MITSUBISHI MICROCOMPUTERS

M37902FCCHP, M37902FGCHP, M37902FJCHP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER

DESCRIPTION

These are single-chip microcomputers designed with high-performance CMOS silicon gate technology, including the internal flash memory. These microcomputers support the 7900 Series instruction set, which are enhanced and expanded instruction set and are upper-compatible with the 7700/7751 Series instruction set.

The CPU of these microcomputers is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. Also, the bus interface unit of these microcomputers enhances the memory access efficiency to execute instructions fast. Therefore, these microcomputers are suitable for office, business, and industrial equipment controller that require high-speed processing of large data.

For the internal flash memory, single-power-supply programming and erasure, using a PROM programmer or the control by the central processing unit (CPU), is supported. Also, each of these microcomputers has the memory area dedicated for storing a certain software which controls programming and erasure (reprogramming control software). Therefore, on these microcomputers, the program can easily be changed even after they are mounted on the board.

DISTINCTIVE FEATURES

<Microcomputer mode>

- Number of basic machine instructions 203
- Memory

  [M37902FCCHP]

  Flash memory (User ROM area) .... 120 Kbytes

  RAM .... 4096 bytes

  [M37902FGCHP]

  Flash memory (User ROM area) .... 248 Kbytes

  RAM .... 6144 bytes

  [M37902FJCHP]

  Flash memory (User ROM area) .... 498 Kbytes

  RAM .... 12288 bytes

  [All of the above computers]

  Flash memory (Boot ROM area) .... 16 Kbytes
- Instruction execution time

  The fastest instruction at 26 MHz frequency .... 38 ns
- Single power supply .... 5 V ± 0.5 V
- Interrupts .... 6 external sources, 16 internal sources, 7 levels
- Multi-functional 16-bit timer .... 5 + 3
- Serial I/O (UART or Clock synchronous) 2
- 10-bit A-D converter .... 8-channel inputs
- 8-bit D-A converter .... 3-channel outputs
- Real-time output

  .... 4 bits × 2 channels, or 6 bits × 1 channel + 2 bits × 1 channel
- 12-bit watchdog timer
- Programmable input/output (ports P0–P8, P10, P11) 84

<Flash memory mode>

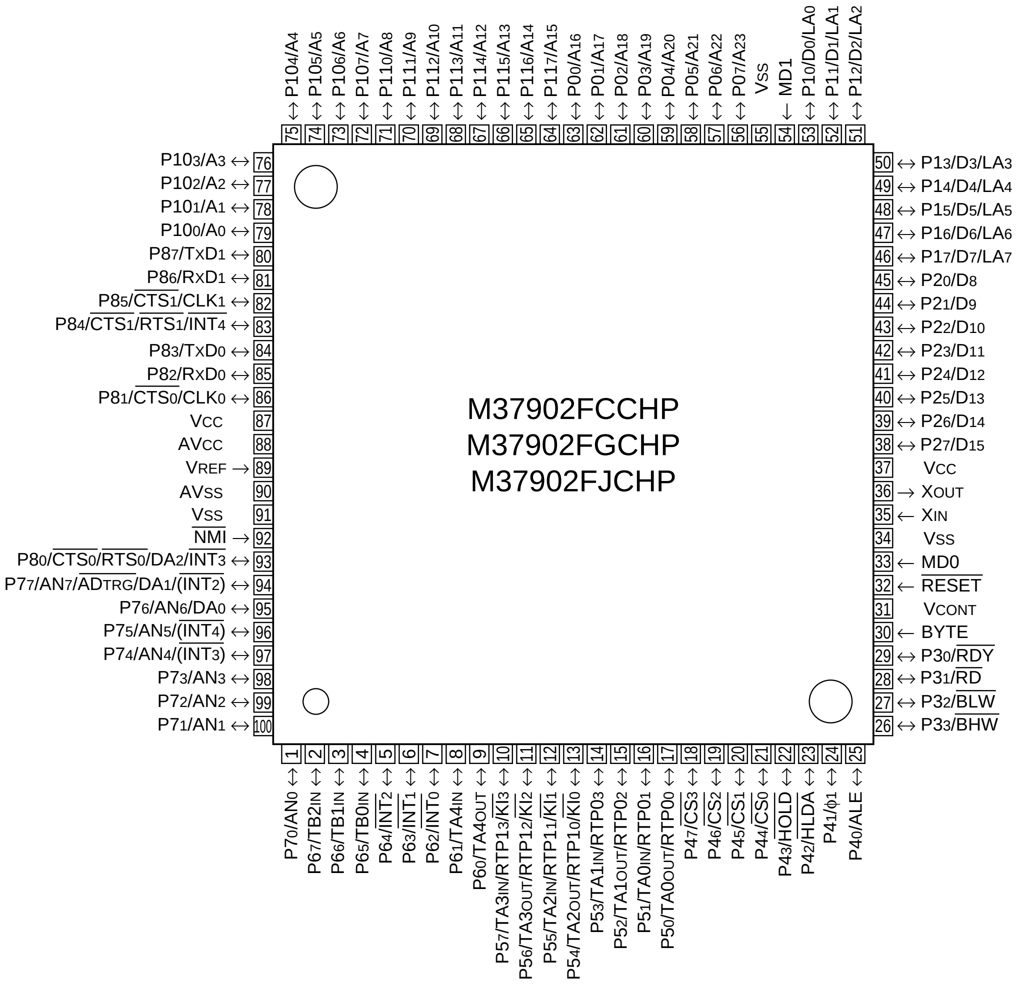
- Power supply voltage .... 5 V ± 0.5 V
- Programming/Erase voltage .... 5 V ± 0.5 V
- Programming method .... Programming in a unit of 256 bytes
- Erase method .... Block erase or Total erase

  (Data protection per block is enabled.)
- Programming/Erase control by software command
- Maximum number of reprograms 100

APPLICATION

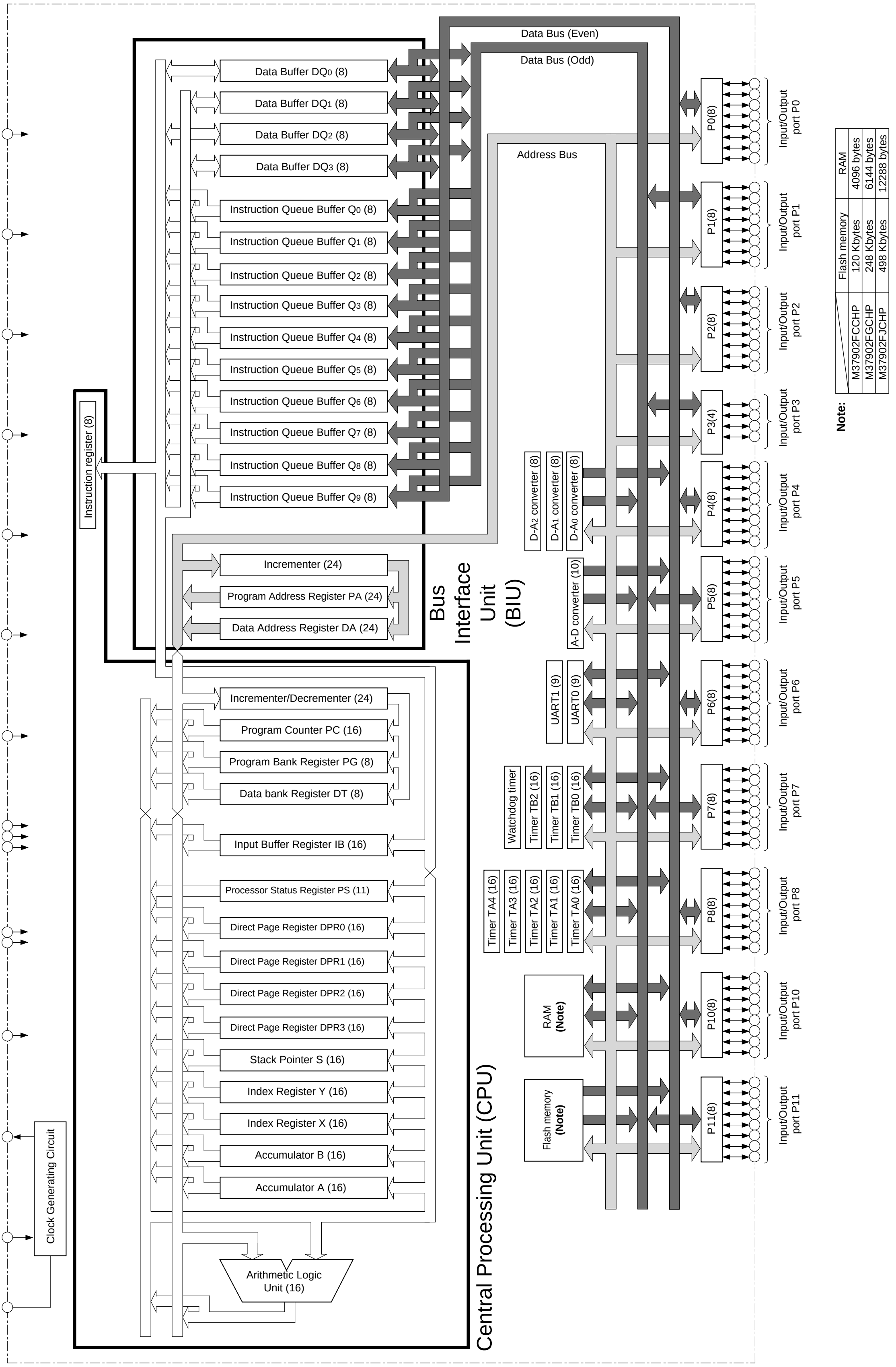
Control devices for personal computer peripheral equipment such as CD-ROM drives, DVD-ROM drives, hard disk drives, high density FDD, printers

M37902FxCHP PIN CONFIGURATION (TOP VIEW)



Outline 100P6Q-A

BLOCK DIAGRAM



Note:

	Flash memory	RAM
M37902FCCHP	120 Kbytes	4096 bytes
M37902FGCHP	248 Kbytes	6144 bytes
M37902FJCHP	498 Kbytes	12288 bytes

FUNCTIONS (Microcomputer mode)

Parameter		Functions
Number of basic machine instructions		203
Instruction execution time		38 ns (the fastest instruction at $f(f_{sys})$ = 26 MHz)
External clock input frequency $f(X_{IN})$		26 MHz (Max.)
System clock frequency $f(f_{sys})$		26 MHz (Max.)
Memory size	Flash memory (User ROM area)	**(Note)**
	RAM	**(Note)**
	Flash memory (Boot ROM area)	16 Kbytes
Programmable input/output ports	P0–P2, P4–P8, P10, P11	8-bit × 10
	P3	4-bit × 1
Multi-functional timers	TA0–TA4	16-bit × 5
	TB0–TB2	16-bit × 3
Serial I/O	UART0 and UART1	(UART or Clock synchronous serial I/O) × 2
A-D converter		10-bit successive approximation method × 1 (8 channels)
D-A converter		8-bit × 3
Watchdog timer		12-bit × 1
Chip-select wait control		Chip select area × 4 ($\overline{CS_0}$–$\overline{CS_3}$). A bus cycle type and bus width can be set for each chip select area.
Real-time output		4 bits × 2 channels; or 6 bits × 1 channel + 2 bits × 1 channel
Interrupts	Maskable interrups	5 external types, 13 internal types. Each interrupt can be set to a priority level within the range of 0–7 by software.
	Non-maskable interrups	1 external type, 3 internal types.
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz crystal resonator).
PLL frequency multiplier		The following multiplication methods are available: double, triple, and quadruple.
Power supply voltage		5 V±0.5 V
Power dissipation		150 mW (at $f(f_{sys})$ = 26 MHz, Typ., PLL frequency multiplier stopped)
Ports' input/output characteristics	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Up to 16 Mbytes. Note that bank FF_{16} is a reserved area.
Operating ambient temperature range		–20 to 85 °C
Device structure		CMOS high-performance silicon gate process
Package		100-pin plastic molded QFP

