies according to the timing of a parity error signal detection. Fig.GA-23 shows the timing of the parity error signal output.

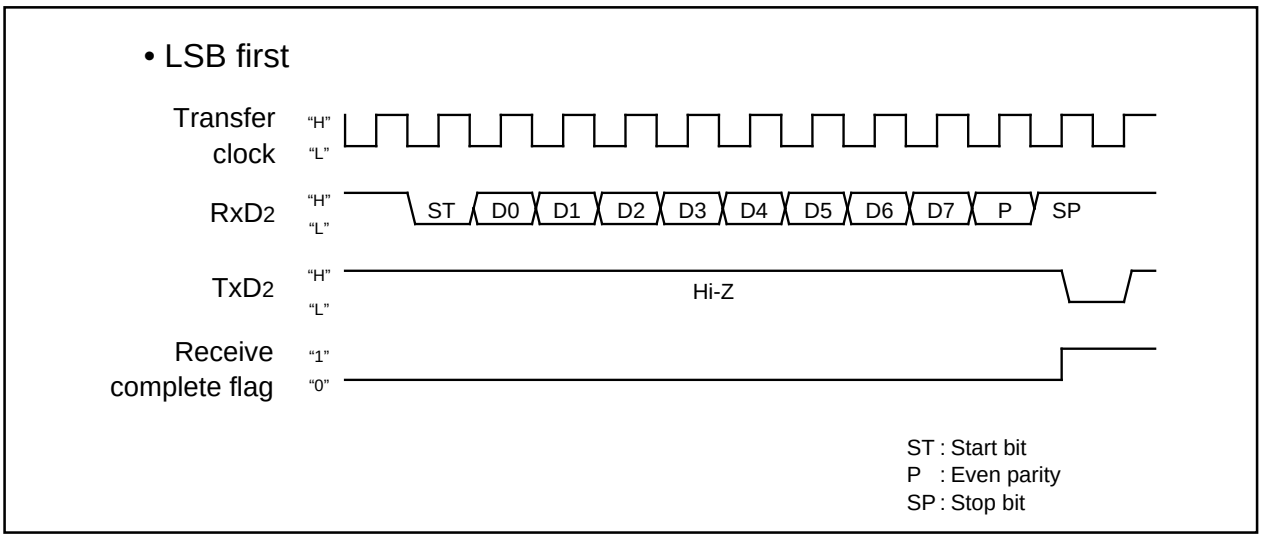


Fig.GA-23 Timing of the parity error signal output

(b) Direct format/inverse format

Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, data are output from TxD2 beginning with D0. If you choose the inverse format, data are inverted and output from TxD2 beginning with D7.

Fig.GA-24 shows the SIM interface format.

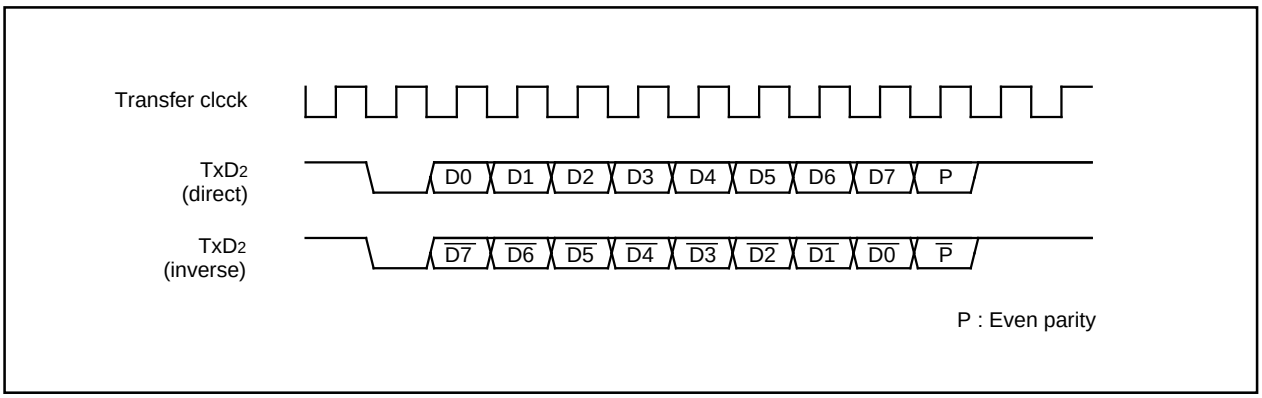


Fig.GA-24 SIM interface format

Fig.GA-25 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

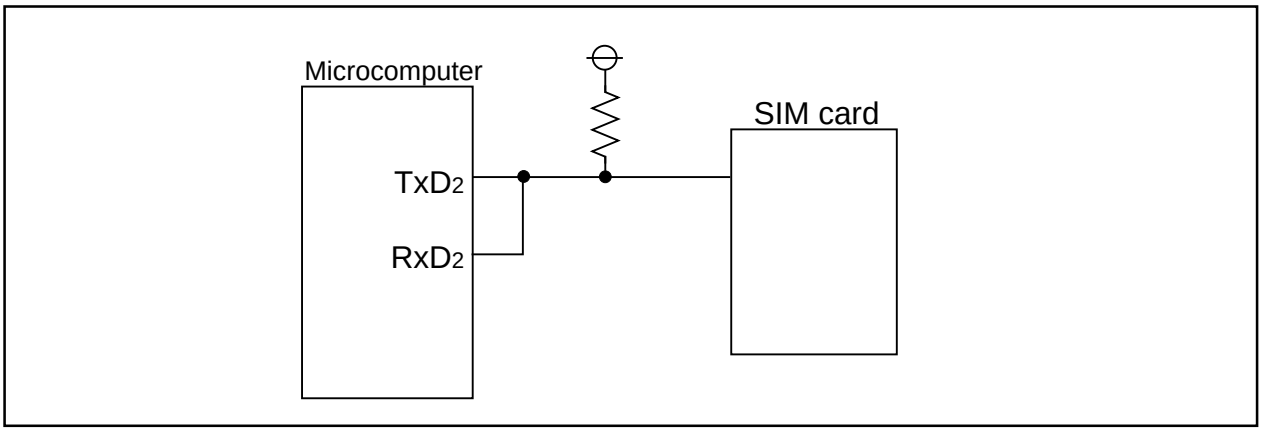


Fig.GA-25 Connecting the SIM interface

UART2 Special Mode Register

The UART2 special mode register (address 0377_{16}) is used to control UART2 in various ways. Fig.GA-26 shows the UART2 special mode register.

UART2 special mode register

b7	b6	b5	b4	b3	b2	b1	b0
0				0	0	0	0

Symbol: U2SMR
Address: 0377_{16}
When reset: 00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
Reserved bits		Must always be “0”		—	—
ABSCS	Bus collision detect sampling clock selection bit	Must always be “0”	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	○	○
ACSE	Auto clear function selection bit of transmit enable bit	Must always be “0”	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition selection bit	Must always be “0”	0 : Ordinary 1 : Falling edge of RxD2	○	○
Reserved bit		Must always be “0”		—	—

Fig.GA-26 UART2 special mode register

Some other functions added are explained here. Fig.GA-27 shows their workings.

Bit 4 of the UART2 special mode register is used as the bus collision detect sampling clock selection bit. The bus collision detect interrupt occurs when the RxD2 level and TxD2 level do not match, but the nonconformity is detected in synchronization with the rising edge of the transfer clock signal if the bit is set to “0”. If this bit is set to “1”, the nonconformity is detected at the timing of the overflow of timer A0 rather than at the rising edge of the transfer clock.

Bit 5 of the UART2 special mode register is used as the auto clear function selection bit of transmit enable bit. Setting this bit to “1” automatically resets the transmit enable bit to “0” when “1” is set in the bus collision detect interrupt request bit (nonconformity).

Bit 6 of the UART2 special mode register is used as the transmit start condition selection bit. Setting this bit to “1” starts the TxD transmission in synchronization with the falling edge of the RxD pin.

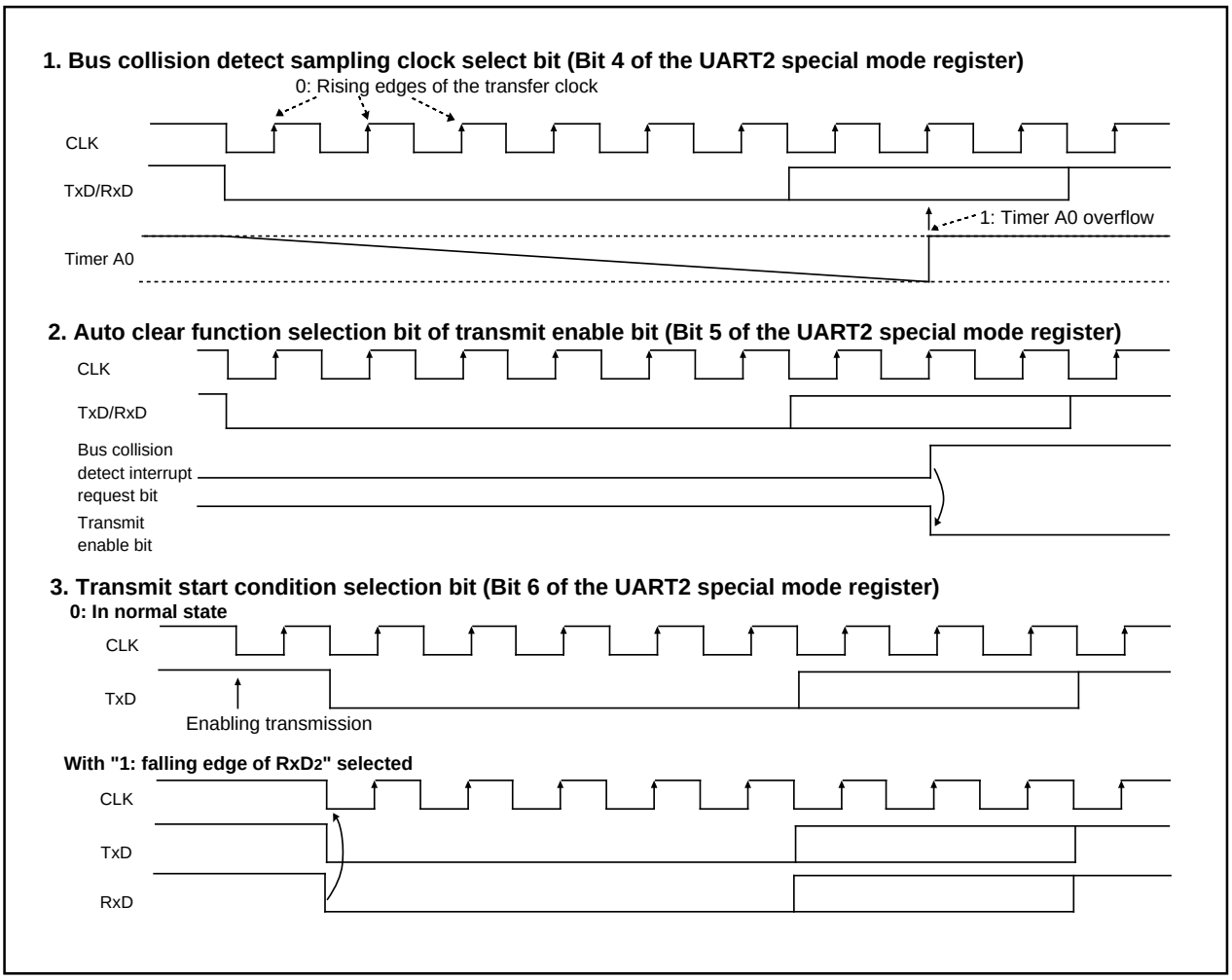


Fig.GA-27 Some other functions added

S I/O3, 4

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

Fig.GA-28 shows the S I/O 3, 4 block diagram, and Fig.GA-29 shows the S I/O 3, 4 control register.

Table.GA-9 shows the specifications of S I/O 3, 4.

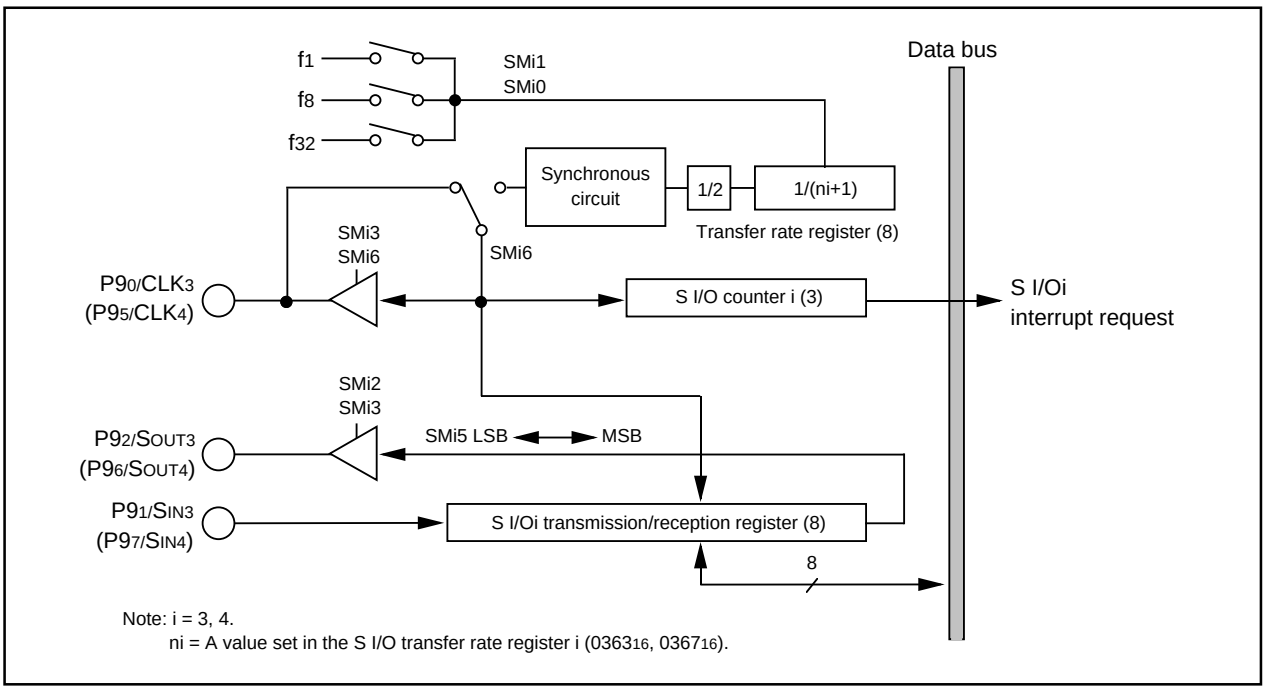
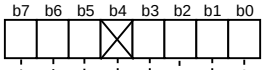


Fig.GA-28 S I/O3, 4 block diagram

S I/Oi control register (i = 3, 4) (Note 1)

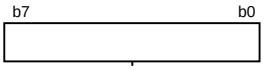


Symbol	Address	When reset
SiC	0362_{16}, 0366_{16}	40_{16}

Bit symbol	Bit name	Description	R	W
SMi0	Internal synchronous clock selection bits	b1 b0 0 0 : Selecting f_1 0 1 : Selecting f_8 1 0 : Selecting f_{32} 1 1 : Not to be used	○	○
SMi1			○	○
SMi2	SOUTi output disable bit	0 : SOUTi output 1 : SOUTi output disable (high impedance)	○	○
SMi3	S I/Oi port selection bit (Note 2)	0 : Input-output port 1 : SOUTi output, CLK function	○	○
	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".		—	—
SMi5	Transfer direction selection bit lect bit„û	0 : LSB first 1 : MSB first	○	○
SMi6	Synchronous clock selection bit (Note 2)	0 : External clock 1 : Internal clock	○	○
SMi7	SOUTi initial value set bit	Effective when SMi3 = 0 0 : L output 1 : H output	○	○

Note 1: Set "1" in bit 2 of the protection register ($000A_{16}$) in advance to write to the S I/Oi control register (i = 3, 4).
Note 2: When SI/Oi port selection bit (i= 3, 4) is set to "0" as for I/O port, set the synchronous clock selection bit to "1".

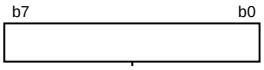
SI/Oi bit rate generator



Symbol	Address	When reset
S3BRG	0363_{16}	Indeterminate
S4BRG	0367_{16}	Indeterminate

Indeterminate	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}		○

SI/Oi transmission/reception register



Symbol	Address	When reset
S3TRR	0360_{16}	Indeterminate
S4TRR	0364_{16}	Indeterminate

Indeterminate	R	W
Transmission/reception starts by writing data to this register. After transmission/reception finishes, reception data is input.	○	○

Fig.GA-29 S I/O3, 4 control registers

Table.GA-9 Specifications of S I/O3, 4

Item	Specifications
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• With the internal clock selected (bit 6 of 0362_{16}, 0366_{16} = "1"): f1/2(ni+1), f8/2(ni+1), f32/2(ni+1) (Note 1) • With the external clock selected (bit 6 of 0362_{16}, 0366_{16} = 0):Input from the CLKi terminal (Note 2)
Conditions for transmission/ reception start	• To start transmit/reception, the following requirements must be met: - Select the synchronous clock (use bit 6 of 0362_{16}, 0366_{16}). Select a frequency dividing ratio if the internal clock has been selected (use bits 0 and 1 of 0362_{16}, 0366_{16}). - SOUTi initial value set bit (use bit 7 of 0362_{16}, 0366_{16})= 1. - S I/Oi port select bit (bit 3 of 0362_{16}, 0366_{16}) = 1. - Select the transfer direction (use bit 5 of 0362_{16}, 0366_{16}) - Write transfer data to SI/Oi transmission/reception register(0360_{16}, 0364_{16}) • To use S I/Oi interrupts, the following requirements must be met: - S I/Oi interrupt request bit (bit 3 of 0049_{16}, 0048_{16}) = 0.
Interrupt request generation timing	• At the rising edge of the last transfer clock (Note3)
Select function	• LSB first or MSB first selection Whether transmission/reception begins with bit 0 (LSB) or bit 7 (MSB) can be selected. • The SOUTi default value setting function If the transfer clock is selected to external clock, the output level of SOUTi pin can be selected when it is not in transferring please refer to Fig.GA-30.
Precaution	• The SI/Oi (i=3,4) is different from UART0 to 2 that the register and buffer can not be separated, so don't write the next transfer data to the transmission/reception register(0360_{16}, 0364_{16}) during transferring. • If the transfer clock is selected to internal clock, at the end of transferring, the SOUTi holds the last data during the last 1/2 transfer clock, and then to high impedance. If the transmission/reception register(0360_{16}, 0364_{16}) is written during the period, the SOUTi becomes the high impedance right the writing ,the data hold time will be shortened.

Note 1: n is a value from 00_{16} through FF_{16} set in the S I/Oi transfer rate register (i = 3, 4).

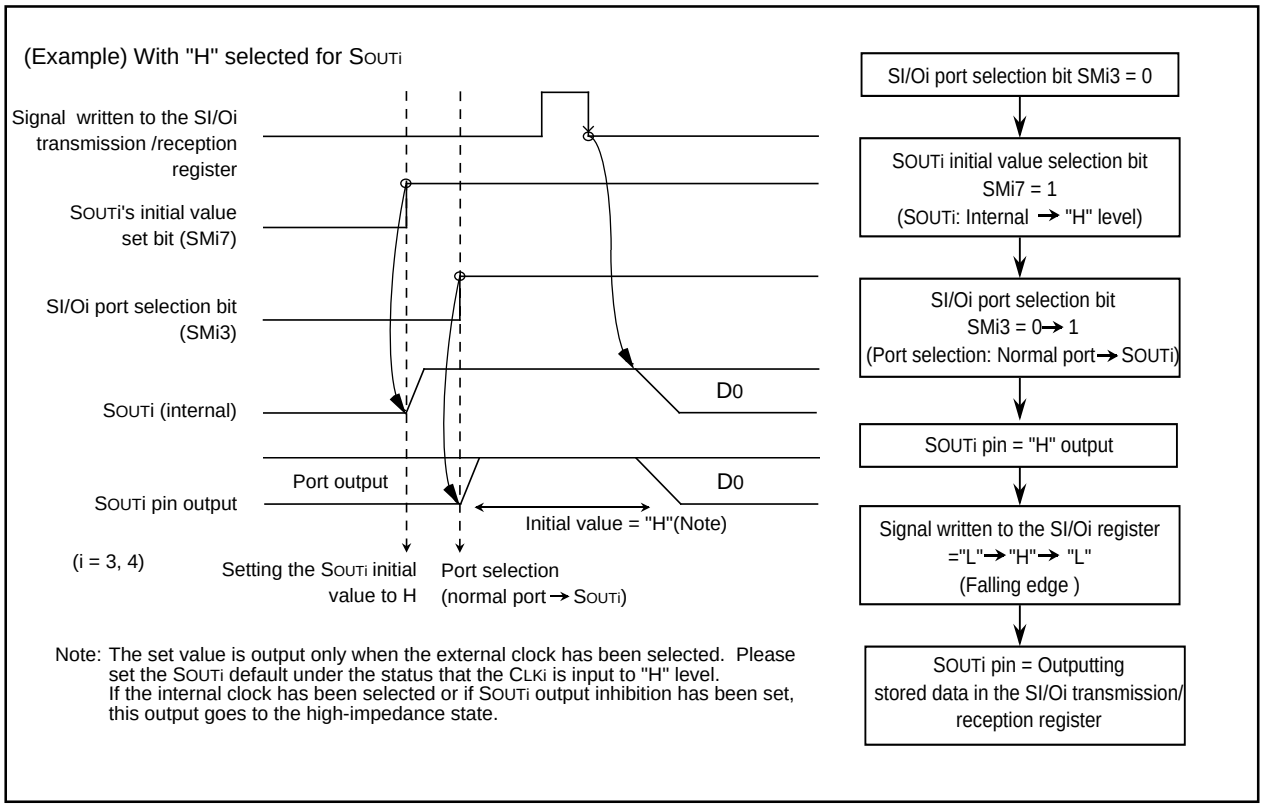
Note 2: With the external clock selected:

- Please write to the SI/Oi transmission/reception register(0360_{16}, 0364_{16}) under the status that the CLKi pin is input to "H" level. Also please write to the bit 7(SOUTi default value setting bit) under the status that the CLKi pin is input to "H" level.
- The S I/Oi circuit keeps on with the shift operation as long as the synchronous clock is entered in it, so stop the synchronous clock at the instant when it counts to eight. The internal clock, if selected, automatically stops.

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

■ Functions for setting an SOUTi initial value

In carrying out transmission, the output level of the SOUTi pin as it is before transmitting 1-bit data can be set either to “H” or to “L”. Fig.GA-30 shows the timing chart for setting an SOUTi initial value and how to set it.

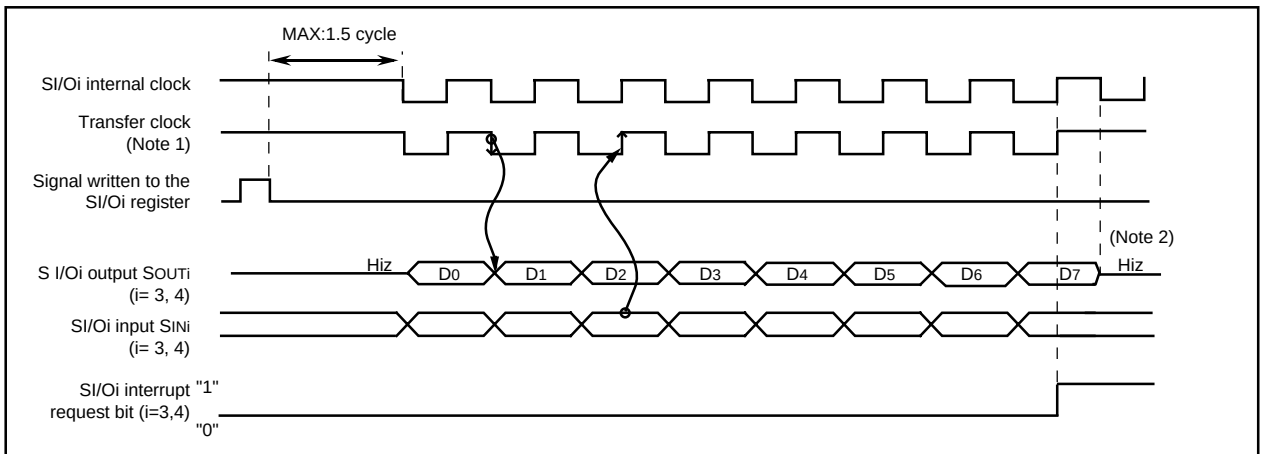


Note: The set value is output only when the external clock has been selected. Please set the SOUTi default under the status that the CLKi is input to "H" level.
If the internal clock has been selected or if SOUTi output inhibition has been set, this output goes to the high-impedance state.

Fig.GA-30 Timing chart for setting SOUTi’s initial value and how to set it

■ S I/Oi operation timing

Fig.GA-31 shows the S I/Oi operation timing



Note 1: With the internal clock selected for the transfer clock, the frequency dividing ratio can be selected using bits 0 and 1 of the SI/Oi control register (i = 3,4). (No frequency division, 8-division frequency, 32-division frequency.)
Note 2: With the internal clock selected for the transfer clock, the SOUTi (i = 3,4) pin becomes to the high-impedance state after the transfer finishes.
Note 3: The figure shows when the port selection bit of SOUTi (i = 3,4) is set to "1".

Fig.GA-31 S I/Oi operation timing chart

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P10_0$ to $P10_7$, $P9_5$, and $P9_6$ also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address $03D7_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect V_{REF}.

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

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

Table.JA-1 Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage	0V to AV_{CC} (V_{CC})
Operating clock ϕ_{AD} (Note1)	f_{AD}/divide-by-2 of f_{AD}/divide-by-4 of f_{AD}, $f_{AD}=f(X_{IN})$
Resolution	8-bit or 10-bit (selectable)
Absolute precision	• 8-bit resolution ±2LSB • 10-bit resolution ±6LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8pins (AN_0 to AN_7) + 2pins (ANEX0 and ANEX1)
A-D conversion start condition	• Software trigger A-D conversion starts when the A-D conversion start flag changes to “1” • External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is “1” and the $\overline{ADTRG}$/$P9_7$ input changes from “H” to “L”
Conversion speed per pin	• Without sample and hold function 8-bit resolution : 49 ϕ_{AD} cycles 10-bit resolution : 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution : 28 ϕ_{AD} cycles 10-bit resolution : 33 ϕ_{AD} cycles

Note 1: Without sample and hold function,set the ϕ_{AD} frequency to 250kHz min.
With the sample and hold fucntion, set the ϕ_{AD} frequency to 1MHz min.

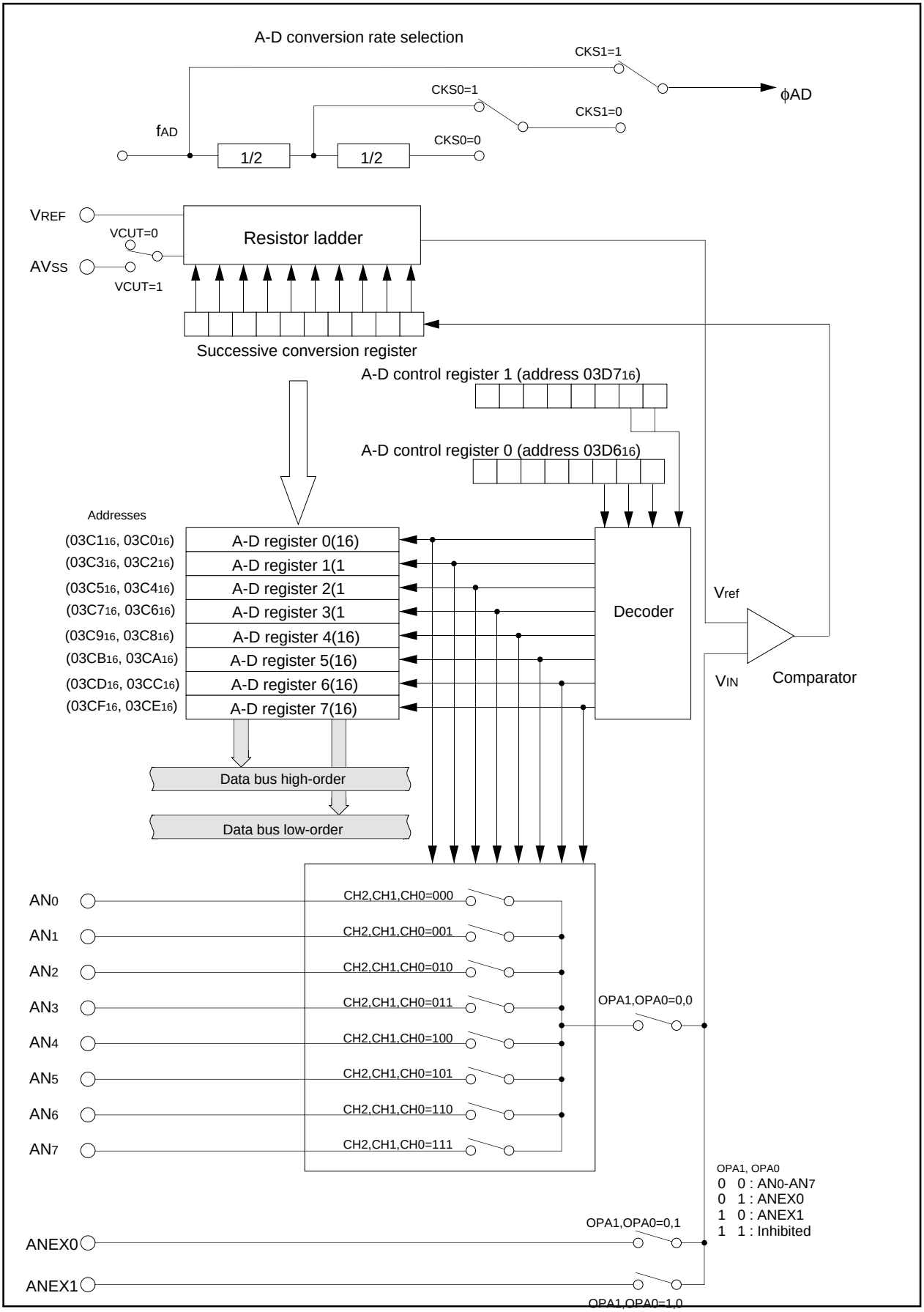


Fig.JA-1 Block diagram of A-D converter

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin selection bits	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode selection bits 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1 (Note 2)	○	○
MD1			○	○
TRG	Trigger selection bit	0 : Software trigger 1 : $\overline{AD}_{TRG}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency selection bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin selection bits	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins)	○	○
SCAN1		When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	○	○
MD2	A-D operation mode selection bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode selection bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency selection bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
OPA0	ANEX0,1 selection bits	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : Inhibited	○	○
OPA1			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

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

A-D control register 2 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
X	X	X	X	0	0	0	

Symbol	Address	When reset
ADCON2	$03D4_{16}$	$0000XXX0_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bit		Always set to “0”.	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

(b15) b7	(b8) b0	b7	b0
X	X		

Symbol	Address	When reset
ADi(i=0 to 7)	$03C0_{16}$ to $03CF_{16}$	Indeterminate

Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.	–	–

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

(1) One-shot mode

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

Table.JA-2 One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin selection bits is used for one A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected) • Writing "0" to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
			0	0				ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin selection bits	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode selection bits 0	b4 b3 0 0 : One-shot mode (Note 2)	O	O
MD1			O	O
TRG	Trigger selection bit	0 : Software trigger 1 : $\overline{AD_{TRG}}$ trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency selection bit 0	0: $f_{AD}/4$ is selected 1: $f_{AD}/2$ is selected	O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
		1			0			ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin selection bits	Invalid in one-shot mode	O	O
SCAN1			O	O
MD2	A-D operation mode selection bit 1	0 : Any mode other than repeat sweep mode 1	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency selection bit 1	0: $f_{AD}/2$ or $f_{AD}/4$ is selected 1: f_{AD} is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
OPA0	ANEX0,1 selection bis	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : Inhibited	O	O
OPA1			O	O

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

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

(2) Repeat mode

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

Table.JA-3 Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin selection bits is used for repeated A-D conversion
Star condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	Not generated
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	1			

Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin selection bits	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode selection bits 0	b4 b3 0 1 : Repeat mode (Note 2)	○	○
MD1			○	○
TRG	Trigger selection bit	0 : Software trigger 1 : $\overline{AD_{TRG}}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency selection bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin selection bits	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode selection bit 1	Should be "0" in this mode	○	○
BITS	8/10-bit mode selection bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency selection bit 1	0: $f_{AD}/2$ or $f_{AD}/4$ is selected 1: f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	ANEX0,1 selection bis	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : Inhibited	○	○
OPA1			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

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

(3) Single sweep mode

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

Table.JA-4 Single sweep mode specifications

Item	Specification
Function	The pins selected by the A-D sweep pin selection bits are used for one-by-one A-D conversion
Start condition	Writing "1" to A-D converter start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected) • Writing "0" to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN_0 and AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)
Reading of result of A-D converter	Read A-D registers corresponding to selected pins

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	0			

Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin selection bits	Invalid in single sweep mode	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode selection bits 0	b4 b3 1 0 : Single sweep mode	O	O
MD1			O	O
TRG	Trigger selection bit	0 : Software trigger 1 : $\overline{AD_{TRG}}$ trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency selection bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	O	O

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	Should be "0" in this mode	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency selection bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
OPA0	ANEX0,1 selection bis	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : Inhibited	O	O
OPA1			O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

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

(4) Repeat sweep mode 0

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

Table.JA-5 Repeat sweep mode 0 specifications

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

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			1	1			

Symbol	Address	When reset
ADCON0	03D6₁₆	00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin selection bits	Invalid in repeat sweep mode 0	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode selection bits 0	b4 b3 1 1 : Repeat sweep mode 0	O	O
MD1			O	O
TRG	Trigger selection bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency selection bit 0	0 : fAD/4 is selected 1 : fAD/2 is selected	O	O

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol	Address	When reset
ADCON1	03D7₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin selection bits	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode selection bits 1	Should be "0" in this mode	O	O
BITS	8/10-bit mode selection bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency selection bit 1	0 : fAD/2 or fAD/4 is selected 1 : fAD is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
OPA0	ANEX0,1 selection bis	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : Inhibited	O	O
OPA1			O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

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

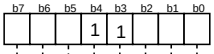
(5) Repeat sweep mode 1

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

Table.JA-6 Repeat sweep mode 1 specifications

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin selection bits Example : AN_0 selected $AN_0 \rightarrow AN_1 \rightarrow AN_0 \rightarrow AN_2 \rightarrow AN_0 \rightarrow AN_3$, etc
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	Not generated
Input pin	AN_0 (1 pin), AN_0 and AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins)
Reading of result of A-D converter	Read A-D registers corresponding to selected pins (at any time)

A-D control register 0 (Note)



Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin selection bits	Invalid in repeat sweep mode 1	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode selection bits 0	b4 b3 1 1 : Repeat sweep mode 1	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	O	O

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin selection bits	When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode selection bit 1	1 : Repeat sweep mode 1	O	O
BITS	8/10-bit mode selection bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency selection bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	O	O
VCUT	Vref connect bit	1 : Vref connected	O	O
OPA0	ANEX0,1 selection bis	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : Inhibited	O	O
OPA1			O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

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

(a) Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4$_{16}$) to “1”. When sample and hold is selected, the rage of conversion of each pin increases. As a result, a 28 f_{AD} cycle is achieved with 8-bit resolution and 33f_{AD} with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

(b) Extended analog input pins

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

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

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

D-A Converter

This is an 8-bit, R-2R type D-A converter. The microcomputer contains two independent D-A converters of this type.

DA conversion is performed when a value is written to the corresponding D-A register. Bits 0 and 1 (D-A output enable bits) of the D-A control register decide if the result of conversion is to be output. Do not set the corresponding port to output mode if D-A conversion is to be performed. If D-A output is enabled, the pull-up of corresponding port is inhibited.

Output analog voltage (V) is determined by a set value n (n : decimal) in the D-A register.

$V = V_{REF} \times n / 256$ (n = 0 to 255)

V_{REF} : reference voltage

Table.JB-1lists the performance of the D-A converter. Fig.JB-1 shows the block diagram of the D-A converter. Fig.JB-2 shows the D-A control register. Fig.JB-3 shows the D-A converter equivalent circuit.

Table.JB-1 Performance of D-A converter

Item	Performance
Conversion type	R-2R type
Resolution	8 bits
Analog output pins	2 channels

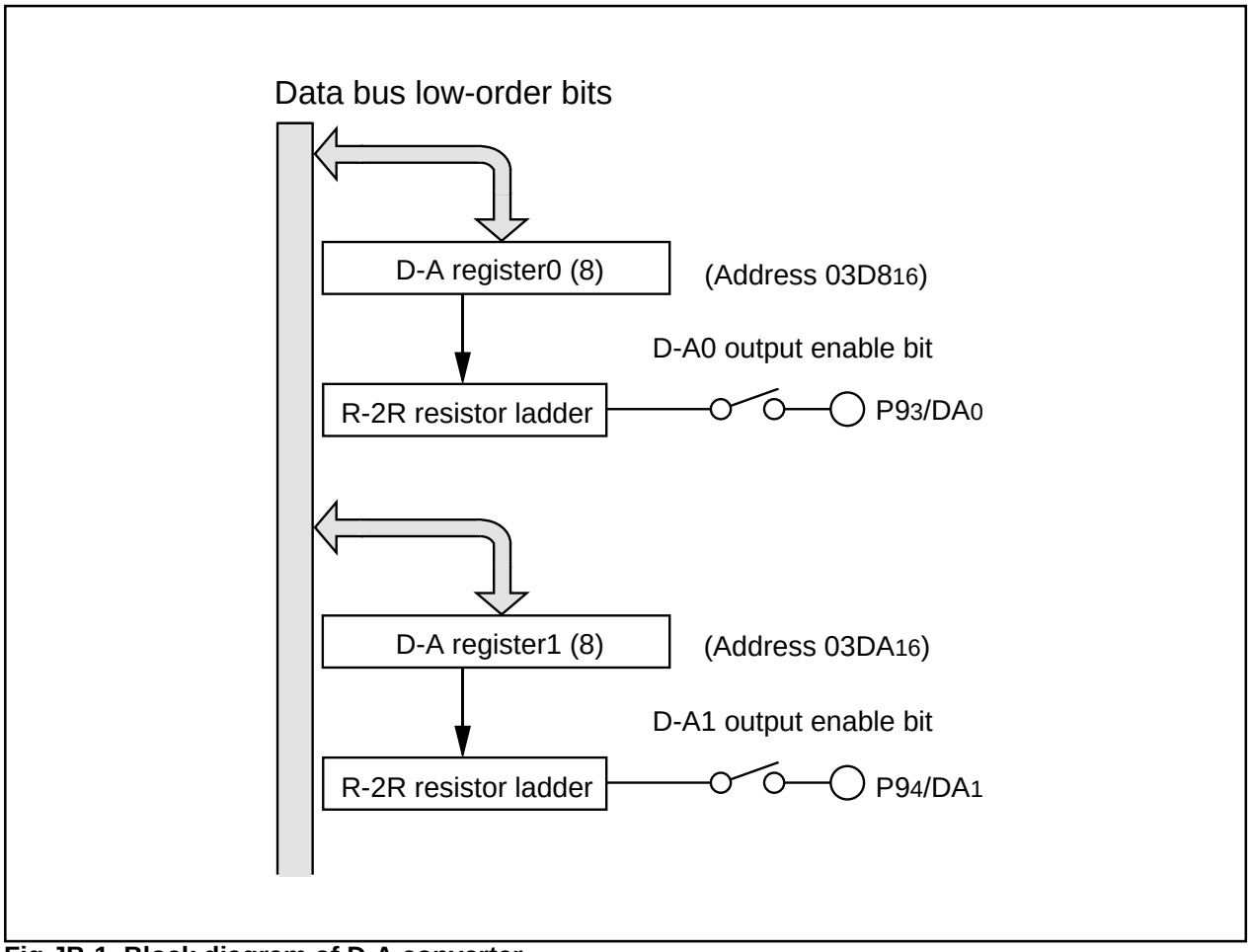


Fig.JB-1 Block diagram of D-A converter

D-A control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
DACON	03DC16	0016

Bit symbol	Bit name	Function	R	W
DA0E	D-A0 output enable bit	0 : Output disabled 1 : Output enabled	O	O
DA1E	D-A1 output enable bit	0 : Output disabled 1 : Output enabled	O	O
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”			–	–

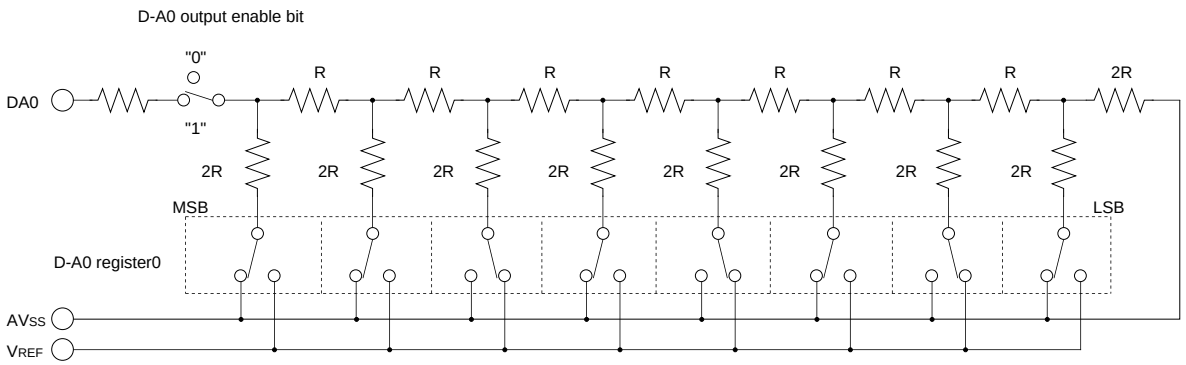
D-A register

b7 b0

Symbol	Address	When reset
DAi (i = 0,1)	03D816, 03DA16	Indeterminate

Function	R	W
Output value of D-A conversion	O	O

Fig.JB-2 D-A control register



Note 1: The above diagram shows an instance in which the D-A register is assigned 2A16.
Note 2: The same circuit as this is also used for D-A1.
Note 3: To reduce the current consumption when the D-A converter is not used, set the D-A output enable bit to 0 and set the D-A register to 0016
Do not let current flows through R-2R resistors.

Fig.JB-3 D-A converter equivalent circuit

Comparator Circuit

Comparator Configuration

A comparator circuit consists of a switch tree, ladder resistance, comparators, comparator control circuit, the comparator control register (address 03DE16), comparator data register (address 03DF16), and analog signal input pins(P50 - P57). The analog input pins (P50 - P57) are shared with the usual digital port I/O pins.
The comparator control register is a 4-bit register and can generate internal analog voltages in steps of 1/16 Vcc with the contens of bits 0 to 3. In Table.JC-2 contents of bits 0 to 3 of the comparator control register and corresponding internal analog voltage generated are indicated. The compared result of the analog input voltage and internal analog voltage is stored in the comparator data register. The value of comparator control register can not be read out.

Comparator Operation

In order to perform comparator operation, first, set the port P5 direction register (address 03EB16) to "0", as P5 can be used as the analog input pins. Then write a digital value, which corresponds to the internal analog voltage to be compared, to bits 0 to 3 of the comparator control register (address 03DE16) . The voltage comparison starts immediately by the writing operation. After 14 cycles of 1/2 main clock (the time needed for comparison), the compared result of the comparator is stored in the comparator data register (address 03DF16). Each bit of this register becomes as follows depending on the status of corresponding P50 to P57 pins:

When analog input voltage > internal analog voltage, it is "1".

When analog input voltage < internal analog voltage, it is "0".

For comparing once more, it is necessary to write data into comparator control register again even if the internal analog voltage is the same.
To read the result, wait 14 or more cycles after the comparator operation starts.
During the 14 cycles of the comparison, the ladder resistance is turned on and the reference voltage is generated. When the comparator is not in operation, the ladder resistance is off. Therefore, unnecessary consumption is prevented.
The comparison is accomplished by capacitive coupling. If the clock frequency is too low, electric charge will be lost. While the comparator is in operation, the clock frequency must be 1MHz or higher. During this time, do not execute a STP instruction, a WIT instruction, or an I/O instruction for port P5.

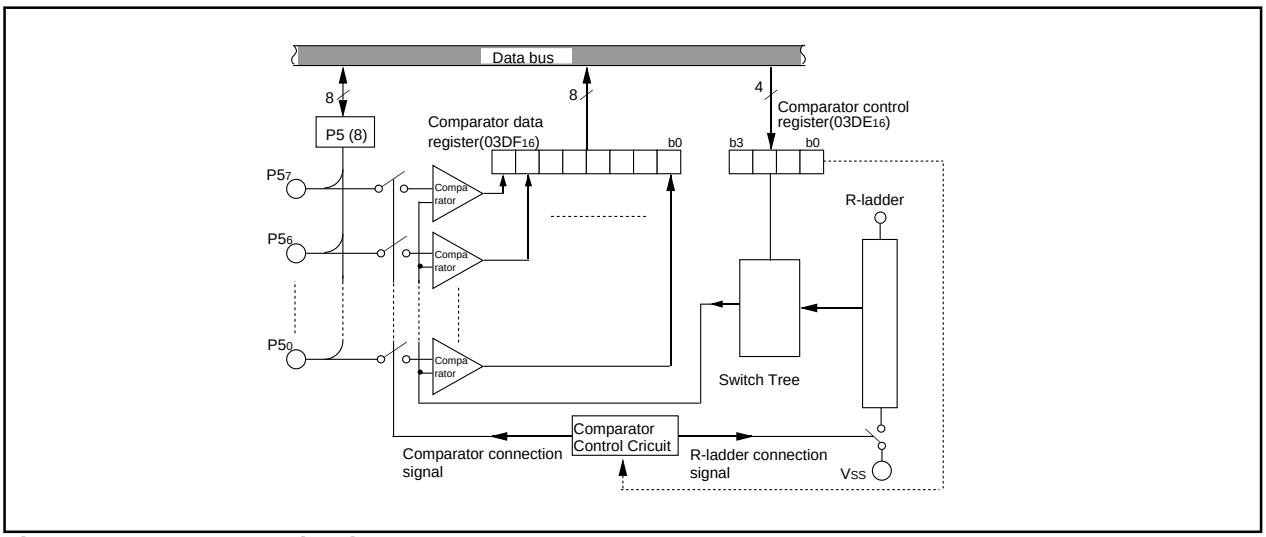


Fig.JC-1 Comparator circuit

Table.JC-1 Correspondence of internal analog voltage and contents of bits 0 to 3 of the comparator control register.

Comparator Control Register Contents of bit 0 to 3	Internal Analog Voltage
0 0 0 0 LSB	1 / 32•Vcc
0 0 0 1	1 / 16•Vcc + 1 / 32•Vcc
0 0 1 0	2 / 16•Vcc + 1 / 32•Vcc
0 0 1 1	3 / 16•Vcc + 1 / 32•Vcc
0 1 0 0	4 / 16•Vcc + 1 / 32•Vcc
0 1 0 1	5 / 16•Vcc + 1 / 32•Vcc
0 1 1 0	6 / 16•Vcc + 1 / 32•Vcc
0 1 1 1	7 / 16•Vcc + 1 / 32•Vcc
1 0 0 0	8 / 16•Vcc + 1 / 32•Vcc
1 0 0 1	9 / 16•Vcc + 1 / 32•Vcc
1 0 1 0	10 / 16•Vcc + 1 / 32•Vcc
1 0 1 1	11 / 16•Vcc + 1 / 32•Vcc
1 1 0 0	12 / 16•Vcc + 1 / 32•Vcc
1 1 0 1	13 / 16•Vcc + 1 / 32•Vcc
1 1 1 0	14 / 16•Vcc + 1 / 32•Vcc
1 1 1 1	15 / 16•Vcc + 1 / 32•Vcc

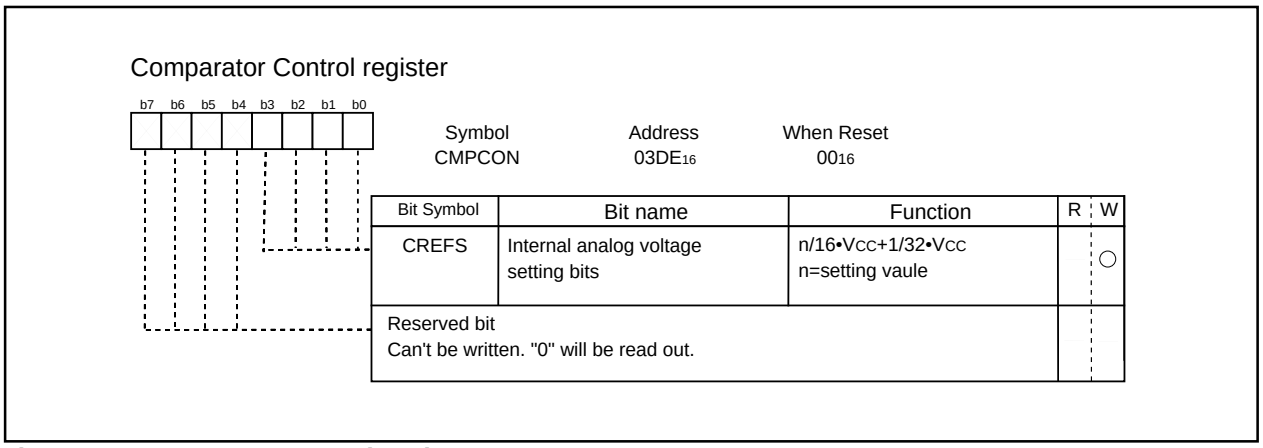


Fig.JC-2 Comparator control register

Pulse Width Modulation (PWM) Output Circuit

The M16C/6K group has four PWM output circuits, PWM0 to PWM3, with 14-bit resolution.