Note:

Flash memory (User ROM area)	M37902FCCHP	120 Kbytes
	M37902FGCHP	248 Kbytes
	M37902FJCHP	498 Kbytes
RAM	M37902FCCHP	4096 bytes
	M37902FGCHP	6144 bytes
	M37902FJCHP	12288 bytes

FUNCTIONS (Flash memory mode)

Parameter		Functions
Power supply voltage		5 V±0.5 V (in the flash memory parallel I/O mode, 3.3 V±0.3 V)
Programming/Erase voltage		5 V±0.5 V (in the flash memory parallel I/O mode, 3.3 V±0.3 V)
Flash memory mode		3 modes: parallel I/O, serial I/O, and CPU reprogramming modes
Block division for erasure	User ROM area	**(Note 1)**
	Boot ROM area	1 block (16 Kbytes × 1) **(Note 2)**
Programming method		Programmed per page (in a unit of 256 Kbytes)
	Flash memory parallel I/O mode	User ROM area + Boot ROM area
	Flash memory serial I/O mode	User ROM area
	Flash memory CPU reprogramming mode	User ROM area
Erase method		Total erase/Block erase
	Flash memory parallel I/O mode	User ROM area + Boot ROM area
	Flash memory serial I/O mode	User ROM area
	Flash memory CPU reprogramming mode	User ROM area
Programming/Erase control		Programming/Erase control by software commands
Data protection method		Protected per block, by using a lock bit.
Number of commands		8 commands
Maximum number of reprograms		100

Notes 1:

User ROM area	M37902FCCHP	5 blocks (8 Kbytes × 3, 32 Kbytes × 1, 64 Kbytes × 1), total 120 Kbytes
	M37902FGCHP	7 blocks (8 Kbytes × 3, 32 Kbytes × 1, 64 Kbytes × 3), total 248 Kbytes
	M37902FJCHP	11 blocks (2 Kbytes × 1, 8 Kbytes × 2, 32 Kbytes × 1, 64 Kbytes × 7), total 498 Kbytes

2: On shipment, our reprogramming control firmware for the flash memory serial I/O mode has been stored into the boot ROM area. Note that the boot ROM area can be erased/programmed only in the flash memory parallel I/O mode.

PIN DESCRIPTION (MICROCOMPUTER MODE)

Pin	Name	Input/ Output	Functions
Vcc, Vss	Power supply input	—	Apply 5 V±0.5 V to Vcc, and 0 V to Vss.
MD0	MD0	Input	This pin controls the processor mode. Connect this pin to V$_{SS}$ for the single-chip mode or memory expansion mode, and V$_{CC}$ for the microprocessor mode.
MD1	MD1	Input	Connect this pin to Vss.
$\overline{\text{RESET}}$	Reset input	Input	The microcomputer is reset when “L” level is applied to this pin.
X$_{IN}$	Clock input	Input	These are input and output pins of the internal clock generating circuit. Connect a ceramic or quartz- crystal resonator between the X$_{IN}$ and X$_{OUT}$ pins. When an external clock is used, the clock source should be connected to the X$_{IN}$ pin, and the X$_{OUT}$ pin should be left open.
X$_{OUT}$	Clock output	Output	
BYTE	External data bus width select input	Input	This pin determines whether the external data bus has an 8-bit width or 16-bit width for the memory expansion mode or microprocessor mode. The width is 16 bits when “L” signal is input, and 8 bits when “H” signal is input. When BYTE = Vss level, by the register setting, the external data bus for each of areas $\overline{\text{CS}_1}$ to $\overline{\text{CS}_3}$ can have a width of 8 bits.
V$_{CONT}$	Filter circuit connection	—	When using the PLL frequency multiplier, connect this pin to the filter circuit. When not using, this pin should be left open.
AVcc, AVss	Analog power supply input	—	Power supply input pins for the A-D converter and the D-A converter. Connect AVcc to Vcc, and AVss to Vss externally.
V$_{REF}$	Reference voltage input	Input	This is the reference voltage input pin for the A-D converter and the D-A converter.
P0$_0$–P0$_7$	I/O port P0	I/O	■ In single-chip mode Port P0 is an 8-bit I/O port. This port has an I/O direction register, and each pin can be programmed for input or output. These pins enter the input mode at reset. ■ In memory expansion and microprocessor modes Address (A$_{16}$–A$_{23}$) is output. These pins also function as I/O port pins according to the register setting.
P1$_0$–P1$_7$	I/O port P1	I/O	■ In single-chip mode These pins have the same functions as port P0. ■ In memory expansion and microprocessor modes The low-order 8 bits of data (D$_0$–D$_7$) are input/output. When the external data bus has an 8-bit width, address (LA$_0$–LA$_7$) output and data (D$_0$–D$_7$) input/output can be performed with the time-sharing method, according to the register setting.
P2$_0$–P2$_7$	I/O port P2	I/O	■ In single-chip mode or When 8-bit external data bus is used in memory expansion mode and microprocessor mode These pins have the same functions as port P0. ■ When the 16-bit external data bus is used in memory expansion or microprocessor mode The high-order 8 bits of data (D$_8$–D$_{15}$) are input or output.
P3$_0$–P3$_3$	I/O port P3	I/O	■ In single-chip mode These pins have the same functions as port P0. ■ In memory expansion mode P3$_0$ functions as an I/O port pin; and P3$_1$, P3$_2$, and P3$_3$ function as the output pins of $\overline{\text{RD}}$, $\overline{\text{BLW}}$, $\overline{\text{BHW}}$, respectively. P3$_0$ also functions as an output pin of $\overline{\text{RDY}}$ according to the register setting. When the external data bus has a width of 8 bits, the $\overline{\text{BHW}}$ pin functions as an I/O port pin (P3$_3$). ■ In microprocessor mode P3$_0$ functions as an input pin of $\overline{\text{RDY}}$; and P3$_1$,P3$_2$, P3$_3$ function as the output pins of $\overline{\text{RD}}$, $\overline{\text{BLW}}$, $\overline{\text{BHW}}$, respectively. P3$_0$ also functions as an I/O port pin according to the register setting. When the external data bus has a width of 8 bits, the $\overline{\text{BHW}}$ pin functions as an I/O port pin (P3$_3$).
P4$_0$–P4$_7$	I/O port P4	I/O	■ In single-chip mode These pins have the same functions as port P0. ■ In memory expansion mode P4$_0$–P4$_7$ function as I/O port pins. According to the register setting, these pins function as output pins or input pins of ALE, ϕ_1, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$, $\overline{\text{CS}_0}$–$\overline{\text{CS}_3}$, respectively. ■ In microprocessor mode P4$_0$–P4$_4$ function as output or input pins of ALE, ϕ_1, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$, $\overline{\text{CS}_0}$, and P4$_5$–P4$_7$ as I/O port pins, respectively. According to the register setting, P4$_0$–P4$_3$ also function as I/O port pins, and P4$_5$–P4$_7$ as output pins of $\overline{\text{CS}_1}$–$\overline{\text{CS}_3}$.

Pin	Name	Input/ Output	Functions
P5$_0$–P5$_7$	I/O port P5	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as I/O pins for timers A0–A3, output pins for the real-time output, and input pins for the key-input interrupt.
P6$_0$–P6$_7$	I/O port P6	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as I/O pins for timer A4, input pins for external interrupt inputs $\overline{\text{INT}}_0$–$\overline{\text{INT}}_2$, and input pins for timers B0–B2.
P7$_0$–P7$_7$	I/O port P7	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as input pins for the A-D converter, output pins for the D-A converter, and input pins for $\overline{\text{INT}}_2$, $\overline{\text{INT}}_3$, and $\overline{\text{INT}}_4$.
P8$_0$–P8$_7$	I/O port P8	I/O	In addition to having the same functions as port P0 in the single-chip mode, these pins also function as I/O pins for UART0, UART1, output pins for D-A converter, and input pins for $\overline{\text{INT}}_3$ and $\overline{\text{INT}}_4$.
P10$_0$–P10$_7$	I/O port P10	I/O	■ In single-chip mode These pins have the same functions as port P0. ■ In memory expansion and microprocessor modes Address (A$_0$–A$_7$) is output.
P11$_0$–P11$_7$	I/O port P11	I/O	■ In single-chip mode These pins have the same functions as port P0. ■ In memory expansion and microprocessor modes Address (A$_8$–A$_{15}$) is output. Also, these pins function as I/O port pins according to the register setting.
$\overline{\text{NMI}}$	Non-maskable interrupt	Input	This pin is for a non-maskable interrupt.

PIN DESCRIPTION (FLASH MEMORY SERIAL I/O MODE)

Pin	Name	Input /Output	Functions
V_{CC}, V_{SS}	Power supply input	—	Apply 5 V ± 0.5 V to Vcc, and 0 V to Vss.
MD0	MD0	Input	Connect this pin to Vss.
MD1	MD1	Input	Connect this pin to Vss via a resistor of 10 kΩ to 100 kΩ.
$\overline{\text{RESET}}$	Reset input	Input	The reset input pin.
X_{IN}	Clock input	Input	Connect a ceramic resonator between the X_{IN} and X_{OUT} pins, or input an external clock from the X_{IN} pin with the X_{OUT} pin left open.
X_{OUT}	Clock output	Output	
BYTE	BYTE	Input	Connect this pin to Vcc or Vss. (This is not used in the flash memory serial I/O mode.)
V_{CONT}	Filter circuit connection	—	Connect this pin to the filter circuit, or leave this pin open. (This is not used in the flash memory serial I/O mode.)
AVcc, AVss	Analog supply input	—	Connect AVcc to Vcc, and AVss to Vss.
V_{REF}	Reference voltage input	Input	Input an arbitrary level within the range of V_{SS}–V_{CC}. (This is not used in the flash memory serial I/O mode.)
$P0_0$–$P0_7$	Input port P0	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P1_0$–$P1_7$	Input port P1	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P2_0$–$P2_7$	Input port P2	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P3_0$–$P3_3$	Input port P3	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P4_0$, $P4_4$– $P4_7$	Input port P4	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P4_1$	SCLK input	Input	This is an input pin for a serial clock.
$P4_2$	SDA I/O	I/O	This is an I/O pin for serial data. Connect this pin to V_{CC} via a resistor (about 1 kΩ).
$P4_3$	BUSY output	Output	This is an output pin for the BUSY signal.
$P5_0$–$P5_7$	Input port P5	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P6_0$–$P6_7$	Input port P6	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P7_0$–$P7_7$	Input port P7	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P8_0$–$P8_7$	Input port P8	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P10_0$–$P10_7$	Input port P10	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$P11_0$–$P11_7$	Input port P11	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
$\overline{\text{NMI}}$	Non-maskable interrupt	Input	Input “H”, or leave this pin open.

BASIC FUNCTION BLOCKS

These microcomputers contain the following devices on the single chip: the flash memory, RAM, CPU, bus interface unit, and peripheral devices such as the interrupt control circuit, timers, serial I/O, A-D converter, D-A converter, I/O ports, clock generating circuit, etc.

MEMORY

Figures 1 to 3 show the memory maps. The address space is 16 Mbytes from addresses 0_{16} to $FFFFFF_{16}$. The address space is divided into 64-Kbyte units called banks. The banks are numbered from 0_{16} to FF_{16}. Bank FF_{16} is a reserved area for the development support tool. Therefore, do not use bank FF_{16}.

Internal flash memory and internal RAM are assigned as shown in Figures 1 to 3.

Addresses $FFC0_{16}$ to $FFFF_{16}$ contain the RESET and the interrupt vector addresses, and the interrupt vectors are stored there.

For details, refer to the section on interrupts.

Assigned to addresses 0_{16} to FF_{16} are peripheral devices such as I/O ports, A-D converter, D-A converter, UART, timers, interrupt control registers, etc. Figures 7 and 8 show the location of SFRs.

For the flash memory in the boot ROM area, refer to the section on the flash memory mode.

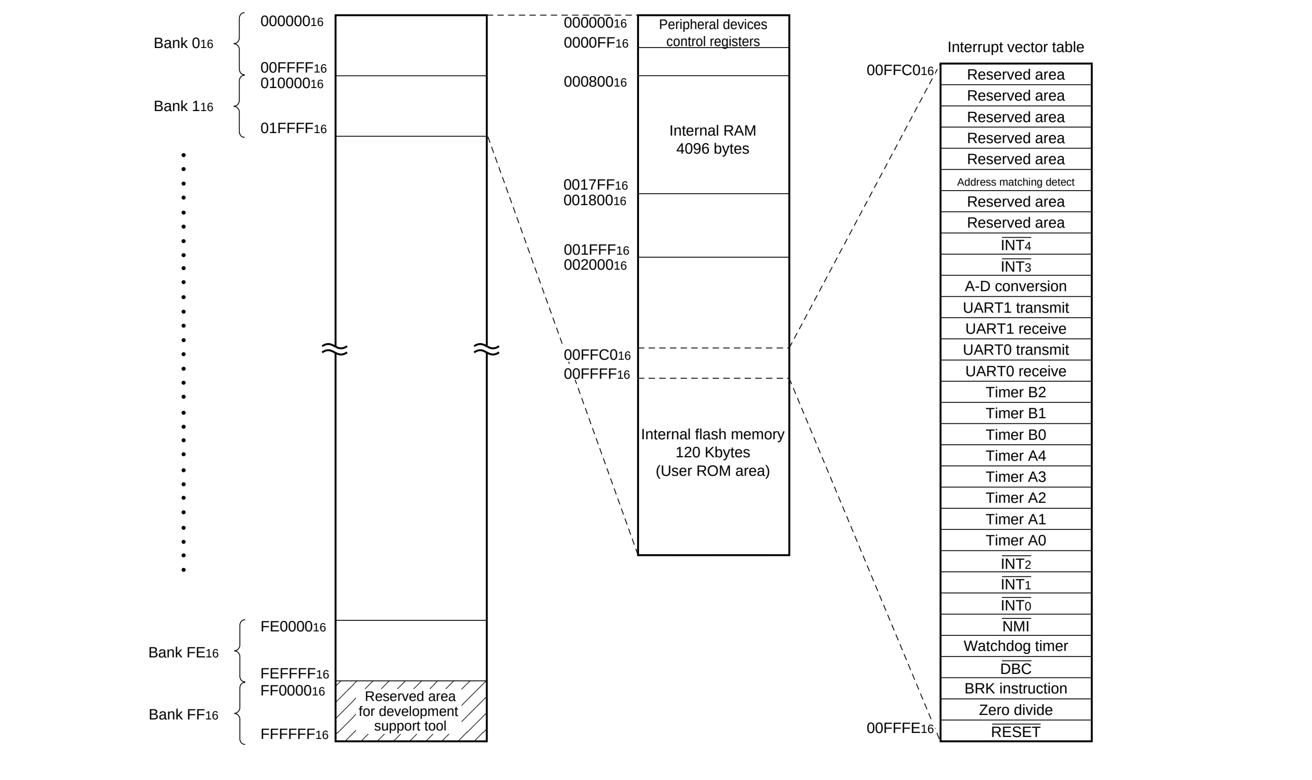


Fig. 1 Memory map of M37902FCCHP (Single-chip mode)

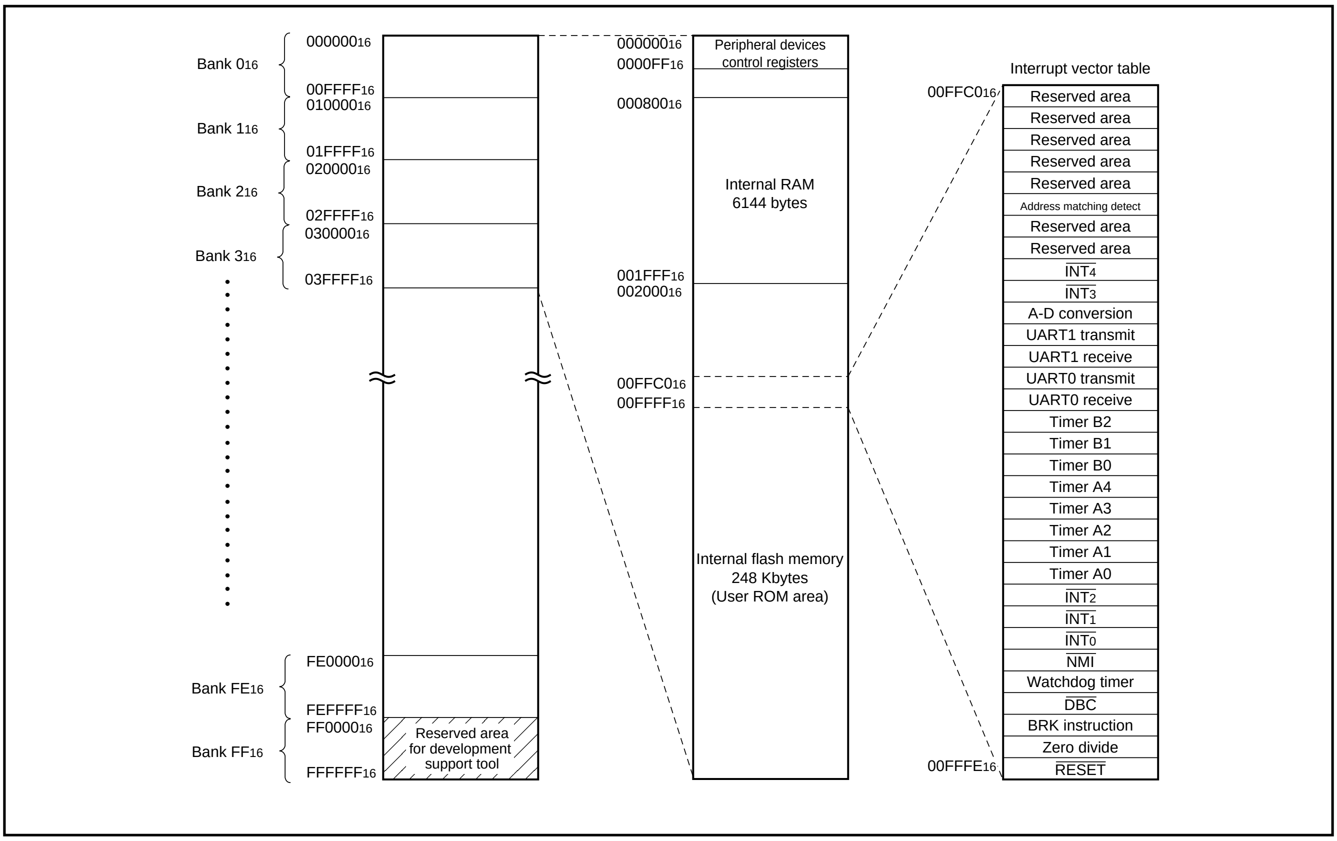


Fig. 2 Memory map of M37902FGCHP (Single-chip mode)

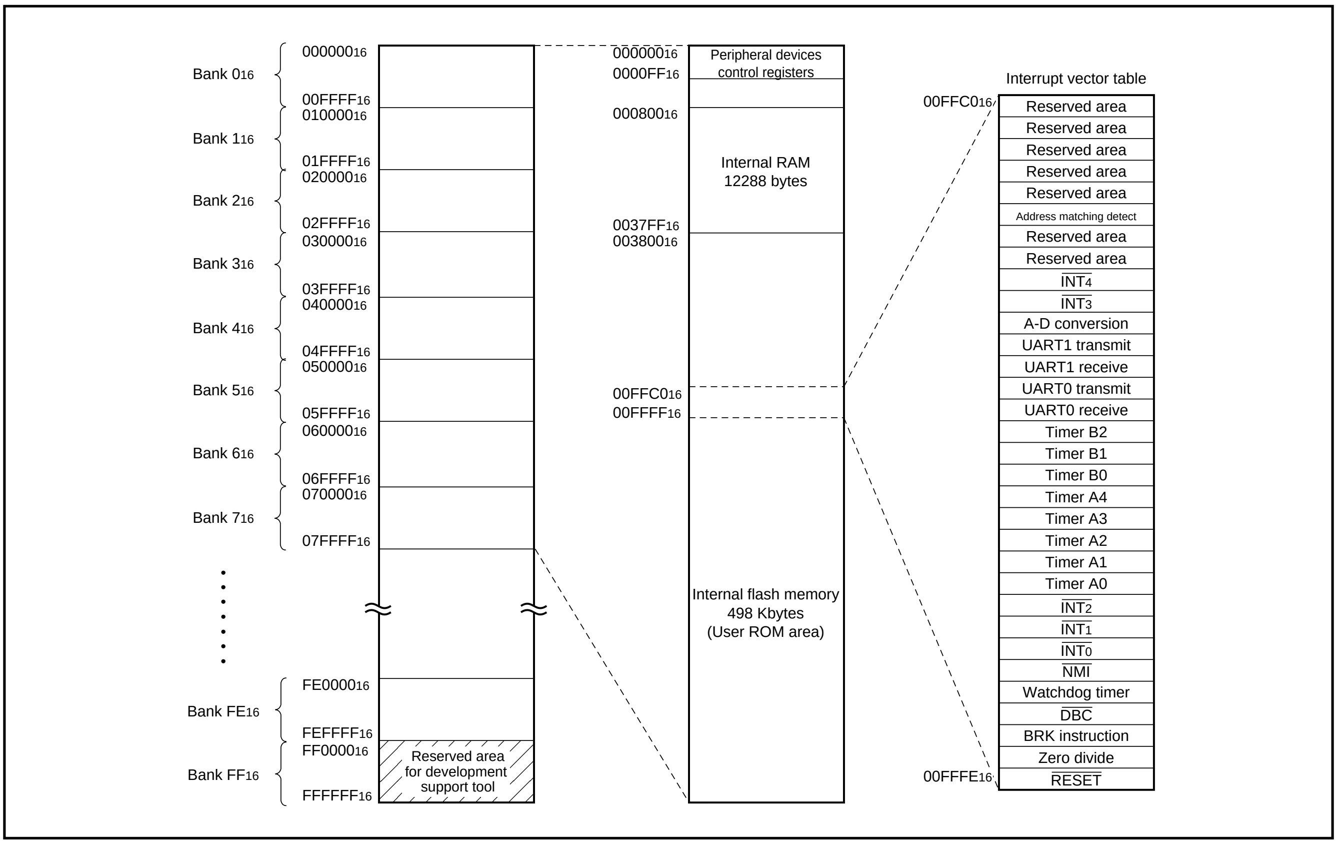


Fig. 3 Memory map of M37902FJCHP (Single-chip mode)