They operate independently. When the oscillation frequency X_{IN}= 8MHz, the minimum resolution bit width is 250 ns and the cycle period is 4096 µs. The PWM timing generator supplies a PWM control signal based on the X_{IN} clock.

The following explanation assumes X_{IN} = 8 MHz.

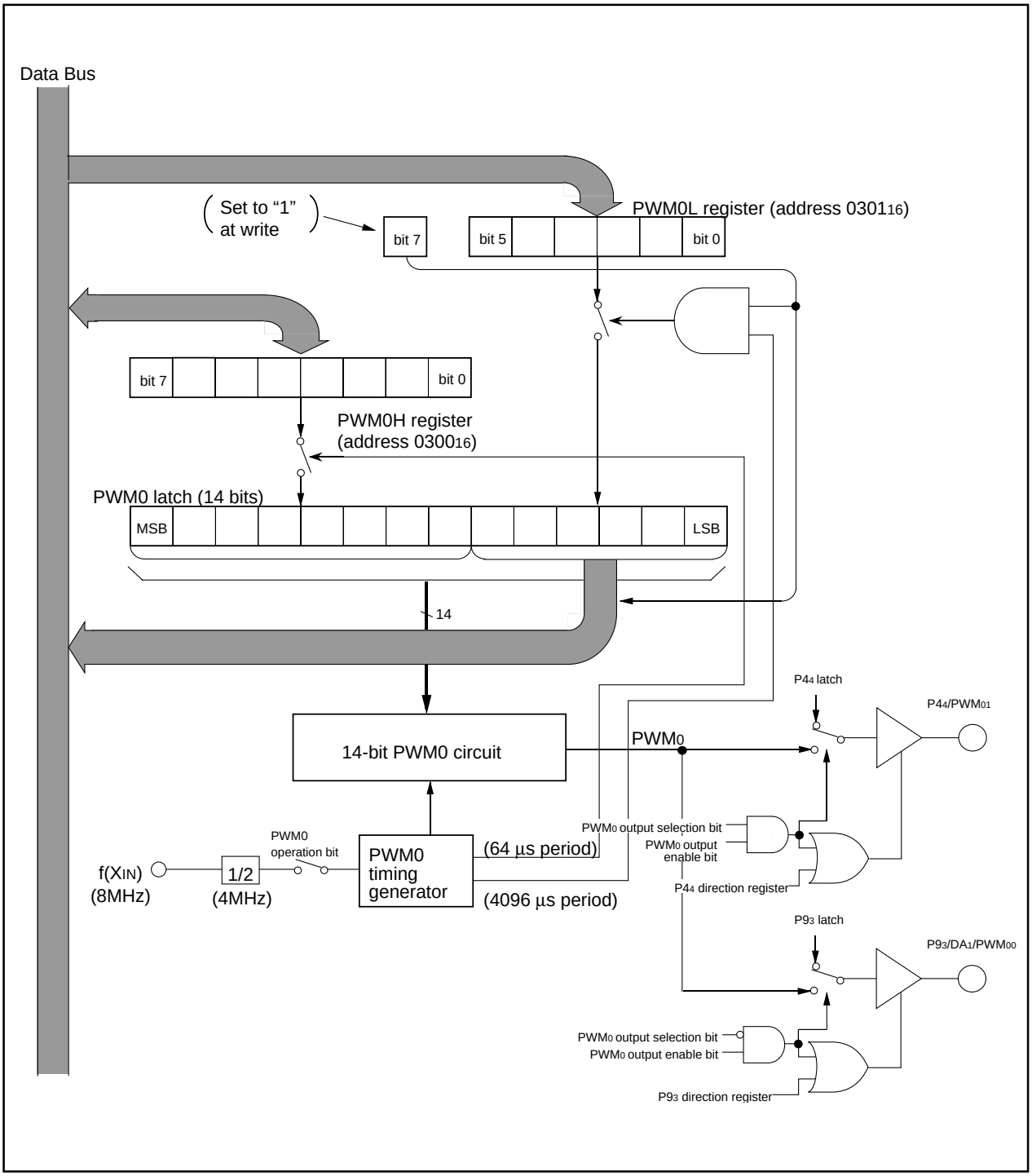


Fig.LA-1 PWM blobk diagram(PWM0)

Data Setup (PWM0)

The PWM_0 output pin shares with $P9_3$ or $P4_4$. The PWM_0 output pin is selected from either $P9_3/PWM_{00}$ or $P4_4/PWM_{01}$ by bit 0 of PWM control register 0 (address 0308_{16}). The PWM_0 output is enabled by setting bit 4 of PWM control register (address 0308_{16}) to "1". The PWM operation starts by setting bit 0 of PWM control register 1 (address 0309_{16}) to "1". The high-order eight bits of output data are set in the PWM0H register (address 0300_{16}) and the low-order six bits are set in the PWM0L register (address 0301_{16}). PWM1 to PWM3 is set as the same way.

PWM Operation

The 14-bit PWM data is divided into the low-order six bits and the high-order eight bits in the PWM latch. The high-order eight bits of data determine how long an “H”-level signal is output during each sub-period. There are 64 sub-periods in each period, and each sub-period is 256 X τ (64 μs) long. The signal is “H” for a length equal to N times τ, where τ is the minimum resolution (250 ns). “H” or “L” of the bit in the ADD part shown in Fig. LA-2 is added to this “H” duration by the contents of the low-order 6-bit data according to the rule in Table.LA-1. That is, only in the sub-period tm shown by Table.LA-1 in the PWM cycle period T = 64t, its “H” duration is lengthened to the minimum resolution τ added to the length of other periods.

For example, if the high-order eight bits of the 14-bit data are 03_{16} and the low-order six bits are 05_{16}, the length of the “H”-level output in sub-periods t8, t24, t32, t40, and t56 is 4 τ, and its length is 3 τ in all other sub-periods. Time at the “H” level of each sub-period almost becomes equal, because the time becomes length set in the high-order 8 bits or becomes the value plus τ, and this sub-period t (= 64 μs approximate 15.6 kHz) becomes cycle period approximately.

Transfer From Register to Latch

Data written to the PWML register is transferred to the PWM latch at each PWM period (every 4096 μs) and data written to the PWMH register is transferred to the PWM latch at each sub-period (every 64 μs). The signal which is output to the PWM output pin corresponds to the contents of this latch. A read from the PWML gets the latch content. However, bit 7 of the PWML register indicates whether the transfer to the PWM latch is completed; the transfer is completed when bit 7 is “0” and it is not done when bit 7 is “1.”

Table.LA-1 Relationship between low-order 6 bits of data and period set by the ADD bit.

Low-order 6 bits of data (PWML)	Sub-periods tm Lengthened (m=0 to 63)
0 0 0 0 0 0 LSB	None
0 0 0 0 0 1	m=32
0 0 0 0 1 0	m=16,48
0 0 0 1 0 0	m=8,24,40,56
0 0 1 0 0 0	m=4,12,20,28,36,44,52,60
0 1 0 0 0 0	m=2,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62
1 0 0 0 0 0	m=1,3,5,7.....63

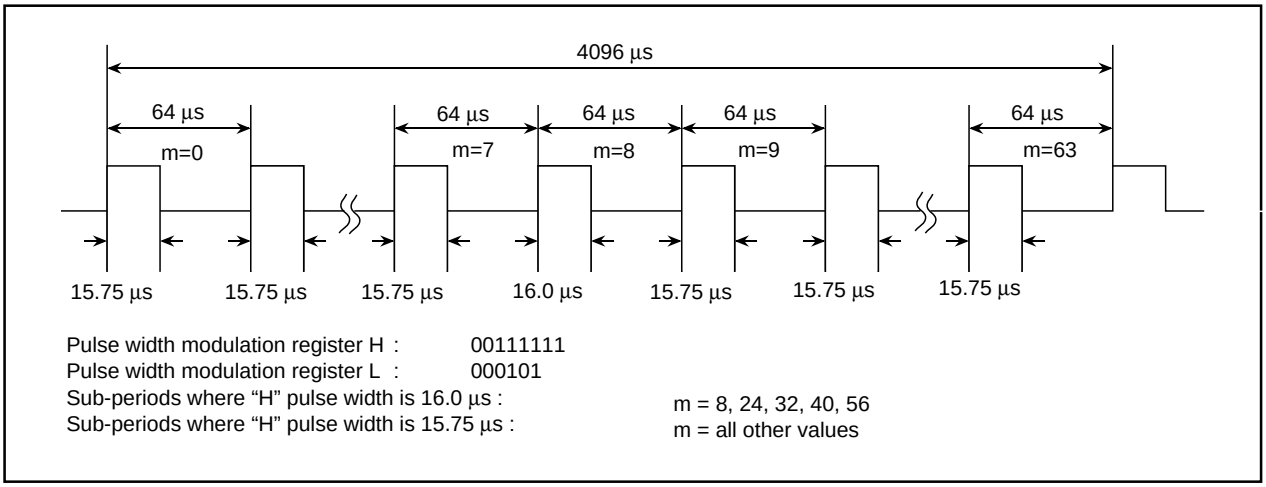


Fig.LA-2 PWM timing

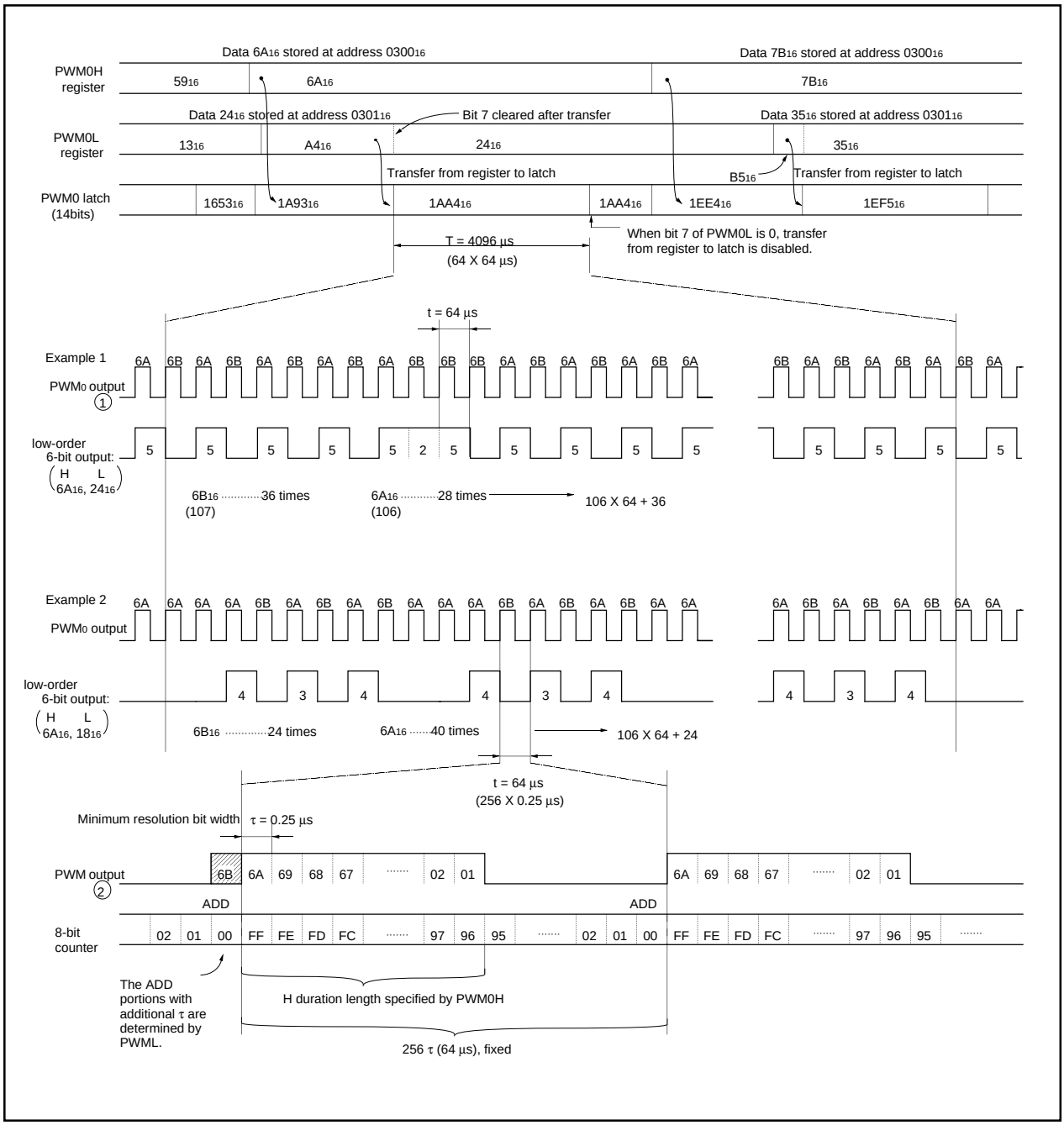
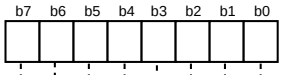


Fig.LA-3 14-bit PWM timing (PWM0)

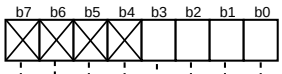
PWM control register 0



Symbol	Address	When reset
PWMCON0	0308_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
PWMSEL0	PWM0 output pin selection bit	0 : $P9_3$/PWM_{00} 1 : $P4_4$/PWM_{01}	○	○
PWMSEL1	PWM1 output pin selection bit	0 : $P9_4$/PWM_{10} 1 : $P4_5$/PWM_{11}	○	○
PWMSEL2	PWM2 output pin selection bit	0 : $P9_5$/PWM_{20} 1 : $P4_6$/PWM_{21}	○	○
PWMSEL3	PWM3 output pin selection bit	0 : $P9_6$/PWM_{30} 1 : $P4_7$/PWM_{31}	○	○
PWMEN0	PWM0 output enable bit	0 : disable 1 : enable	○	○
PWMEN1	PWM1 output enable bit	0 : disable 1 : enable	○	○
PWMEN2	PWM2 output enable bit	0 : disable 1 : enable	○	○
PWMEN3	PWM3 output enable bit	0 : disable 1 : enable	○	○

PWM control register 1



Symbol	Address	When reset
PWMCON1	0309_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
PWMST0	PWM0 operation bit	0 : operation stop 1 : operation start	○	○
PWMST1	PWM1 operation bit	0 : operation stop 1 : operation start	○	○
PWMST2	PWM2 operation bit	0 : operation stop 1 : operation start	○	○
PWMST3	PWM3 operation bit	0 : operation stop 1 : operation start	○	○
Reserved bit Can't be written. "0" will be read out.			—	—

Fig.LA-4 PWM control registers

LPC Bus Interface

LPC bus interface is based on Intel Low Pin Count (LPC) Interface Specification, Revision 1.0. It is I/O cycle data transfer format of serial communication. 4 channels are built in. The function of data bus buffer and data bus buffer status are almost the same as that of MELPS8-41 series. It can be written in or read out (as slave mode) by the control signals from host CPU side. The LPC bus interface functionality block diagram is shown in Figure GF-2. LPC data bus buffer functional Input / Output ports ($P3_0$-$P3_6$) are shared with GPIO port.

The setting of bit3 (LPC bus buffer enable bit) of LPC control register ($02D6_{16}$) is as below:

0: General purpose Input / Output port

1: LPC bus buffer functional Input / Output port

The enabling of channel of LPC bus buffer is controlled by bits 4-7 (LPC bus buffer 0-3 enable bits) of LPC control register ($02D6_{16}$). The slave address (16 bits) of LPC bus buffer channel 0 is fixed on 0060h, 0064h. The slave addresses (16 bits) of LPC bus buffer channel 1-3 are definable by setting LPC 1-3 address register H, L ($02D0_{16}$ to $02D5_{16}$). The setting value of bit2 of LPC1-3 address register (A2) L will not be decoded. The bit is “0” when read from slave CPU. The A2 status of slave address is latched to XA2 flag when written by host CPU. The input buffer full interrupt is generated when written in the data by host CPU. The Output buffer empty interrupt is generated when read out the data by host CPU. As shown in GF-1, the input buffer full interrupt request and output buffer empty interrupt request are switched by bit6, 7 of ISA control register0.

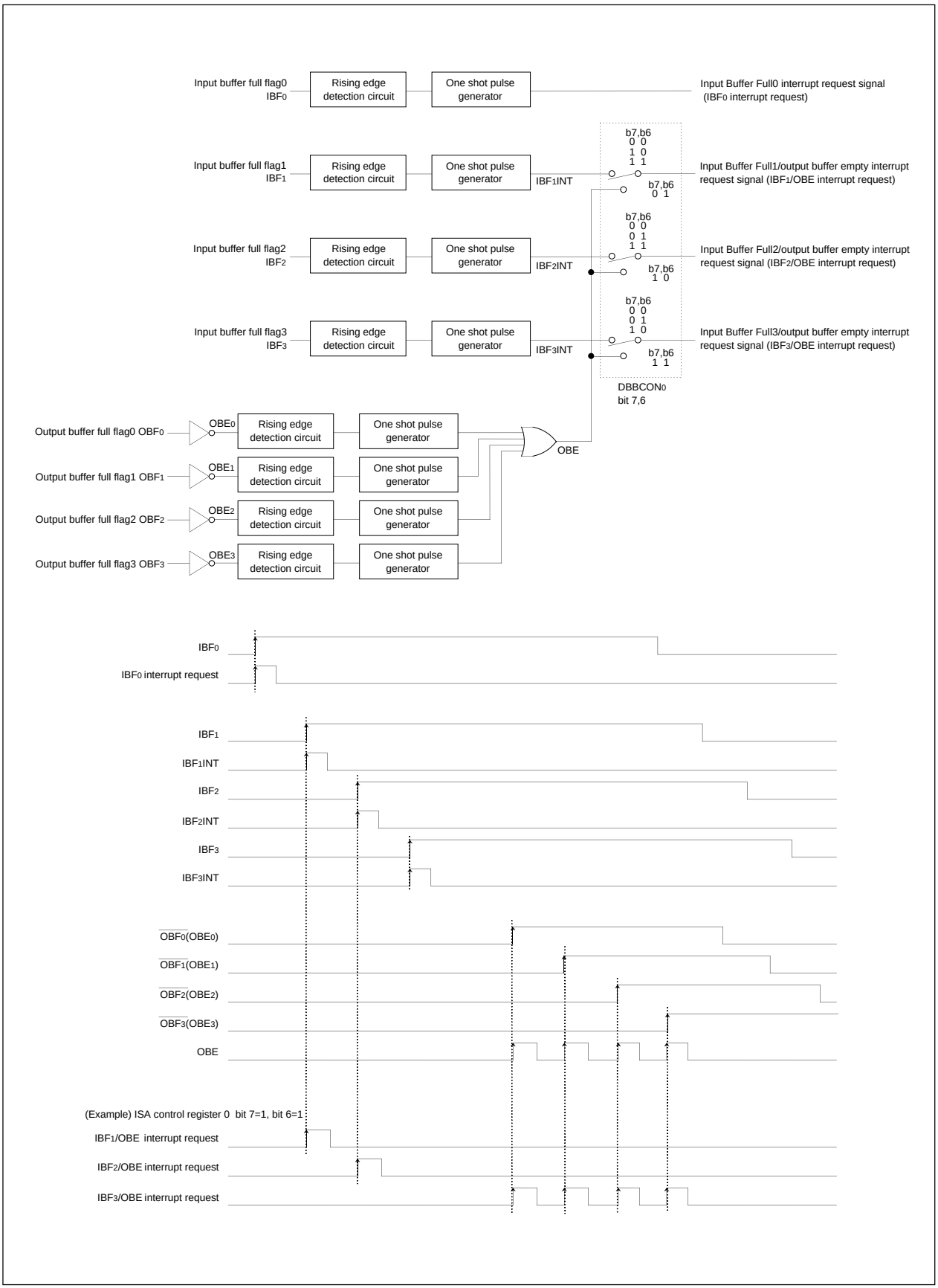


Fig.GF-1 Interrupt, request, circuit of Data Bus Buffer

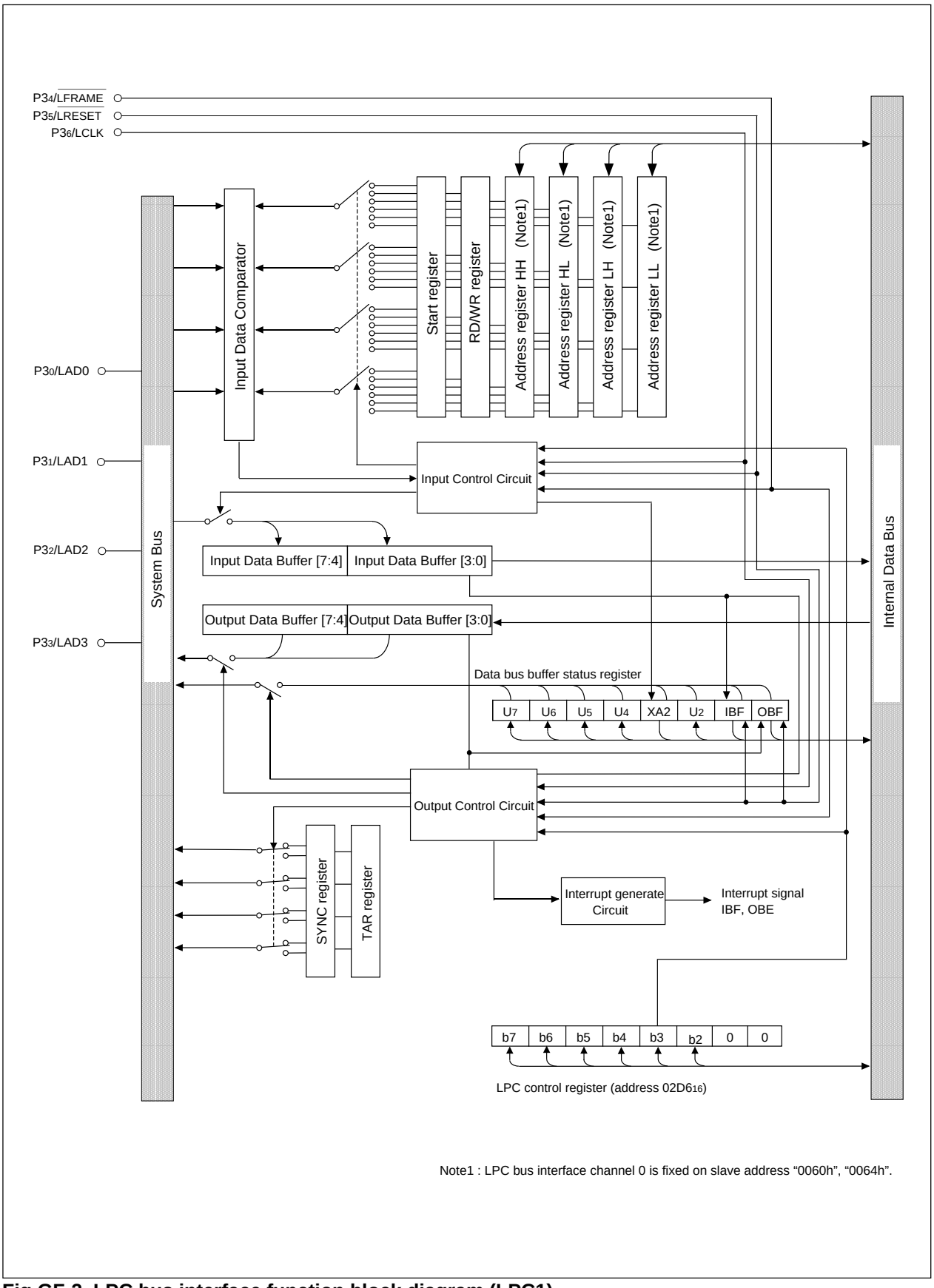


Fig.GF-2 LPC bus interface function block diagram (LPC1)

Figure GF-3: ISA control registers
Figure GF-4: Data bus buffer status register
Figure GF-5, 6: LPC related registers

Data bus buffer status register (DBBSTS0-DBBSTS3)

This is 8-bit register.
The bit 0, 1, 3 are read only bits and indicatie the status of data bus buffer.
Bit 2, 4, 5, 6, 7 are user definable and flags which can be read and written by software. The data bus buffer status register can be read out by host CPU when the slave address (16 bit) bit2 (A2) is high.

• Output buffer full flag (OBF)

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

• Input buffer full flag (IBF)

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

• XA2 flag (XA2)

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

Input data bus buffer register (DBBIN0-DBBIN3)

When there is a write request from host CPU, the data on the data bus will be latched to DBBIN0-3. The data of DBBIN0-3 can be read out from data bus buffer registers (Address:$02C0_{16}$, $02C2_{16}$, $02C4_{16}$, $02C6_{16}$) in SFR field.

Output data bus buffer register (DBBOUT0-DBBOUT3)

When writing data to data bus buffer registers (Address: $02C0_{16}$, $02C2_{16}$, $02C4_{16}$, $02C6_{16}$), the data will be transferred to DBBOUT0-3 automatically. The data of DBBOUT0-3 will be output to the data bus when there is a read request from host CPU and the status of bit2 (A2) of slave address (16 bits) is low.

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

The slave address (16 bits) of LPC bus buffer channel 0 is fixed on 0060h, 0064h.
The slave addresses (16 bits) of LPC bus buffer channel 1-3 are definable by setting LPC1-3 address registers H/L ($02D0_{16}$ to $02D5_{16}$). The settings are for slave address upper 8 bits and lower 8 bits. And these registers can be set and cleared in any time.
The bit 2 of LPC 1-3 address L is not decoded regardless of the setting value. When slave CPU reads LPC1-3 address registers, the bit2 (A2) of address low byte will be fixed to "0". The bit2 (A2) status of slave address is latched to XA2 flag when written by host CPU. The slave addresses that are already set in these registers will be used for comparing with the addresses to be received.

LPC control register (LPCCON)

• LPC bus interface enable bit (LPCBEN)
"0": $P3_0$ -$P3_6$ use as GPIO
"1": $P3_0$ -$P3_6$ use as LPC bus interface
• LPC bus buffer 0 enable bit (LPCEN0)
"0": LPC bus buffer0 disable
"1": LPC bus buffer0 enable
• LPC bus buffer 1 enable bit (LPCEN1)
"0": LPC bus buffer1 disable
"1": LPC bus buffer1 enable
• LPC bus buffer 2 enable bit (LPCEN2)
"0": LPC bus buffer2 disable
"1": LPC bus buffer2 enable
• LPC bus buffer 3 enable bit (LPCEN3)
"0": LPC bus buffer3 disable
"1": LPC bus buffer3 enable

• LPC software reset bit (LPCSR)
By setting the bit to "1", LPC interface is reset by the same status as $\overline{\text{LRESET}}$="L". After 1.5 cycles of BCLK at writing "1", reset is released and the bit becomes "0".
Nothing happens if "0" is set.

• SYNC output selection bits (SYNCSEL0,SYNCSEL1)
Table.GF-1 shows the content of SYNC output selected by SYNC output selection bits.

Table GF-1 SYNC output

SYNCSEL1	SYNCSEL0	SYNC cycle	SYNC output			
			1st cycle	2nd cycle	3rd cycle	4th cycle
0	0	1	0000_2	—	—	—
0	1	4	0110_2	0110_2	0110_2	0000_2
1	0	1	1010_2	—	—	—
1	1	4	0110_2	0110_2	0110_2	1010_2

ISA control register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	0	0	0

Symbol	Address	When reset
DBBCON0	$02C8_{16}$	00000000_2

Bit symbol	Bit name	Function	R	W
Nothing is assigned. Cannot be written. The value is "0" in reading.			–	–
IBFSEL	IBF/OBE interrupt request select bit	b7 b6 (Note 1) 0 0 : OBE disable 0 1 : OBE enable, IBF_1 disable 1 0 : OBE enable, IBF_2 disable 1 1 : OBE enable, IBF_3 disable	○	○

Note 1: By setting these two bits, one of IBF_1 to IBF_3 interrupt requests will be switched to OBE interrupt request.
There is no relative between IBF_0 interrupt request and these two bits.

Interrupt \ b7, b6	0 , 0	0 , 1	1 , 0	1 , 1
IBF_0 interrupt	IBF_0	IBF_0	IBF_0	IBF_0
IBF_1 interrupt	IBF_1	OBE	IBF_1	IBF_1
IBF_2 interrupt	IBF_2	IBF_2	OBE	IBF_2
IBF_3 interrupt	IBF_3	IBF_3	IBF_3	OBE

ISA control register 1

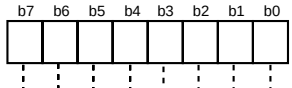
b7	b6	b5	b4	b3	b2	b1	b0
0	0						

Symbol	Address	When reset
DBBCON1	$02C9_{16}$	00000000_2

Bit symbol	Bit name	Function	R	W
OBF0SEL	OBF0 output selection bit	0 : OBF_{00} enable 1 : OBF_{01} enable	○	○
$OBF_{00}EN$	OBF_{00} output enable bit	0 : $P4_0$ as GPIO 1 : $P4_0$ as OBF_{00} output	○	○
$OBF_{01}EN$	OBF_{01} output enable bit	0 : $P4_3$ as GPIO 1 : $P4_3$ as OBF_{01} output	○	○
OBF_1EN	OBF_1 output enable bit	0 : $P4_4$ as GPIO 1 : $P4_4$ as OBF_1 output	○	○
OBF_2EN	OBF_2 output enable bit	0 : $P4_5$ as GPIO 1 : $P4_5$ as OBF_2 output	○	○
OBF_3EN	OBF_3 output enable bit	0 : $P4_6$ as GPIO 1 : $P4_6$ as OBF_3 output	○	○
Nothing is assigned. Cannot be written. The value is "0" in reading.			–	–

Fig.GF-3 ISA control registers

Data bus buffer status register

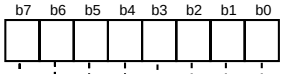


Symbol	Address	When reset
DBBSTS0	$02C1_{16}$	00000000_2
DBBSTS1	$02C3_{16}$	00000000_2
DBBSTS2	$02C5_{16}$	00000000_2
DBBSTS3	$02C7_{16}$	00000000_2

Bit symbol	Bit name	Function	R	W
OBF	Output buffer full flag	0 : Buffer empty 1 : Buffer full	○	×
IBF	Input buffer full flag	0 : Buffer empty 1 : Buffer full	○	×
U2	User definable flag	This flag can be freely defined by user	○	○
XA2	XA2 flag	This flag is indication the A2 status of the 16 bit slave address when IBF flag is set	○	×
U4	User definable flag	This flag can be freely defined by user	○	○
U5			○	○
U6			○	○
U7			○	○

Fig.GF-4 Data bus buffer status register

LPCi address register L (i=1,2,3) (Note2)

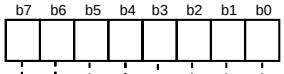


Symbol	Address	When reset
LPC1ADL	$02D0_{16}$	00000000_2
LPC2ADL	$02D2_{16}$	00000000_2
LPC3ADL	$02D4_{16}$	00000000_2

Bit symbol	Bit name	R	W
LPCSAD0	Slave address0	○	○
LPCSAD1	Slave address1	○	○
LPCSAD2	Slave address2 (Note1)	○	×
LPCSAD3	Slave address3	○	○
LPCSAD4	Slave address4	○	○
LPCSAD5	Slave address5	○	○
LPCSAD6	Slave address6	○	○
LPCSAD7	Slave address7	○	○

Note1: Always returns “0” when read, even if writing “1” to this bit.
Note2: Do not set the same 16 bits slave address in each channel.

LPCi address register H (i=1,2,3)

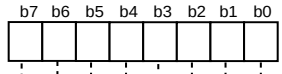


Symbol	Address	When reset
LPC1ADH	$02D1_{16}$	00000000_2
LPC2ADH	$02D3_{16}$	00000000_2
LPC3ADH	$02D5_{16}$	00000000_2

Bit symbol	Bit name	R	W
LPCSAD8	Slave address8	○	○
LPCSAD9	Slave address9	○	○
LPCSAD10	Slave address10	○	○
LPCSAD11	Slave address11	○	○
LPCSAD12	Slave address12	○	○
LPCSAD13	Slave address13	○	○
LPCSAD14	Slave address14	○	○
LPCSAD15	Slave address15	○	○

Fig.GF-5 LPC related registers

LPC control register



Symbol	Address	When reset
LPCCON	$02D6_{16}$	00000000_2

Bit symbol	Bit name	Function	R	W
SYNCSEL0	SYNC output selection bits	b1 b0 0 0: OK 0 1: Long & OK 1 0: Err 1 1: Long & Err	○	○
SYNCSEL1				
LPCSR	LPC software reset bit	0 : The release of reset (Note) 1 : Reset	○	○
LPCBEN	LPC interface enable bit	0 : $P3_0$ - $P3_6$ as GPIO 1 : LPC bus buffer enable	○	○
LPCEN0	LPC bus buffer 0 enable bit	0 : LPC bus buffer 0 disable 1 : LPC bus buffer 0 enable	○	○
LPCEN1	LPC bus buffer 1 enable bit	0 : LPC bus buffer 1 disable 1 : LPC bus buffer 1 enable	○	○
LPCEN2	LPC bus buffer 2 enable bit	0 : LPC bus buffer 2 disable 1 : LPC bus buffer 2 enable	○	○
LPCEN3	LPC bus buffer 3 enable bit	0 : LPC bus buffer 3 disable 1 : LPC bus buffer 3 enable	○	○

Note: For LPC software reset, the bit will automatically return to “0” after writing “1”.

Fig.GF-6 LPC control register

Basic operation of LPC bus interface

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

Setting steps for using LPC bus interface is explained below.

• Setting bit3 (LPC interface enable bit) of LPC control register($02D6_{16}$) to "1"

• Choosing which LPC bus buffer channel will be used

• Setting "1" to bits 4-7 (LPC bus buffer 0-3 enable bit) of LPC control register ($02D6_{16}$).

• The 16-bit slave address of LPC bus buffer channel is defined by writing 16-bit slave address to LPC 1-3 address registers ($02D0_{16}$ - $02D5_{16}$). If channel 1-3 LPC bus buffer is chosen, set the address to the corresponding address register.

• Selecting IBF/ OBE interrupt in ISA control register0 ($02C8_{16}$)

• Selecting OBF output port in ISA control register1 ($02C9_{16}$)

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

Writing timing is shown in Figure GF-8.

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

• 1st cycle : When $\overline{\text{LFRAME}}$ is "Low", sending "0000_2 " to LAD[3:0] for communication start frame detecting.

• 2nd cycle : When $\overline{\text{LFRAME}}$ is "High", sending "$001X_2$ " to LAD[3:0] for write frame detecting.

• From 3rd cycle to 6th cycle: These four cycles are detecting for 16 bits slave address.

3rd cycle: The slave address which is from host is written to slave address register [15:12] through LAD[3:0]

4th cycle: The slave address which is from host is written to slave address register [11:8] through LAD[3:0]

5th cycle: The slave address which is from host is written to slave address register [7:4] through LAD[3:0]

6th cycle: The slave address which is from host is written to slave address register [3:0] through LAD[3:0]

• 7th and 8th cycles are used for one data byte transfer.

7th cycle: The data which is from host is written to input data buffer[3:0] through LAD[3:0]

8th cycle: The data which is from host is written to input data buffer[7:4] through LAD[3:0]

• 9th and 10thcycles are for changing the communication direction from host→slave to slave→host

9th cycle: Host outputs "1111_2 " to LAD[3:0]

10thcycle: The LAD[3:0] will be set to Hi-Z by HOST to switch the communication direction.

• 11th cycle: The "0000_2 " (SYNC OK) is output to LAD[3:0] for acknowledge.

• 12th cycle: The "1111_2 " is output to LAD[3:0]. The XA2 and IBF flag are set. IBF interrupt signal is generated.

• 13th cycle: The LAD[3:0] will be set to Hi-Z by slave to switch the communication direction.

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

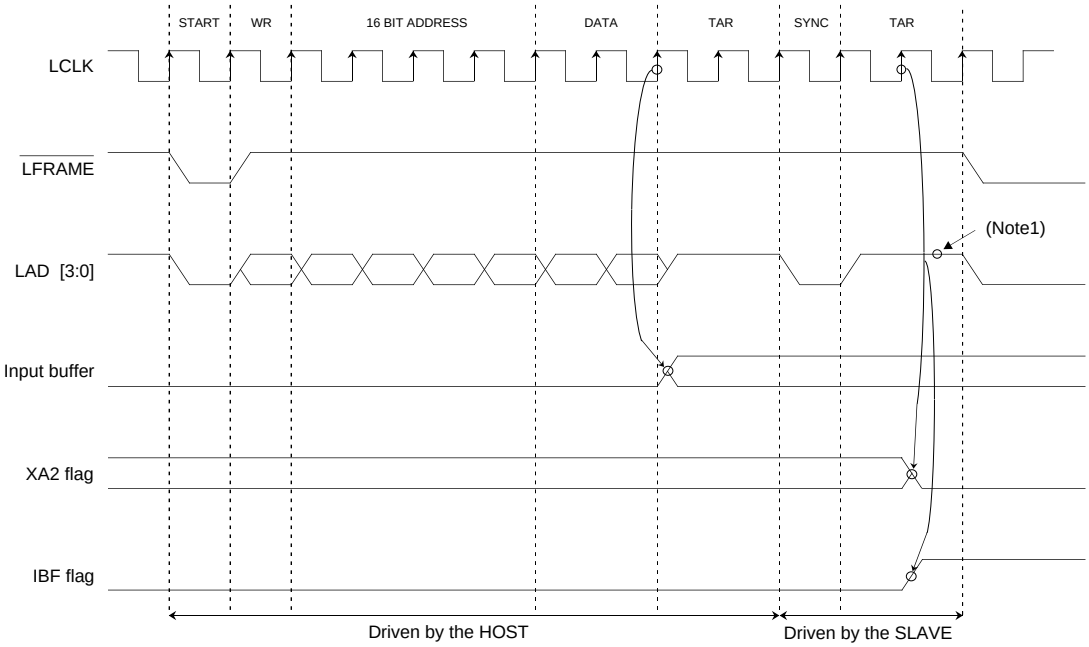
<2> Example for I/O reading cycle from HOST

Reading timing is shown in Figure GF-9.

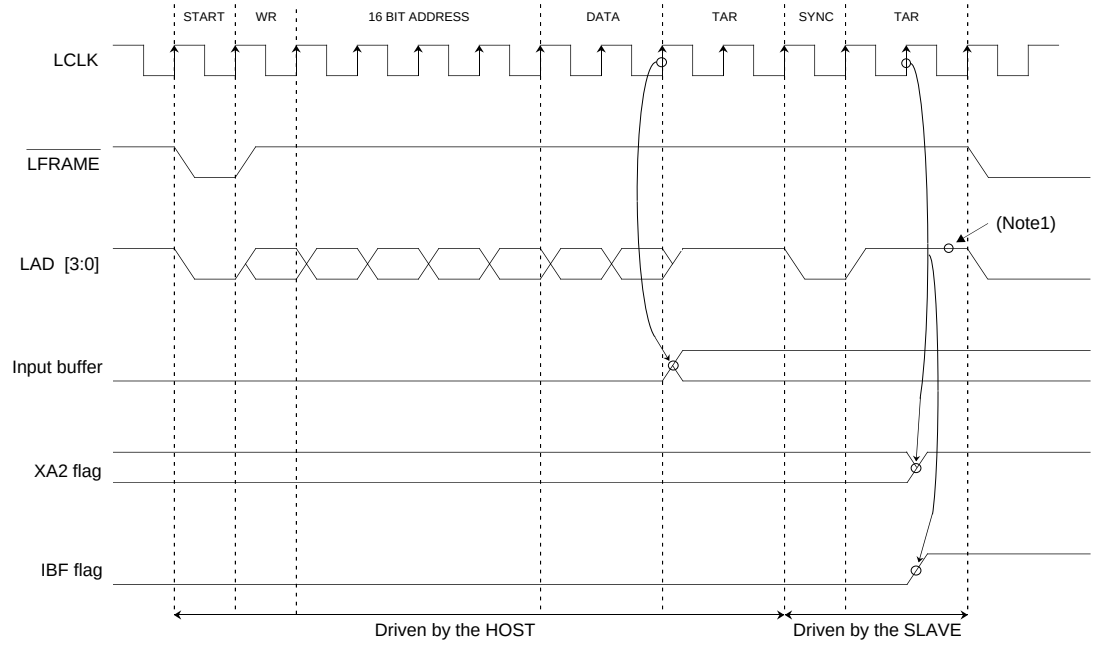
The basic communication cycles of LPC I/O protocol are 13 cycles. The data of LAD[3:0] will be read by the rising edge of LCLK. Communication will start from $\overline{\text{LFRAME}}$ falling edge.

- 1st cycle: When $\overline{\text{LFRAME}}$ is "Low", sending "0000_2 " to LAD[3:0] for communication start detecting.
- 2ndcycle: When $\overline{\text{LFRAME}}$ is "High", the host send "$000X_2$ " on LAD[3:0] to inform the cycle type as I/O read.
- From 3rd cycle to 6thcycle: These four cycles are detecting for 16 bits slave address.
  3rdcycle: The slave address which is from host is written to slave address register [15:12] throughLAD[3:0]
  4thcycle: The slave address which is from host is written to slave address register [11:8] throughLAD[3:0]
  5thcycle: The slave address which is from host is written to slave address register [7:4] throughLAD[3:0]
  6thcycle: The slave address which is from host is written to slave address register [3:0] throughLAD[3:0]
- 7th and 8thcycles are used for changing the communication direction from host→slave to slave→host
  7thcycle: Host is output "1111_2 " to LAD[3:0]
  8thcycle: The LAD[3:0] will be set to Hi-Z by HOST to switch the communication direction.
- 9thcycle : The "0000_2 " (SYNC OK) is output to LAD[3:0] for acknowledge.
- 10th and 11thcycles are for output 8 bits data from output data buffer or output 8 bits data from status register.
  10thcycle: Sending output data buffer [3:0] to LAD[3:0] or sending data of status register [3:0] to LAD[3:0]
  11thcycle: Sending output data buffer [7:4] to LAD[3:0] or sending data of status register [7:4] to LAD[3:0].
- 12thcycle: The "1111_2 " is output to LAD[3:0]. The OBF flag is cleared and OBE interrupt signal is generated.
- 13thcycle: The LAD[3:0] will be set to Hi-Z by slave to switch the communication direction. OBF flag will be set when 8 bits data are written to output data buffer by slave CPU.

● Data WR (I/O write cycle)



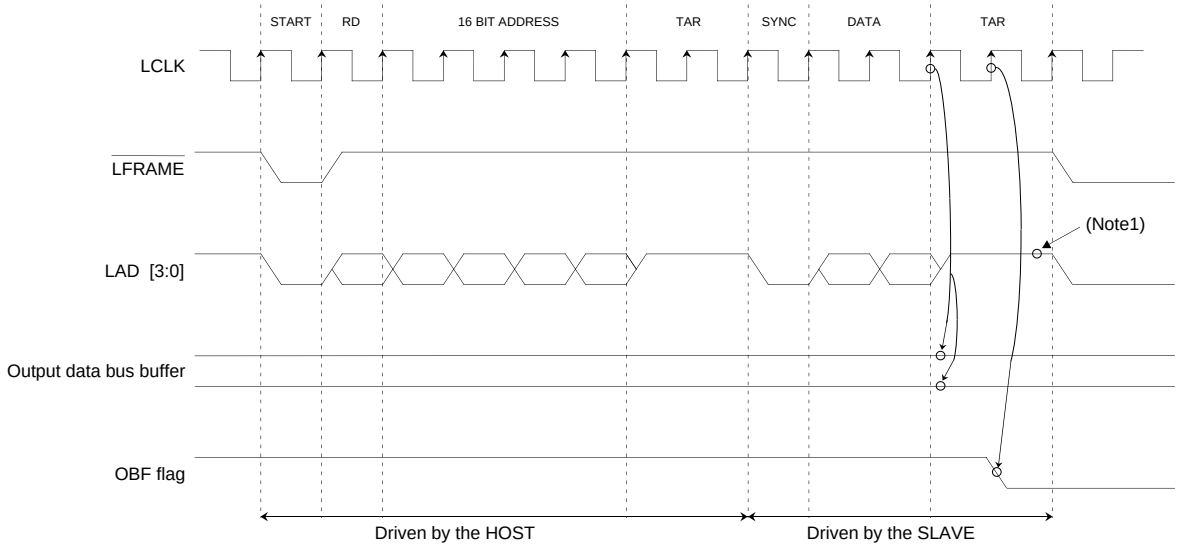
● Command WR (I/O write cycle)



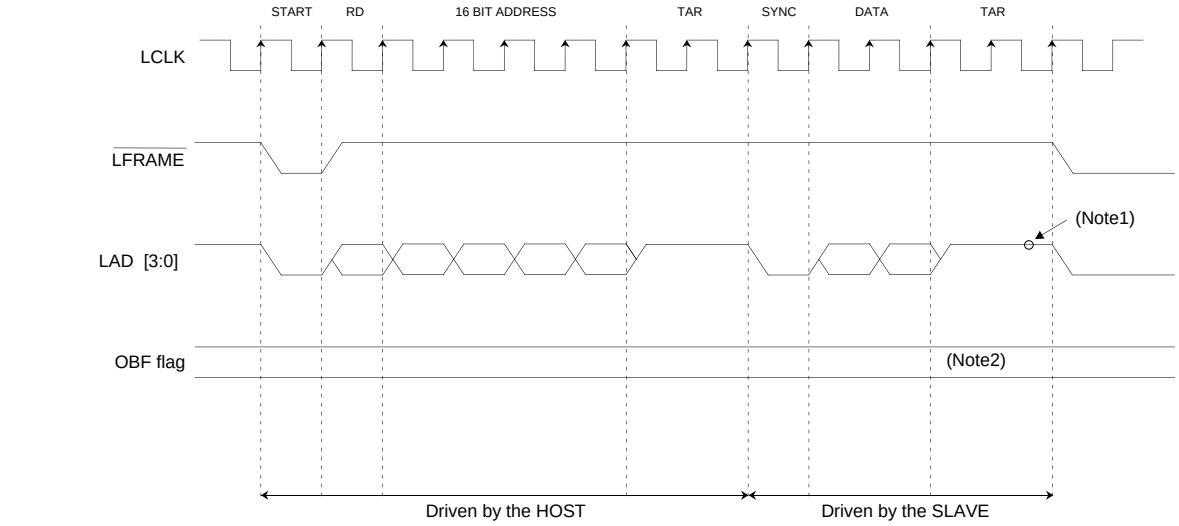
Note1 : LAD_0 to LAD_3 pins remain Hi-Z after transfer completion

Fig.GF-7 Data and Command write timing figure

● Data RD (I/O read cycle)



● Status RD (I/O read cycle)



Note1 : LAD_0 to LAD_3 pins become Hi-Z after transfer completion.
Note2 : OBF flag does not change.