Address (Hexadecimal notation)	
000000$_{16}$	Reserved area **(Note)**
000001$_{16}$	Reserved area **(Note)**
000002$_{16}$	Port P0 register
000003$_{16}$	Port P1 register
000004$_{16}$	Port P0 direction register
000005$_{16}$	Port P1 direction register
000006$_{16}$	Port P2 register
000007$_{16}$	Port P3 register
000008$_{16}$	Port P2 direction register
000009$_{16}$	Port P3 direction register
00000A$_{16}$	Port P4 register
00000B$_{16}$	Port P5 register
00000C$_{16}$	Port P4 direction register
00000D$_{16}$	Port P5 direction register
00000E$_{16}$	Port P6 register
00000F$_{16}$	Port P7 register
000010$_{16}$	Port P6 direction register
000011$_{16}$	Port P7 direction register
000012$_{16}$	Port P8 register
000013$_{16}$	
000014$_{16}$	Port P8 direction register
000015$_{16}$	
000016$_{16}$	Port P10 register
000017$_{16}$	Port P11 register
000018$_{16}$	Port P10 direction register
000019$_{16}$	Port P11 direction register
00001A$_{16}$	
00001B$_{16}$	
00001C$_{16}$	
00001D$_{16}$	
00001E$_{16}$	A-D control register 0
00001F$_{16}$	A-D control register 1
000020$_{16}$	A-D register 0
000021$_{16}$	
000022$_{16}$	A-D register 1
000023$_{16}$	
000024$_{16}$	A-D register 2
000025$_{16}$	
000026$_{16}$	A-D register 3
000027$_{16}$	
000028$_{16}$	A-D register 4
000029$_{16}$	
00002A$_{16}$	A-D register 5
00002B$_{16}$	
00002C$_{16}$	A-D register 6
00002D$_{16}$	
00002E$_{16}$	A-D register 7
00002F$_{16}$	
000030$_{16}$	UART0 transmit/receive mode register
000031$_{16}$	UART0 baud rate register (BRG0)
000032$_{16}$	UART0 transmit buffer register
000033$_{16}$	
000034$_{16}$	UART0 transmit/receive control register 0
000035$_{16}$	UART0 transmit/receive control register 1
000036$_{16}$	UART0 receive buffer register
000037$_{16}$	
000038$_{16}$	UART1 transmit/receive mode register
000039$_{16}$	UART1 baud rate register (BRG1)
00003A$_{16}$	UART1 transmit buffer register
00003B$_{16}$	
00003C$_{16}$	UART1 transmit/receive control register 0
00003D$_{16}$	UART1 transmit/receive control register 1
00003E$_{16}$	UART1 receive buffer register
00003F$_{16}$	

Address (Hexadecimal notation)	
000040$_{16}$	Count start register
000041$_{16}$	
000042$_{16}$	One-shot start register
000043$_{16}$	
000044$_{16}$	Up-down register
000045$_{16}$	Timer A clock division select register
000046$_{16}$	Timer A0 register
000047$_{16}$	
000048$_{16}$	Timer A1 register
000049$_{16}$	
00004A$_{16}$	Timer A2 register
00004B$_{16}$	
00004C$_{16}$	Timer A3 register
00004D$_{16}$	
00004E$_{16}$	Timer A4 register
00004F$_{16}$	
000050$_{16}$	Timer B0 register
000051$_{16}$	
000052$_{16}$	Timer B1 register
000053$_{16}$	
000054$_{16}$	Timer B2 register
000055$_{16}$	
000056$_{16}$	Timer A0 mode register
000057$_{16}$	Timer A1 mode register
000058$_{16}$	Timer A2 mode register
000059$_{16}$	Timer A3 mode register
00005A$_{16}$	Timer A4 mode register
00005B$_{16}$	Timer B0 mode register
00005C$_{16}$	Timer B1 mode register
00005D$_{16}$	Timer B2 mode register
00005E$_{16}$	Processor mode register 0
00005F$_{16}$	Processor mode register 1
000060$_{16}$	Watchdog timer register
000061$_{16}$	Watchdog timer frequency select register
000062$_{16}$	Particular function select register 0
000063$_{16}$	Particular function select register 1
000064$_{16}$	Particular function select register 2
000065$_{16}$	Reserved area **(Note)**
000066$_{16}$	Debug control register 0
000067$_{16}$	Debug control register 1
000068$_{16}$	Address comparison register 0
000069$_{16}$	
00006A$_{16}$	
00006B$_{16}$	Address comparison register 1
00006C$_{16}$	
00006D$_{16}$	
00006E$_{16}$	$\overline{\mathrm{INT}_3}$ interrupt control register
00006F$_{16}$	$\overline{\mathrm{INT}_4}$ interrupt control register
000070$_{16}$	A-D conversion interrupt control register
000071$_{16}$	UART0 transmit interrupt control register
000072$_{16}$	UART0 receive interrupt control register
000073$_{16}$	UART1 transmit interrupt control register
000074$_{16}$	UART1 receive interrupt control register
000075$_{16}$	Timer A0 interrupt control register
000076$_{16}$	Timer A1 interrupt control register
000077$_{16}$	Timer A2 interrupt control register
000078$_{16}$	Timer A3 interrupt control register
000079$_{16}$	Timer A4 interrupt control register
00007A$_{16}$	Timer B0 interrupt control register
00007B$_{16}$	Timer B1 interrupt control register
00007C$_{16}$	Timer B2 interrupt control register
00007D$_{16}$	$\overline{\mathrm{INT}_0}$ interrupt control register
00007E$_{16}$	$\overline{\mathrm{INT}_1}$ interrupt control register
00007F$_{16}$	$\overline{\mathrm{INT}_2}$ interrupt control register

Note: Do not write to this address.

Fig. 7 Location of SFRs (1)

Address (Hexadecimal notation)		Address (Hexadecimal notation)	
000080$_{16}$	$\overline{CS_0}$ control register L	0000C0$_{16}$	
000081$_{16}$	$\overline{CS_0}$ control register H	0000C1$_{16}$	
000082$_{16}$	$\overline{CS_1}$ control register L	0000C2$_{16}$	
000083$_{16}$	$\overline{CS_1}$ control register H	0000C3$_{16}$	
000084$_{16}$	$\overline{CS_2}$ control register L	0000C4$_{16}$	
000085$_{16}$	$\overline{CS_2}$ control register H	0000C5$_{16}$	
000086$_{16}$	$\overline{CS_3}$ control register L	0000C6$_{16}$	
000087$_{16}$	$\overline{CS_3}$ control register H	0000C7$_{16}$	
000088$_{16}$		0000C8$_{16}$	
000089$_{16}$		0000C9$_{16}$	
00008A$_{16}$	Area $\overline{CS_0}$ start address register	0000CA$_{16}$	
00008B$_{16}$		0000CB$_{16}$	
00008C$_{16}$	Area $\overline{CS_1}$ start address register	0000CC$_{16}$	
00008D$_{16}$		0000CD$_{16}$	
00008E$_{16}$	Area $\overline{CS_2}$ start address register	0000CE$_{16}$	
00008F$_{16}$		0000CF$_{16}$	
000090$_{16}$	Area $\overline{CS_3}$ start address register	0000D0$_{16}$	
000091$_{16}$		0000D1$_{16}$	
000092$_{16}$	Port function control register	0000D2$_{16}$	
000093$_{16}$		0000D3$_{16}$	
000094$_{16}$	External interrupt input control register	0000D4$_{16}$	
000095$_{16}$	External interrupt input read-out register	0000D5$_{16}$	
000096$_{16}$	D-A control register	0000D6$_{16}$	
000097$_{16}$		0000D7$_{16}$	
000098$_{16}$	D-A register 0	0000D8$_{16}$	
000099$_{16}$	D-A register 1	0000D9$_{16}$	
00009A$_{16}$	D-A register 2	0000DA$_{16}$	
00009B$_{16}$		0000DB$_{16}$	
00009C$_{16}$	Reserved area **(Note)**	0000DC$_{16}$	
00009D$_{16}$	Reserved area **(Note)**	0000DD$_{16}$	
00009E$_{16}$	Flash memory control register	0000DE$_{16}$	
00009F$_{16}$		0000DF$_{16}$	
0000A0$_{16}$	Real-time output control register	0000E0$_{16}$	
0000A1$_{16}$		0000E1$_{16}$	
0000A2$_{16}$	Pulse output data register 0	0000E2$_{16}$	
0000A3$_{16}$		0000E3$_{16}$	
0000A4$_{16}$	Pulse output data register 1	0000E4$_{16}$	
0000A5$_{16}$		0000E5$_{16}$	
0000A6$_{16}$		0000E6$_{16}$	
0000A7$_{16}$		0000E7$_{16}$	
0000A8$_{16}$		0000E8$_{16}$	
0000A9$_{16}$		0000E9$_{16}$	
0000AA$_{16}$		0000EA$_{16}$	
0000AB$_{16}$		0000EB$_{16}$	
0000AC$_{16}$	Serial I/O pin control register	0000EC$_{16}$	
0000AD$_{16}$		0000ED$_{16}$	
0000AE$_{16}$		0000EE$_{16}$	
0000AF$_{16}$		0000EF$_{16}$	
0000B0$_{16}$		0000F0$_{16}$	
0000B1$_{16}$		0000F1$_{16}$	
0000B2$_{16}$		0000F2$_{16}$	
0000B3$_{16}$		0000F3$_{16}$	
0000B4$_{16}$		0000F4$_{16}$	
0000B5$_{16}$		0000F5$_{16}$	
0000B6$_{16}$		0000F6$_{16}$	
0000B7$_{16}$		0000F7$_{16}$	
0000B8$_{16}$		0000F8$_{16}$	
0000B9$_{16}$		0000F9$_{16}$	
0000BA$_{16}$	Reserved area **(Note)**	0000FA$_{16}$	
0000BB$_{16}$	Reserved area **(Note)**	0000FB$_{16}$	
0000BC$_{16}$	Clock control register	0000FC$_{16}$	
0000BD$_{16}$	Reserved area **(Note)**	0000FD$_{16}$	
0000BE$_{16}$	Reserved area **(Note)**	0000FE$_{16}$	
0000BF$_{16}$	Reserved area **(Note)**	0000FF$_{16}$	

Note: Do not write to this address.

Fig. 8 Location of SFRs (2)

CENTRAL PROCESSING UNIT (CPU)

The CPU has 13 registers and is shown in Figure 9. Each of these registers is described below.

ACCUMULATOR A (A)

Accumulator A is the main register of the microcomputer. It consists of 16 bits and the low-order 8 bits can be used separately. Data length flag m determines whether the register is used as 16-bit register or as 8-bit register. It is used as a 16-bit register when flag m is “0” and as an 8-bit register when flag m is “1”. Flag m is a part of the processor status register (PS) which is described later.
Data operations such as calculations, data transfer, input/output, etc., are executed mainly through accumulator A.

ACCUMULATOR B (B)

Accumulator B has the same functions as accumulator A, but the use of accumulator B requires more instruction bytes and execution cycles than accumulator A.

ACCUMULATOR E

Accumulator E is a 32-bit register and consists of accumulator A (low-order 16 bits) and accumulator B (high-order 16 bits). It is used for 32-bit data processing.

INDEX REGISTER X (X)

Index register X consists of 16 bits and the low-order 8 bits can be used separately. Index register length flag x determines whether the register is used as 16-bit register or as 8-bit register. It is used as a 16-bit register when flag x is “0” and as an 8-bit register when flag x is “1”. Flag x is a part of the processor status register (PS) which is described later.
In index addressing modes in which register X is used as the index register, the contents of this address are added to obtain the real address.
Index register X functions as a pointer register which indicates an address of data table in instructions MVP, MVN, RMPA (Repeat MultiPly and Accumulate).

INDEX REGISTER Y (Y)

Index register Y consists of 16 bits and the low-order 8 bits can be used separately. The index register length flag x determines whether the register is used as 16-bit register or as 8-bit register. It is used as a 16-bit register when flag x is “0” and as an 8-bit register when flag x is “1”. Flag x is a part of the processor status register (PS) which is described later.
In index addressing modes in which register Y is used as the index register, the contents of this address are added to obtain the real address.
Index register Y functions as a pointer register which indicates an address of data table in instructions MVP, MVN, RMPA (Repeat MultiPly and Accumulate).

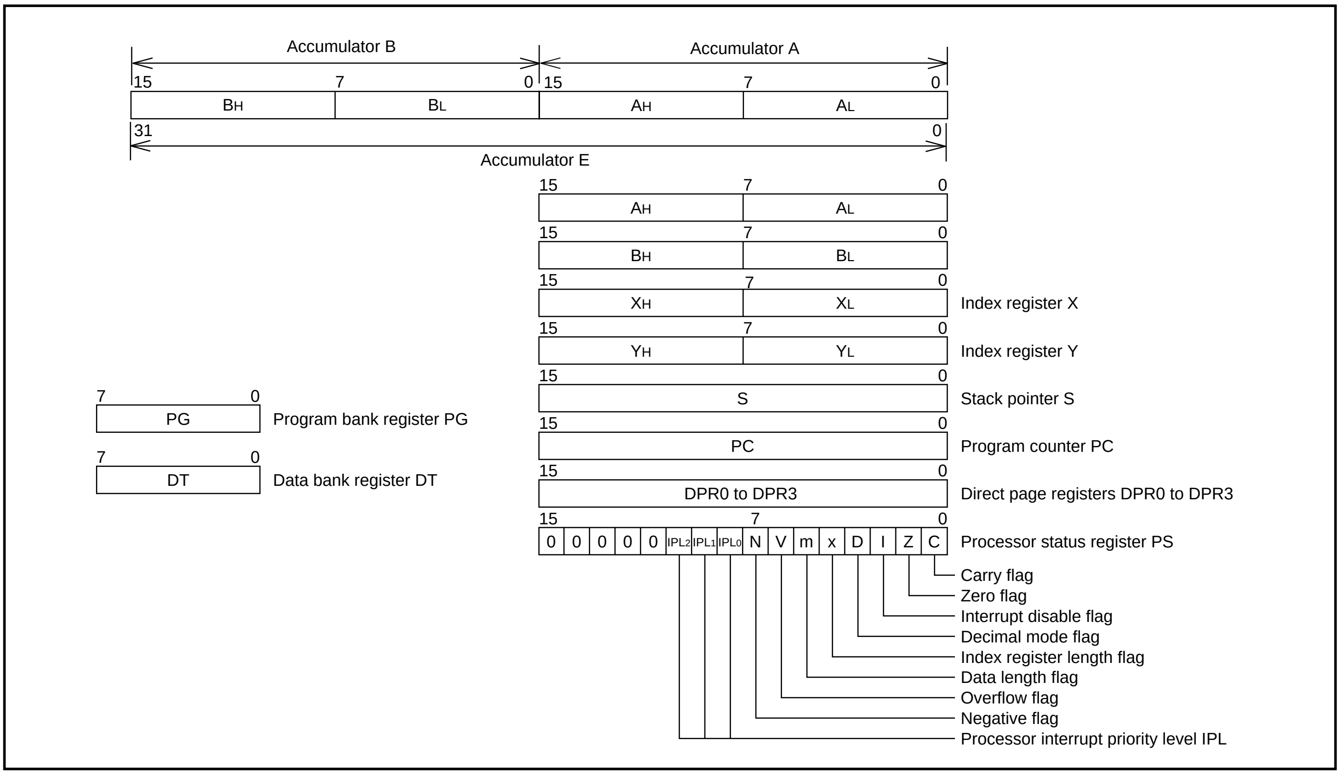


Fig. 9 Register structure

STACK POINTER (S)

Stack pointer (S) is a 16-bit register. It is used during a subroutine call or interrupts. It is also used during stack, stack pointer relative, or stack pointer relative indirect indexed Y addressing mode.

PROGRAM COUNTER (PC)

Program counter (PC) is a 16-bit counter that indicates the low-order 16 bits of the next program memory address to be executed. There is a bus interface unit between the program memory and the CPU, so that the program memory is accessed through bus interface unit. This is described later.

PROGRAM BANK REGISTER (PG)

Program bank register is an 8-bit register that indicates the high-order 8 bits of the next program memory address to be executed. When a carry occurs by incrementing the contents of the program counter, the contents of the program bank register (PG) is increased by 1. Also, when a carry or borrow occurs after adding or subtracting the offset value to or from the contents of the program counter (PC) using the branch instruction, the contents of the program bank register (PG) is increased or decreased by 1, so that programs can be written without worrying about bank boundaries.

DATA BANK REGISTER (DT)

Data bank register (DT) is an 8-bit register. With some addressing modes, the data bank register (DT) is used to specify a part of the memory address. The contents of data bank register (DT) is used as the high-order 8 bits of a 24-bit address. Addressing modes that use the data bank register (DT) are direct indirect, direct indexed X indirect, direct indirect indexed Y, absolute, absolute bit, absolute indexed X, absolute indexed Y, absolute bit relative, and stack pointer relative indirect indexed Y.

DIRECT PAGE REGISTERS 0 to 3 (DPR0 to DPR3)

The direct page register is a 16-bit register. An addressing mode of which name includes ‘direct’ generates an address of data to be accessed, regarding the contents of this register as the base address. The 7900 Series has been expanded direct page registers up to 4 (DPR0 to DPR3), in comparison to the 7700 Series which has the single direct page register. Accordingly, the 7900 Series’s direct addressing method which uses direct page registers differs from that of the 7700 Series. However, the conventional direct addressing method, using only DPR0, is still be selectable, in order to make use of the 7700 Series software property. For more details, refer to the section on the direct page.

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of flags to indicate the result of operation and CPU interrupt levels.
Branch operations can be performed by testing the flags C, Z, V, and N.
The details of each bit of the processor status register are described below.

1. Carry flag (C)

The carry flag contains the carry or borrow generated by the ALU after an arithmetic operation. This flag is also affected by shift and rotate instructions. This flag can be set and reset directly with the SEC and CLC instructions or with the SEP and CLP instructions.

2. Zero flag (Z)

The zero flag is set if the result of an arithmetic operation or data transfer is zero and reset if it is not. This flag can be set and reset directly with the SEP and CLP instructions.

3. Interrupt disable flag (I)

When the interrupt disable flag is set to “1”, all interrupts except watchdog timer, $\overline{\text{NMI}}$, and software interrupt are disabled. This flag is set to “1” automatically when an interrupt is accepted. It can be set and reset directly with the SEI and CLI instructions or SEP and CLP instructions.

4. Decimal mode flag (D)

The decimal mode flag determines whether addition and subtraction are performed as binary or decimal. Binary arithmetic is performed when this flag is “0”. If it is “1”, decimal arithmetic is performed with each word treated as 2- or 4- digit decimal. Arithmetic operation is performed using four digits when data length flag m is “0” and with two digits when it is “1”. Decimal adjust is automatically performed. (Decimal operation is possible only with the ADC and SBC instructions.) This flag can be set and reset with the SEP and CLP instructions.

5. Index register length flag (x)

The index register length flag determines whether index register X and index register Y are used as 16-bit registers or as 8-bit registers. The registers are used as 16-bit registers when flag x is “0” and as 8-bit registers when it is “1”.

This flag can be set and reset with the SEP and CLP instructions.

6. Data length flag (m)

The data length flag determines whether the data length is 16-bit or 8-bit. The data length is 16 bits when flag m is “0” and 8 bits when it is “1”. This flag can be set and reset with the SEM and CLM instructions or with the SEP and CLP instructions.

7. Overflow flag (V)

The overflow flag is valid when addition or subtraction is performed with a word treated as a signed binary number. If data length flag m is “0”, the overflow flag is set when the result of addition or subtraction is outside the range between –32768 and +32767. If data length flag m is “1”, the overflow flag is set when the result of addition or subtraction is outside the range between –128 and +127. It is reset in all other cases. The overflow flag can also be set and reset directly with the SEP, and CLV or CLP instructions.

Additionally, the overflow flag is set when a result of unsigned/signed division exceeds the length of the register where the result is to be stored; the flag is also set when the addition result is outside range of –2147483648 to +2147483647 in the RMPA operation.

8. Negative flag (N)

The negative flag is set when the result of arithmetic operation or data transfer is negative (If data length flag m is “0”, data’s bit 15 is “1”. If data length flag m is “1”, data’s bit 7 is “1”.) It is reset in all other cases. It can also be set and reset with the SEP and CLP instructions.

9. Processor interrupt priority level (IPL)

The processor interrupt priority level (IPL) consists of 3 bits and determines the priority of processor interrupts from level 0 to level 7. Interrupt is enabled when the interrupt priority of the device requesting interrupt (set using the interrupt control register) is higher than the processor interrupt priority. When an interrupt is enabled, the current processor interrupt priority level is saved in a stack and the processor interrupt priority level is replaced by the interrupt priority level of the device requesting the interrupt. Refer to the section on interrupts for more details.

Note: Fix bits 11 to 15 of the processor status register (PS) to “0”.

BANK

In order to effectively use the integrated hardware on the chip, this CPU core uses an address generating method with a 24-bit address split into high-order 8 bits and low-order 16 bits. In other words, the 64 Kbytes specified by the low-order 16 bits are one unit (referred to as "bank"), and the address space is divided into 256 banks (0_{16} to FF_{16}) specified by the high-order 8 bits.

In the program area on the address space, the bank is specified by the program bank register (PG), and the address in the bank is specified by the program counter (PC).

As for each bank boundary, when an overflow has occurred in PC, the contents of PG are incremented by 1. When a borrow has occurred in PC, the contents of PG are decremented by 1. Under the normal conditions, therefore, programming without concern for the bank boundaries is possible. Furthermore, as for the data area on the address space, the bank is specified by the data bank register (DT), and the address in the bank is specified by the operation result by using the various addressing modes **(Note).**

Note: Some addressing modes directly specify a bank.

DIRECT PAGE

The internal memory and control registers for internal peripheral devices, etc. are assigned to bank 0_{16} (addresses 0_{16} to $FFFF_{16}$). The direct page and direct addressing modes have been provided for the effective access to bank 0_{16}. In the 7900 Series, two types of direct addressing modes are available: the conventional direct addressing mode which uses only DPR0, as in the 7700 Series, and the expanded direct addressing mode, which uses up to 4 direct page registers as selected by the user. The addressing mode is selected according to the contents of bit 1 of the processor mode register 1. This bit 1 is cleared to "0" at reset. (In other words, the conventional direct addressing mode is selected.) However, once this bit 1 has been set to "1" by software, this bit cannot be cleared to "0" again, except by reset. That is to say, when one of these two direct addressing modes has been selected just after reset, the selected addressing mode cannot be switched to another one while the program is running.