Fig.GF-8 Data and Status read timing figure

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

Pin name	Name	LPC interface enable bit 02D6$_{16}$ Bit 3	OBF$_0$ output enable bit 02C9$_{16}$ Bit 0	OBF$_{00}$ output enable bit 02C9$_{16}$ Bit 1	OBF$_{01}$ output enable bit 02C9$_{16}$ Bit 2	OBF$_1$ output enable bit 02C9$_{16}$ Bit 3	OBF$_2$ output enable bit 02C9$_{16}$ Bit 4	OBF$_3$ output enable bit 02C9$_{16}$ Bit 5	HOST$_{EN}$ control bit 02C9$_{16}$ Bit 6	Input/ Output	Function
P3$_0$/LAD$_0$	LAD$_0$	1	—	—	—	—	—	—	—	I/O	LPC bus used for transmitting and receiving address, command and data between Host CPU and peripheral devices.
P3$_1$/LAD$_1$	LAD$_1$	1	—	—	—	—	—	—	—	I/O	
P3$_2$/LAD$_2$	LAD$_2$	1	—	—	—	—	—	—	—	I/O	
P3$_3$/LAD$_3$	LAD$_3$	1	—	—	—	—	—	—	—	I/O	
P3$_4$/$\overline{\text{LFRAME}}$	$\overline{\text{LFRAME}}$	1	—	—	—	—	—	—	—	I	It is used for indicating the start of LPC cycle and termination of abnormal communication cycle.
P3$_5$/$\overline{\text{LRESET}}$	$\overline{\text{LRESET}}$	1	—	—	—	—	—	—	—	I	LPC reset signal. LPC bus interface function is reset.
P3$_6$/LCLK	LCLK	1	—	—	—	—	—	—	—	I	LPC synchronous clock signal.
P4$_0$/OBF$_{00}$	OBF$_{00}$	—	0	1	0	—	—	—	—	O	Status output signal. OBF$_{00}$ output.
P4$_3$/OBF$_{01}$	OBF$_{01}$	—	1	0	1	—	—	—	—	O	Status output signal. OBF$_{01}$ output.
P4$_4$/OBF$_1$	OBF$_1$	—	—	—	—	1	—	—	—	O	Status output signal. OBF$_1$ output.
P4$_5$/OBF$_2$	OBF$_2$	—	—	—	—	—	1	—	—	O	Status output signal. OBF$_2$ output.
P4$_6$/OBF$_3$	OBF$_3$	—	—	—	—	—	—	1	—	O	Status output signal. OBF$_3$ output.

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

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

GateA20 output function

The GateA20 pin (port P42) can be controlled by LPC interface function channel 0 in hardware.
Hardware GateA20 is sharing with P42 pin. Setting "1" to bit 0 of GateA20 control reigsiter enables the hardware GateA20 function. The default value of hardware GateA20 is "1".
The GateA20 control register is shown in Fig.GF-9.
When the host CPU writes "D1" command to address 006416, and then writes data to address 006016 in succession, the value of bit 1 of the data will be output to GateA20 pin. The timing is shown in Fig.GF-10.
The GateA20 operation sequences are shown in Fig.GF-11, Fig.GF-12. As shown in the figures, there is no change in input buffer full flag(IBF0) and no input buffer full(IBF) interrupt request, but the input data bus buffer and XA2 flag are changed in these sequences.
The value of the GateA20 output pin will be held till the data next to D1 command is written in. P42 becomes I/O port and the the value of GateA20 becomes "0" when $\overline{\text{LRESET}}$ input is "L". GateA20 will be initialized even if the sequence is executed. However, the GateA20 enable bit will not be changed and GateA20 output pin will be resumed after the $\overline{\text{LRESET}}$ input becomes "H".

GateA20 control register

b7	b6	b5	b4	b3	b2	b1	b0
						0	

Symbol	Address	Reset
GA20CON	002CA16	0016

Bit Symbol	Bit name	Function	R	W
GA20EN	GateA20 enable bit	0 : P42 as GPIO 1 : Hardware GateA20 function enable	○	○
Reserved bit		Must be set to "0"	—	—
Nothing is assigned. Meanless in writing. "0" in reading.			—	—

Fig.GF-9 GateA20 control register

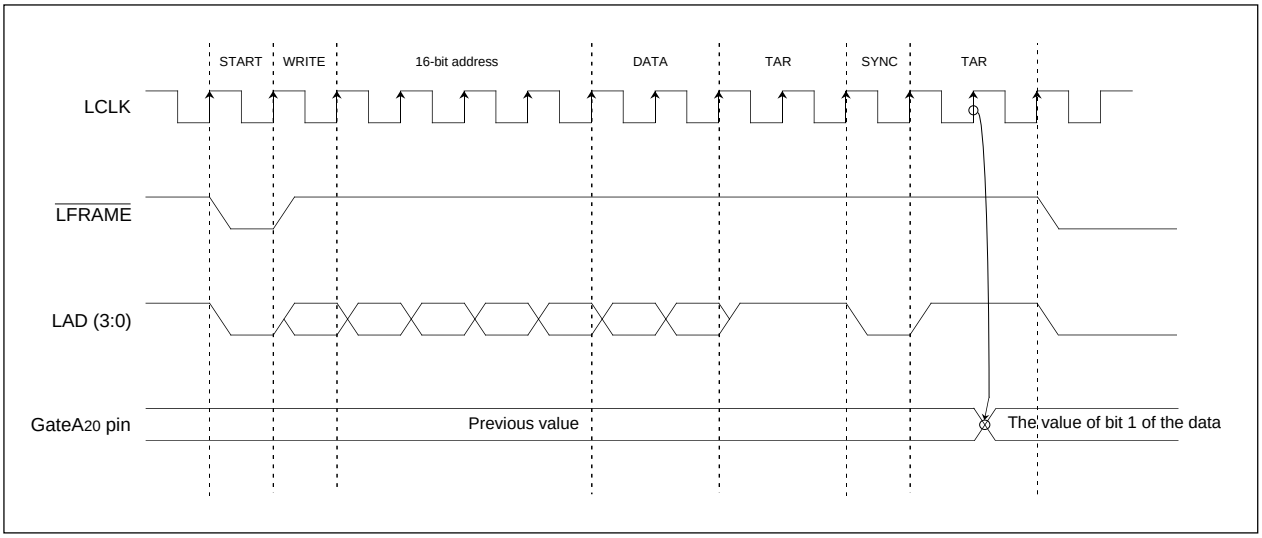


Fig.GF-10 GateA20 output timing

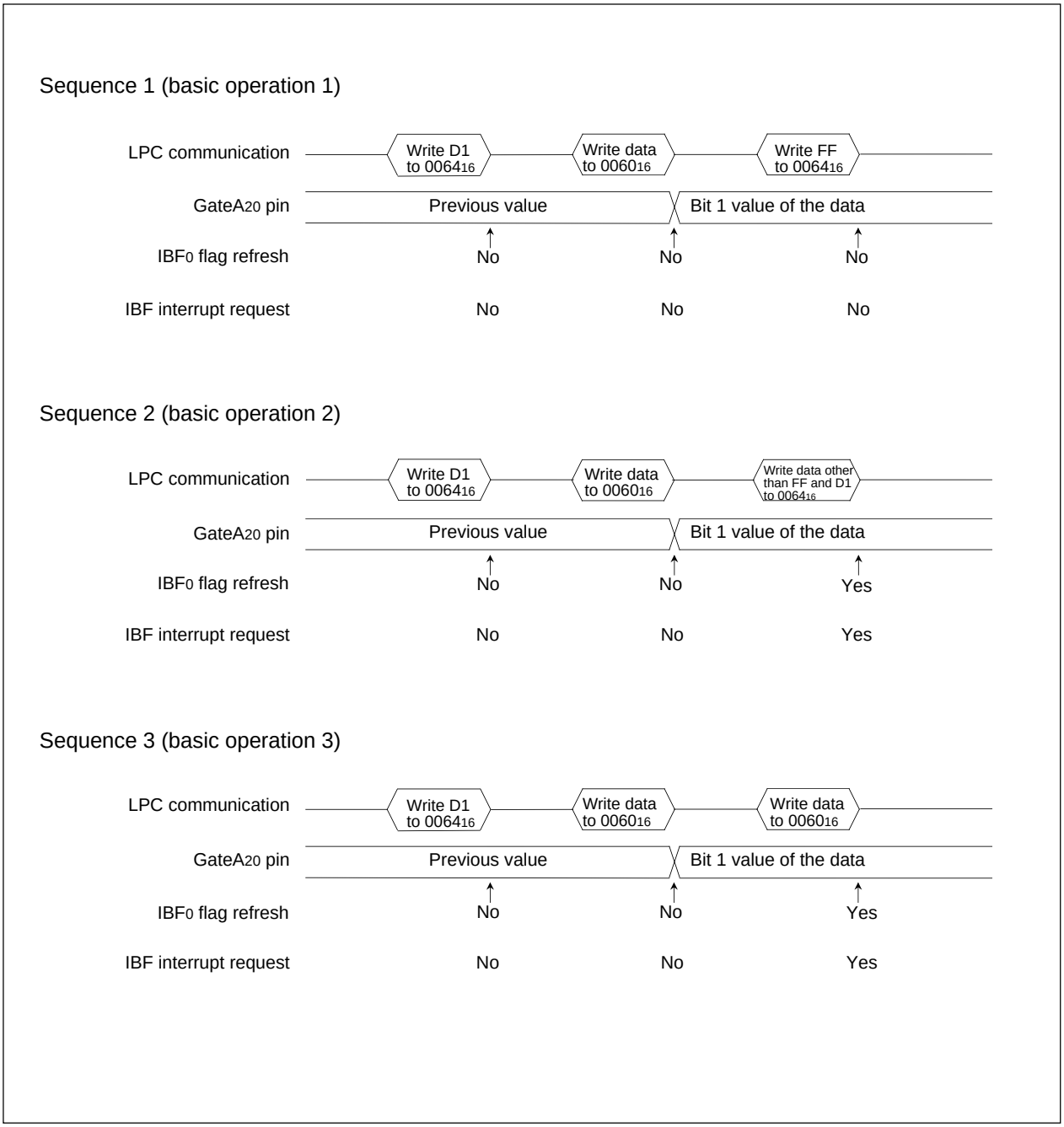


Fig.GF-11 GateA20 operation sequence (1)

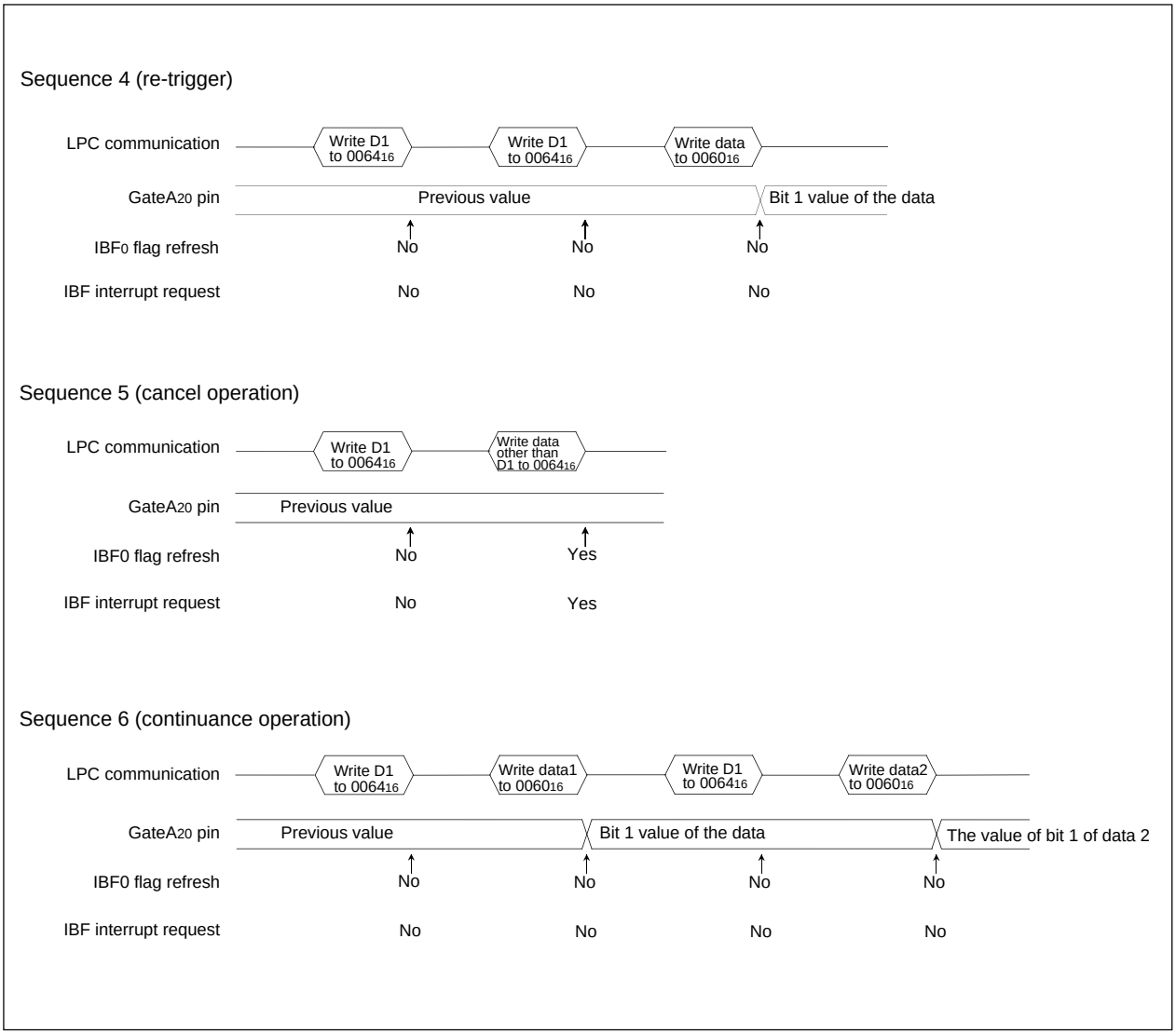


Fig.GF-12 GateA20 operation sequence (2)

Serial Interrupt Output

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

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

Table.SI-1 Specifications of serial interrupt output

Item	Specification
The factors of serial interrupt	The numbers of serial interrupt requests (numbers of channels) that can output simultaneously are 5 factors. Each interrupt factor of each channel is explained as follows. • Channel 0 ① By setting "1" to IRQi request bit (bit 5, 6 i=1,12) of IRQ request register 0, the interrupt request can be generated. ② Synchronized with the rising edge of OBF00 and OBF01 that are the host bus interface internal signals, the serial interrupt request can be generated. • Channel 1-3 ① By setting "1" to IRQ request bit (bit 5) of IRQ request register 1-3, the interrupt request can be generated. ② Synchronized with the rising edge of OBF1-3 that are the host bus interface internal signals, the serial interrupt request can be generated. • Channel 4 By setting "1" to IRQ request bit (bit 5) of IRQ request register 4, the interrupt request can be generated.
The number of frame	• Channel 0 ① Setting the IRQ1 request bit (bit 5) of IRQ request register0 to “1” or detecting the rising edge of OBF00, which is the host bus interface internal signal, selects Frame 1. ② Setting the IRQ12 request bit (bit 6) of IRQ request register0 to “1” or detecting the rising edge of OBF01, which is the host bus interface internal signal, selects Frame 12. • Channel 1-4 Selecting the frame select bit (bit 0-4) of IRQ request register1-4 selects Frame 1-15 or extend Frame 0-10.
Operation clock	The operation synchronized with LCLK (Max. 33MHz). (Note)
Clock restart	Setting the clock restart enable bit (bit 6) of serial interrupt control register0 to “1” requests the clock restart if the clock has stopped or slowed down in serial interrupt output.
Clock stop inhibition	Setting the clock stop inhibition bit (bit 5) of serial interrupt control register0 to “1” requests the inhibition of clock stop if the clock tends to stop or slow down in serial interrupt output.
OBF sync enable	Setting the OBF00, OBF01, OBF1-3 sync enable bit (bit 0-4) of serial interrupt control register0 to “1” enables the OBF synchronization.

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

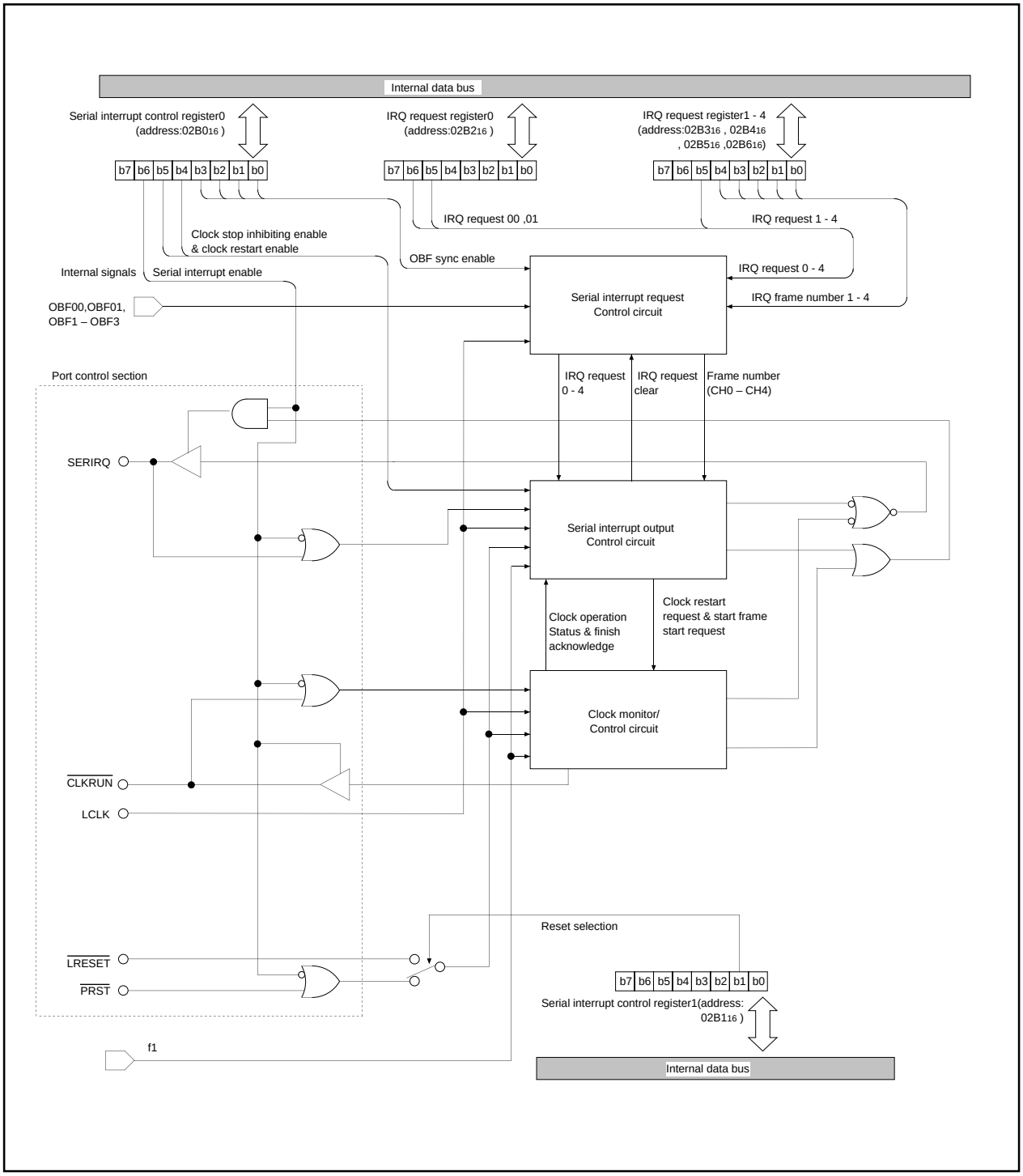


Fig.SI-1 Serial interrupt block chart

(1) Register explanation

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

● IRQ request register0 IRQR0

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

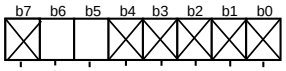
•IRQ1 request bit IR0

Setting the bit to “1” generates the serial interrupt request (Frame 1).
By setting the OBF00 sync enable bit (bit 0) of the serial interrupt control register0 to “1”, the value of IR0 is the same as that of OBF00, which is the host bus interface internal signal. When the internal signal OBF00 is "1", the serial interrupt is generated.
IR0 is cleared to "0" by writing "0" in software.
IR0 can not be cleared to "0" by software when the internal signal OBF00 is "1" if OBF00 sync enable bit is set to "1".

•IRQ12 request bit IR1

Setting the bit to “1” generates the serial interrupt request (Frame 12).
By setting the OBF01 sync enable bit (bit 1) of the serial interrupt control register0 to “1”, the value of IR1 is the same with that of OBF01, which is the host bus interface internal signal. When the internal signal OBF01 is "1", the serial interrupt is generated.
IR1 is cleared to "0" by writing "0" in software.
IR1 can not be cleared to "0" by software when the internal signal OBF01 is "1" if OBF01 sync enable bit is set to "1".

IRQ request register0



Symbol: IRQR0 Address: $02B2_{16}$ When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
Nothing is located. Meaningless in writing, “0” in reading.			–	–
IR0	IRQ1 request bit	0: No IRQ1 request 1: IRQ1 request	○	○
IR1	IRQ12 request bit	0: No IRQ12 request 1: IRQ12 request	○	○
Nothing is located. Meaningless in writing, “0” in reading.			–	–

Fig.SI-2 Configuration of IRQ request register0

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

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

•IRQ request bit IR

Setting the bit to “1” generates the serial interrupt request.
By setting the OBFj sync enable bit (bit2-4, j=1-3) of the serial interrupt control register0 to “1”, the value of IR is the same as that of OBFj, which is the host bus interface internal signal. When the internal signal OBFj is "1", the serial interrupt is generated.
IR is cleared to "0" by writing "0" in software.
IR can not be cleared to "0" by software when the internal signal OBFj is "1" if OBFj sync enable bit is set to "1".

● IRQ select bit IS0-4

The asserting frame is selected.

IRQ request register1-4

Symbol	Address	When reset
IRQR1	02B3₁₆	00₁₆
IRQR2	02B4₁₆	00₁₆
IRQR3	02B5₁₆	00₁₆
IRQR4	02B6₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
IS0	Frame select bit	b4b3b2b1b0 IRQ Frame 0 0 0 0 0 : No SERIRQ output 0 0 0 0 1 : Frame 1 0 0 0 1 0 : Frame 2 0 0 0 1 1 : Frame 3 0 0 1 0 0 : Frame 4 0 0 1 0 1 : Frame 5 0 0 1 1 0 : Frame 6 0 0 1 1 1 : Frame 7 0 1 0 0 0 : Frame 8 0 1 0 0 1 : Frame 9 0 1 0 1 0 : Frame 10 0 1 0 1 1 : Frame 11 0 1 1 0 0 : Frame 12 0 1 1 0 1 : Frame 13 0 1 1 1 0 : Frame 14 0 1 1 1 1 : Frame 15 1 0 0 0 0 : Can’t select 1 0 0 0 1 : Can’t select 1 0 0 1 0 : Can’t select 1 0 0 1 1 : Can’t select 1 0 1 0 0 : Can’t select 1 0 1 0 1 : Extend Frame 0 1 0 1 1 0 : Extend Frame 1 1 0 1 1 1 : Extend Frame 2 1 1 0 0 0 : Extend Frame 3 1 1 0 0 1 : Extend Frame 4 1 1 0 1 0 : Extend Frame 5 1 1 0 1 1 : Extend Frame 6 1 1 1 0 0 : Extend Frame 7 1 1 1 0 1 : Extend Frame 8 1 1 1 1 0 : Extend Frame 9 1 1 1 1 1 : Extend Frame 10	○	○
IS1			○	○
IS2			○	○
IS3			○	○
IS4			○	○
IR	IRQ request bit	0: No IRQ request 1: IRQ request	○	○
Nothing is located. Meaningless in writing, “0” in reading.				

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

● Serial interrupt control register0 SERCON0

The operation condition of serial interrupt is set.

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

By setting the bit to “1”, the rising edge of OBFi output to host bus interface generates the serial interrupt request synchronously.

•Clock stop inhibition bit SUPEN

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

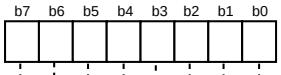
•Clock restart enable bit RUNEN

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

•Serial interrupt enable bit IRQEN

0: SERIRQ, $\overline{\text{PRST}}$, $\overline{\text{CLKRUN}}$ are I/O ports.
1: SERIRQ, $\overline{\text{PRST}}$, $\overline{\text{CLKRUN}}$ are serial interrupt function ports.

Serial interrupt control register0



Symbol	Address	When reset
SERCON0	$02B0_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SEN00	OBF00 sync enable bit	0: Sync inhibition 1: Sync enable	○	○
SEN01	OBF01 sync enable bit	0: Sync inhibition 1: Sync enable	○	○
SEN1	OBF1 sync enable bit	0: Sync inhibition 1: Sync enable	○	○
SEN2	OBF2 sync enable bit	0: Sync inhibition 1: Sync enable	○	○
SEN3	OBF3 sync enable bit	0: Sync inhibition 1: Sync enable	○	○
SUPEN	Clock stop inhibition bit	0: Stop control operation 1: No stop control operation	○	○
RUNEN	Clock restart enable bit	0: No clock restart 1: Clock restart	○	○
IRQEN (Note 1)	Serial interrupt enable bit	0: Serial interrupt inhibition 1: Serial interrupt enable	○	○

Note 1 : By setting this bit to “1”, $P4_3$, $P4_5$ and $P4_6$ are selected as SERIRQI/O, $\overline{\text{PRST}}$ input and $\overline{\text{CLKRUN}}$ I/O ports respectively.
The output of $\overline{\text{CLKRUN}}$ pin is N channel open drain type.
$P4_5/\overline{\text{PRST}}$ functions as GPIO port even when the bit is "1" if bit 1 of serial interrupt control register 1 (Address $02B1_{16}$) is "1".

Fig.SI-4 Configuration of serial interrupt control register0

● Serial interrupt control register1 SERCON1

The register is for setting the pins of serial interrupt.

•Reset selection bit RSEL

0: The input of $\overline{\text{PRST}}$ is the reset signal.
1: The input of $\overline{\text{LRESET}}$ is the reset signal. (Note1)
Note 1: The $\overline{\text{PRST}}$ pin becomes I/O port if setting the bit to “1”.

•OBF0 mergence function

By setting the bit to “1”, the signal, which is logically OR by OBF00 and OBF01 signals from LPC bus interface, will output to IRQ1 and IRQ12 of serial interrupt circuit.
Fig.SI-6 shows the selection circuit controlled by the bit.

With the function, the IRQ1 and IRQ12 request bits can be cleared simultaneously by H/W at the read of output data buffer from system if both IRQ1 and IRQ12 request bits are set in the case that IRQ1 request bit (or IRQ12 request bit) is set after that of IRQ12 (or IRQ1) because of the overwrite to the output data buffer.

Serial interrupt control register 1

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	✕	✕		0		0

Symbol	Address	When reset
SERCON1	$02B1_{16}$	00_{16}

Bit symbol	Bit symbol	Function	R	W
Reserved bit		Must be “0”.	—	—
RSEL	Reset selection bit	0 : Select $\overline{\text{PRST}}$ pin. 1 : Select $\overline{\text{LRESET}}$ pin.	○	○
Reserved bit		Must be “0”.	—	—
OBF0MRG	OBF0 mergence bit	0 : OBF00 and OBF01 are independent. 1 : OBF00 and OBF01 are merged (logical OR).	○	○
Nothing is allocated. Meaningless in writing. “0” in reading.			—	—

Fig.SI-5 Configuration of serial interrupt control register1

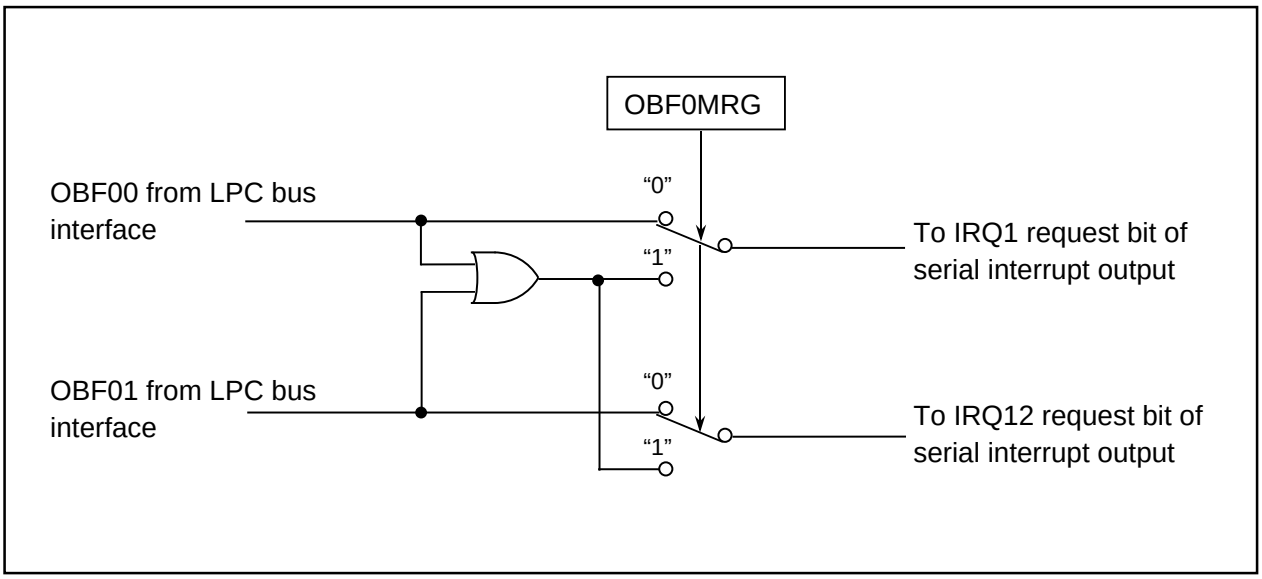
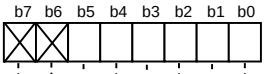


Fig.SI-6 The selection circuit controlled by OBF0MRG

● Serial interrupt control register2 SERCON2

The polarity of serial interrupt output can be selected by bit 0 to bit 5 of serial interrupt control register 2.
When the bit is set to "0":
If there is a request, Hiz-Hiz-Hiz
If there is no request, L-H-Hiz
When the bit is set to "1":
If there is a request, L-H-Hiz
If there is no request, Hiz-Hiz-Hiz
Only the default value of bit 4 (serial interrupt polarity bit 3) of serial interrupt control register 2 after reset is "1".
Fig.SI-7 shows the configuration of serial interrupt control register 2.

Serial interrupt control register 2



Symbol: SERCON2 | Address: $02B7_{16}$ | When reset: 10_{16}

Bit symbol	Bit name	Function	R	W
SERSEL00	Serial interrupt polarity selection bit 00	Select serial interrupt output polarity of IRQ1 in channel 0. (Note)	○	○
SERSEL01	Serial interrupt polarity selection bit 01	Select serial interrupt output polarity of IRQ12 in channel 0. (Note)	○	○
SERSEL1	Serial interrupt polarity selection bit 1	Select serial interrupt output polarity of IRQ in channel 1. (Note)	○	○
SERSEL2	Serial interrupt polarity selection bit 2	Select serial interrupt output polarity of IRQ in channel 2. (Note)	○	○
SERSEL3	Serial interrupt polarity selection bit 3	Select serial interrupt output polarity of IRQ in channel 3. (Note)	○	○
SERSEL4	Serial interrupt polarity selection bit 4	Select serial interrupt output polarity of IRQ in channel 4. (Note)	○	○
Nothing is allocated. Meaningless in writing. "0" in reading.			—	—

Note: "0" : If there is a request Hiz-Hiz-Hiz
If there is no request L-H-Hiz
"1" : If there is a request L-H-Hiz
If there is no request Hiz-Hiz-Hiz

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

(2) The operation of serial interrupt

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

● The timing of serial interrupt cycle

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

① Start frame
The start frame will be detected if the SERIRQ remains “L” in 4-8 clock cycles.
②IRQ data frame
Each IRQ data frame is 3 clock cycles.
•Channel 0-2,4:If the IRQ request bit is "0", then the SERIRQ is driven to “L” during the 1st clock cycle of the corresponding data frame, to “H” during the 2nd clock cycle, to high impedance during the 3rd clock cycle. If the IRQ request bit is "1", then the SERIRQ is high impedance during all of the 3 clock cycles.
•Channel 3:If the IRQ request bit is "0", then the SERIRQ is high impedance during all of the 3 clock cycles. If the IRQ request bit is "1", then the SERIRQ is driven to “L” during the 1st clock cycle of the corresponding data frame, to “H” during the 2nd clock cycle, to high impedance during the 3rd clock cycle.
③Stop frame
The stop frame will be detected if the SERIRQ remains “L” in 2 or 3 clock cycles. The next operation mode is quiet mode if the length of “L” is 2 clock cycles, the continuous mode mode if the length is 3 clock cycles.

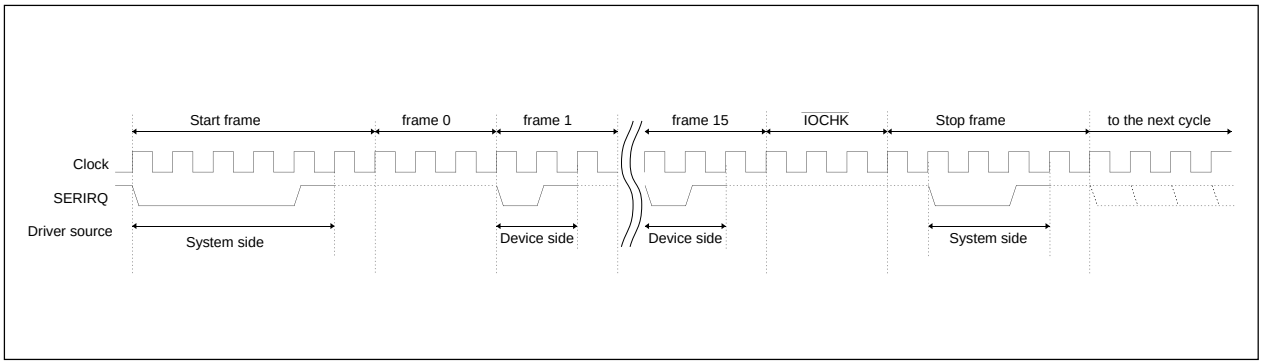


Fig.SI-8 Basic timing of serial interrupt cycle

● Operation mode

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

①Continuous mode

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

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

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

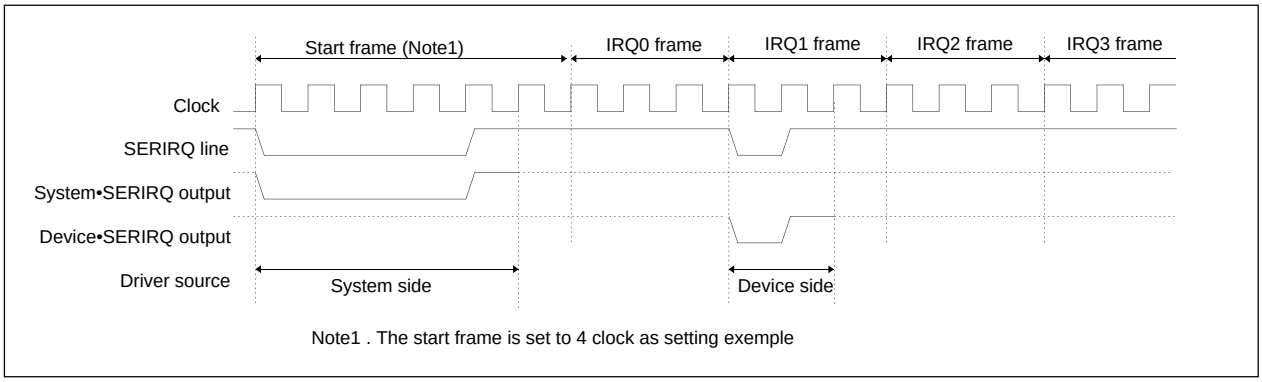


Fig.SI-9 Timing diagram of continuous mode

②Quiet mode

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

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

Note 1: If the sum of length of “L” that is driven by the device in the 1st clock cycle and by the host in the rest clock cycles is within 4-8 clock cycles, the start frame will be detected.
If the sum of length of “L” is less than 4 clock cycles or more than 9 clock cycles, the start frame will not be detected and the next start (the falling edge of SERIRQ) is waited.

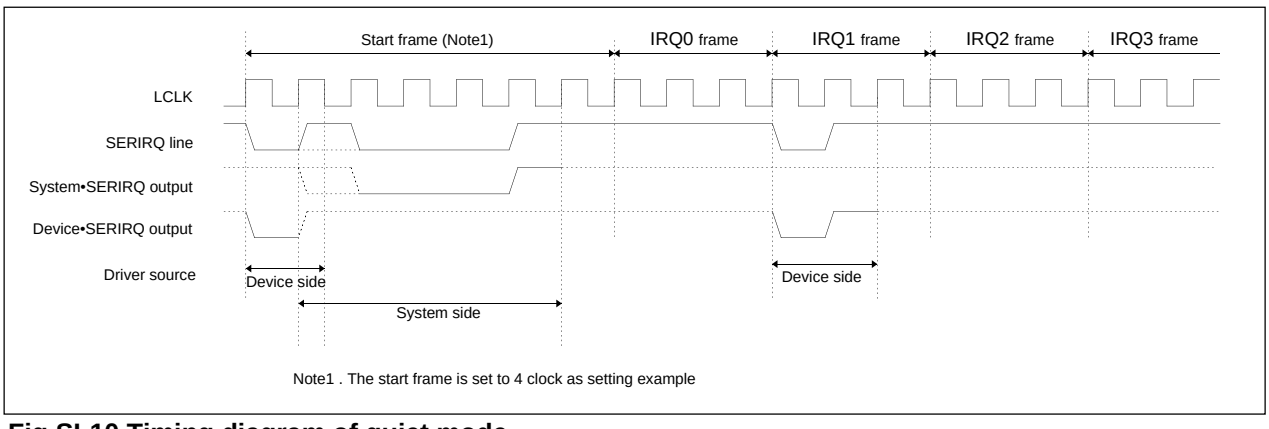


Fig.SI-10 Timing diagram of quiet mode

(3) Clock restart/ stop inhibition request

Asserting the $\overline{\text{CLKRUN}}$ signal can request to restart or maintain the clock which stops or slows down or request the host to tend to stop or slow down.
Fig.SI-11 shows an example of timing of clock restart request, Fig.SI-12 shows an example of timing of clock stop inhibition request.
①Clock restart operation
Setting the clock restart bit of serial interrupt control register0 to “1” will request the clock restart if the clock has slowed down or stopped at serial interrupt request.

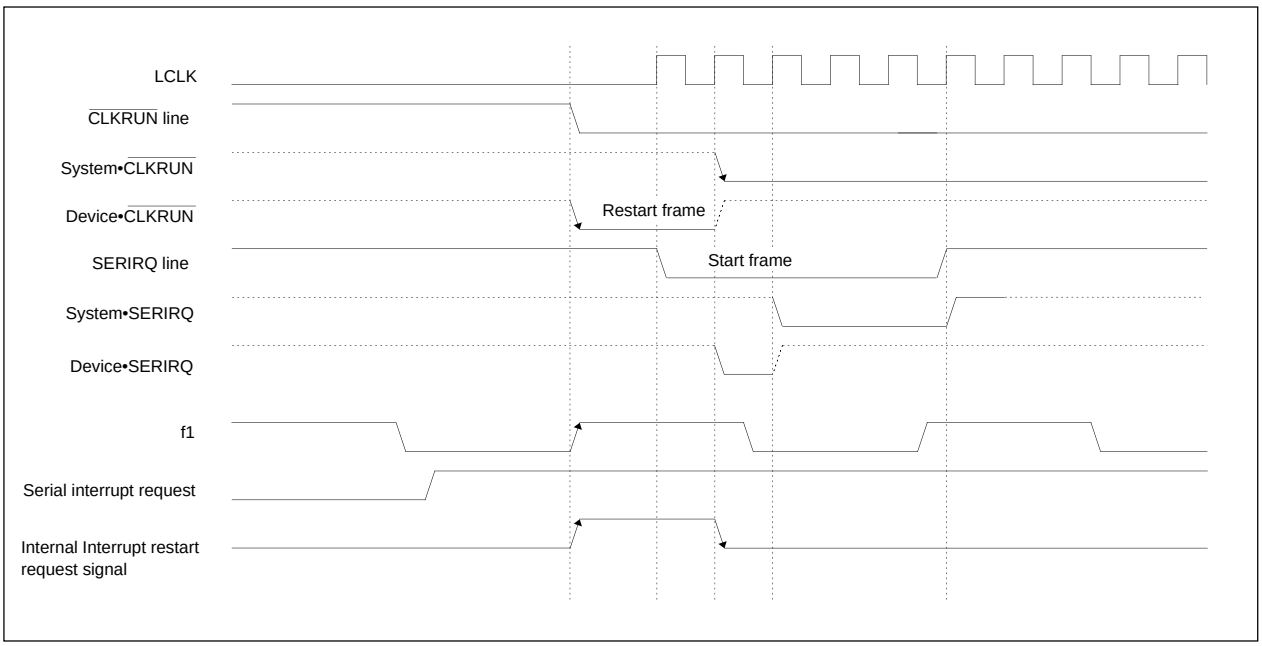


Fig.SI-11 Timing diagram of clock restart request

②Clock stop inhibition request
Setting the clock stop inhibition bit of serial interrupt control register0 to “1” will request the inhibition of clock stop if the clock tends to stop or slow down during all the period of serial interrupt output.

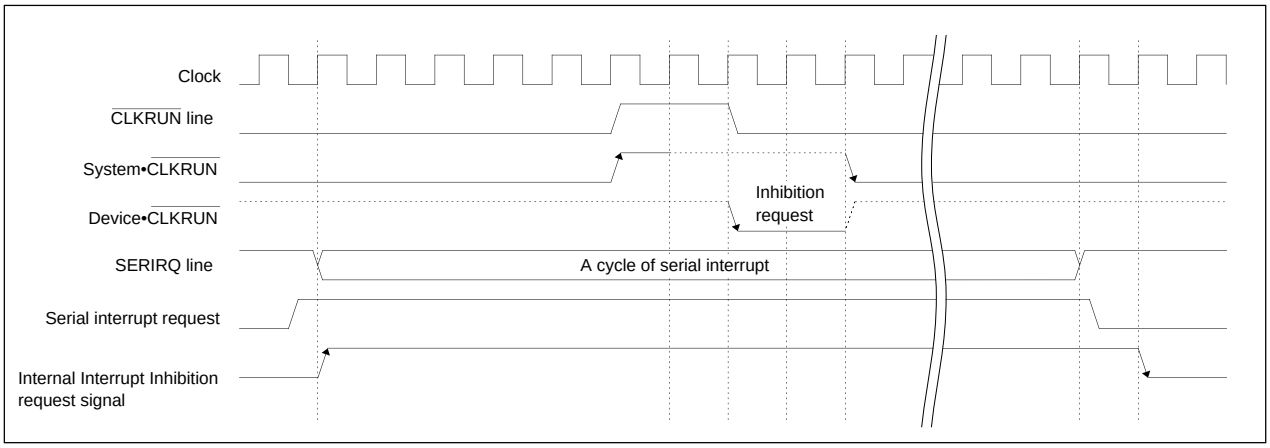


Fig.SI-12 Example of timing of clock stop inhibition request

MULTI-MASTER I2C-BUS INTERFACE

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

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

Item	Function
Format	Based on Philips I2C-BUS standard: 10-bit addressing format 7-bit addressing format High-speed clock mode Standard clock mode
Communication mode	Based on Philips I2C-BUS standard: Master transmission Master reception Slave transmission Slave reception
SCL clock frequency	16.1kHz to 400kHz (at VIIC = 4MHz)

***VIIC=I2C system clock**

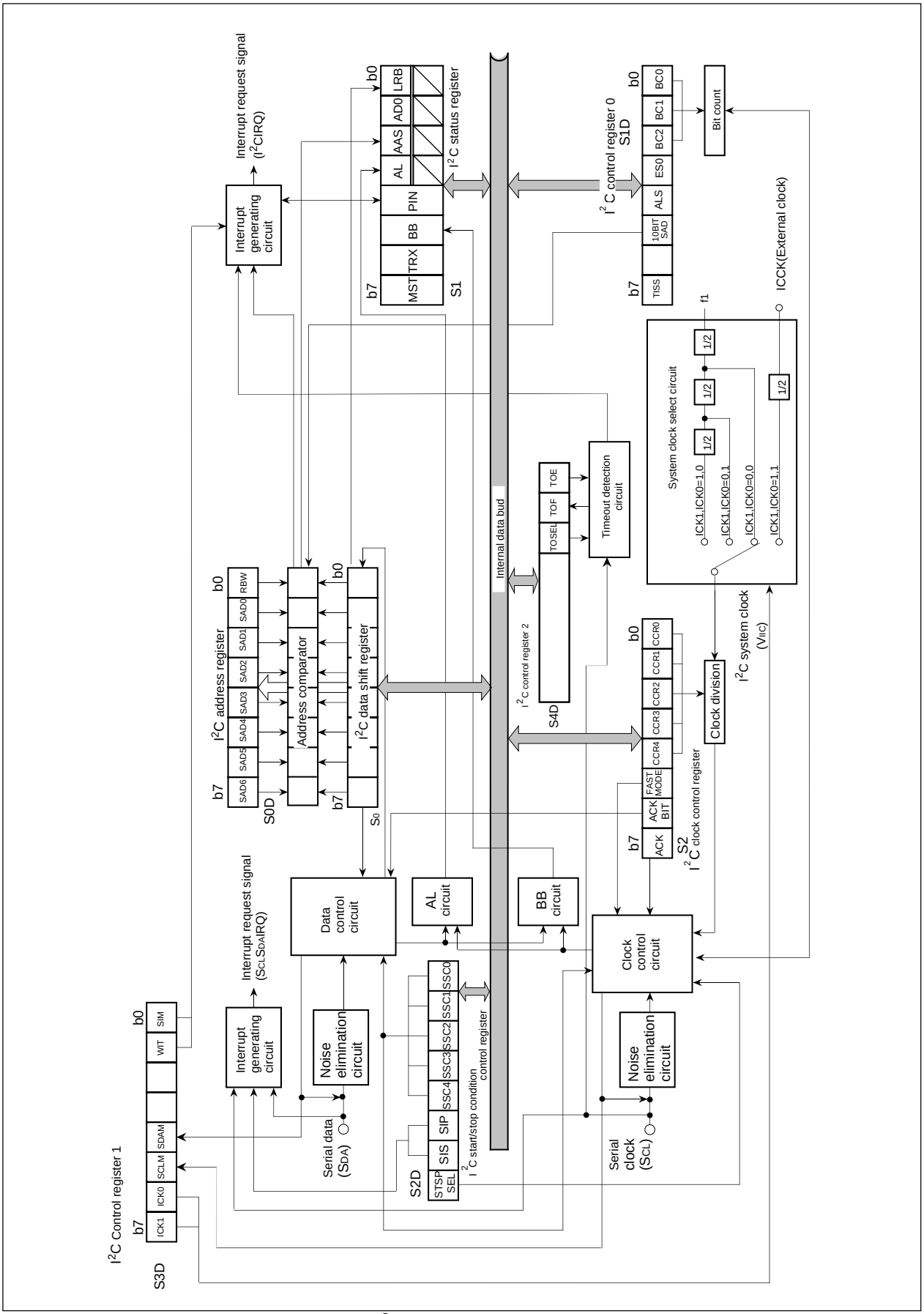


Fig.GC-1 Block diagram of multi-master I2C-BUS interface

I2C Data Shift Register

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

I2C data shift register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
S0i(i=0,1)	032016,033016	--

Function	R	W
Transmission data /receiving data are stored. In the master transmission mode, the start condition/ stop condition are triggered by writing data to the register (refer to the section on the method to generate the start/stop condition). The transmission/receiving are started synchronized with SCL.	○	○ Note

Note The write is only enabled when bus interface enable bit (ES0 bit) is "1".
Because the register is used both for storing transmission data/receiving data, the transmission data should be written after the receiving data are read out before writing transmission data to this register.