■ Conventional direct addressing mode

The direct page area consists of 256-byte space. Its bank address is "00_{16}", and the base address of its low-order 16-bit address is specified by the contents of the direct page register 0 (DPR0). In this conventional direct addressing modes, a value (1 byte) just after an instruction code is regarded as an offset value for the DPR0 contents, and the CPU accesses each address in the direct page area.

■ Expanded direct addressing mode

The direct page area consists of four 64-byte spaces. Their bank address is "00_{16}", and the four base addresses of their low-order 16-bit addresses are respectively specified by the contents of four direct page registers. In this expanded direct addressing mode, a value (1 byte) just after an instruction code is regarded as follows:

- High-order 2 bits: regarded as a selection field for DPR0 to DPR3.
- Low-order 6 bits: regarded as an offset value for the selected direct page register.

Then, the CPU accesses each address in each direct page area:

Refer to "7900 Series Software Manual" for details concerning the various addressing modes which use the direct page area.

Instruction Set

The CPU core of the 7900 Series has an expanded instruction set based on the existing 7700/7751 Series' CPU core. In addition, its source code (mnemonic) has the complete upper compatibility with the 7700 Series instruction set.

For details concerning addressing modes and instruction set, refer to "7900 Series Software Manual".

BUS INTERFACE UNIT

Data transfer between the central processing unit (CPU) and internal memory, internal peripheral devices, or external areas is always performed via the bus interface unit (BIU), which is located between the CPU and the internal buses.

Figure 10 shows the BIU and the bus structure. The CPU and BIU are connected by a dedicated bus, and any transfer between the CPU and BIU is controlled by this dedicated bus.

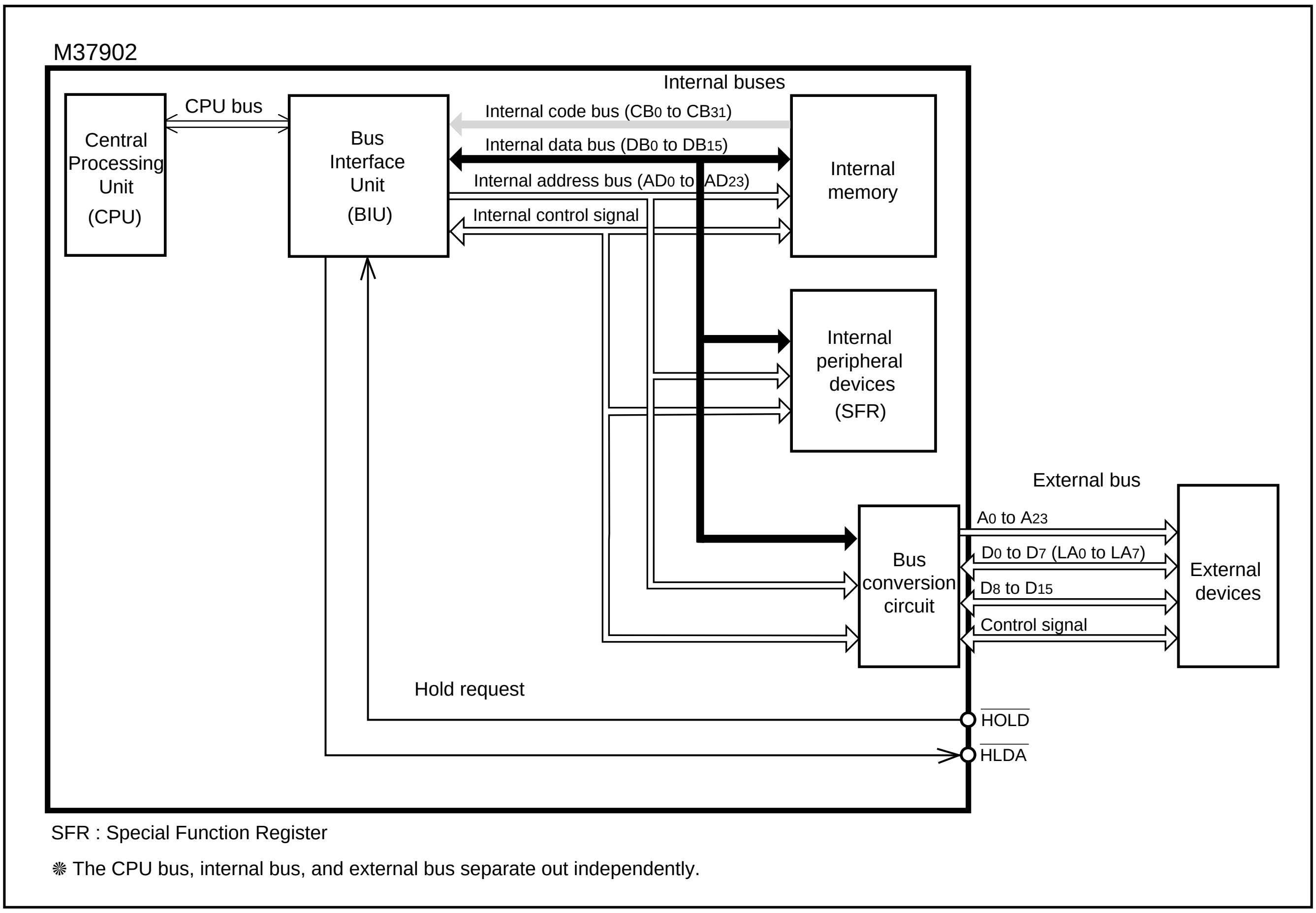
On the other hand, data transfer between the BIU and internal peripheral devices uses the following internal common buses: 32-bit code bus, 16-bit data bus, 24-bit address bus, and control signals.

The bus control method where the code bus and the data bus separate out (hereafter, this method is referred to as the separate code/data bus method) is employed in order to improve data transfer capabilities. As a result, the internal memory is connected to both the code bus and the data bus, and registers of all other internal peripheral devices are connected only to the data bus.

Each width of external buses are as follows: a 24-bit address bus, 16-bit data bus.

The external data bus transfers instruction codes and data. When the code or data access occurs for the external, the external access is performed via the bus conversion circuit.

For details of the connection with the external devices, refer to the section on the processor modes and chip select wait controller described later.



SFR : Special Function Register

❋ The CPU bus, internal bus, and external bus separate out independently.

Fig. 10 BIU and bus structure

BIU structure

The BIU consists of four registers shown in Figure 11. Table 1 lists the functions of each register.

Table 1. Functions of each register

Name	Functions
Program address register	Indicates a storage address for an instruction to be next taken into an instruction queue buffer.
Instruction queue buffer	Temporarily stores an instruction which has been taken from a memory. Consists of 10 bytes.
Data address register	Indicates an address where data will be next read from or written to.
Data buffer	Temporarily stores data which has been read from internal memory, internal peripheral devices, and external areas by the BIU; or temporarily stores data which is to be written to internal memory, internal peripheral devices, and external areas by the CPU. Consists of 32 bits.

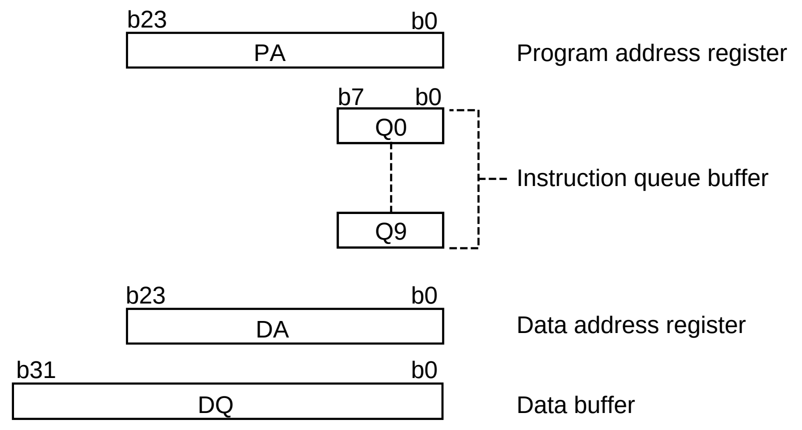


Fig.11 Register structure of BIU

BIU Functions

(1) Instruction prefetch

The BIU has ten instruction queue buffers; each buffer consists of 1 byte. When there is an opening in the bus and the instruction queue buffer, an instruction code is read from the program memory (in other words, the memory where a program is stored) and prefetched into an instruction queue buffer. The prefetched instruction code is transferred from the BIU to the CPU, in response to a request from the CPU, via a dedicated bus.

When a branch occurs as a result of a branch instruction (JMP, BRA, etc.), subroutine call, or interrupt, the contents of the instruction queue buffer are initialized and the BIU reads a new instruction from the branch destination address.

Note that the operations of the BIU instruction prefetch also differ depending on the store addresses for instructions. The store addresses for instructions to be prefetched are categorized as listed in Table 2.

(2) Data read operation

When executing an instruction for reading data from the internal memory, internal peripheral devices, or external areas, at first, the CPU informs the BIU's data address register of the address where the data has been located.

Next, the BIU reads the above data from the specified address, passes it to the data buffer, and then, transfers it to the CPU.

(3) Data write operation

When executing an instruction for writing data into the internal memory, internal peripheral devices, or external area, at first, the CPU informs the BIU's data address register of the address where the data has been located.

Next, the BIU passes the above data to the data buffer register, and then, writes it into the specified address.

(4) Bus cycle

In order for the BIU to execute the above operations (1) through (3), the 24-bit address bus, 32-bit code bus, 16-bit data bus and internal control signals must be appropriately controlled during data transfer between the BIU and internal memory, internal peripheral devices, external areas. This operation is called "bus cycle". The bus cycle is affected by the following conditions at instruction prefetch and data access.

[Instruction prefetch]

- Whether the address area locates in the internal area or the external area.
- When the address area locates in the external area
  ① Whether the external bus width = 16 bits or 8 bits:
  (a) When the external bus width = 16 bits:
  whether the start address for access locates at a 4-byte boundary or at an 8-byte boundary.
  (b) When the external bus width = 8 bits:
  whether the start address for access locates at an even address, a 4-byte boundary or at the 8-byte bound ary.
  ② Whether the prefetch operation is generated by a branch, or not.
  ③ Number of waits
  ④ Whether the burst ROM access is specified or not.

Table 2. Store addresses for instructions to be prefetched

	Low-order 3 bits of store address for instruction		
	AD_2 (A_2)	AD_1 (A_1)	AD_0 (A_0)
Even address	X	X	0
4-byte boundary	X	0	0
8-byte boundary	0	0	0

X: 0 or 1

[Data Access]

- Whether the address area locates in the internal area or the external area.
- Length of data to be transferred: byte, word, double word
- When the address area locates in the external area:
  ① Whether the external bus width = 16 bits or 8 bits:
  ② Number of waits

The BIU controls the bus cycle depending on the above conditions. Figures 12 to 16 show the bus cycle waveform examples for instruction prefetch and data access.

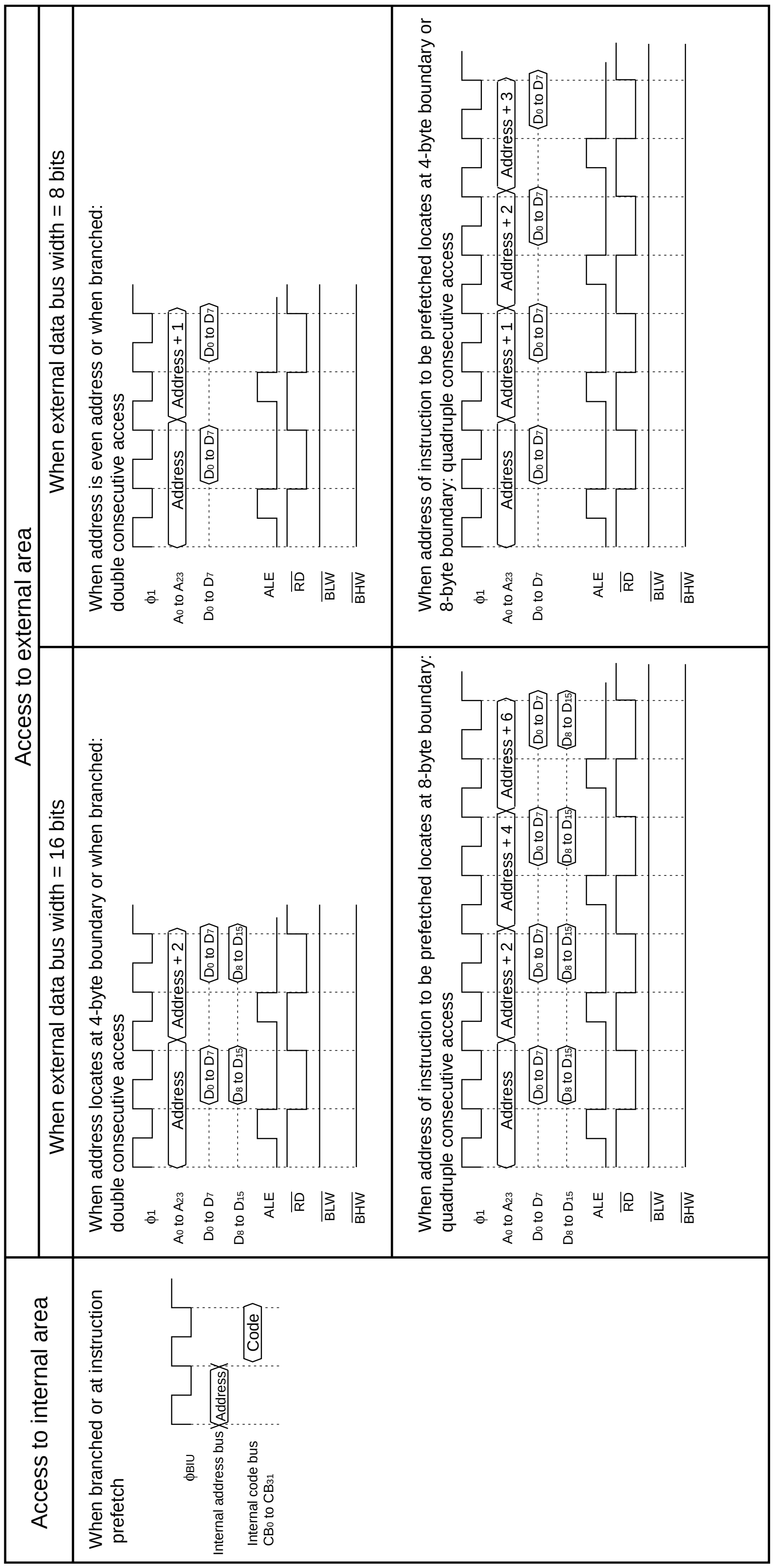

Access to internal area
When branched or at instruction prefetch
φBIU
Internal address bus
Address
Internal code bus CB0 to CB31
Code
Access to external area
When external data bus width = 16 bits
When address locates at 4-byte boundary or when branched: double consecutive access
φ1
A0 to A23
D0 to D7
D8 to D15
ALE
RD
BLW
BHW
Address
Address + 2
D0 to D7
D8 to D15
When address of instruction to be prefetched locates at 8-byte boundary: quadruple consecutive access
Address
Address + 2
Address + 4
Address + 6
When external data bus width = 8 bits
When address is even address or when branched: double consecutive access
Address
Address + 1
When address of instruction to be prefetched locates at 4-byte boundary or 8-byte boundary: quadruple consecutive access
Address
Address + 1
Address + 2
Address + 3

Fig. 12 Bus cycle waveform example for instruction prefetch

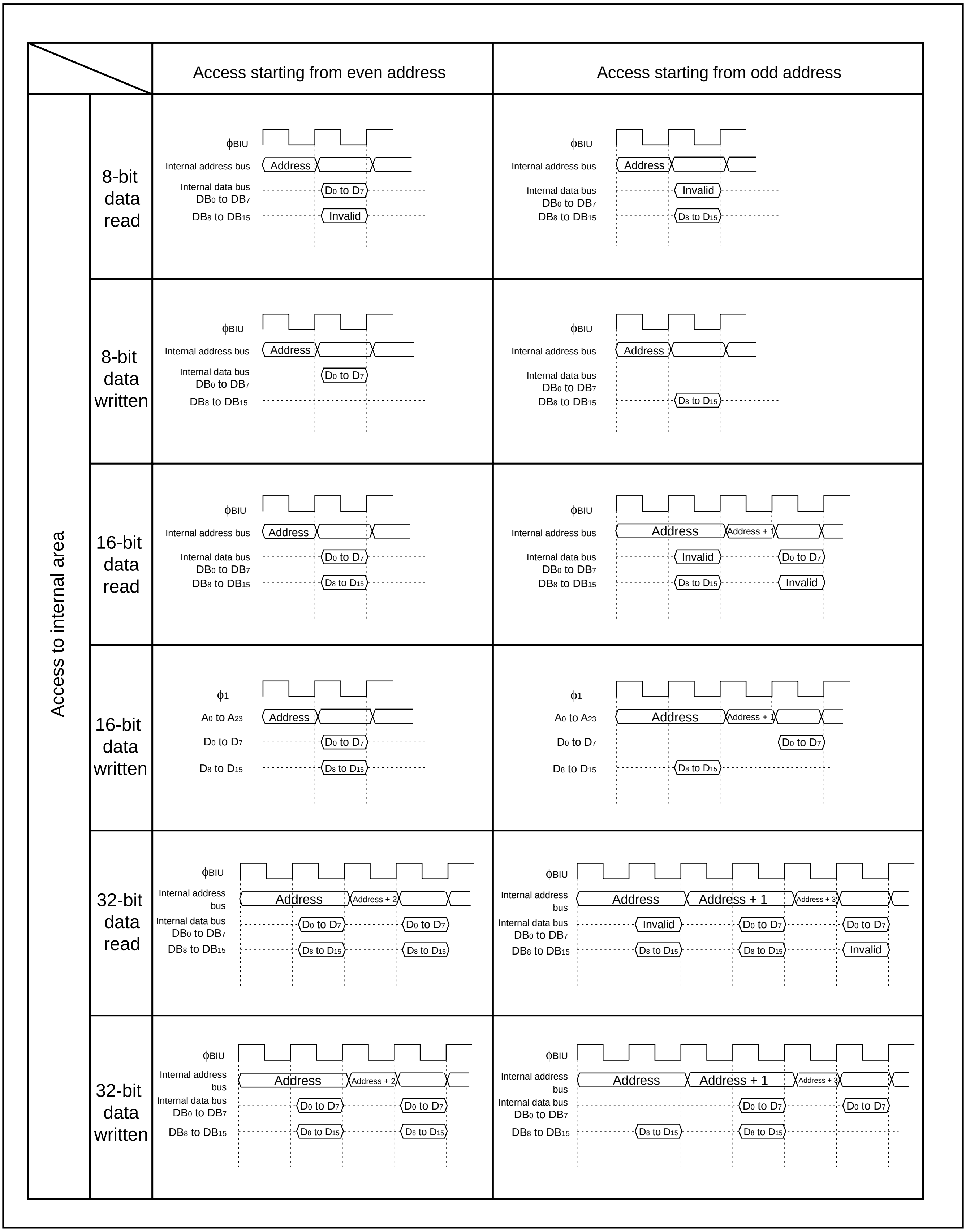


Fig. 13 Bus cycle waveform example for data access (access to internal area)

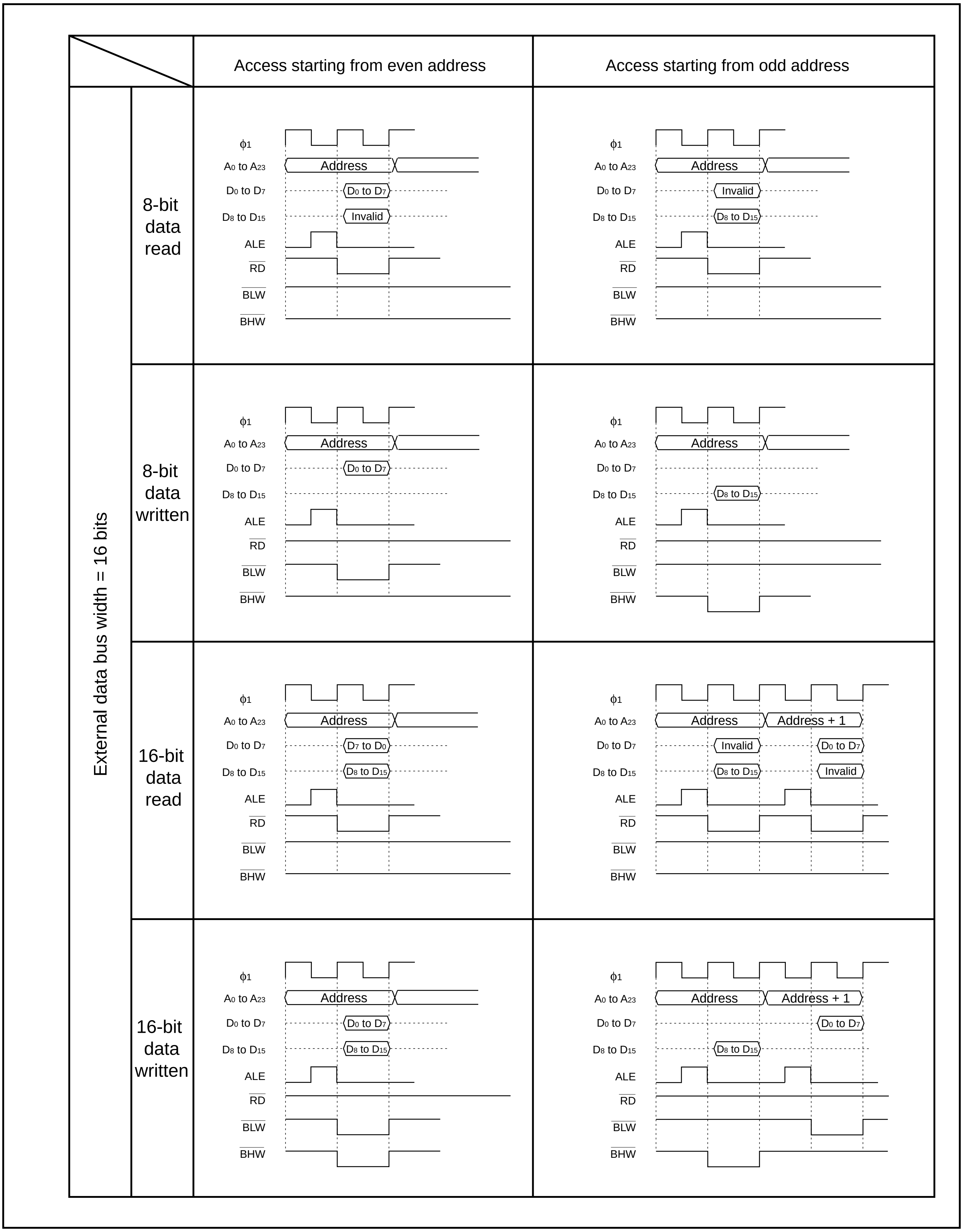


Fig. 14 Bus cycle waveform example for data access (access to external area) (1)