Fig.GC-2 I2C data shift register

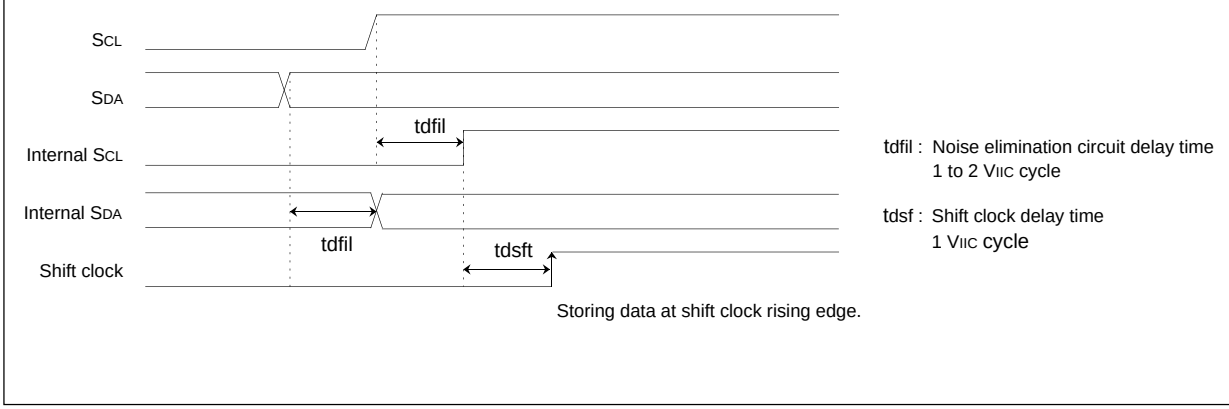


Fig.GC-3 The timing of receiving data stored to I2C data shift register

I2C Address Register

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

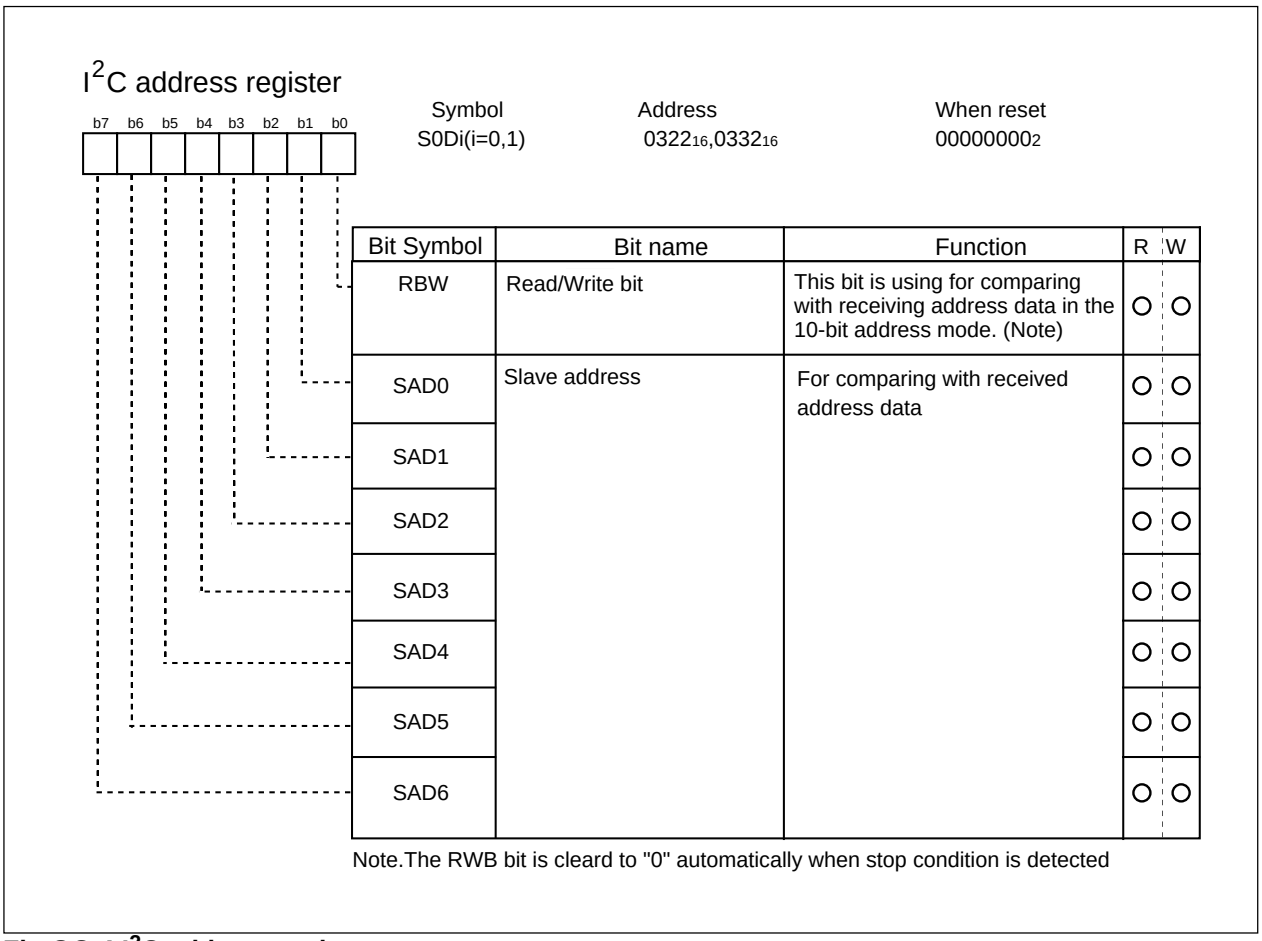
•Bit 0: Read/write bit (RBW)

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

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

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

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



Bit Symbol	Bit name	Function	R	W
RBW	Read/Write bit	This bit is using for comparing with receiving address data in the 10-bit address mode. (Note)	○	○
SAD0	Slave address	For comparing with received address data	○	○
SAD1			○	○
SAD2			○	○
SAD3			○	○
SAD4			○	○
SAD5			○	○
SAD6			○	○

Note.The RWB bit is cleard to "0" automatically when stop condition is detected

Fig.GC-4 I2C address register

I2C Clock Control Register

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

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

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

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

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

•Bit 6: ACK bit (ACK BIT)

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

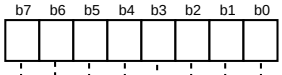
*ACK clock: Clock for acknowledgment

•Bit 7: ACK clock bit (ACK)

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

Note: Except for ACK bit (ACKBIT), do not write data into the I2C clock control register during transfer. If data is written during transfer, the I2C clock generator is reset, so that data cannot be transferred normally.

I^2C clock control register



Symbol	Address	When reset
S2i(i=0,1)	0324_{16},0334_{16}	00000000_2

Bit Symbol	Bit name	Function	R	W
CCR0	SCL frequency control bits	Refer to table.GC-2	○	○
CCR1			○	○
CCR2			○	○
CCR3			○	○
CCR4			○	○
FAST MODE	SCL mode specification bit	0 : Standard clock mode 1 : High-speed clock mode	○	○
ACK BIT	ACK bit	0 : ACK is returned 1 : ACK is not returned	○	○
ACK	ACK clock bit	0 : No ACK clock 1 : ACK clock	○	○

Fig.GC-5 I^2C clock register

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

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

Notes1: Duty of SCL clock output is 50 %. The duty becomes 35 to 45 % only when the high-speed clock mode is selected and CCR value = 5 (400 kHz, at VIIC = 4 MHz). “H” duration of the clock fluctuates from –4 to +2 machine cycles in the standard clock mode, and fluctuates from –2 to +2 machine cycles in the high-speed clock mode. In the case of negative fluctuation, the frequency does not increase because “L” duration is extended instead of “H” duration reduction. These are value when SCL clock synchronization by the synchronous function is not performed. CCR value is the decimal notation value of the SCL frequency control bits CCR4 to CCR0.

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

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

VIIC/(8 X CCR value) Standard clock mode

VIIC/(4 X CCR value) High-speed clock mode (CCR value ≠ 5)

VIIC/(2 X CCR value) High-speed clock mode (CCR value = 5)

Do not set 0 to 2 as CCR value regardless of VIIC frequency.

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

I2C Control Register 0

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

•Bits 0 to 2: Bit counter (BC0–BC2)

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

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

•Bit 3: I2C interface enable bit (ES0)

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

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

1)Set MST = "0", TRX = "0", PIN = "1", BB ="0", AL = "0", AAS = "0", and AD0 = "0", of I2C status register (Address : 032816, 033816)

2)Writing data to I2C data shift register (Address : 032016, 033016) is inhibited.

3)The TOF bit of I2C control register (Address : 032716,033716) is cleared to "0"

4)I2C system clock (VIIC) is stopped and the interval counter, flags are initialized.

•Bit 4: Data format selection bit (ALS)

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

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

This bit selects a slave address specification format. When this bit is set to "0", the 7-bit addressing format is selected. In this case, only the high-order 7 bits (slave address) of the I2C address register (address 032316,033316) are compared with address data. When this bit is set to "1", the 10-bit addressing format is selected, and all the bits of the I2C address register are compared with address data.

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

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

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

Flags are processed as follows:

1)Set MST = "0", TRX = "0", PIN = "1", BB ="0", AL = "0", AAS = "0", and AD0 = "0", of I2C status register (Address : 032816, 033816)

2)The TOF bit of I2C control register (Address : 032716,033716) is cleared to "0"

3)The interval counter, flags are initialized.

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

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

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

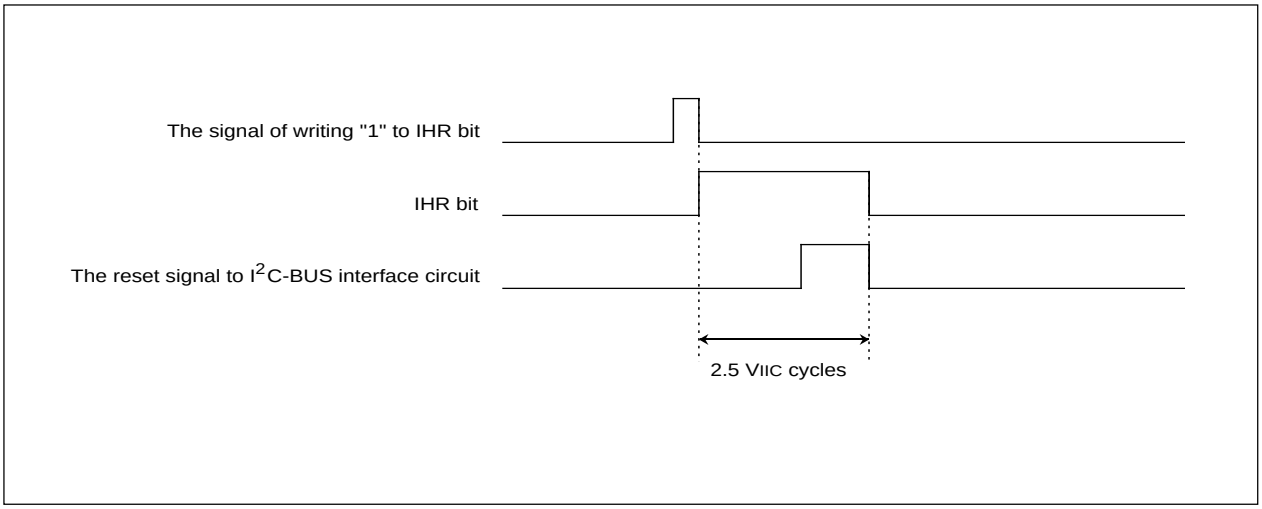


Fig.GC-6 The timing of reset to the I2C-BUS interface circuit

I2C control register 0

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: S1Di(i=0,1) — Address: 0323_{16},0333_{16} — When reset: 00000000_2

Bit Symbol	Bit name	Function	R	W
BC0	Bit counter (Number of transmitting/ receiving bits)	b2 b1 b0 (Note) 0 0 0 : 8 0 0 1 : 7 0 1 0 : 6 0 1 1 : 5 1 0 0 : 4 1 0 1 : 3 1 1 0 : 2 1 1 1 : 1	○	○
BC1			○	○
BC2			○	○
ES0	I2C-BUS interface enable bit	0 : Disable 1 : Enable	○	○
ALS	Data format selection bit	0 : Addressing format 1 : Free data format	○	○
DBIT SAD	Addressing format selection bit	0 : 7-bit addressing format 1 : 10-bit addressing format	○	○
IHR	I2C-BUS interface reset bit	0 : Release of reset (auto) 1 : Reset	○	○
TISS	I2C-BUS interface pin input level selection bit	0 : I2C-BUS input 1 : SMBUS input	○	○

Note In the following status, the bit counter will be cleared automatically
- Start condition/stop condition is detected
- Right after the completion of 1 byte data transmission
- Right after the completion of 1 byte data receiving

Fig.GC-7 I2C control register

I2C Status Register

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

•Bit 0: Last receive bit (LRB)

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

•Bit 1: General call detecting flag (AD0)

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

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

•Bit 2: Slave address comparison flag (AAS)

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

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

• The address data, which following the start conduction, is same with upper bits data of I2C address register(Address 003216,033216)

• A general call is received.

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

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

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

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

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

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

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

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

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

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

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

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

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

•Bit 5: Bus busy flag (BB)

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

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

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

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

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

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

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

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

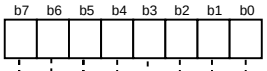
Additionally, the clocks required for data communication are generated on the SCL.

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

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

Note: START condition duplication preventing function The MST, TRX, and BB bits is set to “1” at the same time after confirming that the BB flag is “0” in the procedure of a START condition occurrence. However, when a START condition by an other master device occurs and the BB flag is set to “1” immediately after the contents of the BB flag is confirmed, the START condition duplication preventing function makes the writing to the MST and TRX bits invalid. The duplication preventing function becomes valid from the rising of the BB flag to reception completion of slave address. Refer to the method on the start condition generation in detail.

I2C status register



Symbol: S1i(i=0,1) | Address: 0328_{16},0338_{16} | When reset: $0001000X_2$

Bit symbol	Bit name	Function	R	W
LRB	Last receive bit	0 : Last bit = "0" 1 : Last bit = "1"	○	(Note1)
AD0	General call detecting flag	0 : No general call detected 1 : General call detected	○	(Note1)
AAS	Slave address comparison flag	0 : Address disagreement 1 : Address agreement	○	(Note1)
AL	Arbitration lost detection flag	0 : Not detected 1 : Detected	○	(Note1)
PIN	I2C-BUS interface interrupt request bit	0 : Interrupt request issued 1 : No interrupt request issued	○	(Note2)
BB	Bus busy flag	0 : Bus free 1 : Bus busy	○	(Note1)
TRX	Communication mode specification bits	b7 b6 0 0 : Slave receive mode 0 1 : Slave transmit mode 1 0 : Master receive mode 1 1 : Master transmit mode	○	○ (Note3)
MST			○	○ (Note3)

Note1. This bit is read only if it is used for the status check.
How to write this bit, please refer to start condition/stop condition generating method.
Note2. The bit can be read and only can be written with "0" by software.
Note3. Refer to the method of start condition generation on how to write these bits.
How to write this bit, please refer to start condition/stop condition generating method.

Fig.GC-8 I2C status register

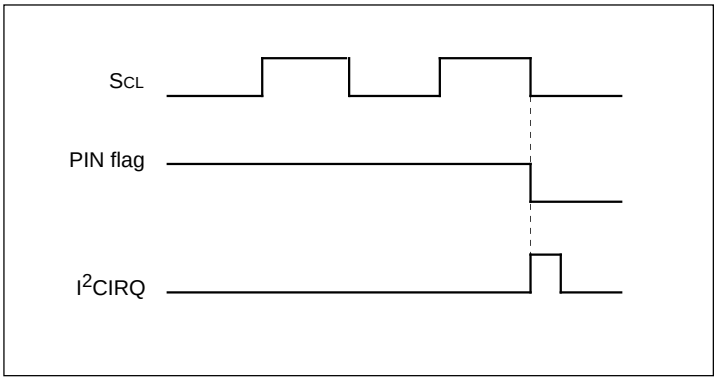


Fig.GC-9 Interrupt request signal generating timing

I2C0, I2C1 control register 1

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

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

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

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

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

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

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

The timing of I2C interrupt generation	The method of communication restart
1)Synchronized with the falling edge of the last data bit clock	The execution of writing to ACKBIT of I2C clock control register. (Do not write to I2C data shift register. The processing of ACK clock would be incorrect.)
2)Synchronized with the falling edge of the ACK clock	The execution of writing to I2C data shift register

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

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

When ES0 bit of I2C control register 0 is set to "1", P61/P63 and P60/P62 function as SCL and SDA respectively. However, if PED is set to "1", SDA functions as output port so as to SCL if PEC is set to "1". In this case, if "0" or "1" is written to the port register, the data can be output on to the I2C-BUS regardless of the internal SCL/SDA output signals. The functions of SCL/SDA are returned back by setting PED/PEC to "1" again.
If the ports are set in input mode, the values on the I2C-BUS can be known by reading the port register regardless of the values of PED and PEC.Table GC-4 shows the port specification.

Table.GC-4 Ports specifications

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

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

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

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

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

Table.GC-5 I2C system clock selecting bits

ICK1	ICK0	I2C system clock
0	0	VIIC = 1 / 2f1
0	1	VIIC = 1 / 4f1
1	0	VIIC = 1 / 8f1
1	1	VIIC = 1 / 2ICCK

Note: f1 = f(XIN)
ICCK = External I2C clock

•The address reception in STOP mode /WAIT mode

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

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

Mode	The setting content
STOP mode	The external clock is selcted as the I2C system clock (ICK1 = 1, ICK0 = 1) and the external I2C clock is supplied by ICCK.
WAIT mode	The external clock is selcted as the I2C system clock (ICK1 = 1, ICK0 = 1) and the external I2C clock is supplied by ICCK.
	Select the peripheral function clock stop bit CMO2 (bit 2 of the system clock control register 0, address : 000616) to the state of not stopping f1,f8,f32 (CMO2 = 0) when in WAIT mode, and then execute the WAIT command.
Low power consumption mode	The external clock is selcted as the I2C system clock (ICK1 = 1, ICK0 = 1) and the external I2C clock is supplied by ICCK.

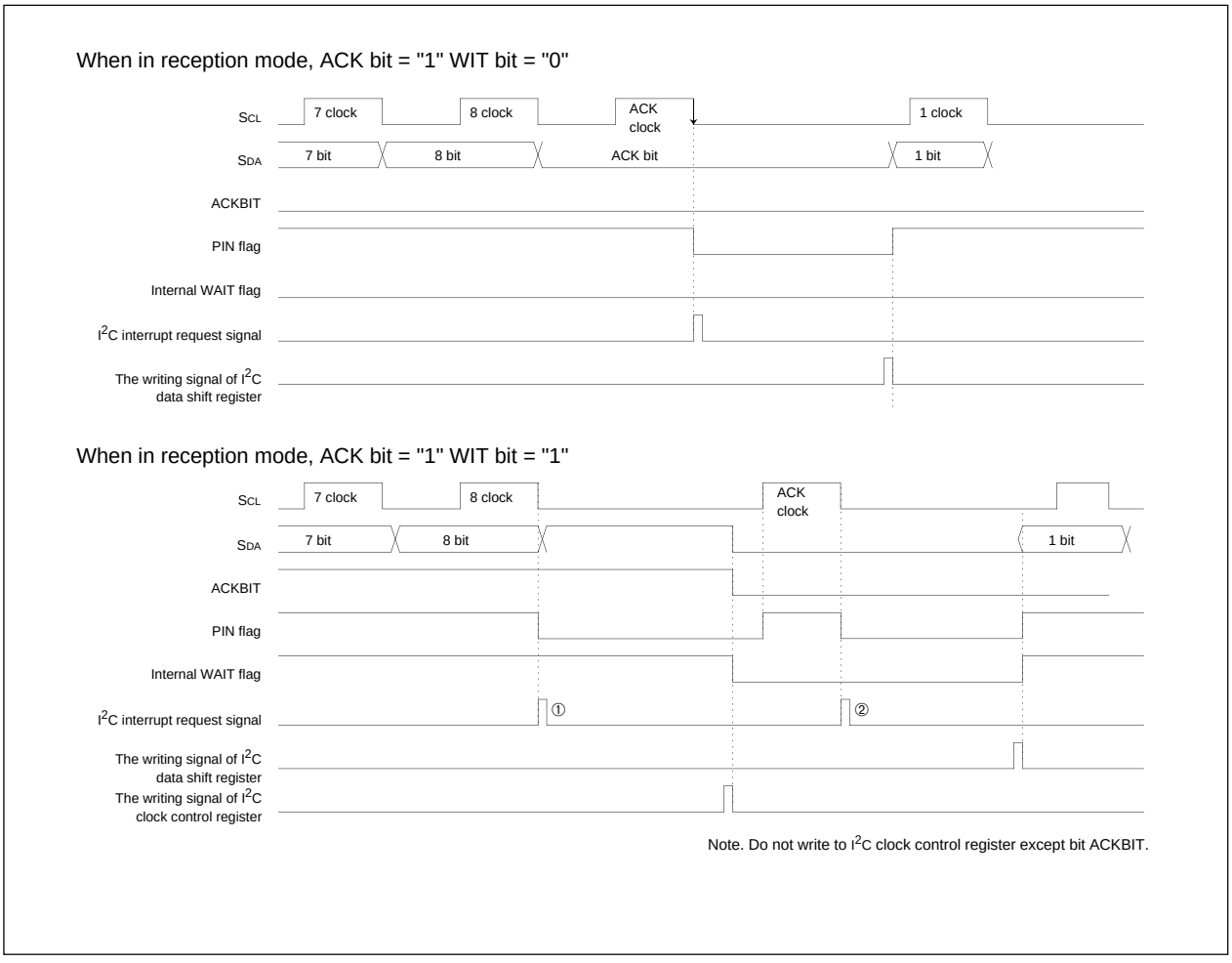


Fig.GC-10 The timing of the interrupt generation at the completion of data reception

I^2C control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
S3Di(i=0,1)	0326_{16},0336_{16}	00110000_2

Bit Symbol	Bit name	Function	R	W
SIM	The interrupt enable bit of STOP condition detection	0 : Disable the interrupt of STOP condition detection 1 : Enable the interrupt of STOP condition detection	○	○
WIT	The interrupt enable bit of at the completion of data reception	0 : Disable 1 : Enable When in NACK setting (ACK bit = "0") please write "0"	○	○
PED	SDAi/Port function switching bit	0 : SDA I/O pin(enable ES0 = 1) 1 : GPIO(enable ES0 = 1)	○	○
PEC	SCLi/Port function switching bit	0 : SCL I/O pin(enable ES0 = 1) 1 : GPIO(enable ES0 = 1)	○	○
SDAM	The logic value monitor bit of SDA output	0 : SDA output logic value = "0" 1 : SDA output logic value = "1"	○	—
SCLM	The logic value monitor bit of SCL output	0 : SCL output logic value = "0" 1 : SCL output logic value = "1"	○	—
ICK0	I^2C system clock selection bits	b7 b6 0 0 : $V_{IIC}=1/2f_1$ 0 1 : $V_{IIC}=1/4f_1$ ✲ $f_1=f(X_{IN})$ 1 0 : $V_{IIC}=1/8f_1$ 1 1 : $V_{IIC}=1/2ICCK$ ✲ ICCK=External I^2C clock	○	○
ICK1			○	○

Fig.GC-11 I^2C control register 1

I2C control register 2

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

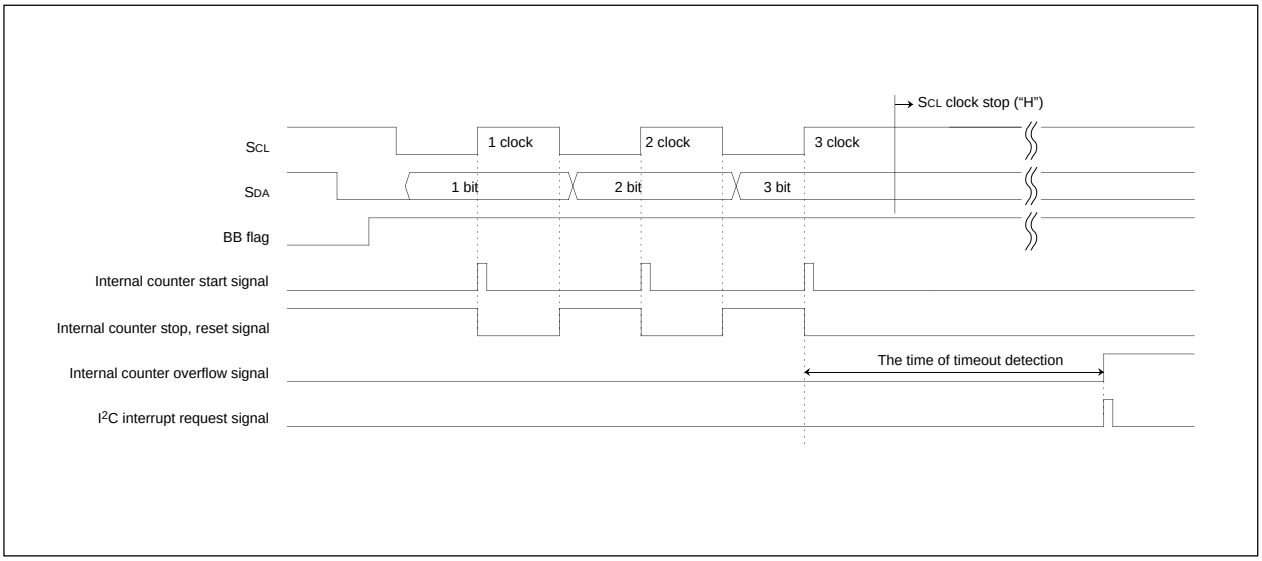


Fig.GC-12 The timing of timeout detection

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

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

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

•Bit1: Time out detection flag (TOF)

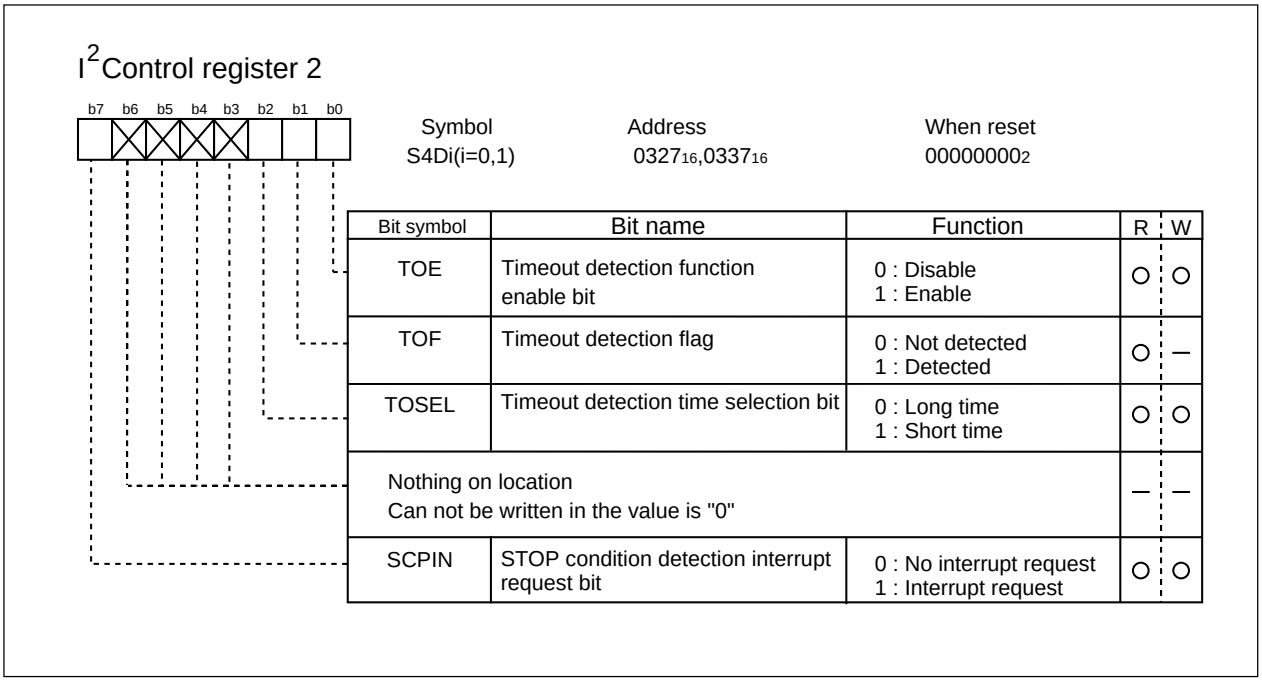
The bit is the flag showing timeout detection status. If the time which is calculated by the internal counter overflows, the time out detection flag (TOF) becomes to “1”, and at the same time the I2C interrupt request signal is generated.

•Bit2: timeout detection time selection bit (TOSEL)

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

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

VIIC(MHz)	Long time mode	Short time mode
4	16.4	4.1
2	32.8	8.2
1	65.6	16.4



Bit symbol	Bit name	Function	R	W
TOE	Timeout detection function enable bit	0 : Disable 1 : Enable	○	○
TOF	Timeout detection flag	0 : Not detected 1 : Detected	○	—
TOSEL	Timeout detection time selection bit	0 : Long time 1 : Short time	○	○
Nothing on location Can not be written in the value is "0"			—	—
SCPIN	STOP condition detection interrupt request bit	0 : No interrupt request 1 : Interrupt request	○	○

Fig.GC-13 I2C control register 2

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

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

I2C START/STOP condition control register

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

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

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

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

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

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

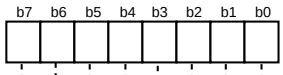
SCL/SDA interrupt pin selection bit selects either SCL pin or SDA pin as SCL/SDA interrupt enable pin.

Note: The SCL/SDA interrupt request may be set when the setting of I2C-BUS interface enable bit ES0 changes. Thus set the interrupt disable before the setting of SCL/SDA interrupt pin polarity selection bit (SIP) and SCL/SDA interrupt selection bit(SIS). After that reset "0" to the interrupt request bit before enabling the interrupt.

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

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

I2C start/stop condition control register



Symbol: S2Di(i=0,1) | Address: 0325₁₆,0335₁₆ | When reset: 00011010₂

Bit Symbol	Bit name	Function	R	W
SSC0	START/STOP condition setting bits	The setting of the detecting condition of START/STOP condition.Refer to Table GC-8. Note: Prohibit the setting of "00000₂" and odd value	○	○
SSC1			○	○
SSC2			○	○
SSC3			○	○
SSC4			○	○
SIP	SCL/SDA interrupt pin polarity selection bit	0 : Active in falling edge 1 : Active in rising edge	○	○
SIS	SCL/SDA interrupt pin selection bit	0 : SDA enable 1 : SCL enable	○	○
STSP SEL	START/STOP condition generation selection bit	0 : Setup/hold time short mode 1 : Setup/hold time long mode	○	○

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

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

Oscillation f(XIN) (MHz)	I2C system clock selection	I2C system clock(MHz)	SSC4-SSC0	SCL release time(cycle)	Setup time (cycle)	Hold time (cycle)
10	1 / 2f1	5	XXX11110	6.2µs (31)	3.2µs (16)	3.0µs (15)
8	1 / 2f1	4	XXX11010	6.75µs(27)	3.5µs (14)	3.25µs(13)
			XXX11000	6.25µs(25)	3.25µs(13)	3.0µs (12)
8	1 / 8f1	1	XXX00100	5.0µs (5)	3.0µs (3)	2.0µs (2)
4	1 / 2f1	2	XXX01100	6.5µs (13)	3.5µs (7)	3.0µs (6)
			XXX01010	5.5µs (11)	3.0µs (6)	2.5µs (5)
2	1 / 2f1	1	XXX00100	5.0µs (5)	3.0µs (3)	2.0µs (2)

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

START Condition Generation Method

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

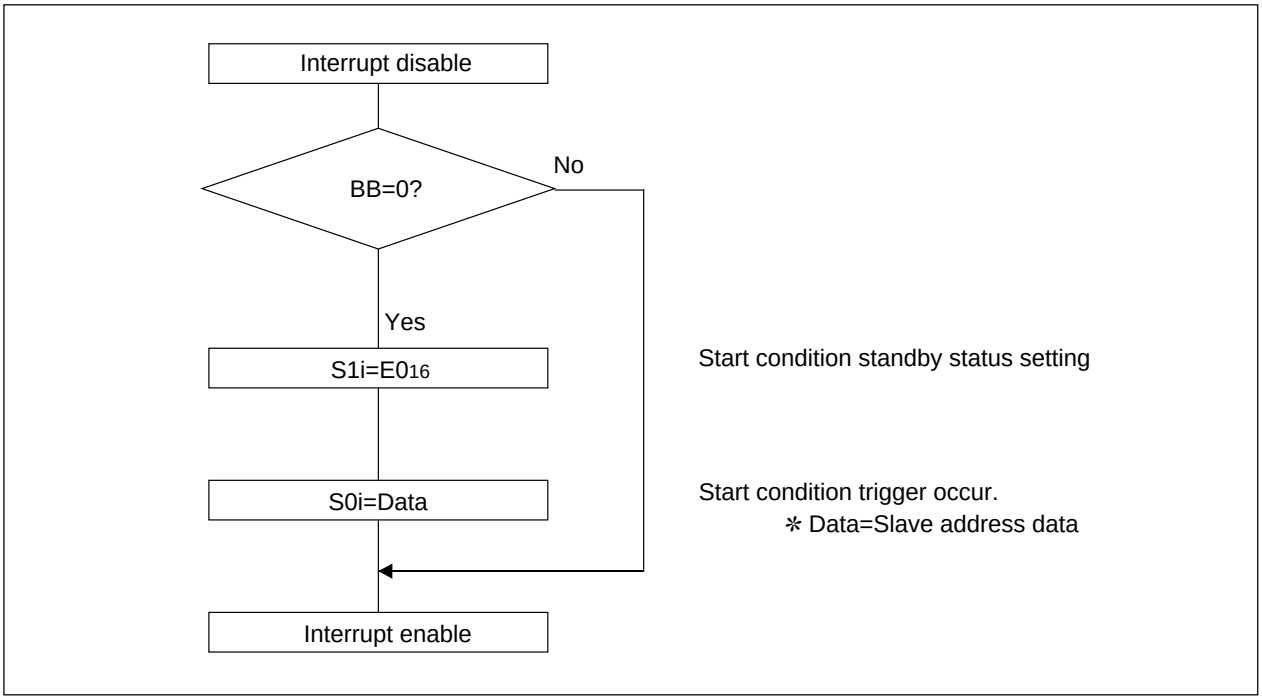


Fig.GC-15 Start condition generation flow chart

Function of protection of duplicate START condition

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

•The prohibition of setting up START condition standby

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

•The prohibition of writing to the I2C data shift register (The prohibition of generating a START condition trigger)

•If the generation of start condition is interrupted, sets the AL flag.

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

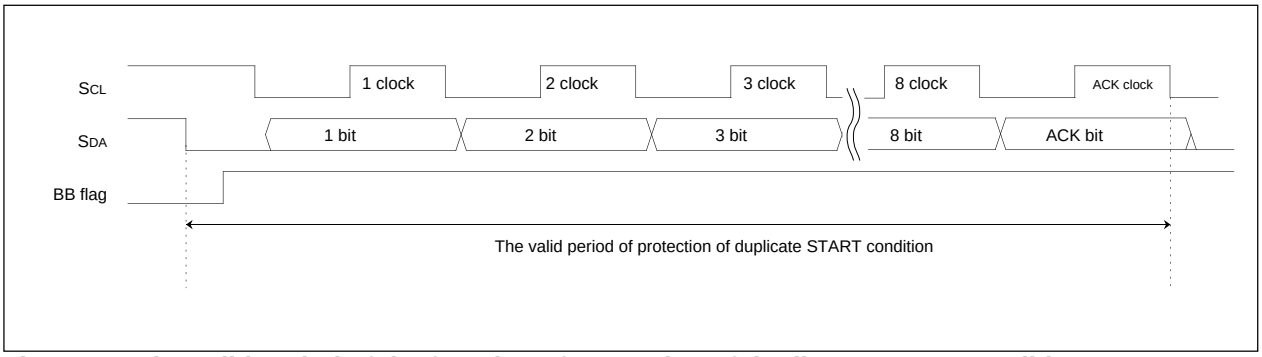


Fig.GC-16 The valid period of the function of protection of duplicate START condition

STOP Condition Generation Method

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

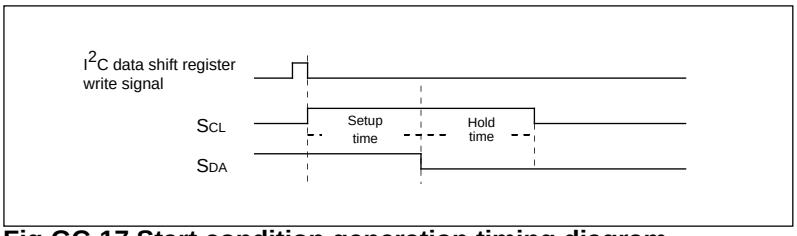


Fig.GC-17 Start condition generation timing diagram

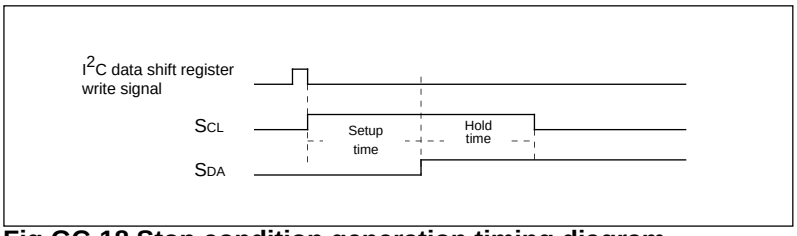


Fig.GC-18 Stop condition generation timing diagram

Table.GC-9 Start/Stop generation timing table

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

Note: VIIC = 4MHz

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

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

The value of the data writing to status register								Function
MST	TRX	BB	PIN	AL	AAS	AS0	LRB	
1	1	1	0	0	0	0	0	Setting up the START condition stand by in master transmission mode
1	1	0	0	0	0	0	0	Setting up the STOP condition stand by in master transmission mode
0/1	0/1	-	0	1	1	1	1	Setting up the communication mode (refer to the description on I2C status register)

START/STOP Condition Detecting Operation

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

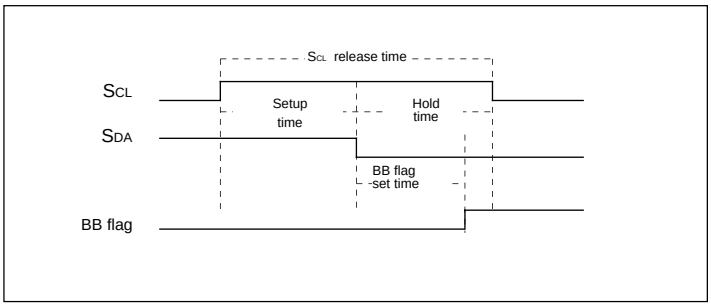


Fig.GC-19 Start condition detection timing diagram

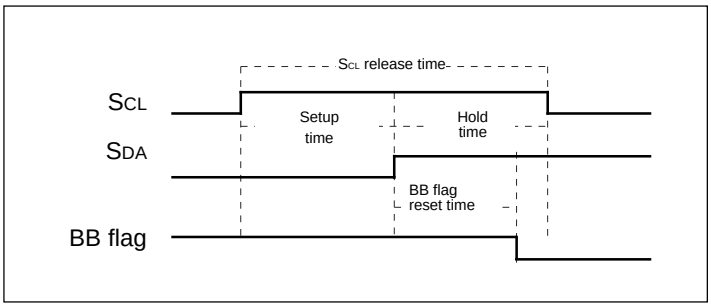


Fig.GC-20 Stop condition detection timing diagram

Table.GC-11 Start/Stop generation timing table

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

Note: Unit : Cycle number of system clock VIIC
SSC value is the decimal notation value of the START/STOP condition set bits SSC4 to SSC0.
Do not set “0” or an odd number to SSC value. The value in parentheses is an example when the I2C START/STOP condition control register is set to “18$_{16}$” at VIIC = 4 MHz.

Address Data Communication

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

(1) 7-bit addressing format

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

(2) 10-bit addressing format

To adapt the 10-bit addressing format, set the DBIT SAD bit of the I2C control register 0 (address032316,033316) to "1". Also set the WIT bit of I2C control register 1 to "1". An address comparison is performed between the first-byte address data transmitted from the master and the 8-bit slave address stored in the I2C address register (address 032216,033216). At the time of this comparison, an address comparison between the RBW bit of the I2C address register (address 032016,033016) and the R/$\overline{W}$ bit which is the last bit of the address data transmitted from the master is made. In the 10-bit addressing mode, the RBW bit which is the last bit of the address data not only specifies the direction of communication for control data, but also is processed as an address data bit.

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

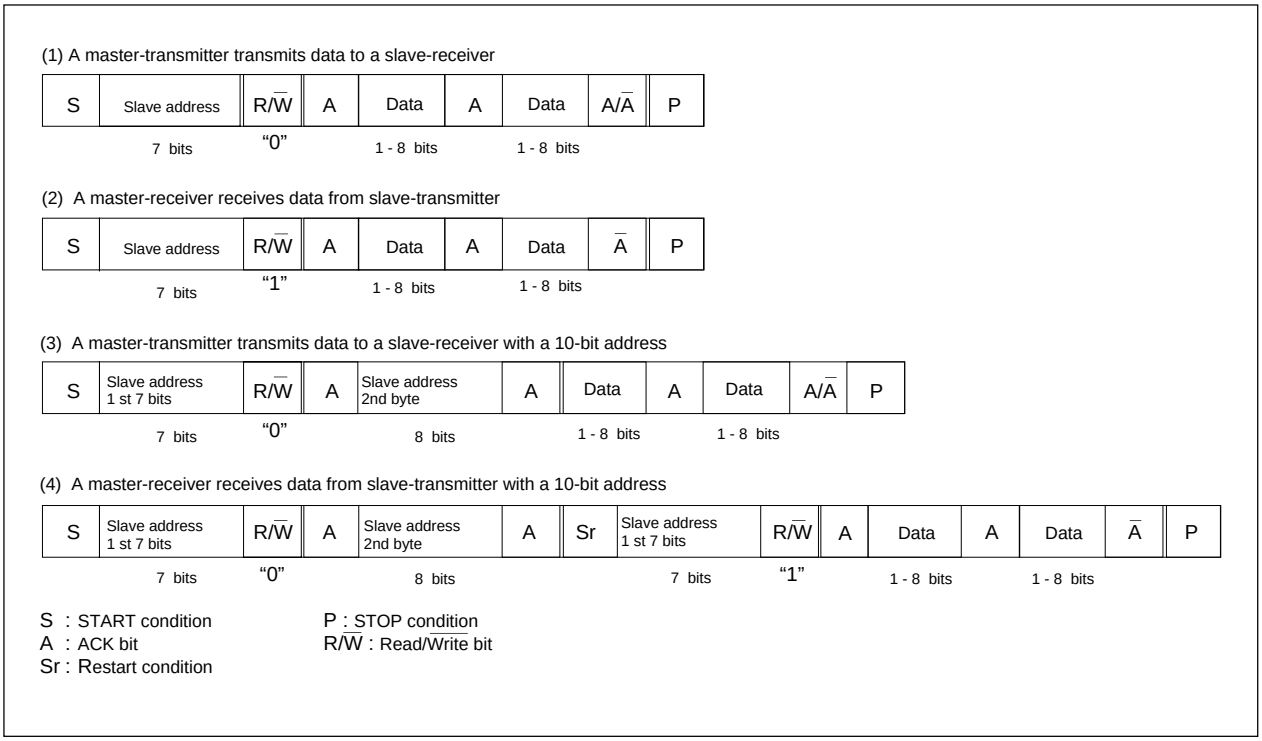


Fig.GC-21 Address data communication format

Example of Master Transmission

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

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

Example of Slave Reception

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

1)Set a slave address in the high-order 7 bits of the I2C address register and "0" in the RBW bit.
2)Set the ACK clock mode and SCL = 400 kHz by setting "0016" in the I2C control register 1 and "A516" in the I2C clock control register respectively. (f(XIN)=8MHz)
3)Set "0016" in the I2C status register so that transmission/reception mode is initialized.
4)Set a communication enable status by setting "0816" in the I2C control register 0.
5)When a START condition is received, an address comparison is performed.
6)•When all transmitted addresses are "0" (general call):
AD0 of the I2C status register is set to "1" and an interrupt request signal occurs.
•When the transmitted addresses agree with the address set in1):
ASS of the I2C status register is set to "1" and an interrupt request signal occurs.
•In the cases other than the above AD0 and AAS of the I2C status register are set to "0" and no interrupt request signal occurs.
7)Set dummy data in the I2C data shift register.
8)After receiving 1 byte data, it returns an ACK automatically and an interrupt request signal occurs.
9)In the case of whether returning an ACK or not by the content of the received control data, set the WIT bit of I2C control register 1 to "1", and after writing dummy data to I2C data shift register, receives the control data.
10)After receiving 1 byte data, an interrupt request signal occurs, set the ACKBIT to "1" or "0" by reading the content of the data shift register. and then returns or does not return an ACK.
11)When receiving control data of more than 1 byte, repeat step 7) 8) or 7) 10).
12)When a STOP condition is detected, the communication ends.

Usage precautions

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

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

•I2C data shift register (S0i : 0320$_{16}$, 0330$_{16}$)

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

•I2C address register (S0Di : address 0322$_{16}$, 0332$_{16}$)

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

•I2C control register 0 (S1Di : address 0323$_{16}$, 0333$_{16}$).

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

•I2C clock control register (S2i : address 0324$_{16}$, 0334$_{16}$)

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

•I2C control register 1 (S3Di : address 0326$_{16}$, 0336$_{16}$)

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

•I2C status register (S1i : address 0328$_{16}$, 0338$_{16}$)

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

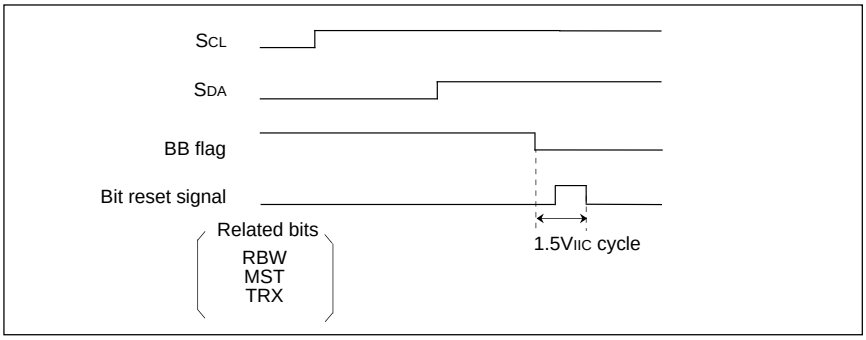


Fig.GC-22 The timing of bit reset (The detection of STOP condition)

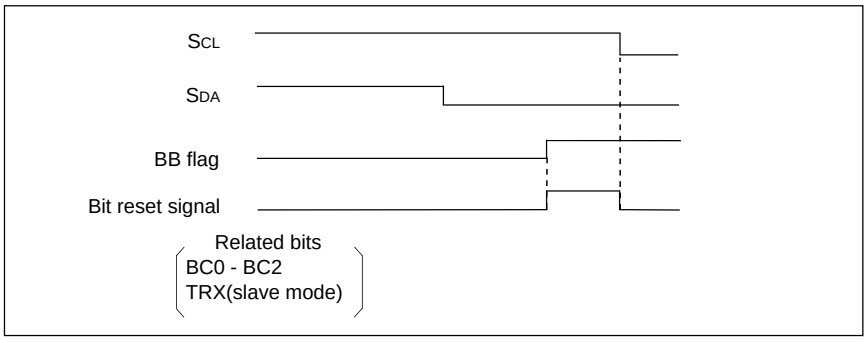


Fig.GC-23 The timing of bit reset (The detection of START condition)

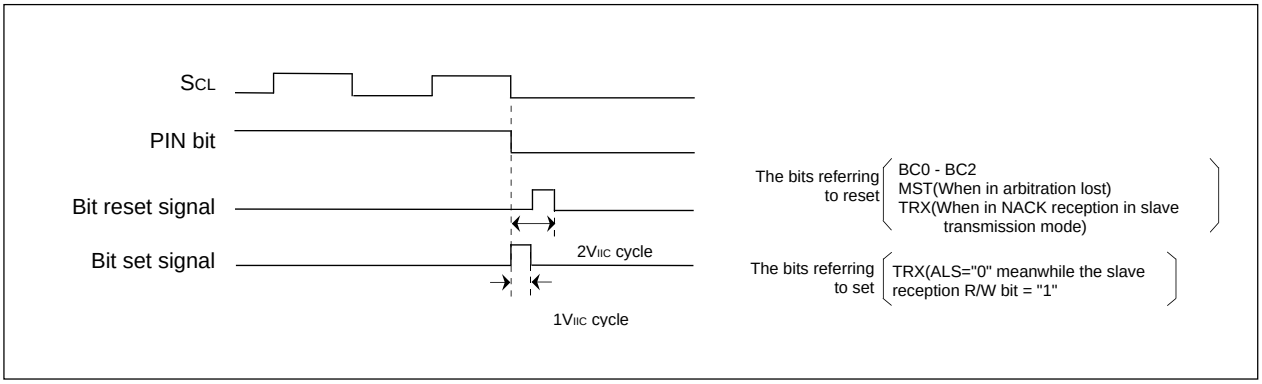


Fig.GC-24 Bit set/reset timing (at the completion of data transfer)

(2) Generation of RESTART condition

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

(3) Iimitation of internal clock 0

The registers of I2C-BUS interface circuit can not be read from or written to if the internal clock up selected to sub clock (XCIN, XCOUT) by system clock selection bit (system clock control register 0, address 000616, CMO7 bit). Please select main clock (XIN, XOUT) in read/write.

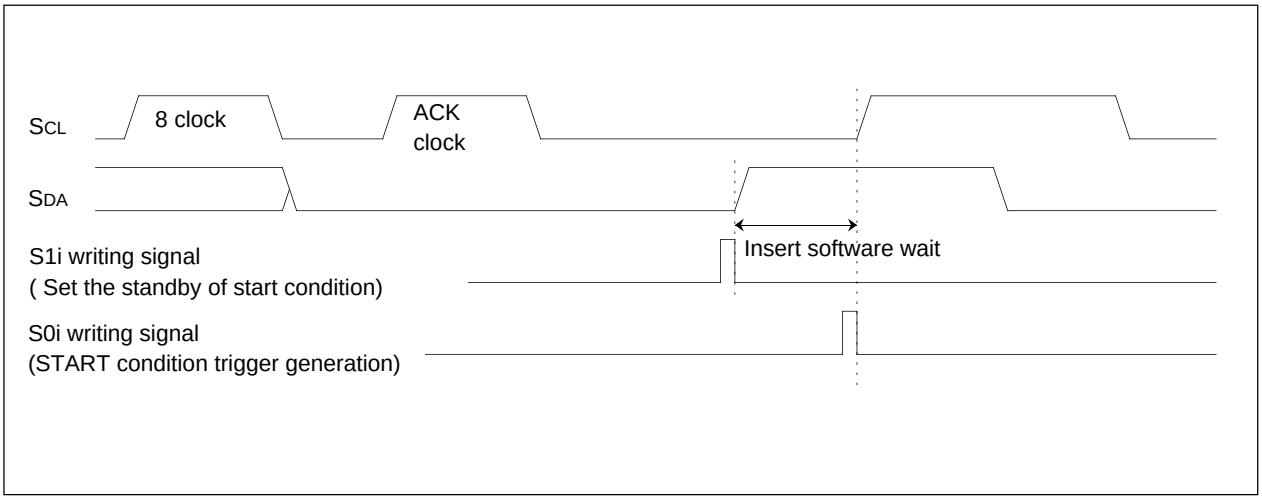


Fig.GC-25 The time of generation of RESTART condition

PS2 Interface

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

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

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

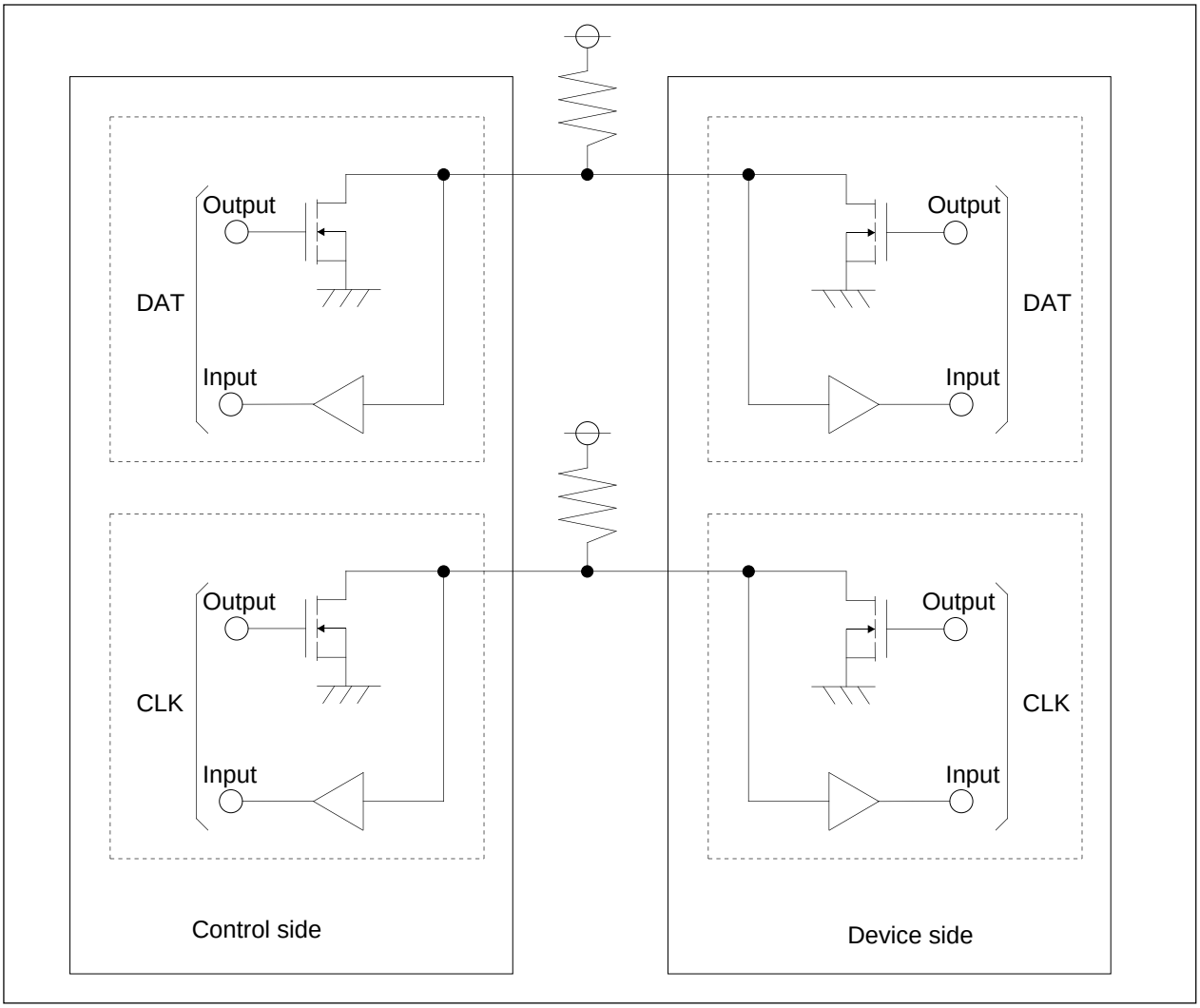


Fig.GK-1 System configuration

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

Table GK-1 Communication specification

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

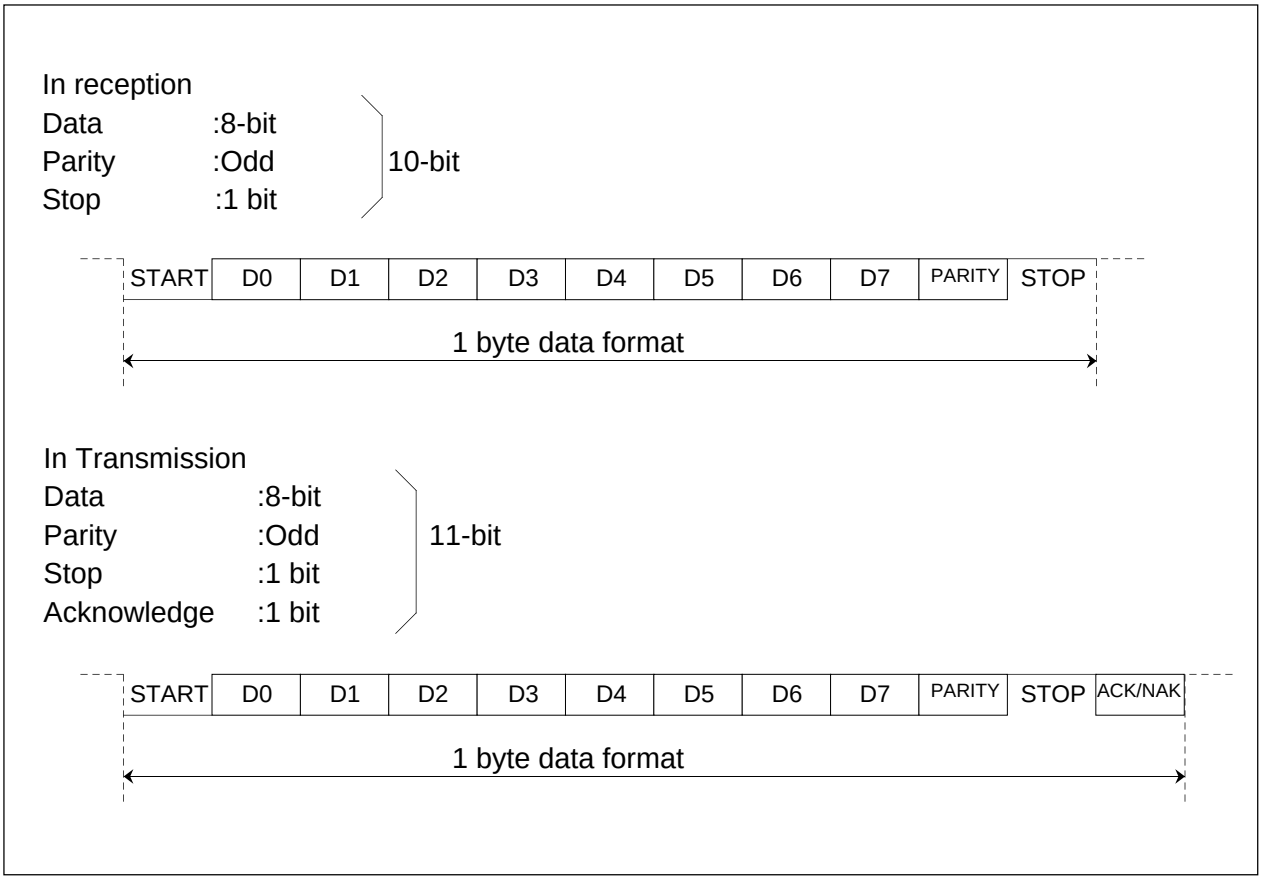


Fig.GK-2 1 byte data format

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

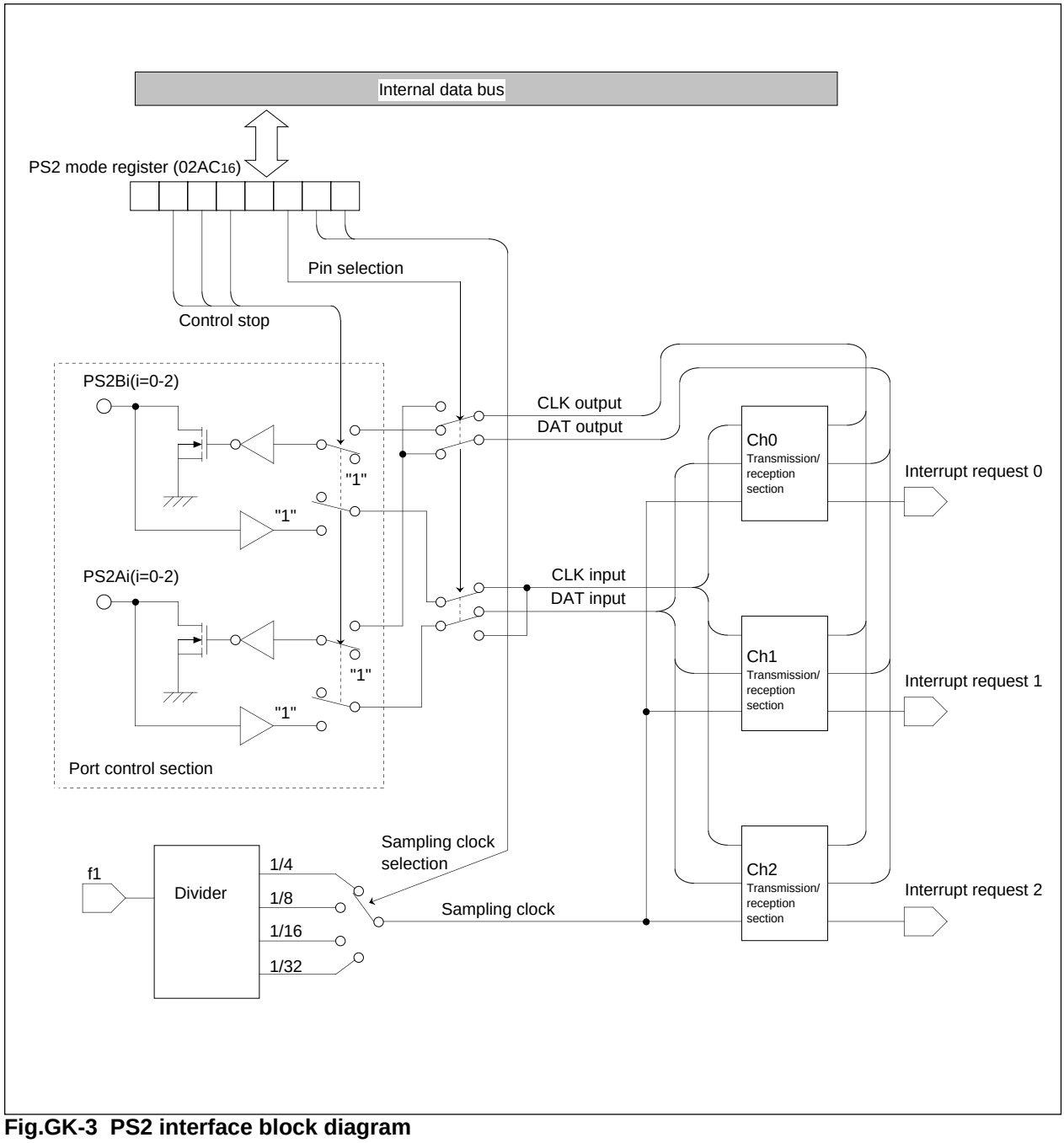


Fig.GK-3 PS2 interface block diagram

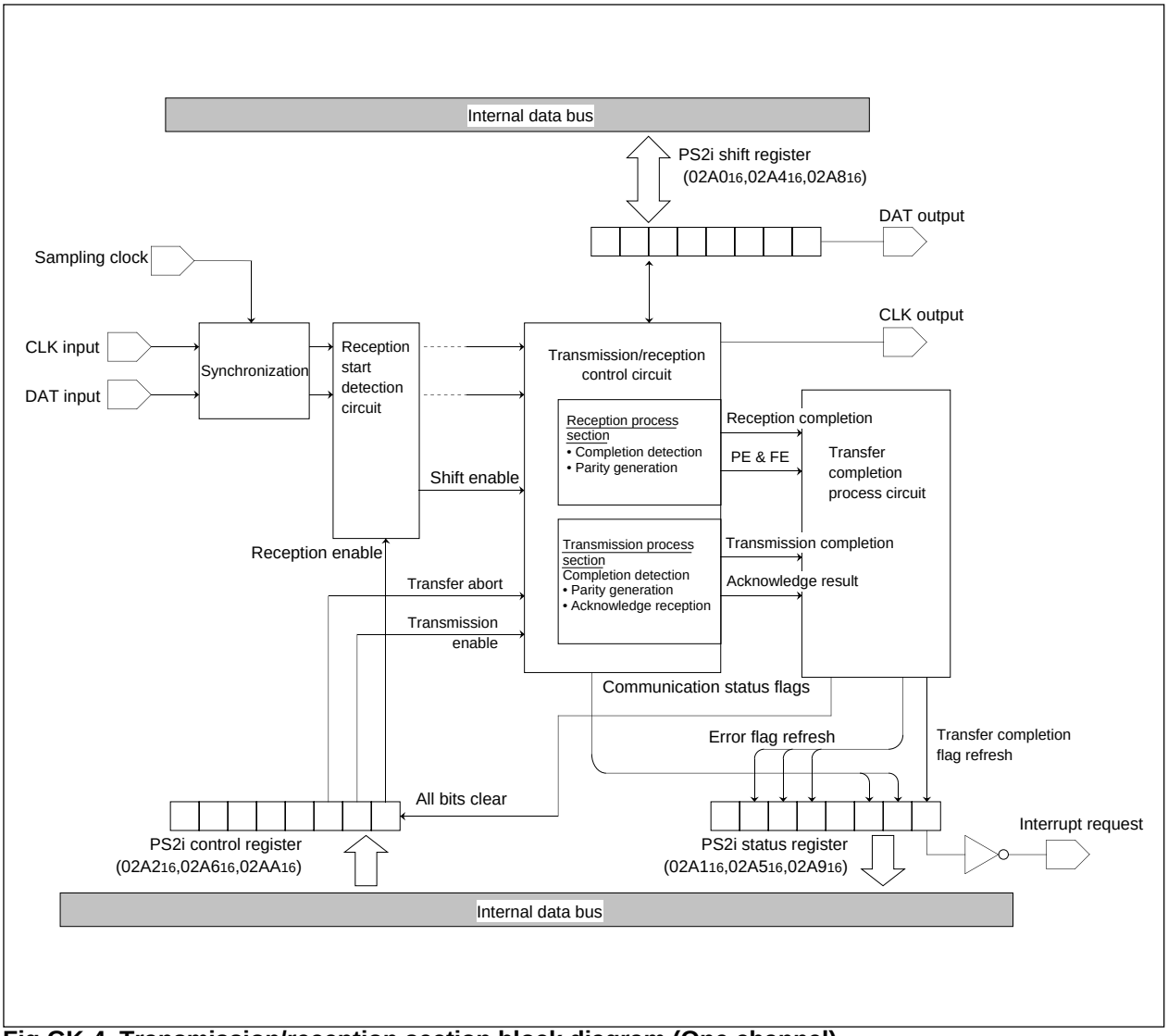


Fig.GK-4 Transmission/reception section block diagram (One channel)

(1) Register description

● PS2i shift register

• Transmission/reception data

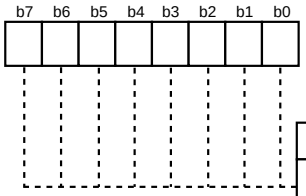
(1) Data reception

The reception data are stored.

(2) Data transmission

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

PS2i Shift register



symbol	Address	Reset value
PS20SR	$02A0_{16}$	00_{16}
PS21SR	$02A4_{16}$	00_{16}
PS22SR	$02A8_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
		Transmission/reception data	○	○

Fig.GK-5 PS2i shift register

● PS2i Control register

• Reception enable bit (REN)

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

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

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

• Transmission enable bit (TEN)

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

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

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

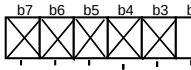
• Transfer abort request bit (RSTOP)

This bit is used to abort the data transfer procession.

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

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

PS2i control register



Symbol	Address	Reset Value
PS20CON	$02A2_{16}$	00_{16}
PS21CON	$02A6_{16}$	00_{16}
PS22CON	$02AA_{16}$	00_{16}

Bit Symbol	Bit name	Function	R	W
REN	Reception enable bit	0 : Disable 1 : Enable	○	○
TEN	Transmission enable bit	0 : Disable 1 : Enable	○	○
RSTOP	Transfer abort request bit	0 : No transfer abort 1 : Transfer abort	○	○
Nothing is allocated. Meaningless in writing, "0" in reading.				

Fig.GK-6 PS2i control register

●PS2i status register

• Transfer completion flag (TI)

The flag is set to “1” at the completion of transmission/reception and the completion of transfer abort.
The flag is cleared at read out from PS2i shift register or when the reception enable bit is changed from “0” to “1”.

• Receiving flag (RF)

The flag is set to “1” during the data reception.
The flag is cleared automatically after the data reception or after the transfer abort.

• Reception abort incognizable flag (CD)

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

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

• Transfer status flag (TS)

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

• Parity error flag (PE)

This bit is set to “1” when parity error occurs in received data.
The flag is cleared at read out from PS2i shift register or when the reception enable bit is changed from “0” to “1”.

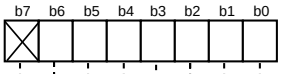
• Framing error / NACK reception flag (FE)

At the completion of reception: The flag is set when the detection of stop bit of reception data fails.
At the completion of transmission: The flag is set when NAK is received from the device side.
The flag is cleared at read out from PS2i shift register, the reception enable bit or the transmission bit is changed from “0” to “1”.

• Transfer abort completion flag (CC)

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

PS2i status register



Symbol	Address	Reset Value
PS20STS	$02A1_{16}$	00_{16}
PS21STS	$02A5_{16}$	00_{16}
PS22STS	$02A9_{16}$	00_{16}

Bit Symbol	Bit name	Function	R	W
TI	Transfer completion flag	0 : Waiting for transfer , During transfer 1 : Communication complete	○	×
RF	Receiving flag	0 : Waiting for reception , Reception complete 1 : During receiving	○	×
CD	Reception abort incognizable flag	0 : Recognizable 1 : Incognizable	○	×
TS	Transfer status flag	0 : Transmission operation 1 : Reception operation	○	×
PE	Parity error flag	0 : No error 1 : Error	○	×
FE	Framing error / NACK reception flag	0 : No error 1 : Error	○	×
CC	Transfer abort completion flag	0 : Not abort 1 : Abort	○	×
Nothing is allocated. Meaningless in writing, "0" in reading.			–	–

Fig.GK-7 PS2i status register

● PS2 mode register

• Sampling clock selection bits (SCK0,1)

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

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

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

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

• Pin selection bit (PSEL)

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

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

• PS2 interface enable bit (PSEN)

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

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

PS2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
0				X			

Symbol	Address	Reset Value
PS2MOD	$02AC_{16}$	00_{16}

Bit Symbol	Bit name	Function	R	W
SCK0	Sampling clock selection bits	00 : Main clock /4 01 : Main clock /8 10 : Main clock /16 11 : Main clock /32	○	○
SCK1				
PSEL	Pin selection bit	0 : PS2B connects to PS2 clock(CLK) 1 : PS2B connects to PS2 data(DAT)	○	○
Nothing is allocated. Meaningless in writing, "0" in reading.				
PSEN0	PS2 interface enable bits	0 : $P7_0$,$P7_3$ as GPIO 1 : $P7_0$,$P7_3$ as PS2A0,PS2B0	○	○
PSEN1		0 : $P7_1$,$P7_4$ as GPIO 1 : $P7_1$,$P7_4$ as PS2A1,PS2B1	○	○
PSEN2		0 : $P7_2$,$P7_5$ as GPIO 1 : $P7_2$,$P7_5$ as PS2A2,PS2B2	○	○
Reserved bit		Must be set to "0".	○	○

Fig.GK-8 PS2 mode register

(2) Operation description

● Basic setting

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

• PS2 interface enable bit

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

• Sampling clock selection bit

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

• External interrupt function support pin (PS2B) selection

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

● Reception operation

Fig.GK-9 shows the reception operation timing.

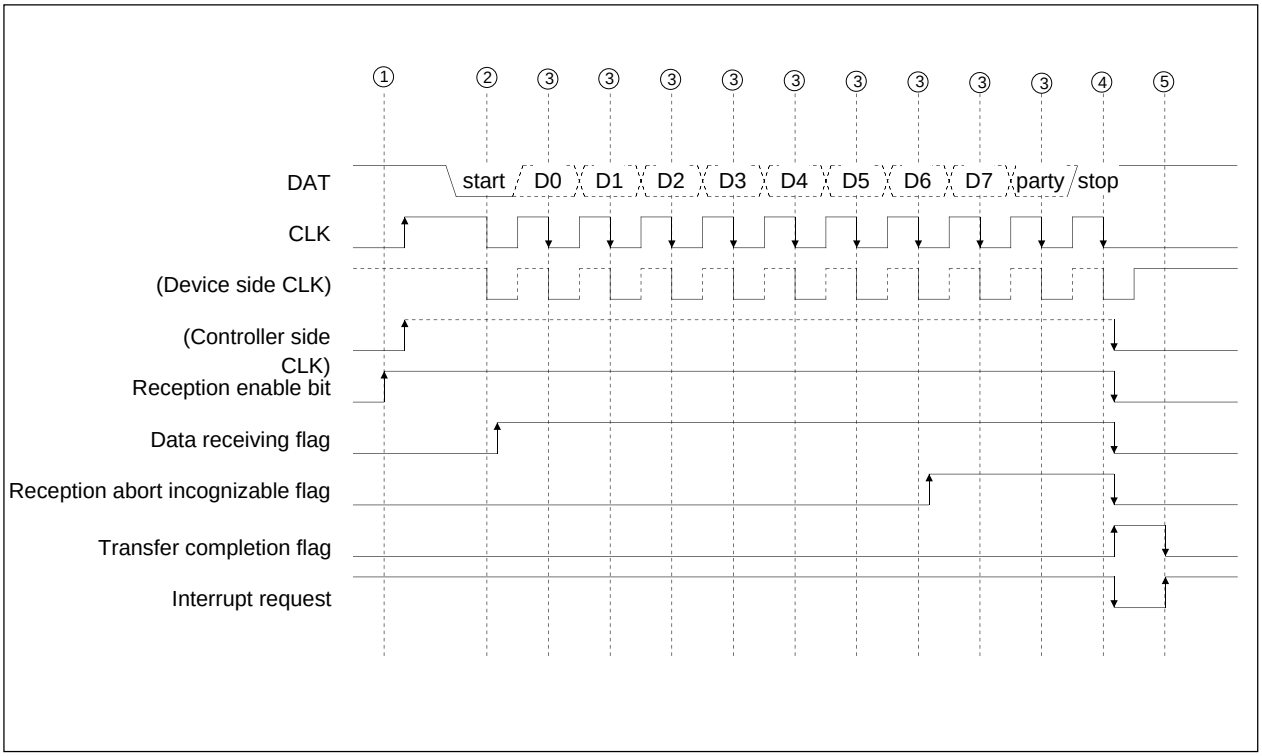


Fig.GK-9 Reception operation timing

(1) Reception enable

The reception operation is enabled by writing 01_{16} (reception enable bit = “1”) to PS2i control register (address : $02A2_{16}$, $02A6_{16}$, $02AA_{16}$). The PS2 clock (CLK) will become “H”.

(2) Reception start

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

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

The content PS2 data (DAT) is read into PS2i shift register (address : $02A0_{16}$, $02A4_{16}$, $02A8_{16}$) sequentially by the falling edge of PS2 clock (CLK). The data transfer sequence is data bit (D0 -D7) then parity bit.

(4) Reception completion (Stop bit Reception completion)

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

(5) Data read out

Read out data from PS2i shift register (address : $02A0_{16}$,$02A4_{16}$,$02A8_{16}$). At this time , the error flags (Bit4 to 6) of and transfer completion flag (bit 0) of PS2i status register (address: $02A1_{16}$, $02A5_{16}$, $02A9_{16}$) will be cleared to “0”.

● Transmission operation

Fig.GK-10 shows the transmission operation timing.

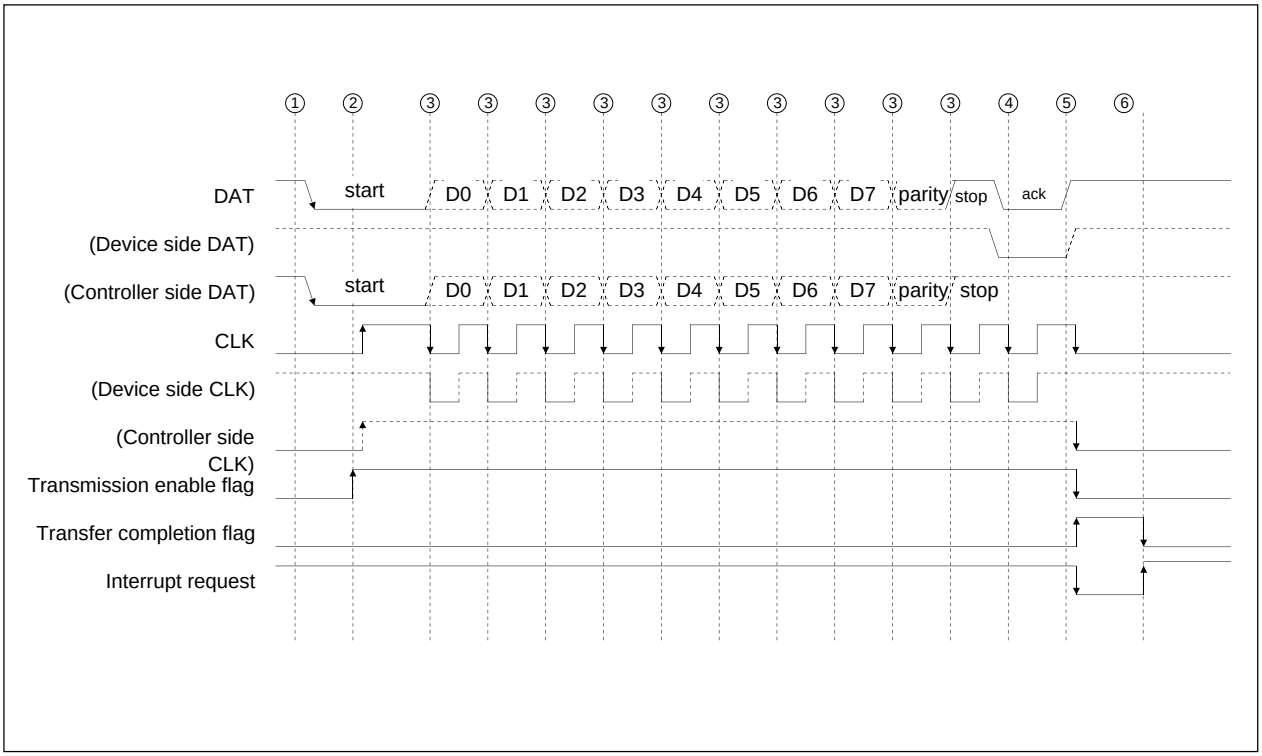


Fig.GK-10 Transmission operation timing

(1) Data writing

Write transmission data to PS2i shift register (address : 02A0$_{16}$, 02A4$_{16}$, 02A8$_{16}$). At this time,PS2 data will become “L” (transmission start).

(2) Transmission enable

Set 02$_{16}$ (Transmission enable bit = “1”) to PS2i control register (address : 02A2$_{16}$, 02A6$_{16}$, 02AA$_{16}$) for enabling transmission operation. At this time , PS2 clock (CLK) will become “H”.

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

The content of PS2i shift register (address : 02A0$_{16}$, 02A4$_{16}$, 02A8$_{16}$) will be output to the PS2 data (DAT) sequentially by the falling edge of PS2 clock (CLK). The sequence of data transfer is data bits (D0 to D7) , Parity bit, and stop bit.

(4) Acknowledge reception

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

(5) Communication completion

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

(6) Status clear

Read out the data from PS2i shift register (address : 02A0$_{16}$, 02A4$_{16}$, 02A8$_{16}$). At this time , the error flags (bits 4 to 6) and transfer completion flag (Bit0) of PS2i status register (address : 02A1$_{16}$, 02A5$_{16}$, 02A9$_{16}$) will be cleared to “0”.

● Transfer abort operation

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

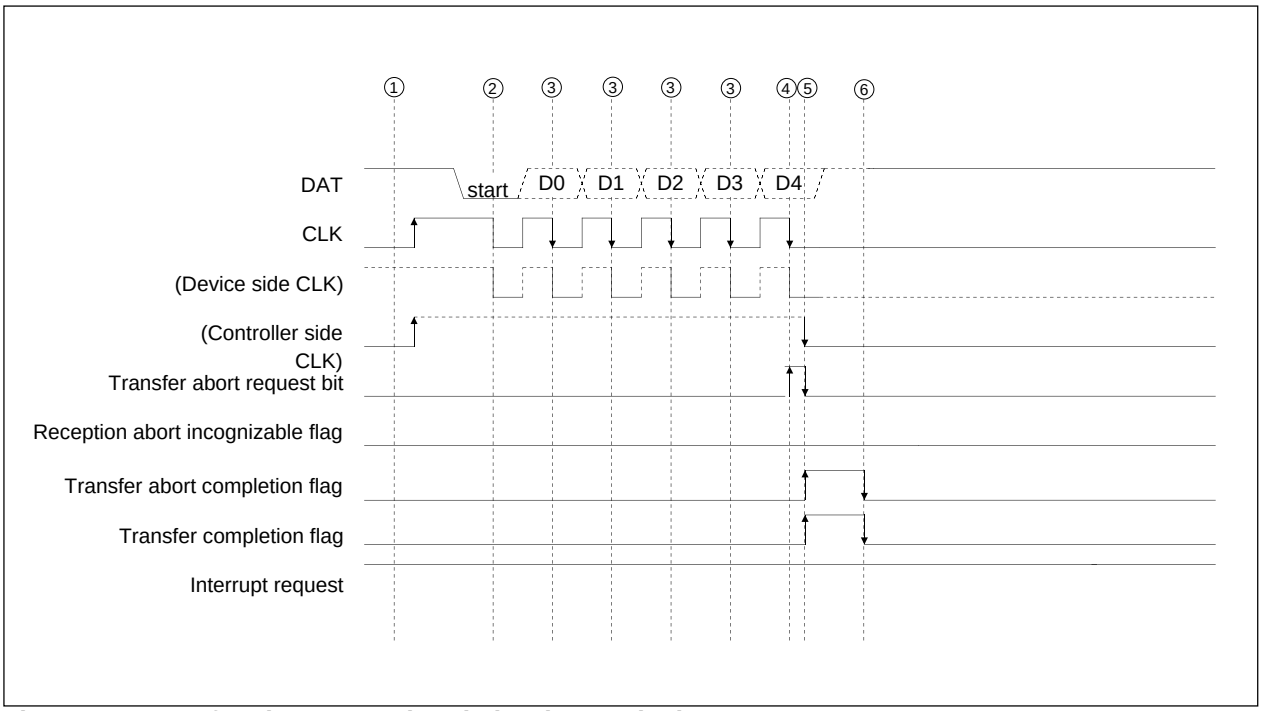


Fig.GK-11 Transfer abort operation timing (reception)

(1) - (3) Data reception operation

(4) Transfer abort request

Set 04_{16} (transfer abort request bit = “1”) to PS2i control register (address : $02A2_{16}$, $02A6_{16}$,$02AA_{16}$).

(5) Transfer abort completion

The transfer abort completion flag (bit 6) and transfer completion flag (bit 0) of PS2i status register (address : $02A1_{16}$, $02A5_{16}$, $02A9_{16}$) are set to “1”, transfer abort request bit (bit 2) of PS2i control register (address : $02A2_{16}$, $02A6_{16}$, $02AA_{16}$) is cleared to “0”. At this time, PS2 clock (CLK) becomes “L” (reception disable status) and interrupt request occurs.

(6) Status clear

By a pseudo read of PS2i shift register (address : $02A0_{16}$, $02A4_{16}$, $02A8_{16}$), the transfer abort completion flag (bit 6) and transfer completion flag (bit 0) of PS2i status register (address : $02A1_{16}$, $02A5_{16}$, $02A9_{16}$) are cleared to “0”.

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

Programmable I/O Ports

There are 129 programmable I/O ports: P0 to P16 (excluding $P8_5$). Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set. The N channel open drain ports $P6_0$ to $P6_3$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P13_0$ to $P13_7$ and $P8_5$ (input only port) do not build internal pull-up resistance.

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

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

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D-A converter), they function as outputs regardless of the contents of the direction registers. When pins are used as the outputs for the D-A converter, do not set the direction registers to output mode. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

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

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

Note: There is no direction register bit for $P8_5$.

(2) Port registers

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

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

(3) Pull-up control registers

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

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

(4) Port control register

Fig.UA-11 shows the configurations of port control register 0, 1. Fig.UA-12 shows the configuraitons of port control register 2, 3. The bit 0 of port control resister 0 is used to read port P1 as follows:

0 : When port P1 is input port, port input level is read.
When port P1 is output port , the contents of port P1 register is read.

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

The P0, P1, $P4_0$ to $P4_6$, P11 and P14 output type, CMOS or N channel open drain, are set by bit 0 to 6 of port control register 1 and bit 0 to 2 of port control register 3.

0 : CMOS output

1 : N channel open drain output

Exception: $P4_2$ output type is N channel open drain if either bit 4 of port control register 1 or bit 2 of port control register 3 is set to “1”.

Bit 7 of port control register1 functions as below

0 : $P4_0$/$P4_3$ output is cleared by software only

1 : $P4_0$/$P4_3$ output is cleared by software or when output buffer 0 is read by host side.

The driving ability of N channel output transistors for $P14_0$ to $P14_3$ can be selected by bit 6 of port control register 2 controls as below:

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

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

(5) Port P4/P7 input register

Fig.UA-13 shows the configurations of P4 and P7 input register.

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

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

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

(6) Port function selection register 0,1

Fig.UA-14 shows the configurations of port function selection register 0,1. The port functions of UART0 to UART2 output, TimerA0 to TimerA2 output, TimerB3,B4 input or external interrupt $\overline{INT6}$ to $\overline{INT12}$ input can be switched by setting these two registers. And by setting bit6,7 of port function selection register 1, the same frequency clock with f(X_{IN}) can be output from $P6_6$ and $P6_7$.

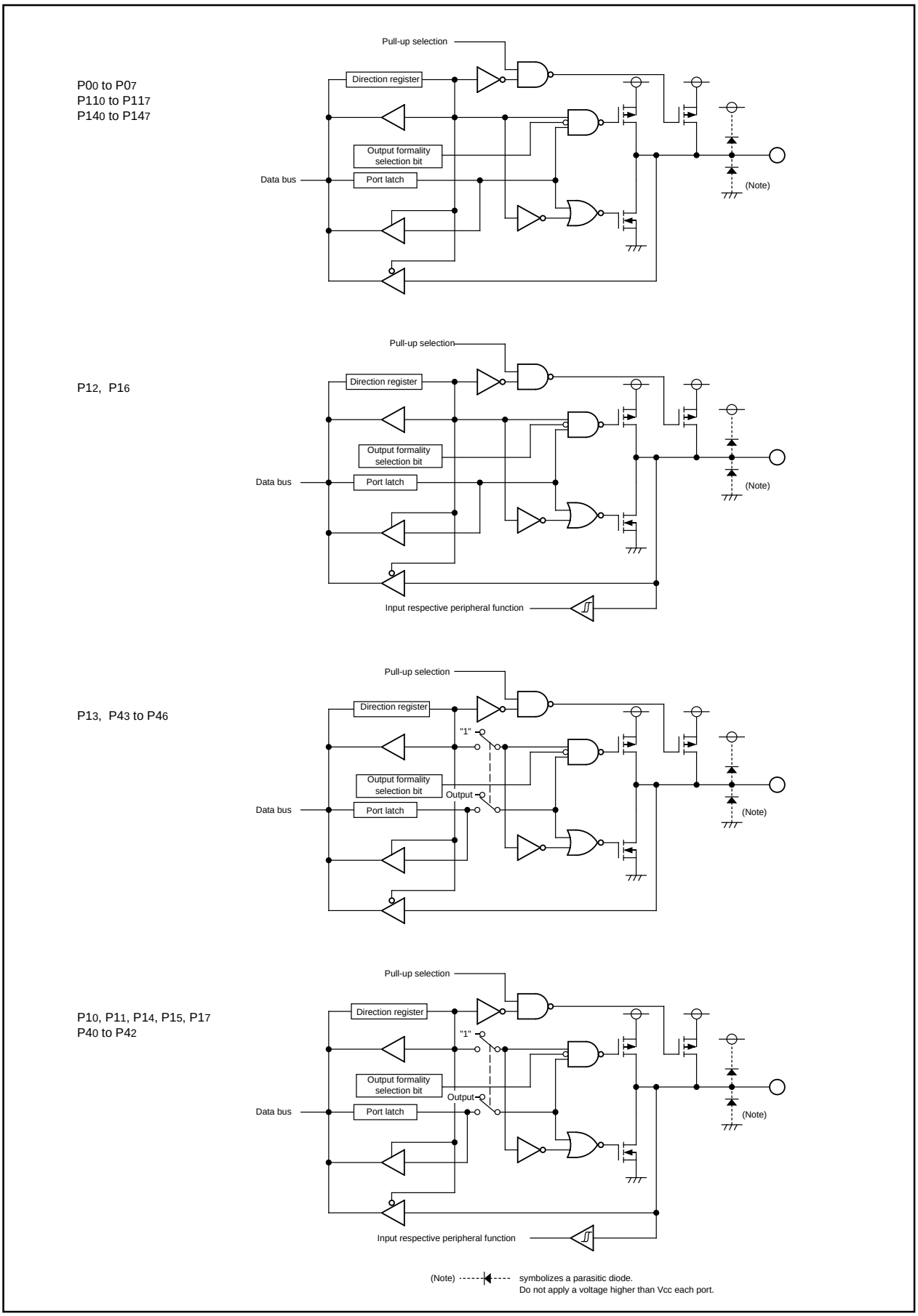
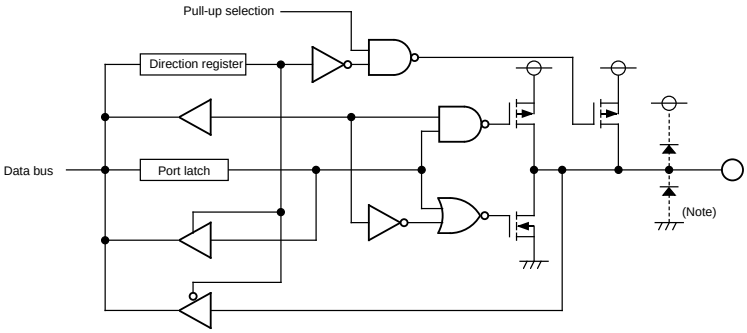
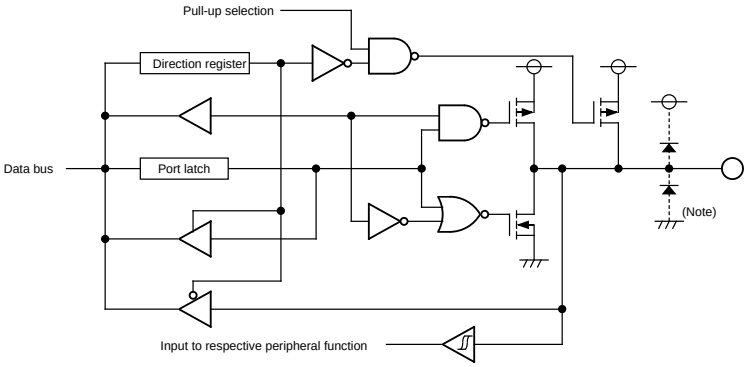


Fig.UA-1 Programmable I/O ports (1)

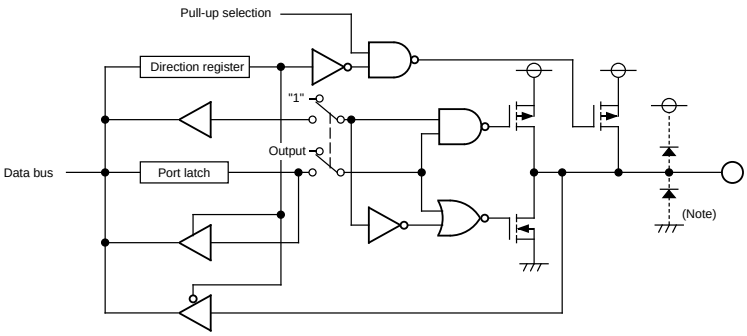
P12$_6$, P12$_7$
P15$_3$ to P15$_7$



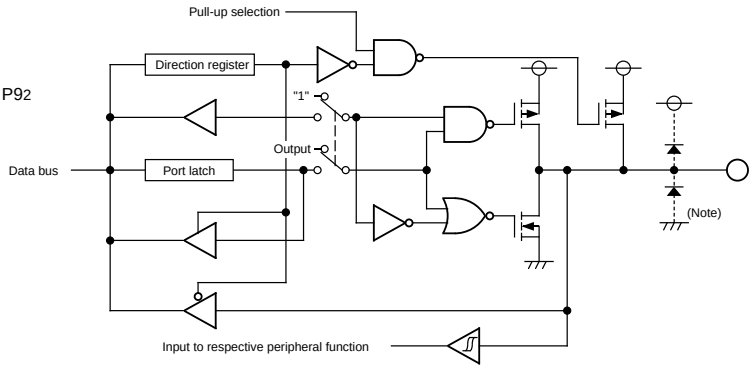
P2$_1$, P2$_4$ to P2$_7$, P5$_0$ to P5$_7$
P9$_1$, P9$_7$
P12$_0$ to P12$_6$
P16$_0$, P16$_1$



P4$_7$



P2$_0$, P2$_2$, P2$_3$, P3$_0$ to P3$_7$
P15$_0$ to P15$_2$, P6$_4$ to P6$_7$, P9$_0$, P9$_2$



Note. ----◄---- symbolizes a parasitic diode.
Do not apply a voltage higher than Vcc to each port.

Fig.UA-2 Programmable I/O ports (2)

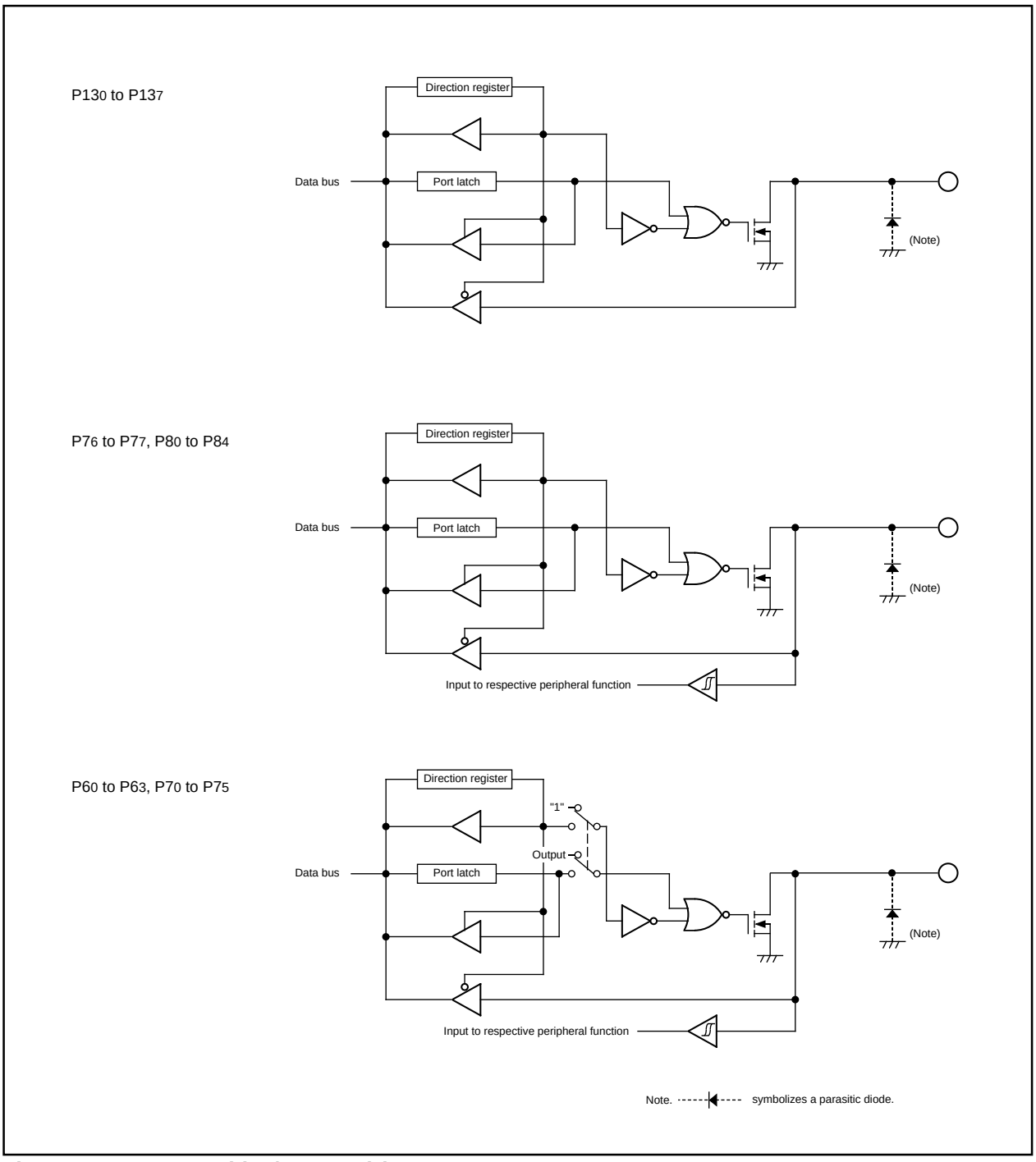
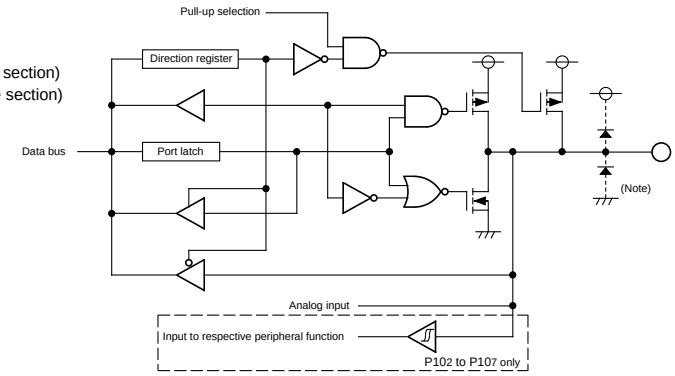
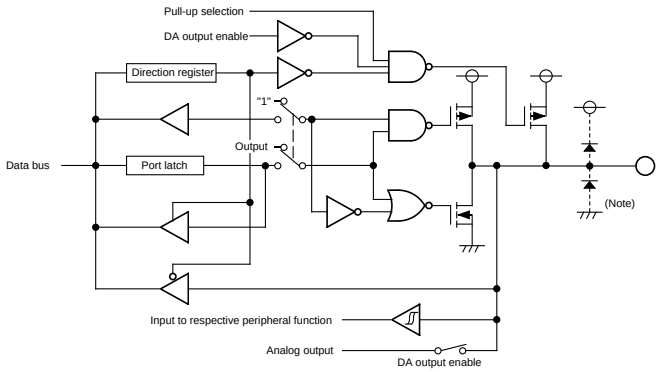


Fig.UA-3 Programmable I/O ports (3)

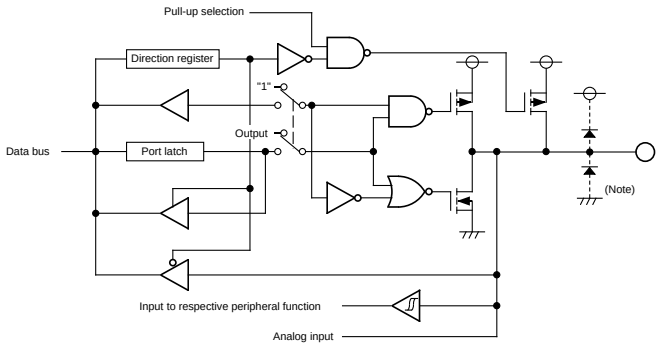
P10$_0$, P10$_1$ (Excepting the short dashes line section)
P10$_2$ to P10$_7$ (Including the short dashes line section)



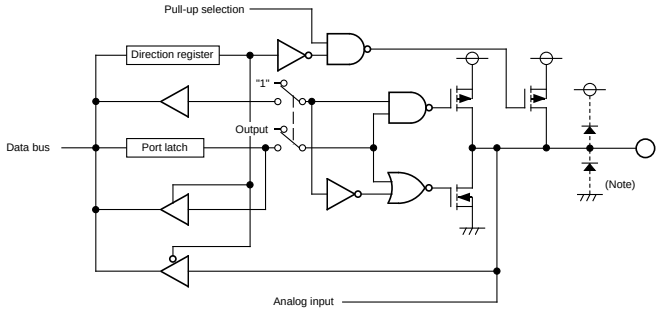
P9$_3$, P9$_4$



P9$_5$



P9$_6$



Note ----|◄---- symbolizes a parasitic diode.
Do not apply a voltage higher than Vcc to each port.

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

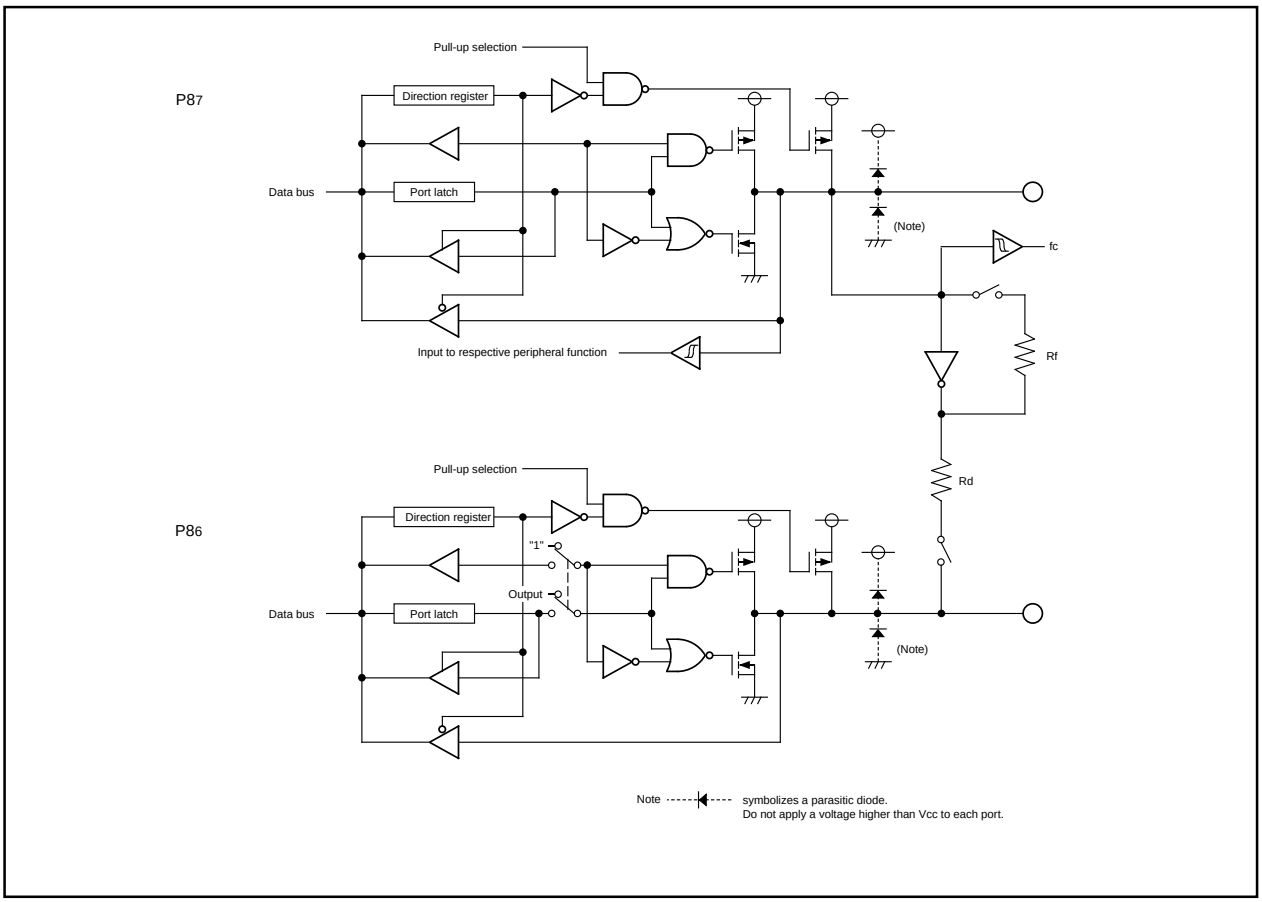


Fig.UA-5 Programmable I/O ports (5)

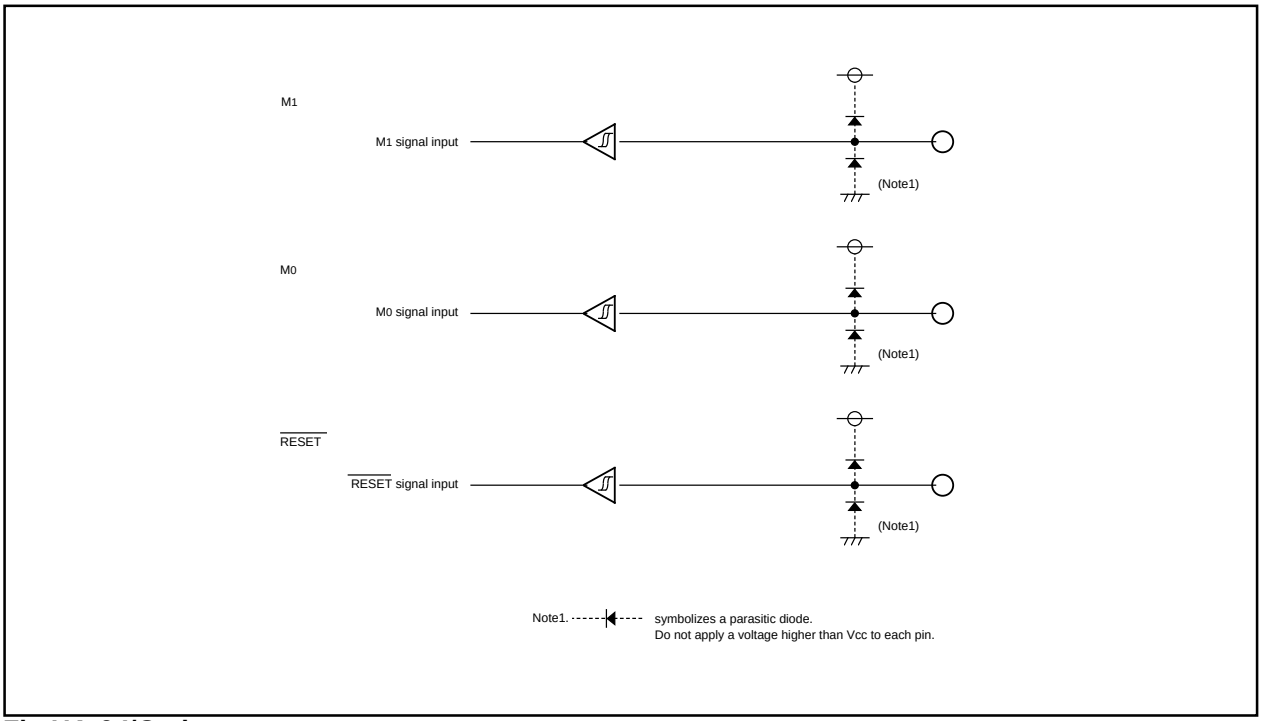
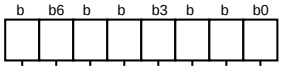


Fig.UA-6 I/O pins

Port Pi direction register (Note1)

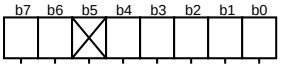


symbol	Address	When reset
PDi(i=0-15, except 8)	03E2$_{16}$,03E3$_{16}$,03E6$_{16}$,03E7$_{16}$,03EA$_{16}$ 03EB$_{16}$,03EE$_{16}$,03EF$_{16}$,03F3$_{16}$,03F6$_{16}$ 02E2$_{16}$,02E3$_{16}$,02E6$_{16}$,02E7$_{16}$,02EA$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Function as an input port) 1 : Output mode (Function as an output port) (i=0-15, except 8)	○	○
PDi_1	Port Pi$_1$ direction register		○	○
PDi_2	Port Pi$_2$ direction register		○	○
PDi_3	Port Pi$_3$ direction register		○	○
PDi_4	Port Pi$_4$ direction register		○	○
PDi_5	Port Pi$_5$ direction register		○	○
PDi_6	Port Pi$_6$ direction register		○	○
PDi_7	Port Pi$_7$ direction register		○	○

Note Set bit 2 of protect register(address 000A$_{16}$) to "1" before rewriting to the port P9 direction register.

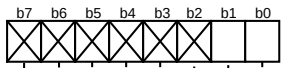
Port P8 direction register



Symbol	Address	When reset
PD8	03F2$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ direction register	0 : Input mode (Function as an input port) 1 : Output mode (Function as an output port)	○	○
PD8_1	Port P8$_1$ direction register		○	○
PD8_2	Port P8$_2$ direction register		○	○
PD8_3	Port P8$_3$ direction register		○	○
PD8_4	Port P8$_4$ direction register		○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			–	–
PD8_6	Port P8$_6$ direction register	0 : Input mode (Function as an input port) 1 : Output mode (Function as an output port)	○	○
PD8_7	Port P8$_7$ direction register		○	○

Port P16 direction register

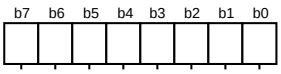


Symbol	Address	When reset
PD16	02EB$_{16}$	XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
PD16_0	Port P16$_0$ direction register	0 : Input mode (Function as an input port) 1 : Output mode (Function as an output port)	○	○
PD16_1	Port P16$_1$ direction register		○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate			–	–

Fig.UA-7 Direction register

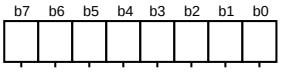
Port Pi register



Symbol	Address	When reset
Pi(i=0 to15 ,except 8)	03E0$_{16}$,03E1$_{16}$,03E4$_{16}$,03E5$_{16}$,03E8$_{16}$ 03E9$_{16}$,03EC$_{16}$,03ED$_{16}$,03F1$_{16}$,03F4$_{16}$ 02E0$_{16}$,02E1$_{16}$,02E4$_{16}$,02E5$_{16}$,02E8$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : "L" level data 1 : "H" level data (i=0-15, except 8)	○	○
Pi_1	Port Pi$_1$ register		○	○
Pi_2	Port Pi$_2$ register		○	○
Pi_3	Port Pi$_3$ register		○	○
Pi_4	Port Pi$_4$ register		○	○
Pi_5	Port Pi$_5$ register		○	○
Pi_6	Port Pi$_6$ register		○	○
Pi_7	Port Pi$_7$ register		○	○

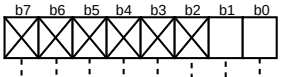
Port P8 register



Symbol	Address	When reset
P8	03F0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
P8_0	Port P8$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : "L" level data 1 : "H" level data	○	○
P8_1	Port P8$_1$ register		○	○
P8_2	Port P8$_2$ register		○	○
P8_3	Port P8$_3$ register		○	○
P8_4	Port P8$_4$ register		○	○
P8_5	Port P8$_5$ register		○	×
P8_6	Port P8$_6$ register		○	○
P8_7	Port P8$_7$ register		○	○

Port P16 register

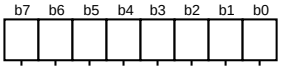


Symbol	Address	When reset
P16	02E9$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
PD16_0	Port P16$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : "L" level data 1 : "H" level data	○	○
PD16_1	Port P16$_1$register		○	○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate			–	–

Fig.UA-8 Port register

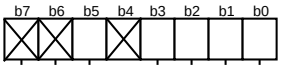
Pull-up control register 0



Symbol	Address	When reset
PUR0	03FC16	0016

Bit symbol	Bit name	Function	R	W
PU00	P00 to P03 pull-up	The corresponding port is pulled high with a pull-up register 0: Not pulled high 1: Pulled high	O	O
PU01	P04 to P07 pull-up		O	O
PU02	P10 to P13 pull-up		O	O
PU03	P14 to P17 pull-up		O	O
PU04	P20 to P23 pull-up		O	O
PU05	P24 to P27 pull-up		O	O
PU06	P30 to P33 pull-up		O	O
PU07	P34 to P37 pull-up		O	O

Pull-up control register 1

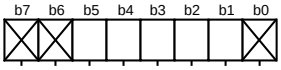


Symbol	Address	When reset
PUR1	03FD16	0016

Bit symbol	Bit name	Function	R	W
PU10	P40 to P43 pull-up	The corresponding port is pulled high with a pull-up register 0: Not pulled high 1: Pulled high	O	O
PU11	P44 to P47 pull-up		O	O
PU12	P50 to P53 pull-up		O	O
PU13	P54 to P57 pull-up		O	O
Nothing is assigned. (Note) Can't write to this bit. The value, if read, turns out to be “0”.			–	–
PU15	P64 to P67 pull-up		O	O
Nothing is assigned. (Note) Can't write to this bit. The value, if read, turns out to be “0”.			–	–

Note.Since P60 to P63 and P70 to P77 are N-channel open drain ports, internal pull-up is not available for them.

Pull-up control register 2



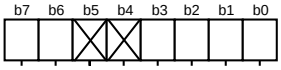
Symbol	Address	When reset
PUR2	03FE16	0016

Bit symbol	Bit name	Function	R	W
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–
PU21	P86, P87 pull-up(Note)	The corresponding port is pulled high with a pull-up register 0: Not pulled high 1: Pulled high	O	O
PU22	P90 to P93 pull-up		O	O
PU23	P94 to P97 pull-up		O	O
PU24	P100 to P103 pull-up		O	O
PU25	P104 to P107 pull-up		O	O
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–
			–	–

Note.Since P80 to P84 are N-channel open drain ports, internal pull-up is not available for them.
And P85 is input port only , also no internal pull-up is available.

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

Pull-up control register 3

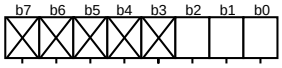


Symbol	Address	When reset
PUR3	02FC16	0016

Bit symbol	Bit name	Function	R	W
PU30	P110 to P113 pull-up	The corresponding port is pulled high with a pull-up register 0 : Not pulled high 1 : Pulled high	○	○
PU31	P114 to P117 pull-up		○	○
PU32	P120 to P123 pull-up		○	○
PU33	P124 to P127 pull-up		○	○
Nothing is assigned. (Note) Can't write to this bit. The value, if read, turns out to be “0”.			–	–
PU36	P140 to P143 pull-up	0 : Not pulled high 1 : Pulled high	○	○
PU37	P144 to P147 pull-up		○	○

Note.Since P130 to P137 are N-channel open drain ports, internal pull-up is not available for them.

Pull-up control register 4

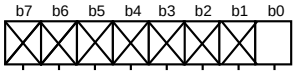


Symbol	Address	When reset
PUR4	02FD16	0016

Bit symbol	Bit name	Function	R	W
PU40	P150 to P153 pull-up	The corresponding port is pulled high with a pull-up register 0 : Not pulled high 1 : Pulled high	○	○
PU41	P154 to P157 pull-up			
PU42	P160, P161 pull-up			
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–

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

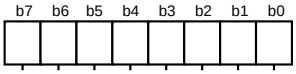
Port control register 0



Symbol	Address	When reset
PCR0	03FF16	0016

Bit Symbol	Bit name	Function	R	W
PCR00	Port P1 control register	0 : When input port, read port input level. When output port, read the contents of port P1 register 1 : Read the contents of port P1 register though input/output port.	○	○
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			—	—

Port control register 1

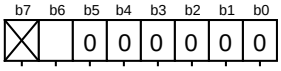


Symbol	Address	When reset
PCR1	02FE16	0016

Bit Symbol	Bit name	Function	R	W
PCR10	Output type selection bit (P00 to P03)	0 : CMOS 1 : N-channel open drain	○	○
PCR11	Output type selection bit (P04-P07)	0 : CMOS 1 : N-channel open drain	○	○
PCR12	Output type selection bit (P10-P13)	0 : CMOS 1 : N-channel open drain	○	○
PCR13	Output type selection bit (P14-P17)	0 : CMOS 1 : N-channel open drain	○	○
PCR14	Output type selection bit (P40-P46)	0 : CMOS 1 : N-channel open drain	○	○
PCR15	Output type selection bit (P110-P113)	0 : CMOS 1 : N-channel open drain	○	○
PCR16	Output type selection bit (P114-P117)	0 : CMOS 1 : N-channel open drain	○	○
PCR17	P40,P43 output clear function selection bit	0 : Cleared by software only 1 : Cleared by software or when output buffer is read by host side.	○	○

Fig.UA-11 Port control register 0, 1

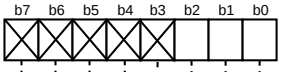
Port control register 2



Symbol: PCR2 | Address: 02FF16 | When reset: 0016

Bit symbol	Bit name	Function	R	W
Reserved bit		Must be set to "0"	O	O
Reserved bit		Must be set to "0"	O	O
Reserved bit		Must be set to "0"	O	O
Reserved bit		Must be set to "0"	O	O
Reserved bit		Must be set to "0"	O	O
Reserved bit		Must be set to "0"	O	O
PCR26	Drive polarity selection bit (P140 to P143)	0 : Low side 1 : High side	O	O
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–

Port control register 3

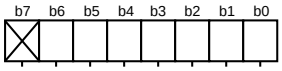


Symbol: PCR3 | Address: 02F716 | When reset: 0016

Bit symbol	Bit name	Function	R	W
PCR30	Output type selection bit (P140 - P143)	0 : CMOS 1 : N-channel open drain	O	O
PCR31	Output type selection bit (P144 - P147)	0 : CMOS 1 : N-channel open drain	O	O
PCR32	Output type selection bit (P42)	0 : CMOS 1 : N-channel open drain	O	O
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–

Fig.UA-12 Port control register 2, 3

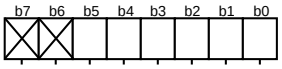
Port P4 input register



Symbol: P4PIN | Address: 02FA16 | When reset: 0XXXXXXX2

Bit Symbol	Bit name	Function	R	W
P4PIN_0	Port P40 input register	For reading P40 pin level	O	–
P4PIN_1	Port P41 input register	For reading P41 pin level	O	–
P4PIN_2	Port P42 input register	For reading P42 pin level	O	–
P4PIN_3	Port P43 input register	For reading P43 pin level	O	–
P4PIN_4	Port P44 input register	For reading P44 pin level	O	–
P4PIN_5	Port P45 input register	For reading P45 pin level	O	–
P4PIN_6	Port P46 input register	For reading P46 pin level	O	–
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–

Port P7 input register

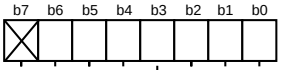


Symbol: P7PIN | Address: 02FB16 | When reset: 00XXXXXX2

Bit Symbol	Bit name	Function	R	W
P7PIN_0	Port P70 input register	For reading P70 pin level	O	–
P7PIN_1	Port P71 input register	For reading P71 pin level	O	–
P7PIN_2	Port P72 input register	For reading P72 pin level	O	–
P7PIN_3	Port P73 input register	For reading P73 pin level	O	–
P7PIN_4	Port P74 input register	For reading P74 pin level	O	–
P7PIN_5	Port P75 input register	For reading P75 pin level	O	–
Nothing is assigned. Can't write to this bit. The value, if read, turns out to be “0”.			–	–

Fig.UA-13 Port P4,P7 input register

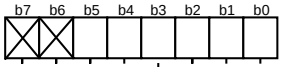
Port function selection register 0



Symbol: PSL0 | Address: 02F8$_{16}$ | When reset: 00$_{16}$

Bit Symbol	Bit name	Function	R	W
PSL00	UART0 I/O pin selection bit	0 : P6$_0$-P6$_3$ 1 : P1$_0$-P1$_3$	O	O
PSL01	UART1 I/O pin selection bit	0 : P6$_4$-P6$_7$ 1 : P1$_4$-P1$_7$	O	O
PSL02	UART2 I/O pin selection bit	0 : P7$_0$-P7$_3$ 1 : P2$_0$-P2$_3$	O	O
PSL03	$\overline{\text{INT7}}$ input pin selection bit	0 : P10$_3$ 1 : P1$_4$	O	O
PSL04	$\overline{\text{INT8}}$ I/O pin selection bit	0 : P10$_4$ 1 : P1$_5$	O	O
PSL05	$\overline{\text{INT9}}$ I/O pin selection bit	0 : P10$_5$ 1 : P1$_6$	O	O
PSL06	$\overline{\text{INT10}}$ I/O pin selection bit	0 : P10$_6$ 1 : P1$_7$	O	O
Reserved bit		Must be set to "0"	O	O

Port function selection register 1



Symbol: PSL1 | Address: 02F9$_{16}$ | When reset: 00$_{16}$

Bit Symbol	Bit name	Function	R	W
PSL10	TA0 output pin selection bit	0 : P4$_0$ 1 : P15$_0$	O	O
PSL11	TA1 output pin selection bit	0 : P4$_1$ 1 : P15$_1$	O	O
PSL12	TA2 output pin selection bit	0 : P4$_2$ 1 : P15$_2$	O	O
PSL13	TB3 output pin selection bit	0 : P9$_3$ 1 : P16$_0$	O	O
PSL14	TB4 output pin selection bit	0 : P9$_4$ 1 : P16$_1$	O	O
PSL15(Note)	$\overline{\text{INT6}}$-$\overline{\text{INT11}}$ input pin switching bit	0 : P9$_7$,P10$_3$-P10$_7$ 1 : P12$_0$,P12$_1$-P12$_5$	O	O
PSL16	P6$_6$: f1 output function selection bit	0 : P6$_6$ as GPIO 1 : P6$_6$ as f1 output	—	—
PSL17	P6$_7$: f1 output function selection bit	0 : P6$_7$ as GPIO 1 : P6$_7$ as f1 output	—	—

Note. If this is set to "1" then port function selection register 1 (address 02F8$_{16}$) bit 3 to bit6 setting will be ignored.

Fig.UA-14 Port function register 0,1

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

Pin name	Connection
Ports P0 to P10 (excluding P8$_5$)	After setting for input mode, connect every pin to V_{SS} or V_{CC} via a resistor; or after setting for output mode, leave these pins open.
Ports P11 to P16	After setting for input mode, connect every pin to V_{SS} or V_{CC} via a resistor; or after setting for output mode, leave these pins open.
X_{OUT} (Note)	Open
$\overline{NMI}$	Connect via resistor to V_{CC} (pull-up)
AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}, M1	Connect to V_{SS}

Note: With external clock input to X_{IN} pin.

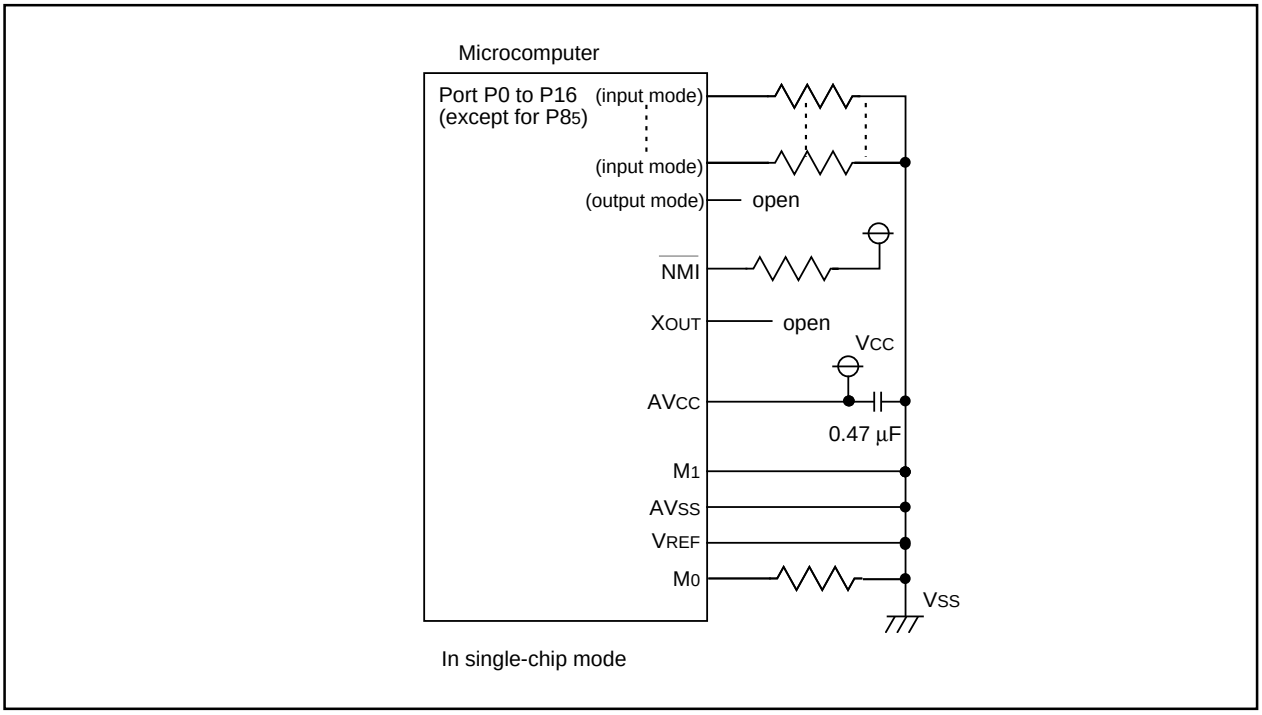


Fig.UA-15 Example connection of unused pins

Usage Precaution

Timer A (timer mode)

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

Timer A (event counter mode)

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

(2) When stop counting in free run type, set timer again.

Timer A (one-shot timer mode)

(1) Setting the count start flag to "0" while a count is in progress causes as follows:
- The counter stops counting and a content of reload register is reloaded.
- The TAiOUT pin outputs "L" level.
- The interrupt request generated and the timer Ai interrupt request bit goes to "1".

(2) The timer Ai interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:
- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

Timer A (pulse width modulation mode)

(1) The timer Ai interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:
- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

(2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TAiOUT pin is outputting an "H" level in this instance, the output level goes to "L", and the timer Ai interrupt request bit goes to "1". If the TAiOUT pin is outputting an "L" level in this instance, the level does not change, and the timer Ai interrupt request bit does not becomes "1".

Timer B (timer mode, event counter mode)

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

Timer B (pulse period/pulse width measurement mode)

(1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to “1”.

(2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

A-D Converter

(1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).
In particular, when the Vref connection bit is changed from “0” to “1”, start A-D conversion after an elapse of 1 μs or longer.

(2) When changing A-D operation mode, select analog input pin again.

(3) Using one-shot mode or single sweep mode
Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)

(4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1
Use the undivided main clock as the internal CPU clock.

Stop Mode and Wait Mode

(1) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to “L” level until main clock oscillation is stabilized.

(2) When switching to either wait mode or stop mode, instructions occupying four bytes either from the WAIT instruction or from the instruction that sets the every-clock stop bit to “1” within the instruction queue are prefetched and then the program stops. So put at least four NOPs in succession either to the WAIT instruction or to the instruction that sets the every-clock stop bit to “1”.

Interrupts

(1) Reading address 00000_{16}
• When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
The interrupt request bit of the certain interrupt written in address 00000_{16} will then be set to “0”.
Reading address 00000_{16} by software sets enabled highest priority interrupt source request bit to “0”.
Though the interrupt is generated, the interrupt routine may not be executed.
Do not read address 00000_{16} by software.

(2) Setting the stack pointer
• The value of the stack pointer immediately after reset is initialized to 0000_{16}. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.
When using the $\overline{\text{NMI}}$ interrupt, initialize the stack point at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.

(3) The $\overline{\text{NMI}}$ interrupt
• As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be disabled. Connect it to the V_{CC} pin via a resistor (pull-up) if unused. Be sure to work on it.
• Do not get either into stop mode with the $\overline{\text{NMI}}$ pin set to “L”.

(4) External interrupt

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

(5) Rewrite the interrupt control register

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

Example 1:

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

Example 2:

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

Example 3:

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

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

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

Instructions : AND, OR, BCLR, BSET

Noise

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

Table.ZA-1 Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
Vcc	Supply voltage		VCC=AVCC	-0.3 to 4.6	V
AVcc	Analog Supply voltage		VCC=AVCC	-0.3 to 4.6	V
VI	Input voltage	$\overline{\text{RESET}}$,M0,M1,VREF,XIN,P00 to P07, P10 to P17,P20 to P27,P30 to P37, P40 to P47,P50 to P57,P64 to P67 P86,P87,P90 to P97,P100 to P107, P110 to P117,P120 to P127, P140 to P147,P150 to P157,P160,P161		-0.3 to Vcc+0.3	V
		P60 to P63,P70 to P77,P80 to P85 P130 to P137		-0.3 to 5.8	V
VO	output voltage	P00 to P07,P10 to P17,P20 to P27, P30 to P37,P40 to P47,P50 to P57, P64 to P67,P86,P87,P90 to P97, P100 to P107,P110 to P117,P120 to P127, P140 to P147,P150 to P157,P160 ,P161,XOUT		-0.3 to Vcc+0.3	V
		P60 to P63,P70 to P77, P80 to P84 P130 to P137		-0.3 to 5.8	V
Pd	Power dissipation		Ta=25 °C	300	mW
Topr	Operating ambient temperature			-20 to 85	°C
Tstg	Storage temperature			-40 to 125	°C

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

Symbol	Parameter			Standard			Unit
				Min.	Typ.	Max.	
Vcc	Supply voltage			3.0	3.3	3.6	V
AVcc	Analog Supply voltage				Vcc		V
Vss	Supply voltage				0		V
AVss	Analog Supply voltage				0		V
V_{IH}	"H" input voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$,$P2_0$ to $P2_7$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_4$ to $P6_7$,$P8_6$,$P8_7$,$P9_0$ to $P9_7$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P14_0$ to $P14_7$, $P15_0$ to $P15_7$, $P16_0$,$P16_1$,X_{IN}, $\overline{RESET}$, M0,M1		0.8Vcc		Vcc	V
		$P6_0$ to $P6_3$, $P7_0$ to $P7_7$,$P8_0$ to $P8_4$,$P8_5$,$P13_0$ to $P13_7$, PSA_0 to PSA_2, PSB_0 to PSB_2		0.8Vcc		5.5	V
		LAD_0 to LAD_3,$\overline{LFARAME}$,LCLK,SERIRQ,$\overline{CLKRUN}$		0.6Vcc		Vcc	V
		SDA_0,SCL_0,SDA_1,SCL_1,$P6_0$ to $P6_3$	I^2C-BUS input level selected	0.7Vcc		5.5	V
			SMBUS input level selected	1.4		5.5	
V_{IL}	"L" input voltage	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$,$P2_0$ to $P2_7$, $P3_0$ to $P3_7$,$P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_4$ to $P6_7$,$P8_6$,$P8_7$,$P9_0$ to $P9_7$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P14_0$ to $P14_7$, $P15_0$ to $P15_7$, $P16_0$,$P16_1$,X_{IN}, $\overline{RESET}$, M0,M1		0		0.2Vcc	V
		$P6_0$ to $P6_3$, $P7_0$ to $P7_7$,$P8_0$ to $P8_4$,$P8_5$,$P13_0$ to $P13_7$, PSA_0 to PSA_2, PSB_0 to PSB_2		0		0.2Vcc	V
		LAD_0 to LAD_3,$\overline{LFARAME}$,LCLK,SERIRQ,$\overline{CLKRUN}$		0		0.2Vcc	V
		SDA_0,SCL_0,SDA_1,SCL_1,$P6_0$ to $P6_3$	I^2C-BUSes input level selected	0		0.3Vcc	V
			SMBUS input level selected	0		0.6	

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

Symbol	Parameter			Standard			Unit
				Min.	Typ.	Max.	
$I_{OH (peak)}$	"H"peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$,$P2_0$ to $P2_7$,$P3_0$ to $P3_7$,$P4_0$ to $P4_7$, $P5_0$ to $P5_7$,$P6_4$ to $P6_7$,$P8_6$ to $P8_7$,$P9_0$ to $P9_7$,$P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P14_0$ to $P14_7$,$P15_0$ to $P15_7$, $P16_0$,$P16_1$				-10.0	mA
$I_{OL (peak)}$	"L"peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$,$P2_0$ to $P2_3$,$P3_0$ to $P3_7$,$P4_0$ to $P4_7$, $P5_0$ to $P5_7$,$P6_0$ to $P6_7$,$P7_6$ to $P7_7$,$P8_0$ to $P8_4$, $P8_6$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$, $P11_0$ to $P11_7$,$P12_0$ to $P12_7$, $P13_0$ to $P13_7$,$P14_4$ to $P14_7$, $P15_0$ to $P15_7$,$P16_0$,$P16_1$				10.0	mA
		$P2_4$ to $P2_7$				20.0	mA
$I_{OL (peak)}$	"L"peak output current	$P14_0$ to $P14_3$	Driven ability:High			20.0	mA
			Driven ability:Low			10.0	mA
$I_{OH (avg)}$	"H" average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$,$P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$,$P5_0$ to $P5_7$,$P6_0$ to $P6_7$,$P8_6$ to $P8_7$,$P9_0$ to $P9_7$, $P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P12_0$ to $P12_7$,$P14_0$ to $P14_7$, $P15_0$ to $P15_7$,$P16_0$,$P16_1$				-5.0	mA
$I_{OL (avg)}$	"L"average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$,$P2_0$ to $P2_3$,$P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$,$P6_0$ to $P6_7$,$P7_6$ to $P7_7$,$P8_0$ to $P8_4$, $P8_6$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$,$P11_0$ to $P11_7$,$P12_0$ to $P12_7$, $P13_0$ to $P13_7$,$P14_0$ to $P14_7$,$P15_0$ to $P15_7$,$P16_0$,$P16_1$				5.0	mA
		$P2_4$ to $P2_7$				15.0	mA
$I_{OL (avg)}$	"L"average output current	$P14_0$ to $P14_3$	Driven ability:High			15.0	mA
			Driven ability:Low			5.0	mA
f (X_{IN})	Main clock input oscillation frequency	Non wait		0		4	MHz
		With wait		0		8	MHz
f (X_{CIN})	Subclock oscillation frequency				32.768	50	kHz

Note1 : The average output current is the average value during the 100ms period limited current.
Note2 : The value are as follow:
The sum of I_{OL} (peak) of P0,P1,P2,$P8_6$ to $P8_7$,P9,P10,P11,$P12_0$ to $P12_6$,$P15_3$ to $P15_7$ P16 should be under 80mA.
The sum of I_{OH} (peak) of P0,P1,P2,$P8_6$ to $P8_7$,P9,P10,P11,$P12_0$ to $P12_6$,$P15_3$ to $P15_7$ P16 should be under 80mA.
The sum of I_{OL} (peak) of P3,P4,P5,P6,P7,$P8_0$ to $P8_4$,P13,P14,$P15_0$ to $P15_2$ should be under 80mA.
The sum of I_{OH} (peak) of P3,P4,P5,$P6_4$ to $P6_7$,P14,$P15_0$ to $P15_2$ should be under 80mA.

Table.ZA-4 Electorical characteristice (referenced to Vcc=3.0V,Vss=0V,Ta=25°C,f(XIN)=8MHz with WAIT)

Symbol	Parameter			Measuring condition	Standard			Unit
					Min.	Typ.	Max.	
VOH	High output voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161		IOH=-1mA	2.5			V
VOH	High output voltage	XOUT	HIGH POWER	IOH=-0.1mA	2.5			V
			LOWPOWER	IOH=-50µA	2.5			
	High output voltage	XCOUT	HIGHPOWER	With no load applied		3.0		V
			LOWPOWER	With no load applied		1.6		
VOL	Low output voltage	P00 to P07, P10 to P17, P20 to P23, P30 to P37, P40 to P47,P50 to P57, P60 to P67, P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161		IOL=1mA			0.5	V
		P24 to P27		VCC=3V, IOL=3mA			0.5	V
VOL	Low output voltage	P140 to P143	HIGH POWER	IOL=3mA			0.5	V
			LOWPOWER	IOL=1mA			0.5	V
VOL	Low output voltage	XOUT	HIGH POWER	IOH=0.1mA			0.5	V
			LOWPOWER	IOH=50µA			0.5	
	Low output voltage	XCOUT	HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
VT+VT-	Hysteresis	TA0IN to TA4IN, TB0IN to TB5IN, $\overline{INT0}$ to $\overline{INT11}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CTS1}$, $\overline{CTS2}$, CLK0, CLK1, CLK2, CLK3, CLK4, SIN3, SIN4, RXD0, RXD1, RXD2, ICCK, $\overline{NMI}$, $\overline{KI00}$ to $\overline{KI07}$			0.2		0.8	
VT+VT-	Hysteresis	$\overline{RESET}$			0.2		1.8	V
IIH	HIGH input current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161, XIN, $\overline{RESET}$, M0, M1		VI=3V			4.0	µA
IIL	Low input current	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P130 to P137, P140 to P147, P150 to P157, P160, P161, XIN, $\overline{RESET}$, M0, M1		VI=0V			-4.0	µA
R PULLUP	Pull-up resisitance	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P86, P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, P140 to P147, P150 to P157, P160, P161		VI=0V	66.0	120.0	500.0	kΩ
R fXIN	Feedback resistance	XIN				3.0		MΩ
R fXCIN	Feedback resistance	XCIN				10.0		MΩ
V RAM	RAM retention voltage			When clock is stopped	2.0			V
I CC	Power supply current	When reset, the output pins are opened , the other pins are connected to Vss.		f(XIN)=8MHz,Square wave without division		12.5	24.25	mA
				f(XCIN)=32kHz,Square wave When operation under RAM		40.0		µA
				f(XCIN)=32kHz,Square wave When operation under Flash memory		300.0		µA
				f(XCIN)=32kHz,With WAIT oscillation capacity High (Note1)		4.5		µA
				f(XCIN)=32kHz,With WAIT oscillation capacity Low (Note1)		2.5		µA
				Ta=25°C When clock is stopprd			3.0	µA
				Ta=85°C When clock is stopprd			60.0	µA

Table.ZA-5 A-D conversion characteristics (referenced to Vcc=AVcc=V_{REF}=3V,Vss=AVss=0V at Ta=25°C,f(X_{IN})=8MHz unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
—	Resolution		V_{REF} =V_{CC}			10	Bits
—	Absolute accuracy	8 bit	V_{REF} =V_{CC}=3V, $Ø_{AD}$=f_{AD}/2			±2	LSB
—		10 bit				±6	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC}	10		40	kΩ
t_{CONV}	Conversion time	8 bit		12.25			µs
		10 bit		14.75			µs
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Table.ZA-6 D-A conversion characteristics (referenced to Vcc=AVcc=V_{REF}=3V,Vss=AVss=0V at Ta=25°C,f(X_{IN})=8MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
-	Resolution				8	Bits
-	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(Note1)			1.0	mA

Note1: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "00_{16}".
The A-D converter's ladder resistance is not included.
When the content of D-A register is not "00", the I_{VREF} will also be sent even if V_{REF} is disconnected.

Table.ZA-7 Comparator characteristics (referenced to Vcc=AVcc=V_{REF}=3V to 3.6V,Vss=AVss=0V at Ta=25°C)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
-	Resolution				4	Bits
-	Absolute accuracy				1/2	LSB
T_{CONV}	Conversion time	when f(X_{IN}) = 8MHz			3.5	µs
		when f(X_{IN}) = 4MHz			7	µs
V_{IA}	Analog input voltage		0		V_{CC}	V
I_{IA}	Analog input current				5.0	µs
R_{LADDER}	Ladder resistance		20	40	50	kΩ

Timing requirements (referenced to Vcc=3V,Vss=0V at Ta=25°C unless otherwise specified)

Table.ZA-8 External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	125		ns
$t_{w(H)}$	External clock input HIGH pulse width	50		ns
$t_{w(L)}$	External clock input LOW pulse width	50		ns
t_r	External clock rising time		18	ns
t_f	External clock falling time		18	ns

Timing requirements (referenced to Vcc=3V,Vss=0V at Ta=25°C unless otherwise specified)

Table.ZA-9 Timer A input (The count input of event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input "H" pulse width	60		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	60		ns

Table.ZA-10 Timer A input (The gating input of timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input "H" pulse width	300		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	300		ns

Table.ZA-11 Timer A input (The external trigger input of one shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input "H" pulse width	150		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	150		ns

Table.ZA-12 Timer A input (The external trigger input of pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input "H" pulse width	150		ns
$t_{w(TAL)}$	TAiIN input "L" pulse width	150		ns

Table.ZA-13 Timer A input (The up down input of event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input "H" pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input "L" pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

Timing requirements (referenced to Vcc=3V,Vss=0V at Ta=25°C unless otherwise specified)

Table.ZA-14 Timer B input (The count input of event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (single edge count)	150		ns
$t_{w(TBH)}$	TBiIN input "H" pulse width (single edge count)	60		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width (single edge count)	60		ns
$t_{c(TB)}$	TBiIN input cycle time (double edge count)	300		ns
$t_{w(TBH)}$	TBiIN input "H" pulse width (double edge count)	160		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width (double edge count)	160		ns

Table.ZA-15 Timer B input (Pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input "H" pulse width	300		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width	300		ns

Table.ZA-16 Timer B input (Pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input "H" pulse width	300		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width	300		ns

Table.ZA-17 A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (The Min. of trigger)	1500		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input "L" pulse width	200		ns

Table.ZA-18 Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input "H" pulse width	150		ns
$t_{w(CKL)}$	CLKi input "L" pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	50		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table.ZA-19 External interrupt $\overline{INTi}$ input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input "H" pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input "L" pulse width	380		ns

Timing requirements (referenced to Vcc = 3.0 to 3.6V, Vss = 0V, Ta =25 °C)

Table. ZA-20 Multi-master I^2C-BUS line

Symbol	Parameter	Standard clock mode		High-speed clock mode		Unit
		Min.	Max.	Min.	Max.	
tBUF	Bus free time	4.7		1.3		µs
tHD;STA	The hold time in start condition	4.0		0.6		µs
tLOW	The hold time in SCL clock "0" status	4.7		1.3		µs
tR	SCL, SDA signals' rising time		1000	20+0.1Cb	300	ns
tHD;DAT	Data hold time	0		0	0.9	µs
tHIGH	The hold time in SCL clock "1" status	4.0		0.6		µs
tF	SCL, SDA signals' falling time		300	20+0.1Cb	300	ns
tsu;DAT	Data setup time	250		100		ns
tsu;STA	The setup time in restart condition	4.7		0.6		µs
tsu;STO	Stop condition setup time	4.0		0.6		µs

Table. ZA-21 PS2 interface (referenced to Vcc = 3.0 to 3.6V, Vss = 0V, Ta =25 °C)

Symbol	Parameter	Standard			Unit
		Min.	Typ.	Max.	
twL	PS2 clock "L" pulse width	30		50	µs
twH	PS2 clock "H" pulse width	30		50	µs
tsu	PS2 data setup time	5			µs
th	PS2 data hold time	0			ns
td	PS2 data delay time			twL-5	µs
tv	PS2 data valid time	0		twL-5	µs

Table. ZA-22 LPC bus interface/serial interrupt output

Symbol	Parameter		Standard			Unit
			Min.	Typ.	Max.	
$t_{C(CLK)}$	LCLK clock input cycle time		30		∞	ns
$t_{WH(CLK)}$	LCLK clock input "H" pulse width		11			ns
$t_{WL(CLK)}$	LCLK clock input "L" pulse width		11			ns
$t_{su(D-C)}$	Input setup time	LAD_3-LAD_0 $\overline{LFRAME}$	9			ns
		SERIRQ,$\overline{CLKRUN}$	7			ns
$t_{h(C-D)}$	LAD_3-LAD_0,SERIRQ,$\overline{CLKRUN}$, $\overline{LFRAME}$ input hold time		0			ns
$t_{V(C-D)}$	LAD_3-LAD_0,SERIRQ,$\overline{CLKRUN}$ output delay time		2		11	ns
$t_{off(A-F)}$	LAD_3-LAD_0,SERIRQ,$\overline{CLKRUN}$ floating output delay time				28	ns

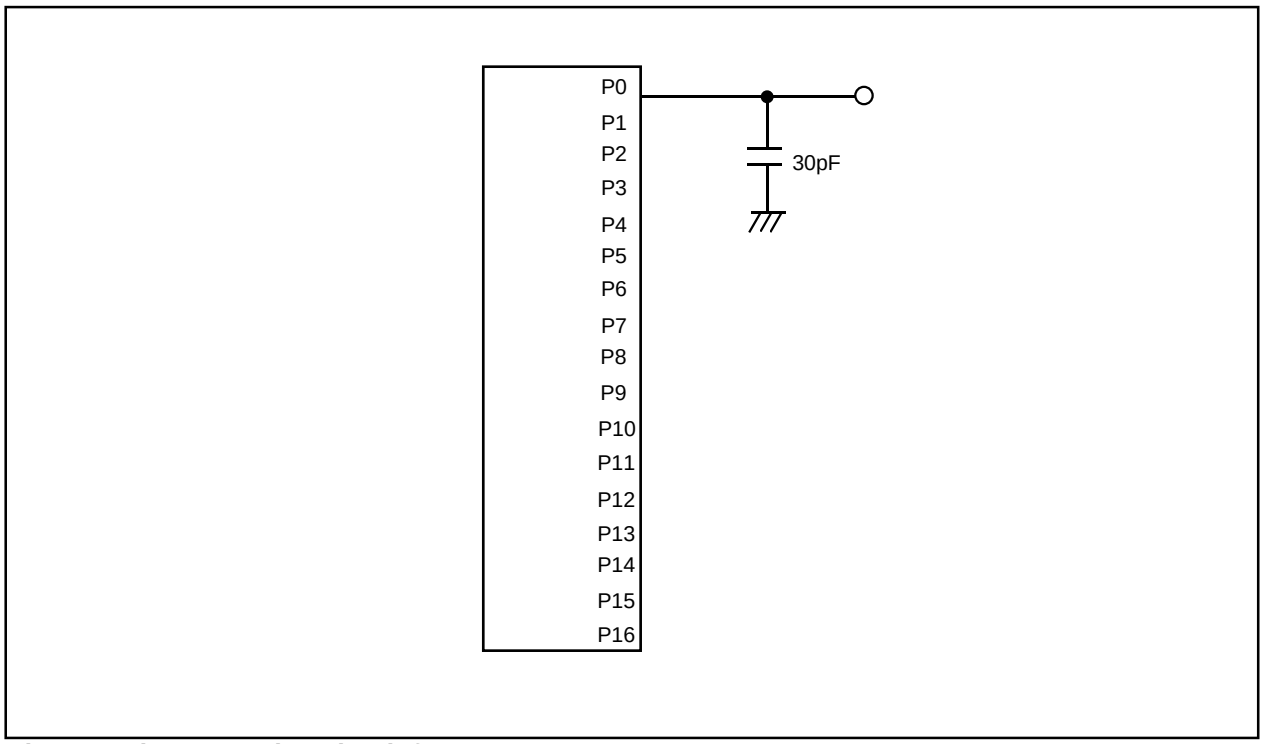


Fig.ZA-1 The measuring circuit for port 0 to port 16

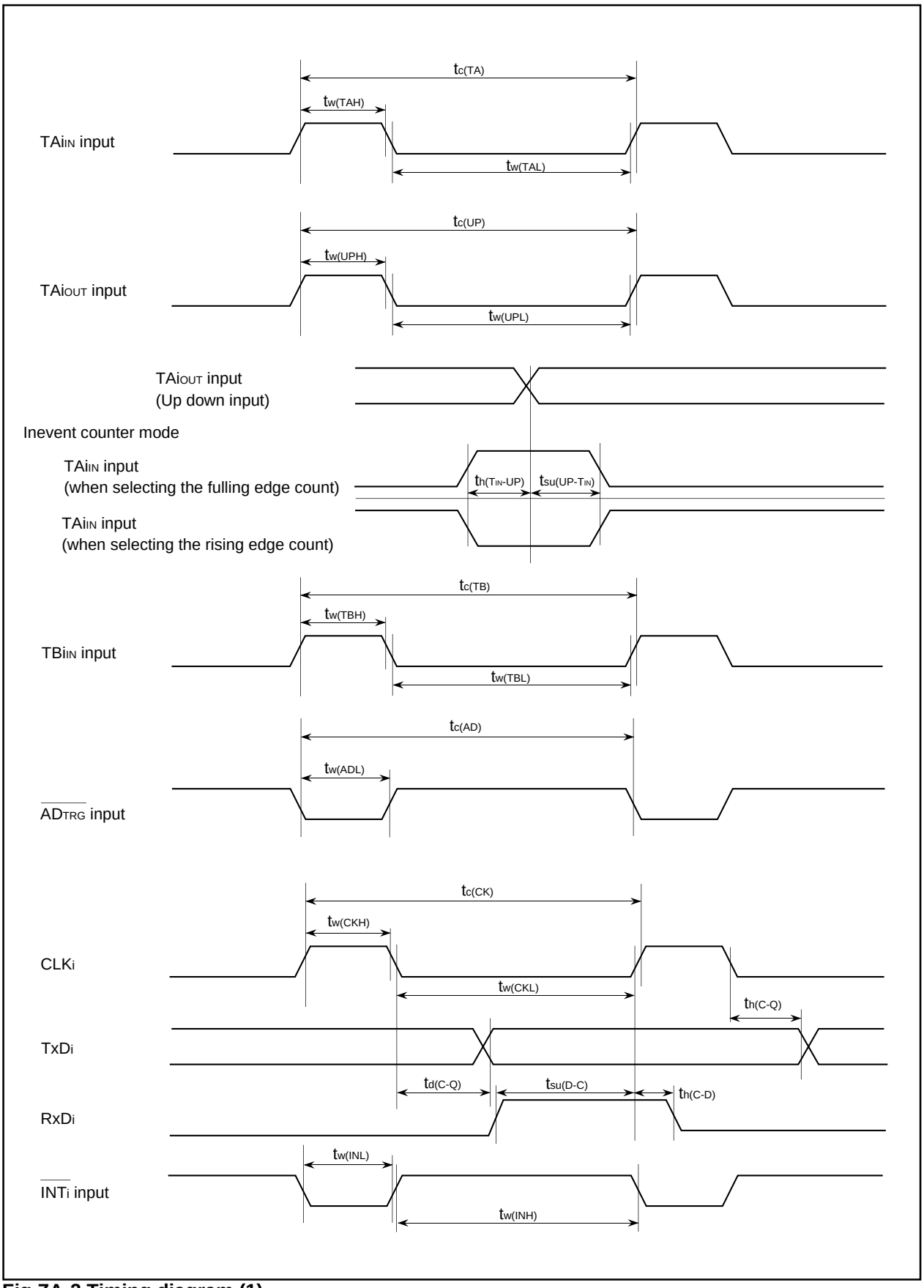


Fig.ZA-2 Timing diagram (1)

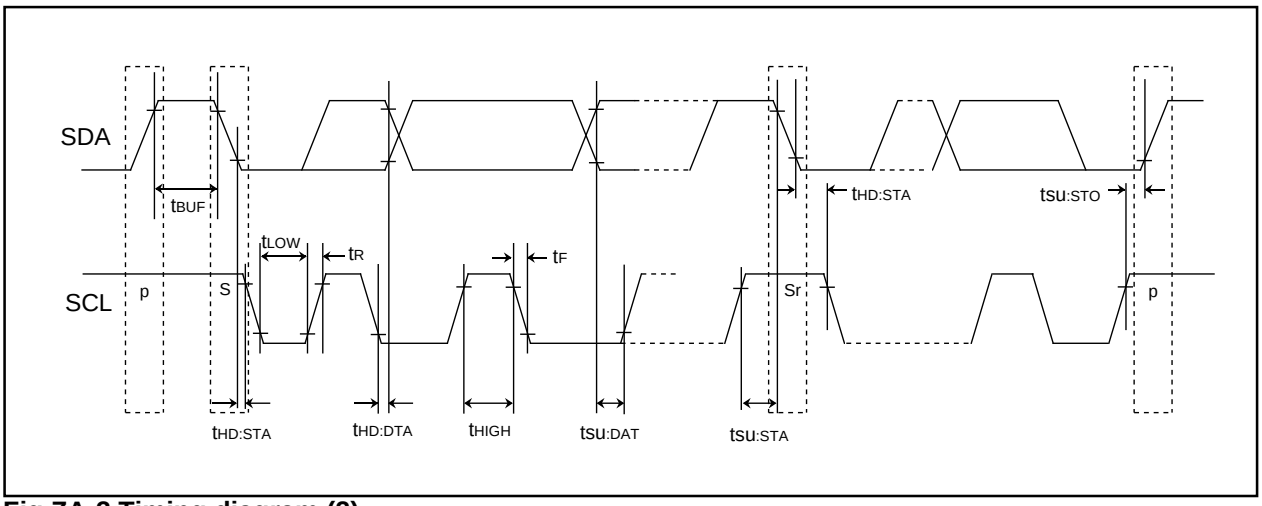
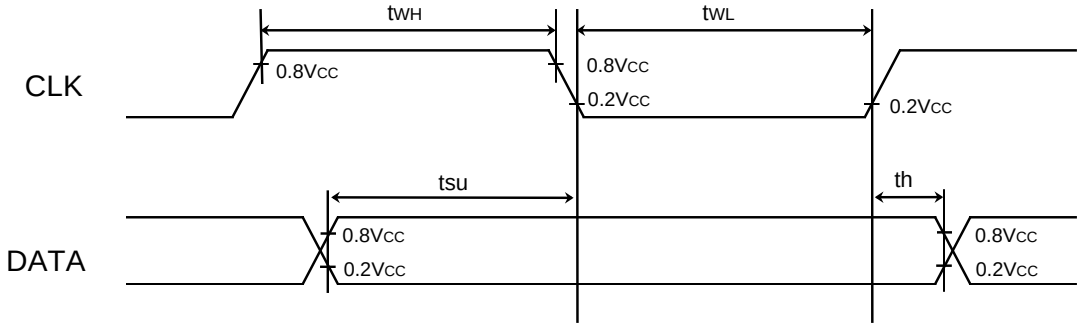


Fig.ZA-3 Timing diagram (2)

PS/2 interface timing diagram

In receiving



In transmitting

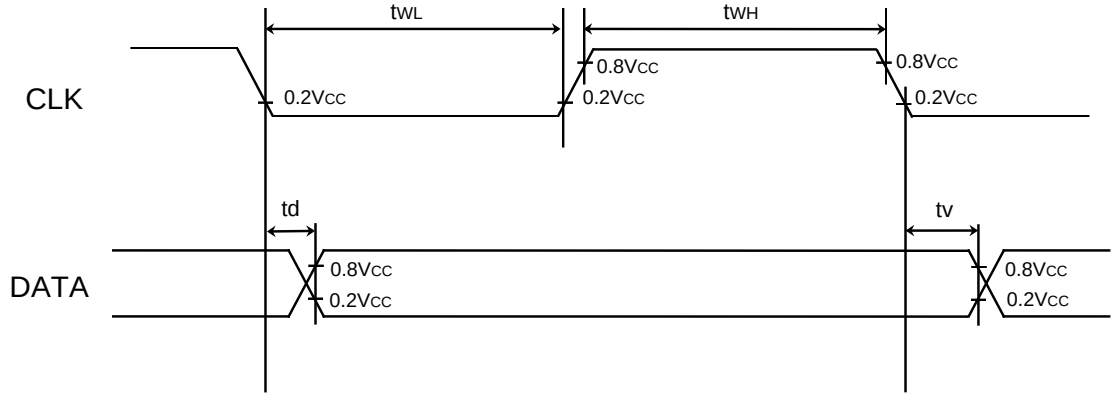


Fig.ZA-4 Timing diagram (3)

LPC bus interface/serial interrupt output timing

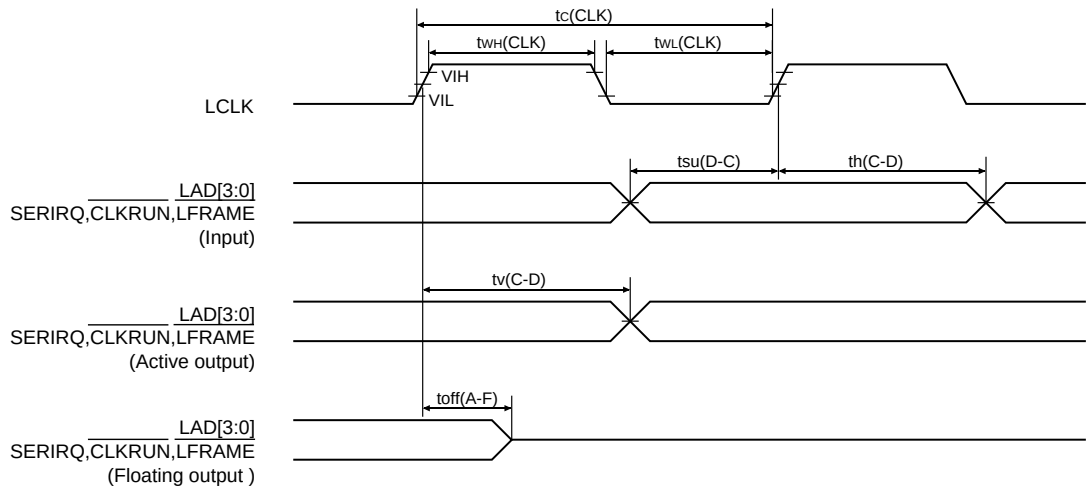


Fig.ZA-5 Timing diagram (4)

Feature Outline

Table AB-1 shows the feature outline of M16C/6K7 (build-in NEW DINOR flash memory version).

Table.AB-1 Feature outline of M16C/6K7 (build-in NEW DINOR flash memory version)

Item		Feature
Power supply voltage		3.0-3.6V (f(X_{IN})=8MHz, 0 Wait)
Flash memory operation mode		3 modes (parallel I/O, standard serial I/O, CPU reprogram)
Erase block division	User ROM area	See Fig.AB-1
	Boot ROM area	1 division (4K bytes) (Note1)
Program method		2-byte unit
Erase method		Block erase
Program/ erase control method		Program/ erase controlled by s/w commands
Number of command		5 commands
Program/ erase count		100 times
ROM code protect		Support for parallel I/O and standard serial I/O modes

Note1: The control program for standard serial I/O mode is stored in boot ROM area when shipping from factory. The area can only be erased or programmed by parallel I/O mode.

Flash Memory

The M16C/6K7 (build-in flash memory version) contains the NEW DINOR type flash memory, which is applied 1 power supplies V_{CC}=3.3V when using CPU reprogram or standard serial I/O mode. For the flash memory, 3 flash memory modes are available in which to read, program and erase. They are parallel I/O mode, standard serial I/O mode and CPU reprogram mode. For parallel I/O mode, a programmer is used. For standard serial I/O and CPU reprogram modes, the flash memory is manipulated by CPU. Each mode is detailed in the pages to follow.

Fig. AB-1 shows that flash memory is divided into several blocks. Erasing is in block unit.

In addition to the ordinary user ROM area there is a boot ROM area to store the control program for the CPU reprogram and standard serial I/O modes. The control program for standard serial I/O mode is stored in boot ROM area when shipping from factory. User can reprogram the program to suit its own application system. The area can only be erased or programmed by parallel I/O mode.

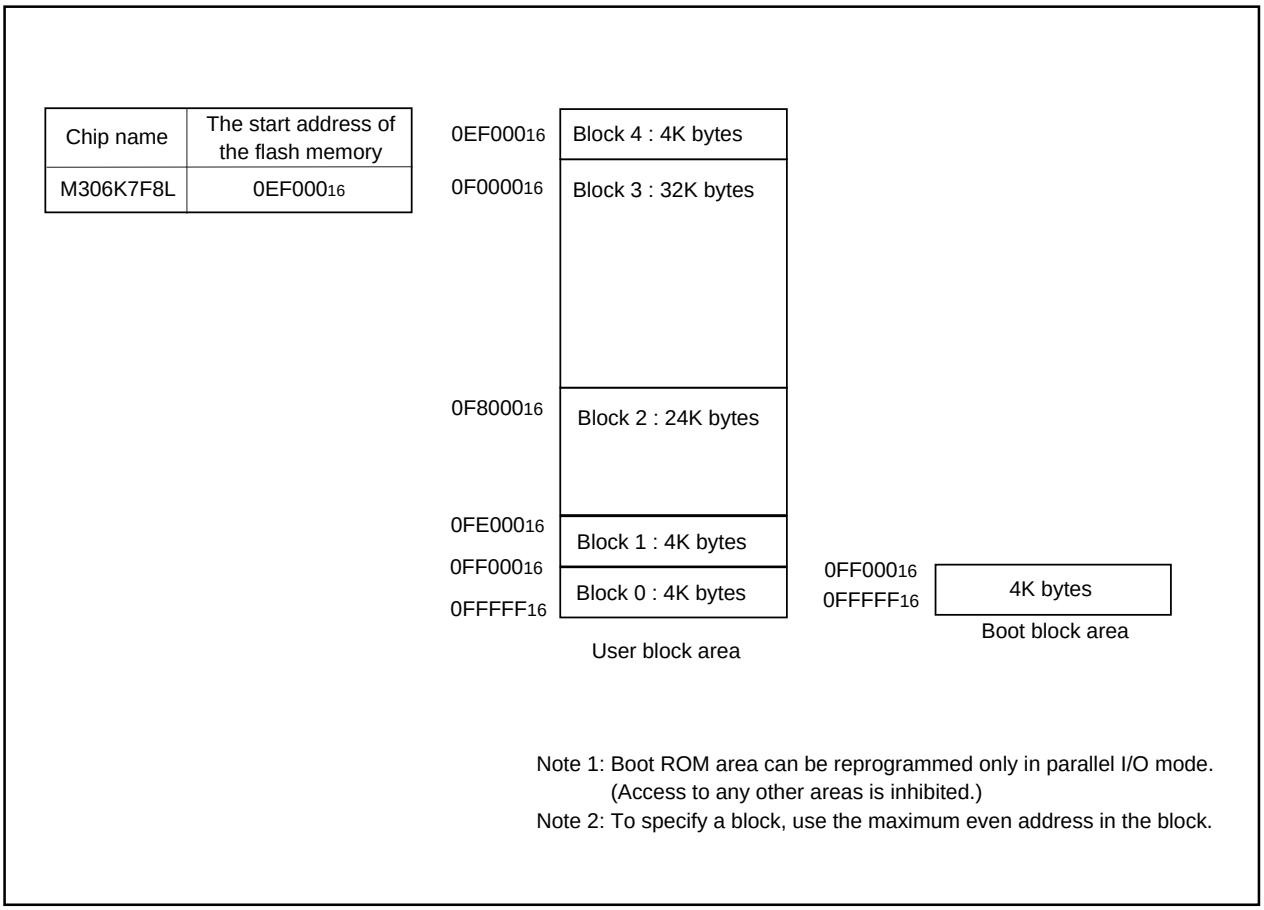


Fig.AB-1 Block diagram of flash memory version

CPU reprogram mode

In CPU reprogram mode, the on-chip flash memory can be operated on (read, program or erase) under the control of CPU.

In CPU reprogram mode, only the user ROM area shown in Fig.AB-1 can be reprogrammed. The boot ROM area cannot be reprogrammed. Make sure the program and block erase commands are issued only for each block of the user ROM area.

The control program for CPU reprogram mode can be stored in either user ROM or boot ROM area. In CPU reprogram mode, because the flash memory cannot be read form CPU, the control program must be transferred to the RAM area before execution.

Microcomputer mode and Boot mode

The control program for CPU reprogram mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes disable.)

See Fig.AB-1 for details about the boot ROM area.

Normal microcomputer mode is entered when reset with pulling “L” of M_0. In this case, the CPU starts operating the control program in user ROM area.

If the microcomputer is reset with M_0 being “H” and M_1 being “L”, the CPU starts operating the control program in boot ROM area. This mode is called as “boot” mode.

Block address

Block address refers to the maximum even address of each block. The address is used in block erase command.

Feature Outline (CPU reprogram mode)

In CPU reprogram mode, the CPU erases, programs and reads the on-chip flash memory as instructed by S/W commands. The reprogram control program must be transferred to the RAM area before it can be executed.

The CPU reprogram mode is accessed by writing “1” to the CPU reprogram select bit (bit 1 in address $03B7_{16}$). S/W commands are accepted once the mode is accessed.

In CPU reprogram mode, the writing and reading of the commands and data should be in even address (“0” for byte address A_0) in 16-bit unit, so the 8-bit unit S/W commands should be written in even address. Commands are ignored with odd address.

Use S/W commands to control flash memory programming and erasing. Whether the programming and erasing operation terminates correctly or in error can be verified by reading the status register.

Fig.BB-1 shows the flash control register.

Bit 0 is the RY/$\overline{BY}$ status flag exclusively used to read the operating status of the flash memory. During programming and erasing operation, it is “0”, otherwise it is “1”.

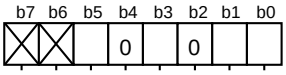
Bit 1 is the CPU reprogram mode select bit. When the bit is set to “1”, CPU reprogram mode is entered S/W commands then can be accessed. In CPU reprogram mode, the CPU cannot access the on-chip flash memory directly. Therefore, use the control program in RAM to write the bit to “1”. To set the bit, it is necessary to write “0” and then write “1” in succession. The bit can be cleared to “0” by only writing the “0”.

Bit 3 is the flash memory reset bit used to reset the control circuit of the on-chip flash memory. The bit is used when exiting the CPU reprogram mode and when flash memory access has failed. When the CPU reprogram mode select bit is “1”, writing “1” to the bit resets the control circuit. To release the reset, it is necessary to set the bit to “0”. If the control circuit is reset while erasing is in progress, the wait for 5 ms is needed so that the flash memory can restore to the normal operation.

Bit 5 of the flash control register 0 is the user ROM select bit. It is enabled only in boot mode. When the bit is set to “1”, the accessed area is switched from boot ROM to user ROM. When CPU reprogram mode is entered in boot mode, please set this bit to “1”. The bit is disabled when program starts in user ROM. Please write the bit with the program that is not located in on-chip flash memory area.

Fig.BB-2 shows a flowchart for the setting/ releasing the CPU reprogram mode.

Flash memory control register

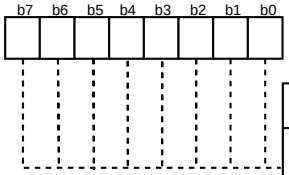


Symbol	Address	When reset
FMR	$03B7_{16}$	$XXXX0001_2$

Bit symbol	Bit name	Function	R	W
FMR0	$\overline{RY/BY}$ signal status bit	0: Busy (be written and erased) 1: Ready	○	×
FMR1	CPU reprogram mode select bit (Note 1)	0: Normal mode 1: CPU reprogram mode	○	○
Reserved bit		Must be “0”.	○	○
FMR3	Flash memory reset bit (Note 2)	0: Normal operation 1: Reset	○	○
Reserved bit		Must be “0”.	○	○
FMR5	User ROM area select bit (Note 3) (Only enabled in boot mode)	0: Access to boot ROM area 1: Access to user ROM area	○	○
Nothing is assigned. When write, set “0”. When read, values is indeterminate.			—	—

Note 1: To write “1” to the bit, it is necessary to write “0” and “1” in succession. Otherwise the bit will not be “1”. Please do not enter interrupt and DMA.
Note 2: It is enabled only CPU reprogram mode select bit is “1”. After setting to “1”(reset), please write “0” in succession.
Note 3: Please write this bit with program that is not located in on-chip flash memory area.

Flash memory identification register

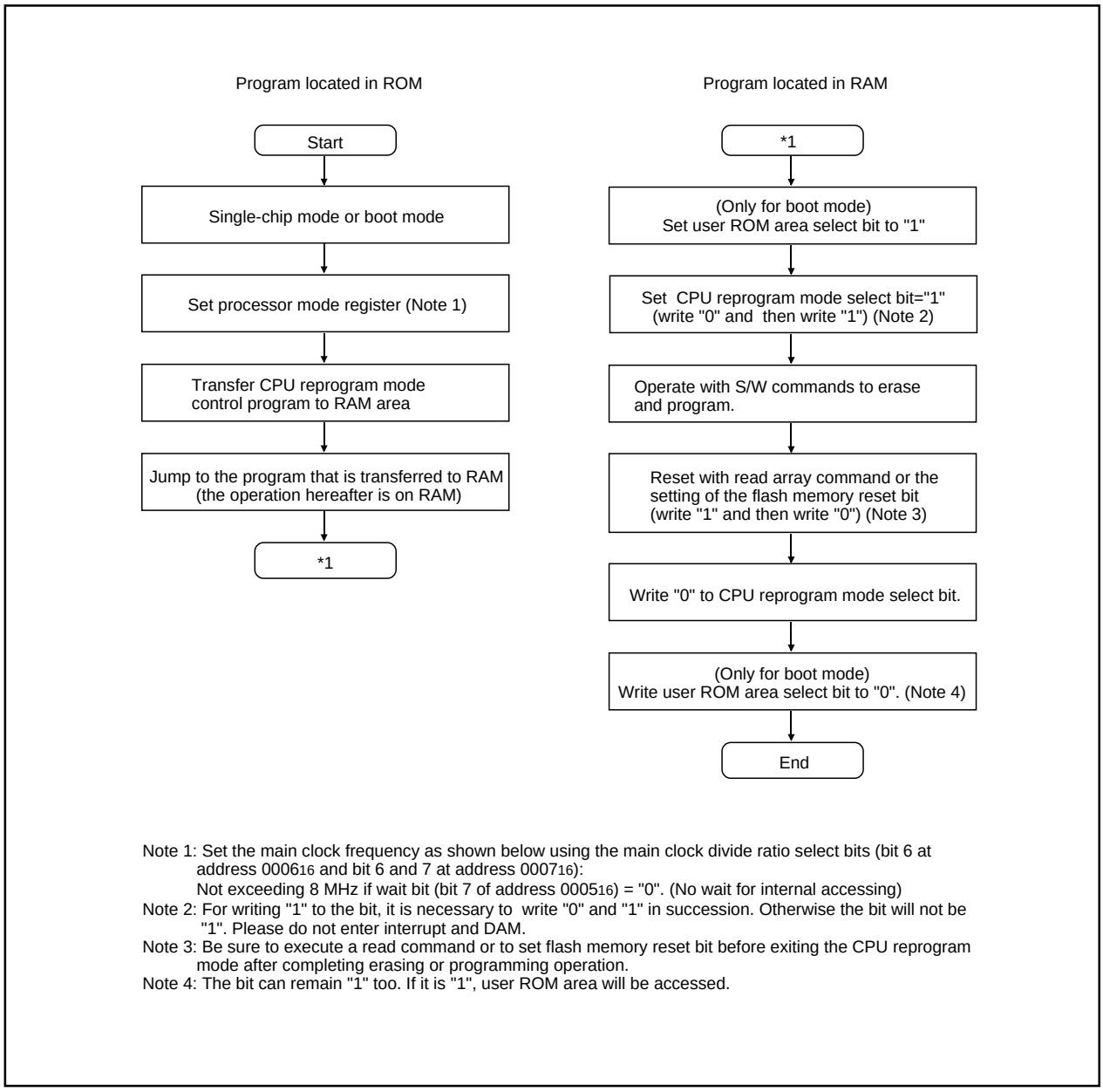


Symbol	Address	When reset
FTR	$03B4_{16}$	00000000_2

Function	R	W
The value after reset 00000000_2 : M16C/6K7 Group $XXXX0001_2$: M16C/6K5 Group (Note)	○	×

Note: Address $03B4_{16}$ of M16C/6K5 Group is the flash memory control register.

Fig.BB-1 The structure of flash memory control register and flash memory identification register



Note 1: Set the main clock frequency as shown below using the main clock divide ratio select bits (bit 6 at address 0006_{16} and bit 6 and 7 at address 0007_{16}):
Not exceeding 8 MHz if wait bit (bit 7 of address 0005_{16}) = "0". (No wait for internal accessing)
Note 2: For writing "1" to the bit, it is necessary to write "0" and "1" in succession. Otherwise the bit will not be "1". Please do not enter interrupt and DAM.
Note 3: Be sure to execute a read command or to set flash memory reset bit before exiting the CPU reprogram mode after completing erasing or programming operation.
Note 4: The bit can remain "1" too. If it is "1", user ROM area will be accessed.

Fig.BB-2 CPU reprogram mode set/reset flowchart

Precautions on CPU reprogram mode

Described below are the precautions to be observed in programming the flash memory in CPU reprogram mode.

(1) Operation speed

During CPU reprogram mode, set the main clock frequency as shown below using the main clock divide ratio select bits (bit 6 at address 0006_{16} and bit 6 and 7 at address 0007_{16}):

Not exceeding 8MHz if wait bit (bit 7 of address 0005_{16}) = “0”. (No wait for internal accessing)

(2) Instructions inhibited against use

The instructions listed below cannot be used during CPU reprogram mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction and BRK instruction

(3) Interrupts inhibited against use

The $\overline{\text{NMI}}$, address match and WDC interrupts cannot be used during CPU reprogram mode because they refer to the internal data of the flash memory. If interrupts have their vectors in the variable vector table, they can be used by transferring the vector into the RAM area.

(4) Reset

The reset is always receivable.

(5) The reprogram in user ROM area

When CPU reprogram mode is entered and the block that the flash reprogram control program is located is being reprogramming, the block may not be reprogrammed correctly if the power supply is suddenly down. It is possible that the flash reprogram cannot be executed again in this case. Thus, it is recommended to use standard serial I/O mode and parallel I/O mode.

Software commands

Table BB-1 lists the S/W commands available.

After setting the CPU reprogram mode select bit to "1", the S/W commands can be used to specify the erasing or programming operation. Note that when entering a S/W command, the upper byte (D_{15}–D_8) is ignored.

The content of each S/W command is explained below.

Table BB-1 List of software commands (CPU reprogram mode)

Command	Cycle number	The 1st bus cycle			The 2nd bus cycle		
		Mode	Address	Data (D_{15}–D_0)	Mode	Address	Data (D_{15}–D_0)
Read array	1	Write	X (Note 5)	FF_{16}			
Read status register	2	Write	X	70_{16}	Read	X	SRD(Note 2)
Clear status register	1	Write	X	50_{16}			
Program	2	Write	X	40_{16}	Write	WA(Note 3)	WD(Note 3)
Block erase	2	Write	X	20_{16}	Write	BA(Note 4)	$D0_{16}$

Note 1: When a S/W command is input, the high-order byte of the data(D_{15}–D_8) is ignored.
Note 2: SRD = Status Register Data
Note 3: WA = Write Address, WD = Write Data
Note 4: BA = Block Address (the maximum even address of the block)
Note 5: "X" can be any even address in user ROM area.

Read Array Command (FF_{16})

Issuing the command code "FF_{16}" in the 1st bus cycle enters the read array mode. When an even address is issued in one of the bus cycle that follows, the content of the address is read out at the data bus (D_{15}–D_0), 16 bits at a time.

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 issued in the 1st bus cycle, the content of the status register is read out at the data bus (D_7–D_0) by a read in the 2nd bus cycle.

The status register is explained in the next section.

Clear Status Register Command (50_{16})

The command is used to clear the bits SR4 and SR5 of the status register after they have been set. These bits indicate that operation has ended in error. To use this command, issue the command code "50_{16}" in the 1st bus cycle.

Program Command (40$_{16}$)

Program operation starts when the command code "40$_{16}$" is issued in the 1st bus cycle. If the address and data are issued in the 2nd bus cycle, program operation (data programming and verification) will start.

Whether the program operation is completed can be conformed by reading the status register or the RY/$\overline{BY}$ status flag. When the program starts, the read status register mode is accessed automatically and the content of the status register can be read on the date bus (D_7–D_0). The status register bit 7 (SR7) is set to "0" at the same time when the program operation starts and is returned to "1" upon the completion of the program operation. In this case, the read status register mode remains active until the Read Array Command (FF$_{16}$) is issued.

The RY/$\overline{BY}$ status flag is "0" during program operation and "1" when the program operation is completed same as the status register bit 7.

After the program, reading the status register can check the result. Refer to the section where the status register is detailed.

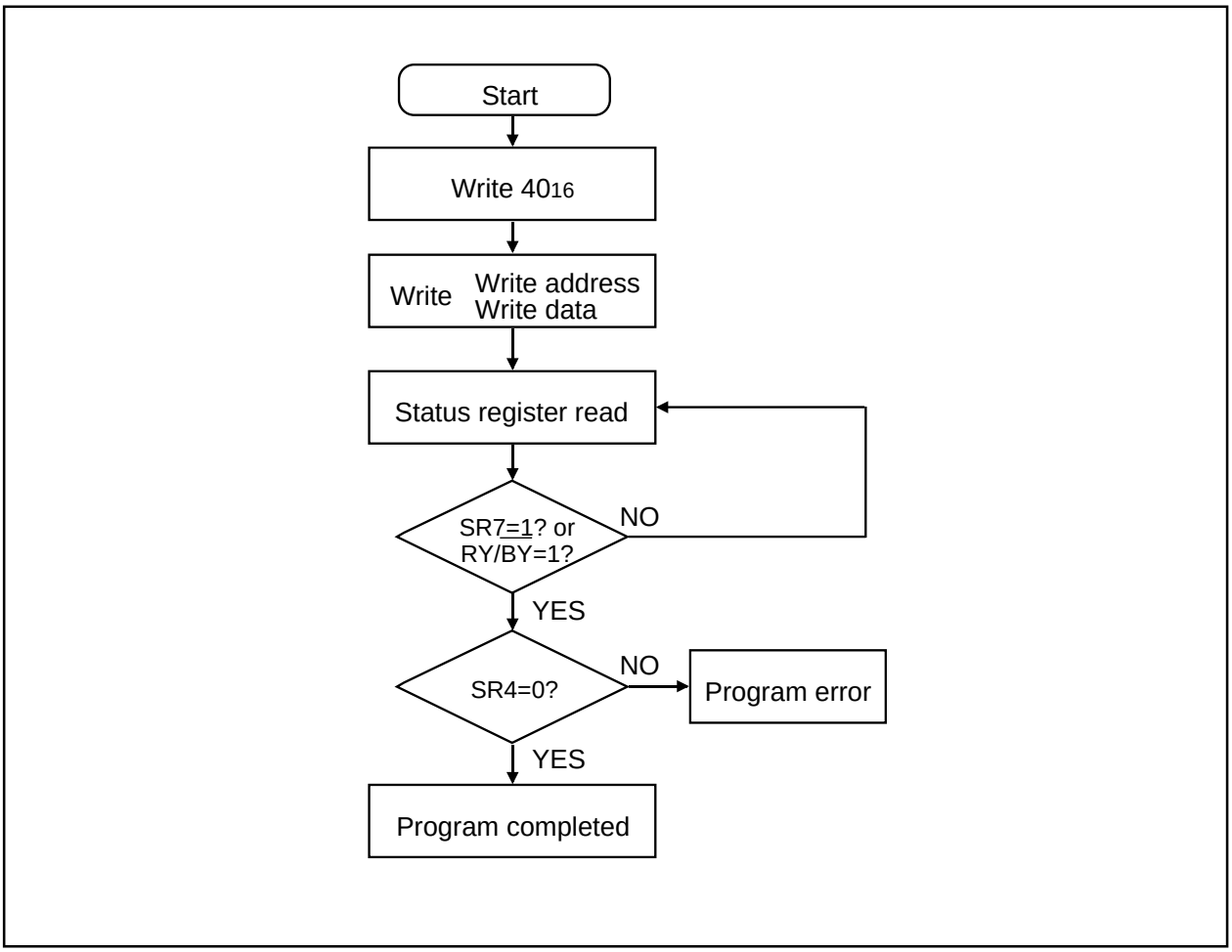


Fig.BB-3 Program flowchart

Block Erase Command (20$_{16}$/D0$_{16}$)

By issuing the command code "20$_{16}$" in the 1st bus cycle and the conformation command code "D0$_{16}$" and block address in the 2nd bus cycle, the erase operation specified by the block address starts (erase and erase verification).

Whether the block erase command is terminated can be conformed by reading the status register or the RY/$\overline{BY}$ status flag. When the block erase operation starts, the read status register mode is accessed automatically and the content of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time when the erase operation starts and is returned to "1" upon the completion of the erase operation. In this case, the read status register mode remains active until the Read Array Command (FF$_{16}$) is written.

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

After the block erase, reading the status register can check the result. Refer to the section where the status register is detailed.

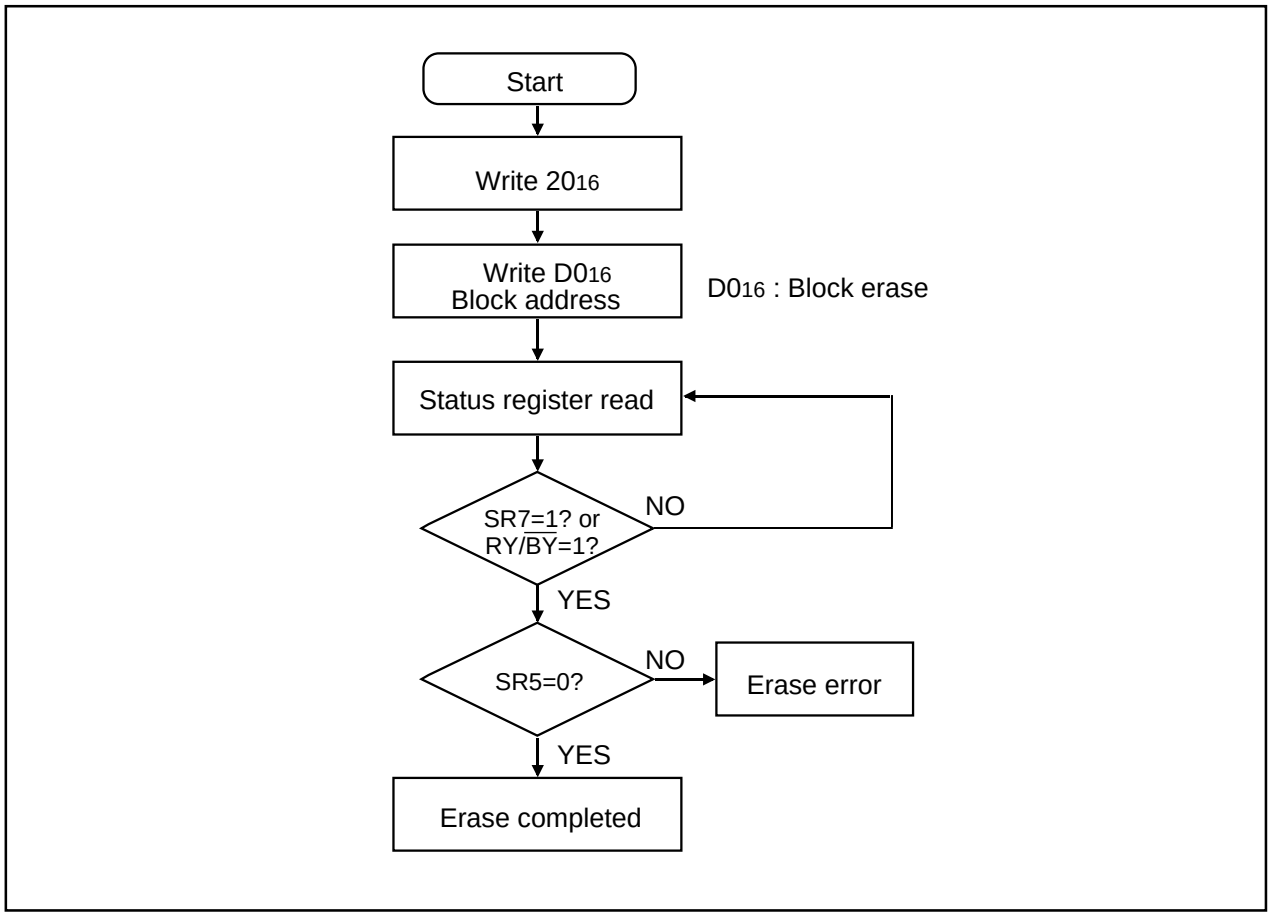


Fig.BB-4 Erase flowchart

Status register

The status register shows the operation status of the flash memory and whether program and erase operations end successfully or not. It can be read in the following conditions.
(1) By reading an arbitrary address from the user ROM area after issuing the read status register command (70_{16})
(2) By reading an arbitrary address from the user ROM area in the period from the start of program or erase operation to the execution of read array command (FF_{16}).
Table BB-2 shows the status register.
The status register can be cleared in the following condition.
(1) By issuing the clear status register command (50_{16}).
(2) After reset, the status register is set to "80_{16}".

Each bit of the register is shows below.

Sequencer status (SR7)

After power-on, the sequencer status is set to "1" (ready).
The bit is set to "0" (busy) during program and erase operations and is set to "1" upon the completion of these operations.

Erase status (SR5)

Erase status indicates the status of erase operation. When erase error occurs, it is set to "1".
The bit becomes "0" when it is cleared.

Program status (SR4)

Program status indicates the status of program operation. When program error occurs, it is set to "1".
The bit becomes "0" when it is cleared.

If "1" is set to SR5 or SR4, the program and block erase operations are not accepted. Before execution of these commands, it is necessary to execute the clear status register command (50_{16}) to clear the status register.
If any S/W commands are not correct, both the SR5 and SR4 are set to "1".

Table BB-2 Definition of each bit of status register

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Sequencer status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Full status check

By performing full status check, the execution result of erase and program operations can be known. Fig.BB-5 shows the full status check flowchart and the method to deal with the error.

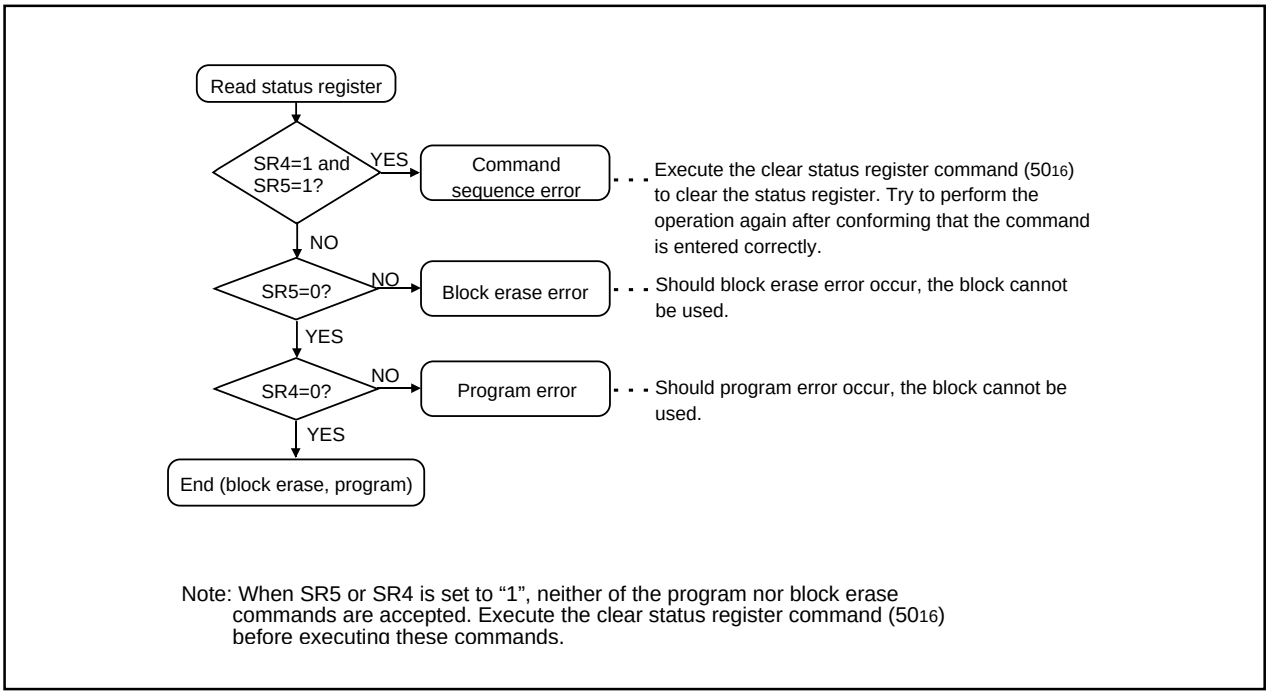


Fig.BB-5 Full status check flowchart and the method to deal with errors

Functions to inhibit rewriting to the on-chip flash memory

To prevent flash memory from being miss-read or miss-written, ROM code protect function for parallel I/O mode and ID code check function for standard serial mode are introduced.

ROM code protect function

ROM code protect function can inhibit readout from or modification to the flash memory by setting the content in ROM code protect control address (0FFFFF16) for parallel I/O mode. Fig.BB-6 shows the content of ROM code protect control address (0FFFFF16). (The address exists in user ROM area.)

If one of the pair of ROM code protect bits is set to “0”, ROM code protect is turned on, so that the flash memory is protected against the readout or modification. ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When both level 1 and level 2 are set, level 2 will be selected.

If both of the two ROM code protect reset bits are set to “00”, ROM code protect is turned off, so that the flash memory can be read out or modified. Once ROM code protect is turned on, the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial mode or other to rewrite these two bits.

ROM code protect control address

b7	b6	b5	b4	b3	b2	b1	b0
						1	1

Symbol	Address	When reset
ROMCP	0FFFFF16	FF16

Bit symbol	Bit name	Function
Reserved bits		Always set these bits to “1”
ROMCP2	ROM code protect level 2 set bits (Note 1,2)	b3 b2 0 0: Protect enabled 0 1: Protect enabled 1 0: Protect enabled 1 1: Protect disabled
ROMCR	ROM code protect reset bits (Note 3)	b5 b4 0 0: Protect removed 0 1: Protect set bits effective 1 0: Protect set bits effective 1 1: Protect set bits effective
ROMCP1	ROM code protect level 1 set bits (Note 1)	b7 b6 0 0: Protect enabled 0 1: Protect enabled 1 0: Protect enabled 1 1: Protect disabled

Note 1: When ROM code protect is turned on, the on-chip flash memory is protected against readout or modification in parallel I/O mode.
Note 2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester,etc. is also inhibited.
Note 3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and level 2. However,Since these bits can not be modified in parallel I/O mode, they should be rewritten in serial I/O mode or other modes.

Fig.BB-6 ROM code protect control address

ID code check function

The function is used in standard serial I/O mode. If the flash memory is not blank, the ID code sent from serial burner is compared with that inside flash memory to check the agreement. It the ID codes do not match, the commands from serial burner are not accepted. Each ID code consists of 8-bit data, the areas of which, beginning from the 1st byte, are 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, 0FFFFB16. Write a program with the ID code at these addresses to the flash memory.

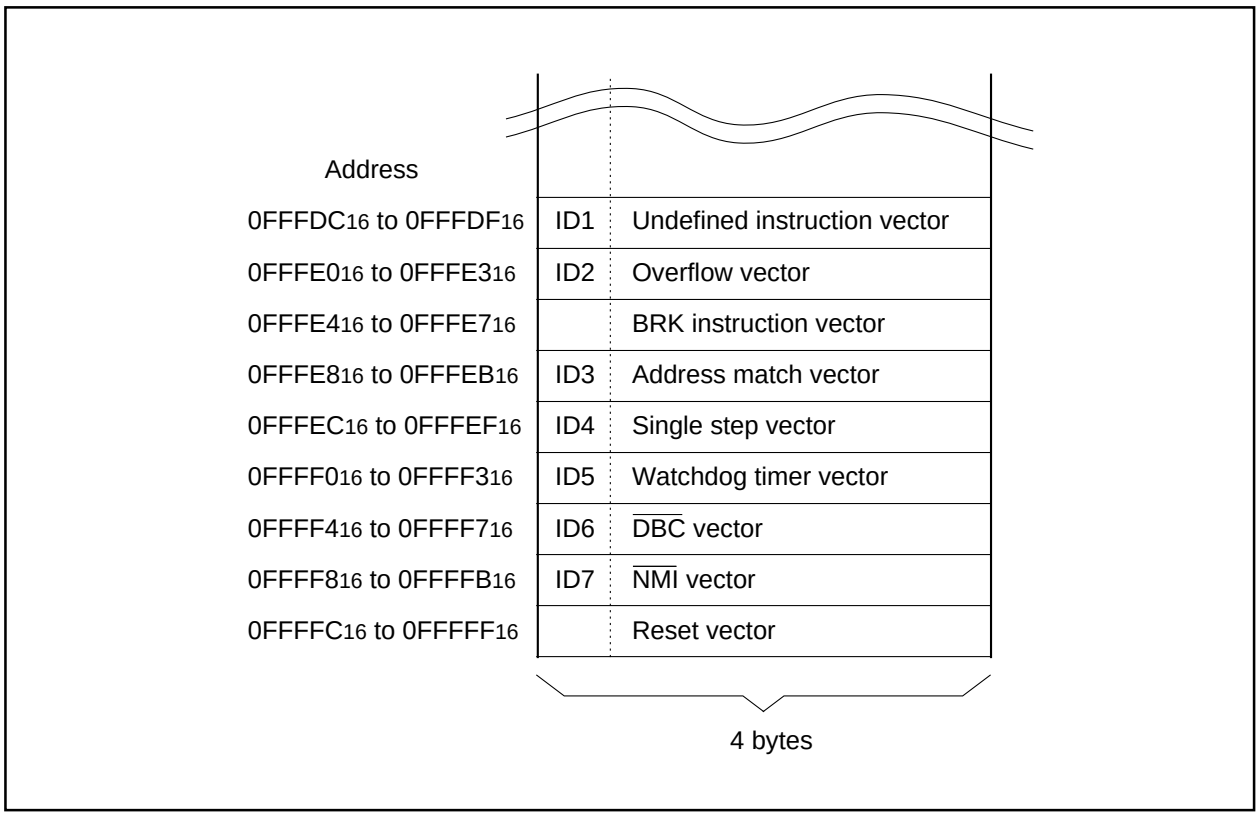


Fig.BB-7 ROM ID code addresses

Parallel I/O mode

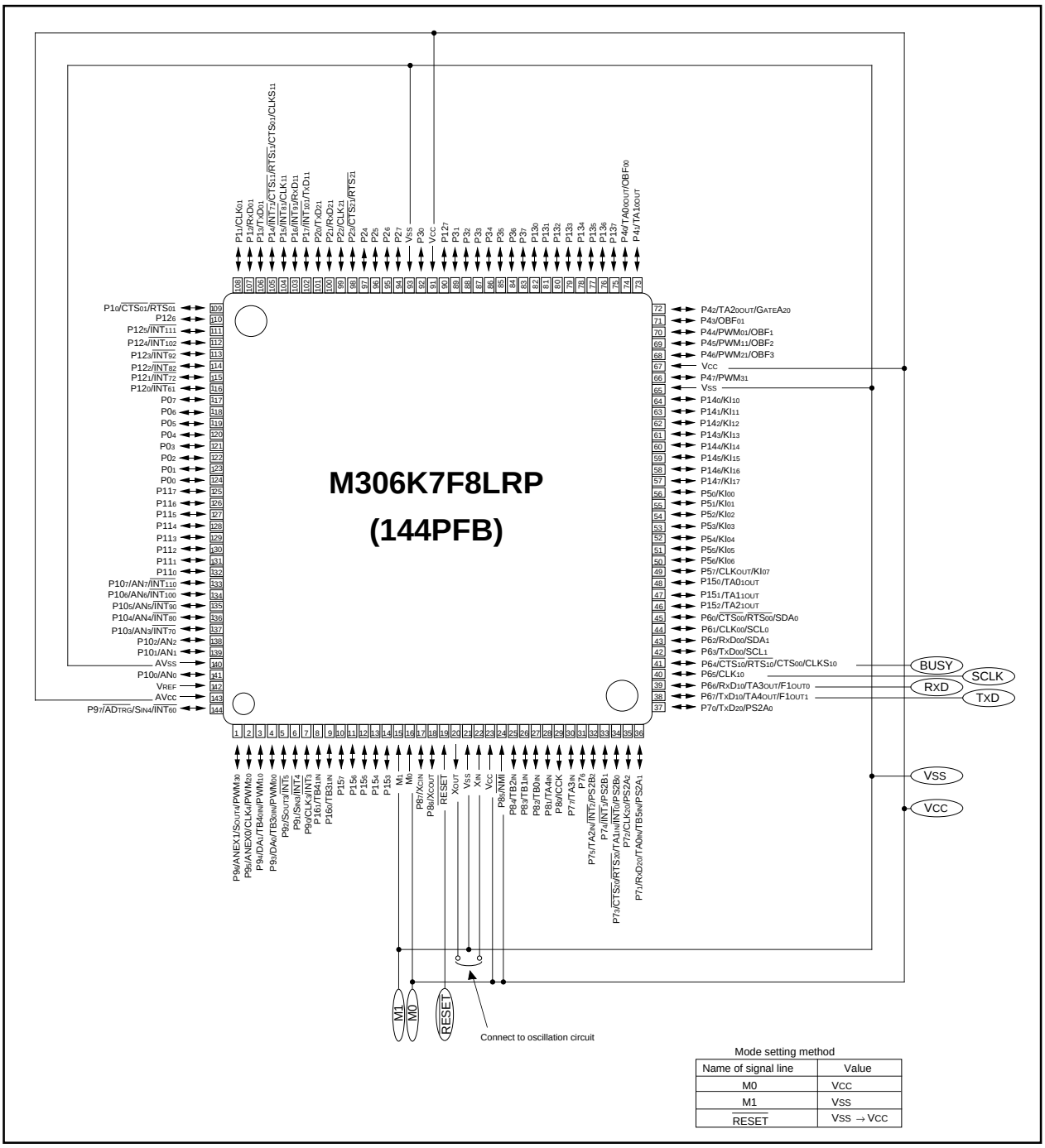
Parallel I/O mode is to input and output the software command, address and data in parallel to access the on-chip flash memory (read, program, erase etc.).
Please use the specific device (programmer) supported for M16C/6K7 Group. Referring to the guideline etc. of each device manufacture for the usage.

User ROM area and boot ROM area

In parallel I/O mode, both user ROM area and boot ROM area showed in Fig.AB-1 can be reprogrammed. The access method to both areas is the same.
The size of boot ROM area is 4K bytes. The addresses are allocated in $0FF000_{16}$– $0FFFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)
In the boot ROM area, erase block operation is applied to only one 4K bytes block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from Mitsubishi factory. Therefore, if the standard serial I/O mode is used, the rewriting to the boot ROM area is not necessary.

Table EE-1 Pin function (Flash memory standard serial I/O mode)

Pin name	Name	I/O	
VCC, VSS	Power supply		Apply 3.3 ± 0.3V to VCC, apply 0V to VSS
M0	M0	I	Connect to VCC
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While reset is "L", 20 cycles or more clocks input to XIN pin are needed.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT. If external clock is used, input it to XIN pin and open the XOUT pin.
XOUT	Clock output	O	
M1	M1	I	Connect to VSS
AVCC, AVSS	Analog power supply		Connect AVSS to VSS, AVCC to VCC
VREF	Reference voltage	I	The input pin of reference voltage of AD converter
$P0_0$–$P0_7$	Input port P0	I	Input "H", "L" or open
$P1_0$–$P1_7$	Input port P1	I	Input "H", "L" or open
$P2_0$–$P2_7$	Input port P2	I	Input "H", "L" or open
$P3_0$–$P3_7$	Input port P3	I	Input "H", "L" or open
$P4_0$–$P4_7$	Input port P4	I	Input "H", "L" or open
$P5_0$–$P5_7$	Input port P5	I	Input "H", "L" or open
$P6_0$–$P6_3$	Input port P6	I	Input "H", "L" or open
$P6_4$	BUSY output	O	The output pin of BUSY signal
$P6_5$	SCLK input	I	The input pin of serial clock
$P6_6$	RXD input	I	The input pin if serial data
$P6_7$	TXD input	O	The output pin of serial data
$P7_0$–$P7_7$	Input port P7	I	Input "H", "L" or open
$P8_0$–$P8_4$ $P8_6$,$P8_7$	Input port P8	I	Input "H", "L" or open
$P8_5$	$\overline{\text{NMI}}$ input	I	Connect to VCC
$P9_0$–$P9_7$	Input port P9	I	The input pin of serial data
$P10_0$–$P10_7$	Input port P10	I	The input pin of serial data
$P11_0$–$P11_7$	Input port P11	I	The input pin of serial data
$P12_0$–$P12_7$	Input port P12	I	The input pin of serial data
$P13_0$–$P13_7$	Input port P13	I	The input pin of serial data
$P14_0$–$P14_7$	Input port P14	I	The input pin of serial data
$P15_0$–$P15_7$	Input port P15	I	The input pin of serial data
$P16_0$,$P16_1$	Input port P16	I	The input pin of serial data



Mode setting method

Name of signal line	Value
M0	VCC
M1	VSS
RESET	VSS → VCC

Fig.EE-1 Pin connections for serial I/O mode

Standard serial I/O mode

The standard serial I/O mode inputs and outputs the S/W commands, addresses and data needed to operate (read, program, erase etc.) the on-chip flash memory with a dedicated serial programmer.

Different from parallel I/O mode, in standard serial mode, CPU controls the flash memory reprogramming (uses the CPU reprogram mode) and the input of serial reprogram data etc. The standard serial I/O mode is started by connecting M_0 to “H”, M_1 to “L” with the release of reset. (To connect M_0 to “L” in normal microcomputer mode.)

This control program is written in boot ROM area when the product is shipped from Mitsubishi factory. Make sure that the standard serial I/O mode cannot be used if boot ROM area is written in parallel I/O mode. Fig EE-1 shows the pin connections for standard serial I/O mode. The input and output of serial data are processed in CLK_{10}, RxD_{10}, TxD_{10}, RTS_{10} (BUSY) 4 pins of the UART1.

The CLK_{10} is clock input pin, which clock is input externally. The TxD_{10} is CMOS output pin. The RTS_{10} (BUSY) pin outputs “L” when ready for reception and outputs “H” when reception starts. The serial data are transferred in 8-bit unit.

In standard serial I/O mode, only the user ROM area shown in Fig.AB-1 can be reprogrammed. Boot ROM area cannot be reprogrammed.

In the standard serial I/O mode, a 7-byte ID code is used. If the flash memory is not blank, commands sent from programmer are not accepted unless the ID code matches.

Outline (standard serial I/O mode)

In standard serial I/O mode, S/W commands, addresses, and data etc. are input and output with the peripheral device (serial programmer) using 4-wire clock-synchronized serial I/O (UART1). In reception, S/W commands, addresses and program data are read from RxD10 pin synchronized with the rising edge of the transfer clock that is input to the CLK10 pin. In transmission, the read data and status are output to TxD10 pin synchronized with the falling edge of the transfer clock.

The TxD10 is CMOS output pin. Transfer is in 8-bit unit with LSB first.

During transmission, reception, erasing and programming, the RTS10 (BUSY) pin is “H”. Accordingly, always start the next transfer after the RTS10 (BUSY) pin becomes “L”.

The read after the input of S/W commands can get memory data and status register. Reading the status register can check the flash memory operation status, the normal/error end of erasing or programming operation.

The following are the explanation of S/W commands, status register etc.

S/W commands

Table EE-2 lists the S/W commands. In standard serial I/O mode, the S/W commands, which transferred from RxD pin, control of erase, program and read etc. The S/W commands in standard serial I/O mode are similar with that in parallel I/O mode. ID check function, download function, version information output function, boot ROM area output function and read check data, 5 commands are added.

Table EE-2 The list of S/W commands (standard serial I/O mode)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte	–	If ID unmatched
1	Page read	FF16	Address (middle)	Address (high)	Data output	Data output	Data output	259th byte data output	Not acceptable
2	Page program	4116	Address (middle)	Address (high)	Data input	Data input	Data input	259th byte data input	Not acceptable
3	Block erase	2016	Address (middle)	Address (high)	D016				Not acceptable
4	Read status register	7016	SRD output	SRD1 output					Acceptable
5	Clear status register	5016							Not acceptable
6	ID check function	F516	Address (low)	Address (middle)	Address (high)	ID size	ID1	–ID7	Acceptable
7	Download function	FA16	Address (low)	Address (high)	Check sum	Data input	No. of times required	–ID7	Not acceptable
8	Version information output function	FB16	Version data output	Version data output	Version data output	Version data output	Version data output	–9th byte Version data output	Acceptable
9	Boot ROM area output function	FC16	Address (middle)	Address (high)	Data output	Data output	Data output	–259th byte data output	Not acceptable
10	Read check data	FD16	Check data (low)	Check data (high)					Not acceptable

Note 1: Shading indicates transfer from flash memory on chip microcomputer to serial programmer. The else indicates transfer from serial programmer to flash memory on chip microcomputer.

Note 2: SRD means status register data. SRD1 means status register 1 data.

Note 3: All commands are acceptable if the flash memory is blank.

The following are the descriptions of S/W commands

Page read command

The command reads the specified page (256 bytes) of the flash memory sequentially one byte at a time. Execute the page read command as following:

(1) Transfer the "FF16" command code in the 1st byte.
(2) Transfer addresses A8– A15 and A16– A23 in the 2nd and 3rd byte respectively.
(3) From the 4th byte onward, data (D7–D0) of the page specified by the address (A23–A8) will be output sequentially from the smallest address sync with the falling edge of the clock.

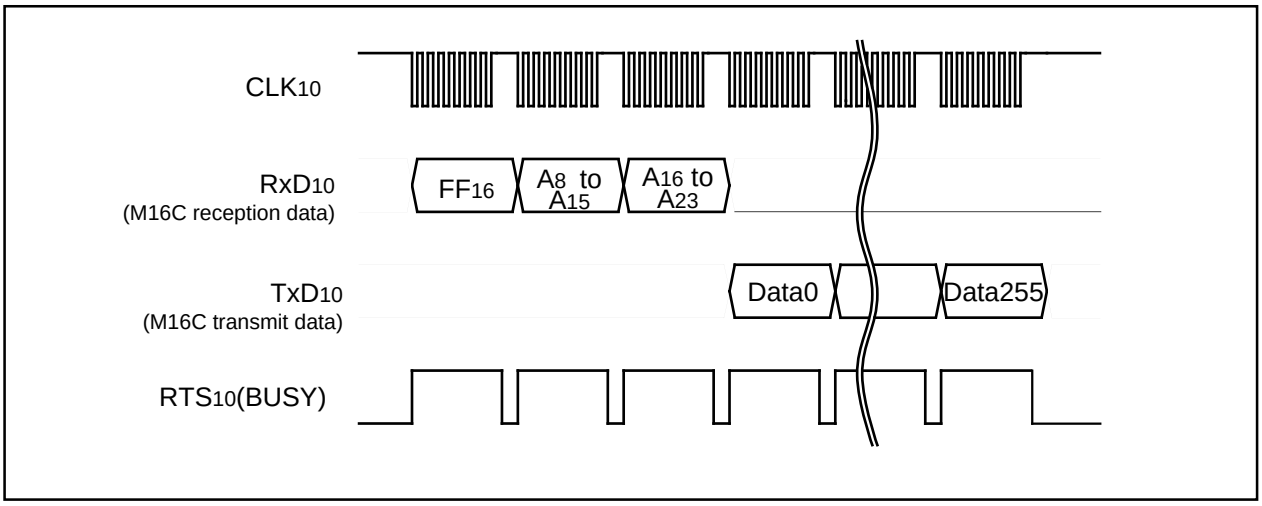


Fig.EE-2 Timing of page read

Read status register command

The command is for reading status information. When command code "7016" is sent in the 1st byte, the contents of status register (SRD) and status register 1 (SRD1) will be output in the 2nd and 3rd byte respectively sync with the falling edge of the clock.

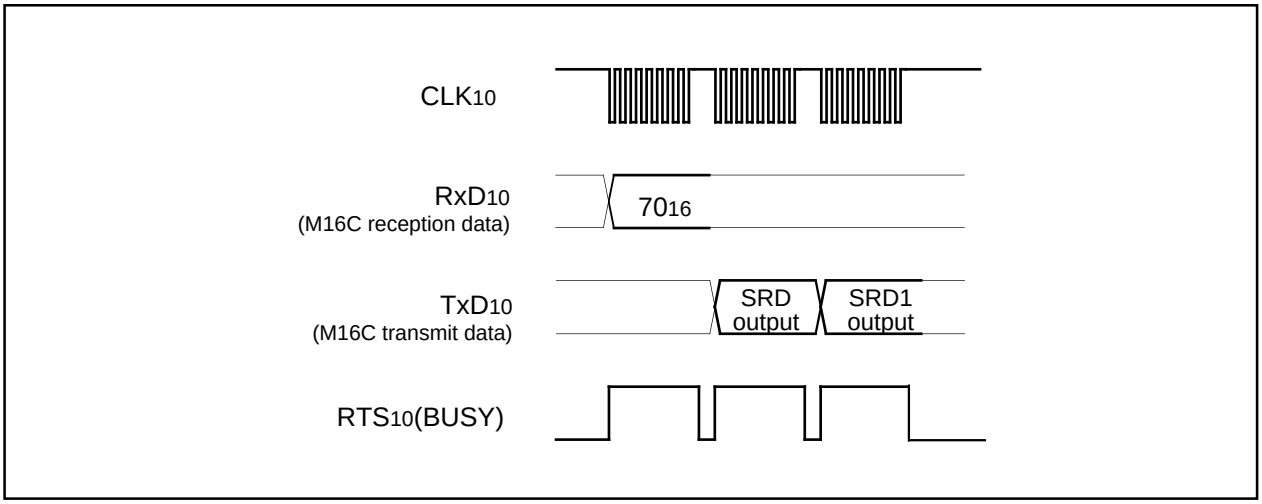


Fig.EE-3 Timing of read status register

Clear status register command

The command clears the bits (SR4–SR5), which are set when operation ended in error. When command code "50_{16}" is sent in the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS_{10} (BUSY) signal changes from "H" to "L".

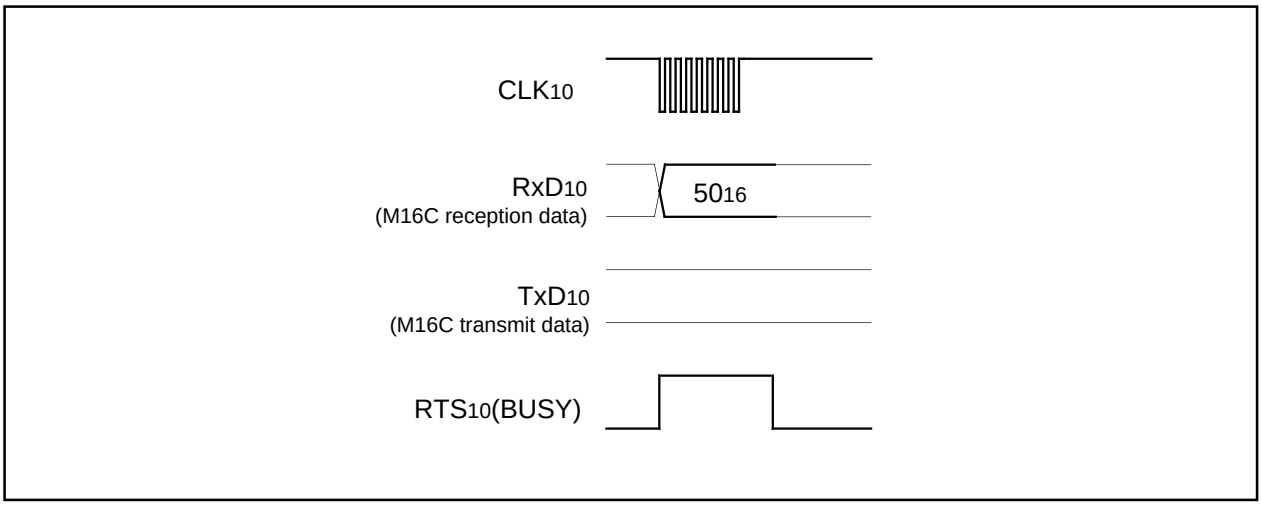


Fig.EE-4 Timing of clear status register

Page program command

The command programs the specified page (256 bytes) of flash memory sequentially one byte a time. Execute the command as follows:

(1) Transfer the command code "41_{16}" in the 1st byte.
(2) Transfer addresses A_{15}–A_8 and A_{23}–A_{16} in the 2nd and 3rd bytes respectively.
(3) From the 4th byte onward, after inputting 256 bytes program data (A_7–A_0) from the smallest address of the specified page, the page program operation will be executed automatically.
When the reception for the next 256 bytes is setup, the RTS_{10} (BUSY) signal changes from "H" to "L".
The result of the page program can be known by reading the status register. For more detail, see the section on the status register.

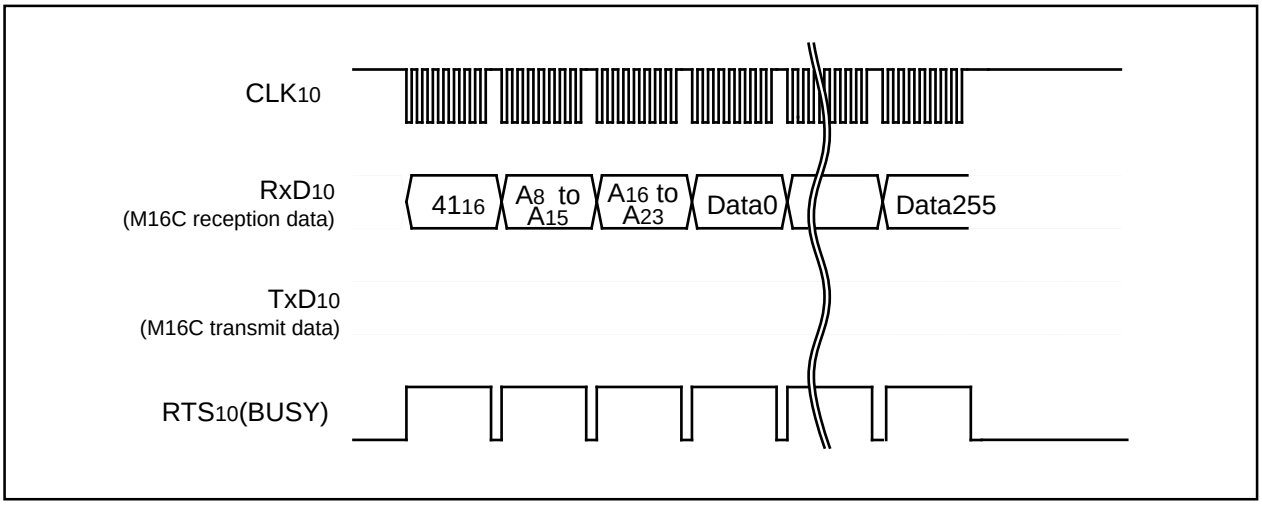


Fig.EE-5 Timing of page program

Block erase command

The command erases the data in the specified block. Execute the command as follows:

(1) Transfer the command code "20_{16}" in the 1st byte.

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

(3) After transferring the verify command code"$D0_{16}$" in the 4th byte, the erase operation starts for the specified block of the flash memory. Issue the biggest address of th2e specified block to A_{23}–A_8.
After the completion of block erase, the RTS_{10} (BUSY) signal changes from "H" to "L". The result of the block erase can be know by reading the status register. For more detail, see the section on the status register.

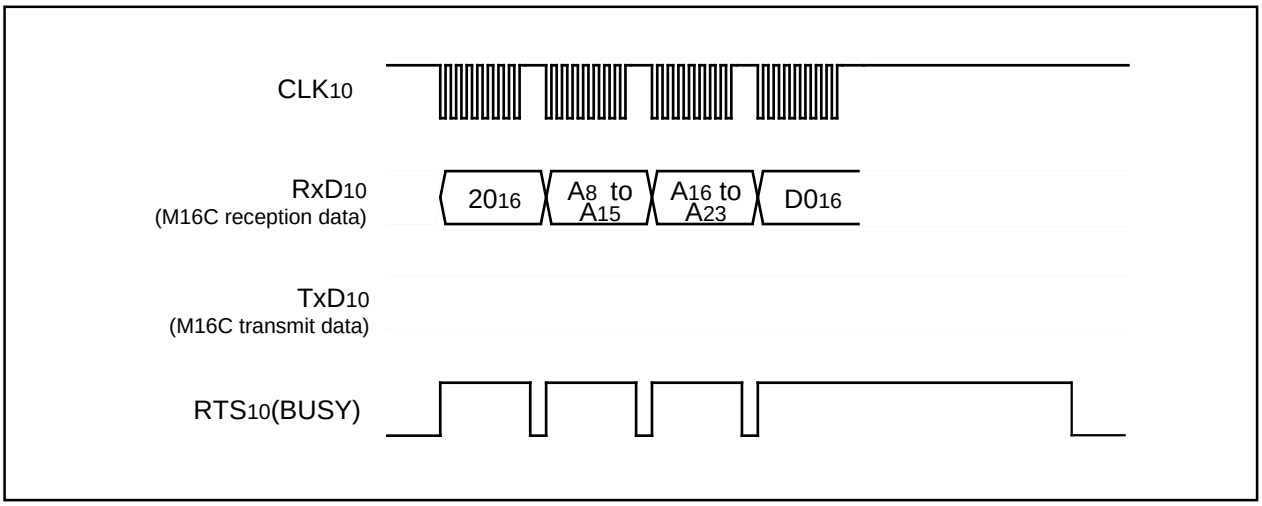


Fig.EE-6 Timing of block erase

Download function

The command downloads an execution program to RAM. Execute the command as follows:

(1) Transfer the command code “FA16” in the 1st byte.

(2) Transfer the program size in the 2nd and 3rd bytes.

(3) Transfer the checksum in the 4th byte. Check sum is calculated from all transferred data from the 5th byte onward.

(4) The execution program is transferred from 5th byte onward.

After the entire program data have been transferred, the downloaded execution program will be executed if the checksum matches.

The program size allowed to transfer varies according to the size of on-chip RAM.

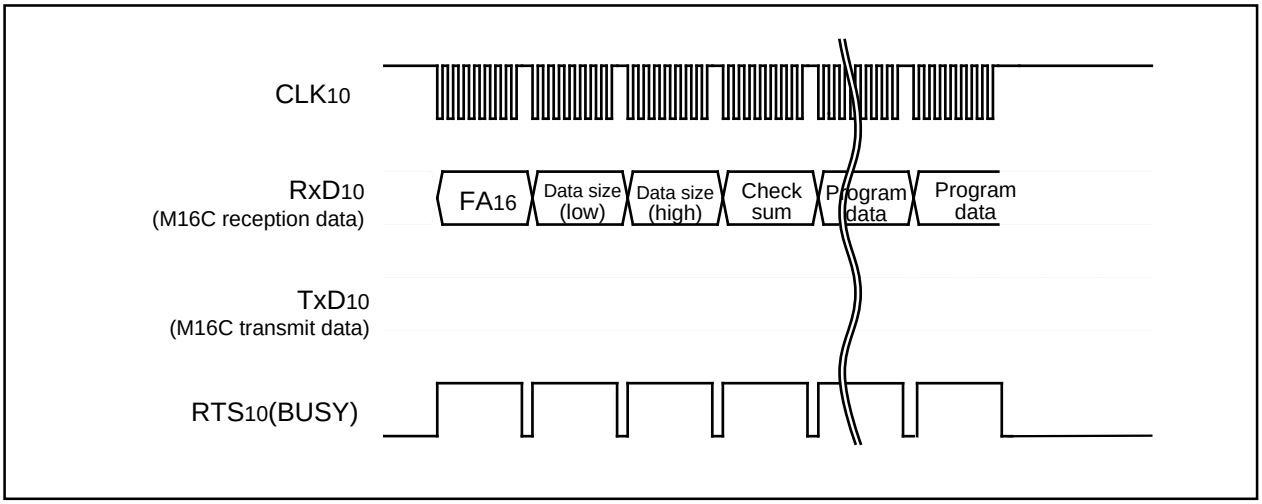


Fig.EE-7 Timing of download function

Version information output function

The version information of the control program stored in boot ROM area can be output by the function. Execute the command as follows:

(1) Transfer the command code "FB_{16}" in the 1st byte.

(2) From the 2nd byte onward, the version information will be output. The information is composed of 8 ASCII character code.

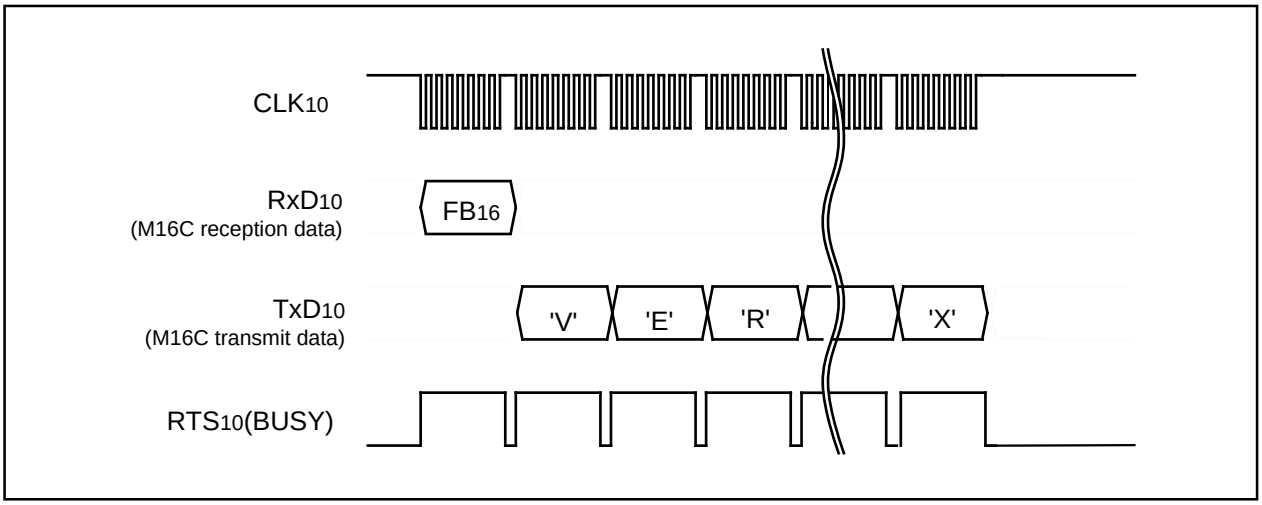


Fig.EE-8 Timing of version information output function

Boot ROM area output function

The control program stored in boot ROM area can be read out in page (256 bytes) unit by the function. Execute the command as follows:

(1) Transfer the command code "FC_{16}" in the 1st byte.

(2) Transfer addresses A_{15}–A_8 and A_{23}–A_{16} in the 2nd and 3rd bytes respectively.
From the 4th byte onward, the data (D_7–D_0) specified in page (256 bytes) address A_{23}–A_8 will be output sequentially from the smallest address in sync with the rising edge of the clock.

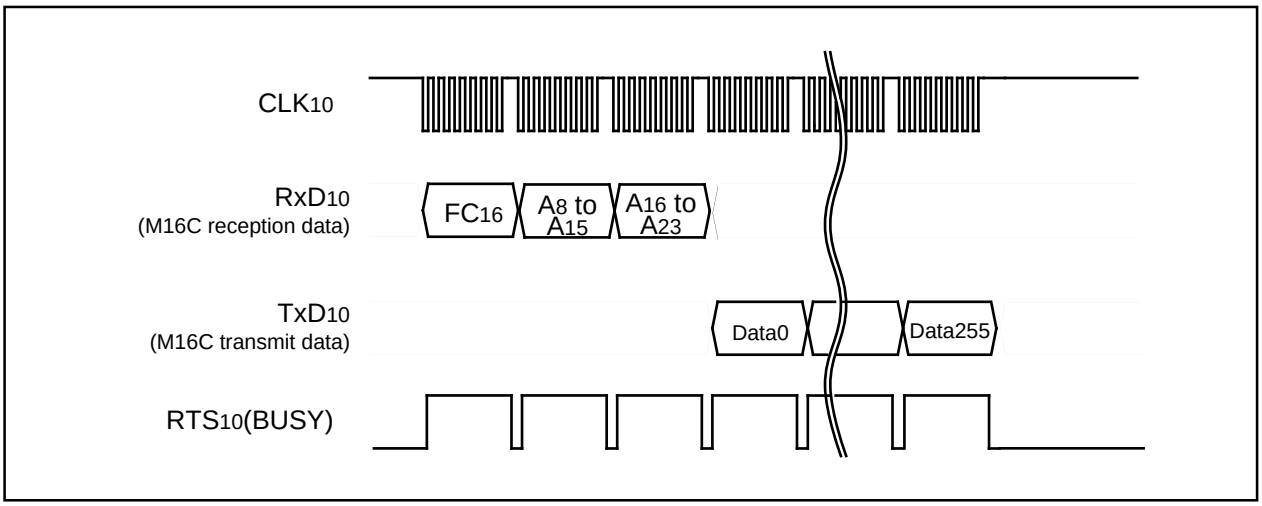


Fig.EE-9 Timing og boot ROM area output function

ID check function

The command checks the ID code. Execute the command as follows:

(1) Transfer the command code "F5$_{16}$" in the 1st byte.

(2) Transfer addresses A$_7$–A$_0$, A$_{15}$–A$_8$ and A$_{23}$–A$_{16}$ of 1st ID code (ID1) in the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of the ID code in the 5th byte.

(4) From the 6th byte onward, transfer the IDs from the 1st ID code (ID1).

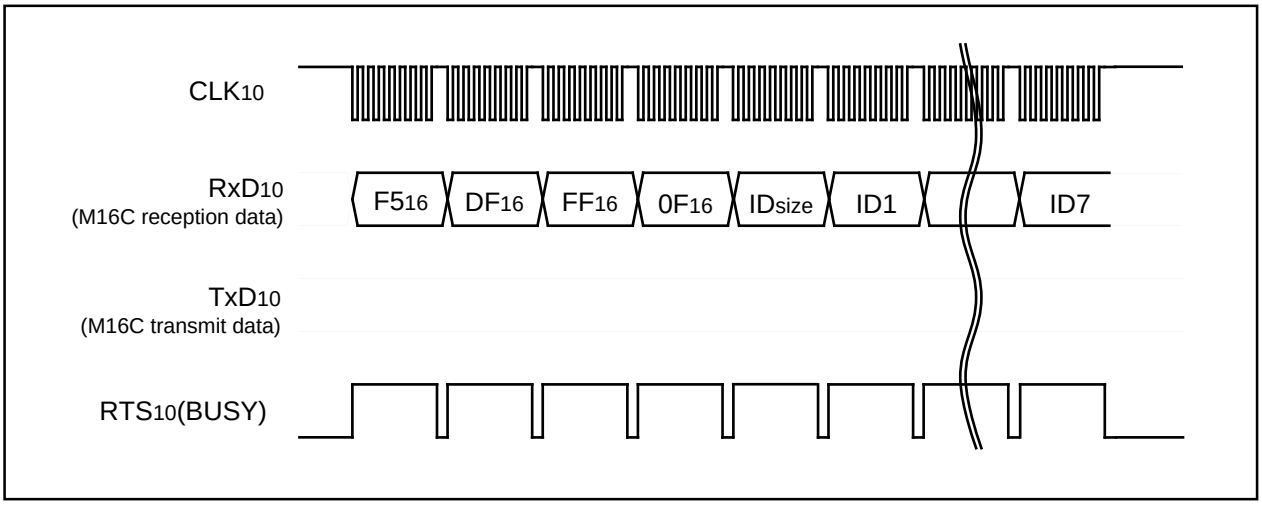


Fig.EE-10 Timing of ID check function

ID code

If the flash memory is not blank, the input ID codes are compared with that written in flash memory. If they do not match, the input commands will not be accepted. Each ID code contains 8 bits data. Beginning from the 1st ID byte, the address of each ID code is 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, 0FFFF7$_{16}$ and 0FFFFB$_{16}$ respectively. Write the program with the ID codes in these addresses to the flash memory.

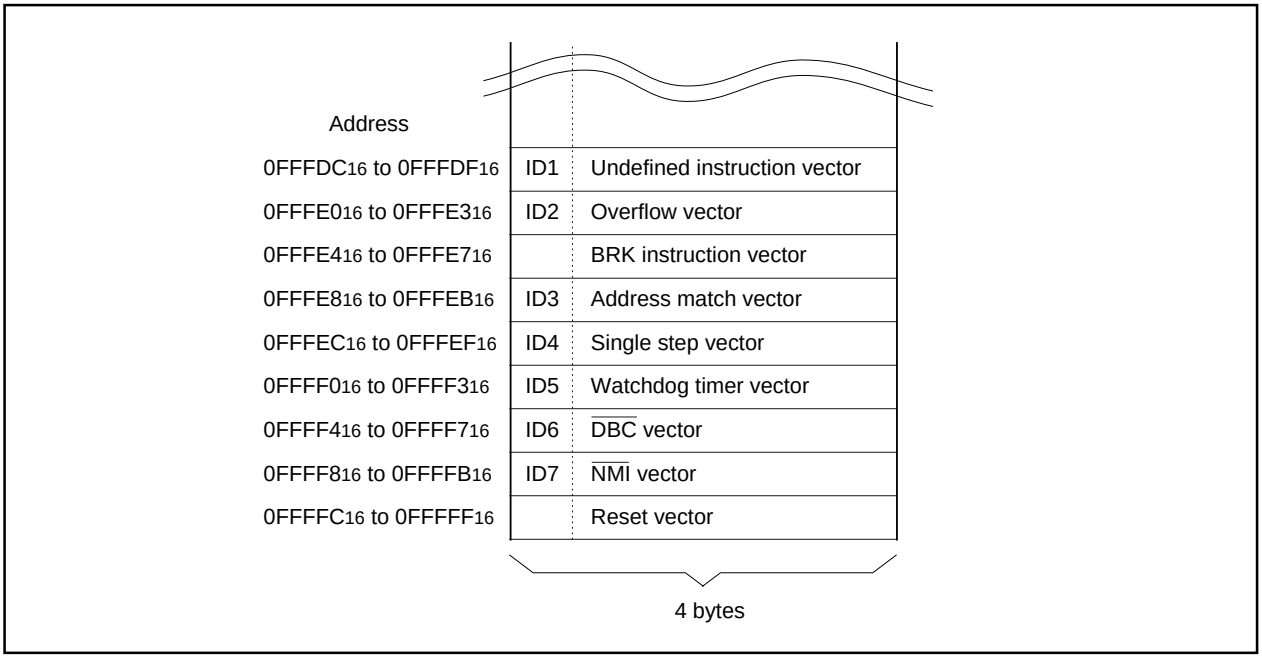


Fig.EE-11 ID code addressed

Read check data

Read check date command is for conforming if the reprogram data sent after page program command have been received correctly.

(1) Transfer the command code "FD_{16}" in the 1st byte.

(2) Transfer check data (low) and check data (high) in the 2nd and 3rd bytes respectively.

When using read check data command, the command should be issued at first to initialize the check data. The next is to issue the page program command and related reprogram data. After that, by issuing the read check data command again, the check data for the reprogram data issued between the two read check data command can be read out.

Adding the reprogram data in byte unit and then calculating the lower 2 bytes of the added data in two's complement gives out the check data.

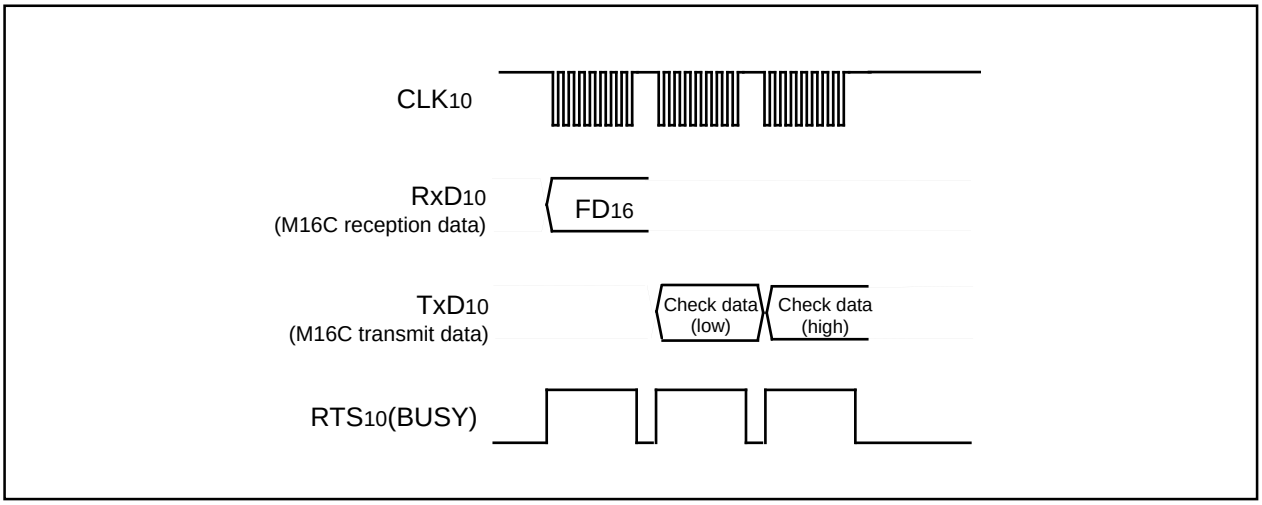


Fig.EE-12 Timing of read check dt command

Status register

Status register indicates if the operation to the flash memory ends successfully or in error. It can be read by issuing the read status register command (70_{16}). The status register can be cleared by issuing the clear status register command (50_{16}).

Table EE-3 shows the definition of each bit of the register.

After reset, status register outputs "80_{16}".

Table EE-3 Status register (SRD)

Symbol	Status	Definition	
		"1"	"0"
SR7 (D_7)	Sequencer status	Ready	Busy
SR6 (D_6)	Reserved	-	-
SR5 (D_5)	Erase status	Terminated in error	Terminated normally
SR4 (D_4)	Program status	Terminated in error	Terminated normally
SR3 (D_3)	Reserved	-	-
SR2 (D_2)	Reserved	-	-
SR1 (D_1)	Reserved	-	-
SR0 (D_0)	Reserved	-	-

Sequencer status (SR7)

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

The bit is set to "0" (busy) during program and erase operations and is set to "1" upon the completion of these operations.

Erase status (SR5)

Erase status indicates the status of erase operation. When erase error occurs, it is set to "1".

The bit becomes "0" when it is cleared.

Program status (SR4)

Program status indicates the status of program operation. When program error occurs, it is set to "1".

The bit becomes "0" when it is cleared.

Status register 1 (SRD1)

Status register 1 indicates the status of serial communication, the result of ID codes comparison, the result of checksum comparison etc. It can be read after SDR by issuing the read status register command (70_{16}). The register can be cleared by issuing the clear status register command (50_{16}).

Table EE-4 shows the definition of each bit of the register.

After power on, status register 1 outputs "00_{16}".

Table EE-4 Status register (SRD1)

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Check sum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 Not verified 01 Verified with mismatch 10 Reserved 11 Verified with match	
SR9 (bit1)	Timeout of data reception	Timeout	Normal operation
SR8 (bit0)	Reserved	-	-

Boot update completed bit (SR15)

The flag indicates that if the control program has been downloaded to RAM with download function.

Check sum match bit (SR12)

The flag indicates if the check sum is matched when downloading the control program with download function.

ID check completed bits (SR11, SR10)

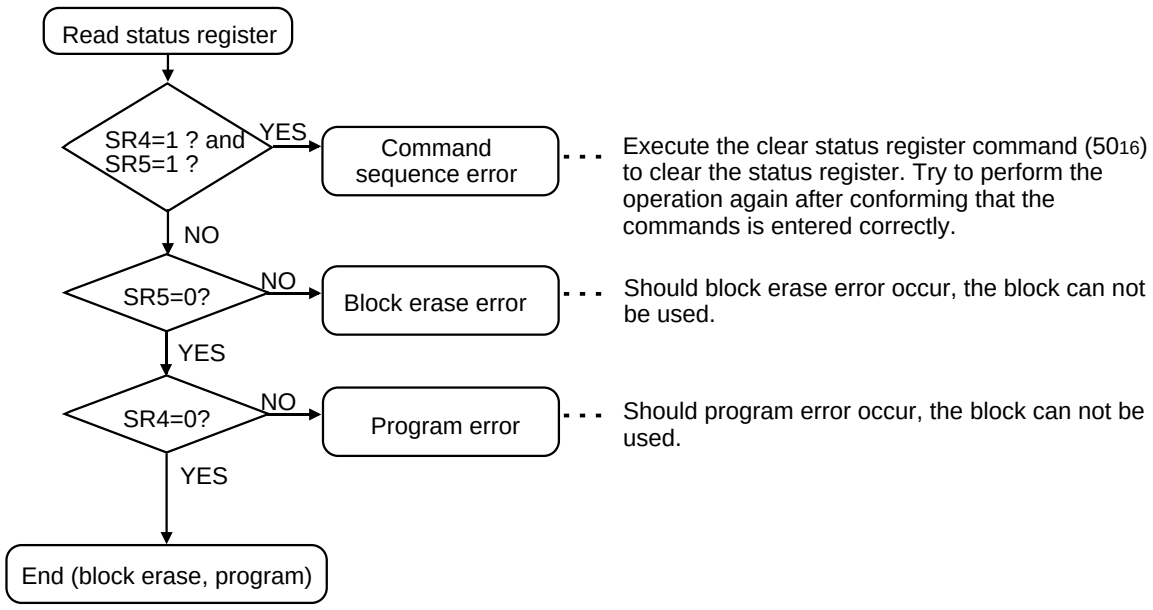
These bits indicate the result of ID checks. Some commands cannot be accepted without the ID checks.

Timeout of data reception bit (SR9)

The flag indicates if timeout occurs during data reception. If the bit is set to "1" during data reception, microcomputer will discard the received data and return to wait state.

Full status check

By performing full status check, the execution result of erase and program operations can be known. Fig.EE-13 shows the full status check flowchart and the method to deal with the error.



Note: When SR5 or SR4 is set to "1", neither of the program nor block erase commands are accepted. Execute the clear status register command (50_{16}) before executing these commands.

Fig.EE-13 Full status check flowchart and the method to deal with errors

Circuit applied for standard serial I/O mode (example)

The figure below shows a circuit applied for standard serial I/O mode. The control pins bary by different programmer. Refer to programmer manual for the detail.

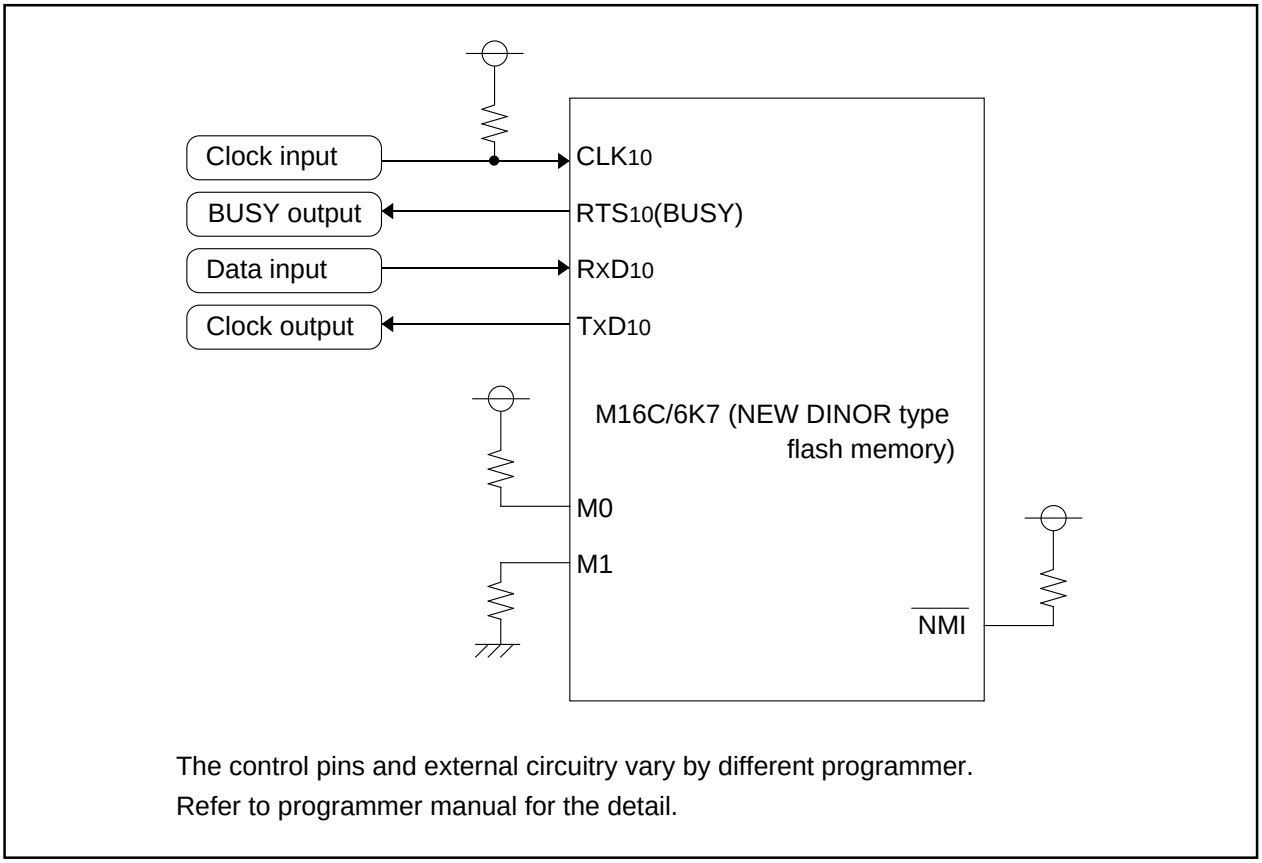


Fig.EE-14 Example circuit applied for the standard serial I/O mode

REVISION HISTORY	M16C / 6K7 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.0	’01.10.23		Misprints, omissions and text styles are revised.
		149	Table GF-1 is added.
		168	Explanation of “•OBF0 mergence function” is added.
			Explanation of Figure SI-5 is revised.
		169	Figure SI-6 is added.
		170	Explanation of “●Serial interrupt control register 2 SERCON2” is added.
			Figure SI-7 is added.

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

RENESAS
Renesas Technology Corp.