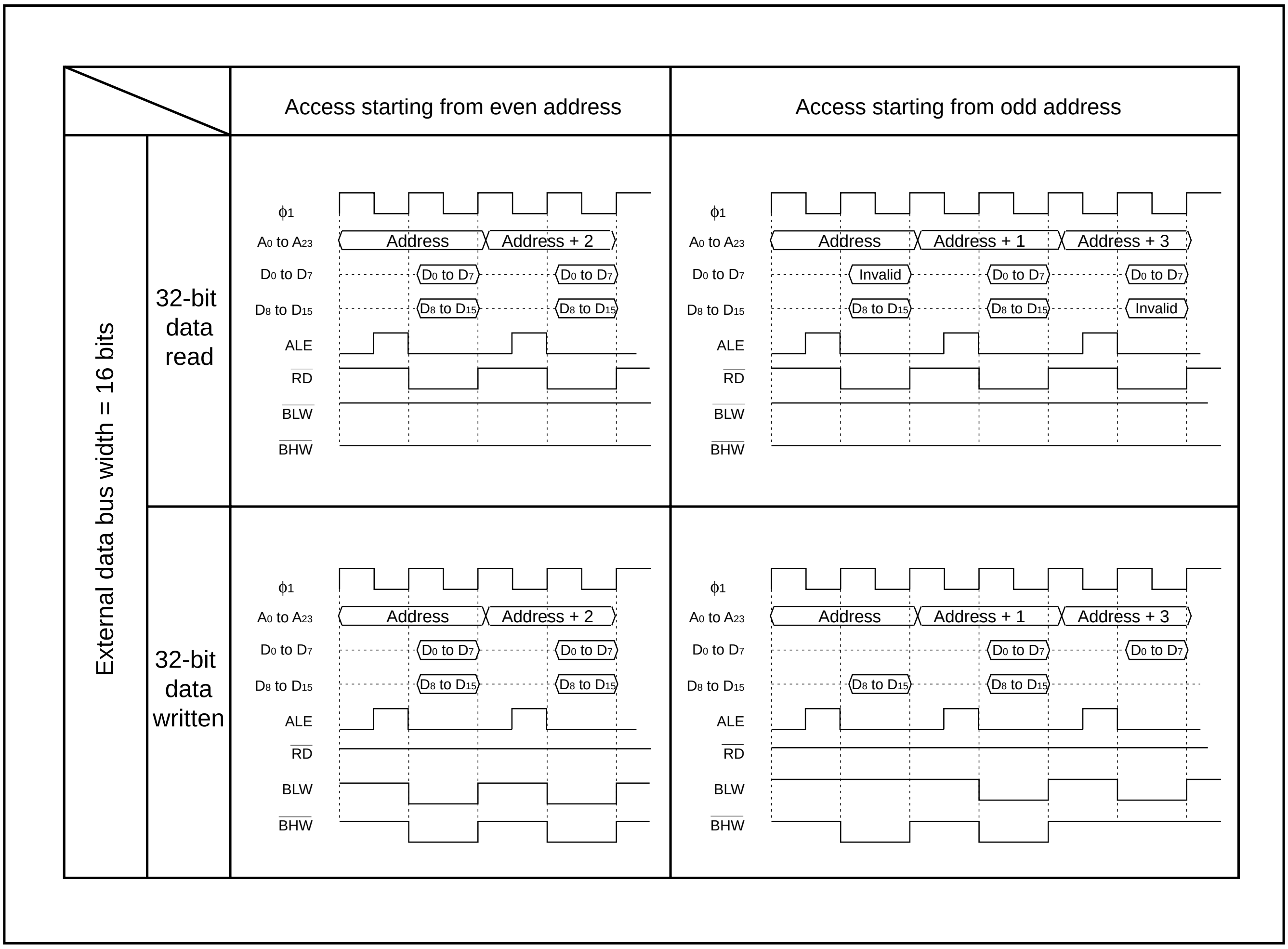
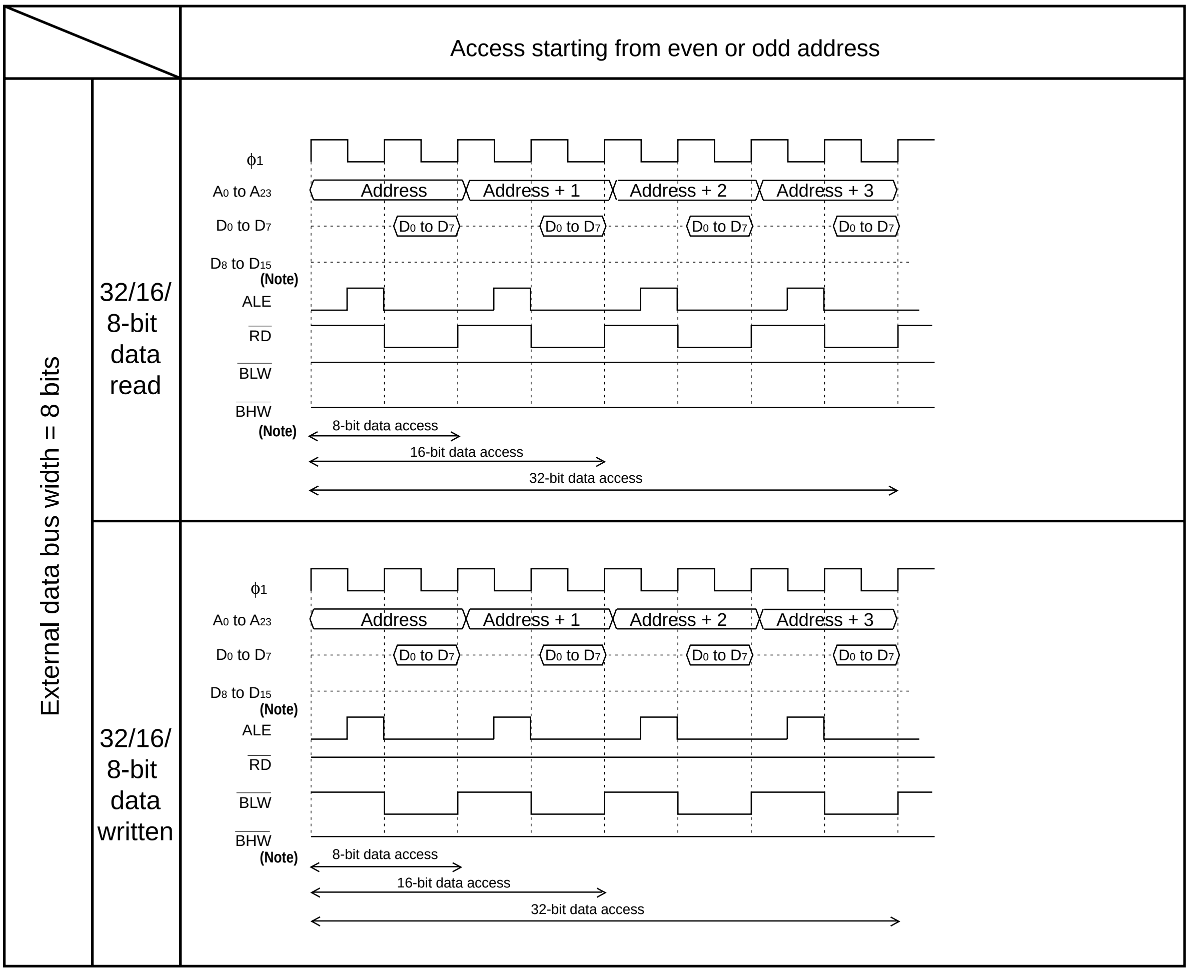


Fig. 15 Bus cycle waveform example for data access (access to external area) (2)

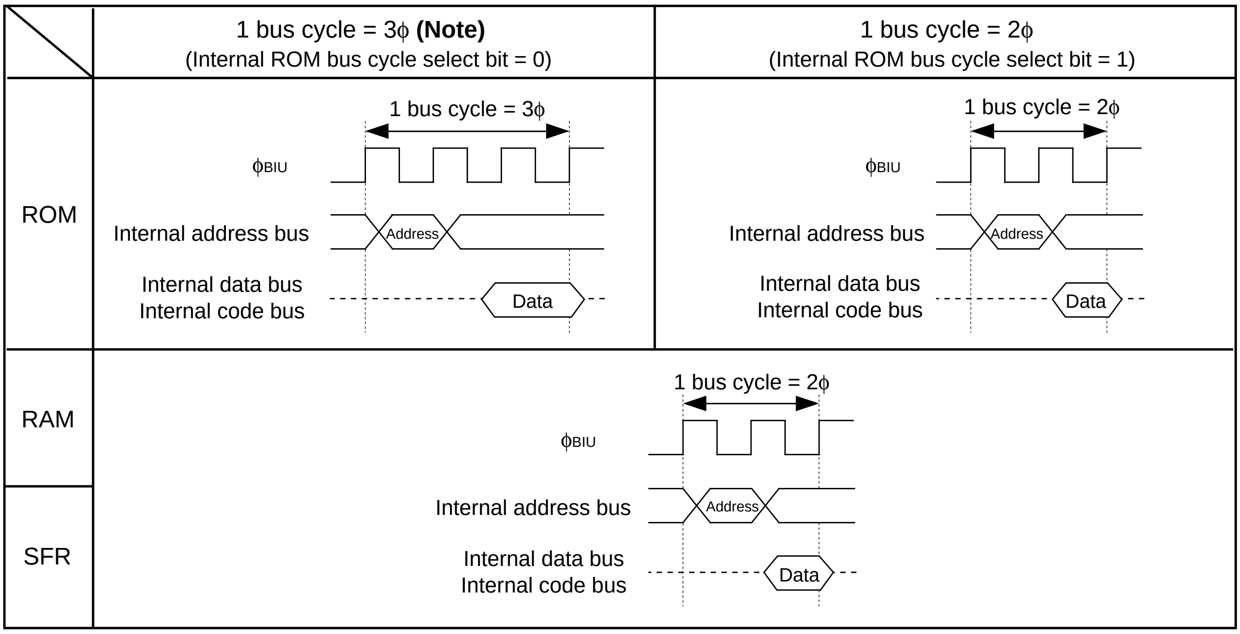


Note: When the voltage level at pin BYTE = “L”, functions as pins D_8 to D_{15} are valid. However, when 8-bit width is selected as the external bus width by the chip select wait controller, the functions as pins D_8 to D_{15} and $\overline{BHW}$ become invalid. (D_8 to D_{15} = floating, $\overline{BHW}$ = “H” output.) When the voltage level at pin BYTE = “H”, these pins function as programmable I/O port (P2, $P3_3$) pins.

Fig. 16 Bus cycle waveform example for data access (access to external area) (3)

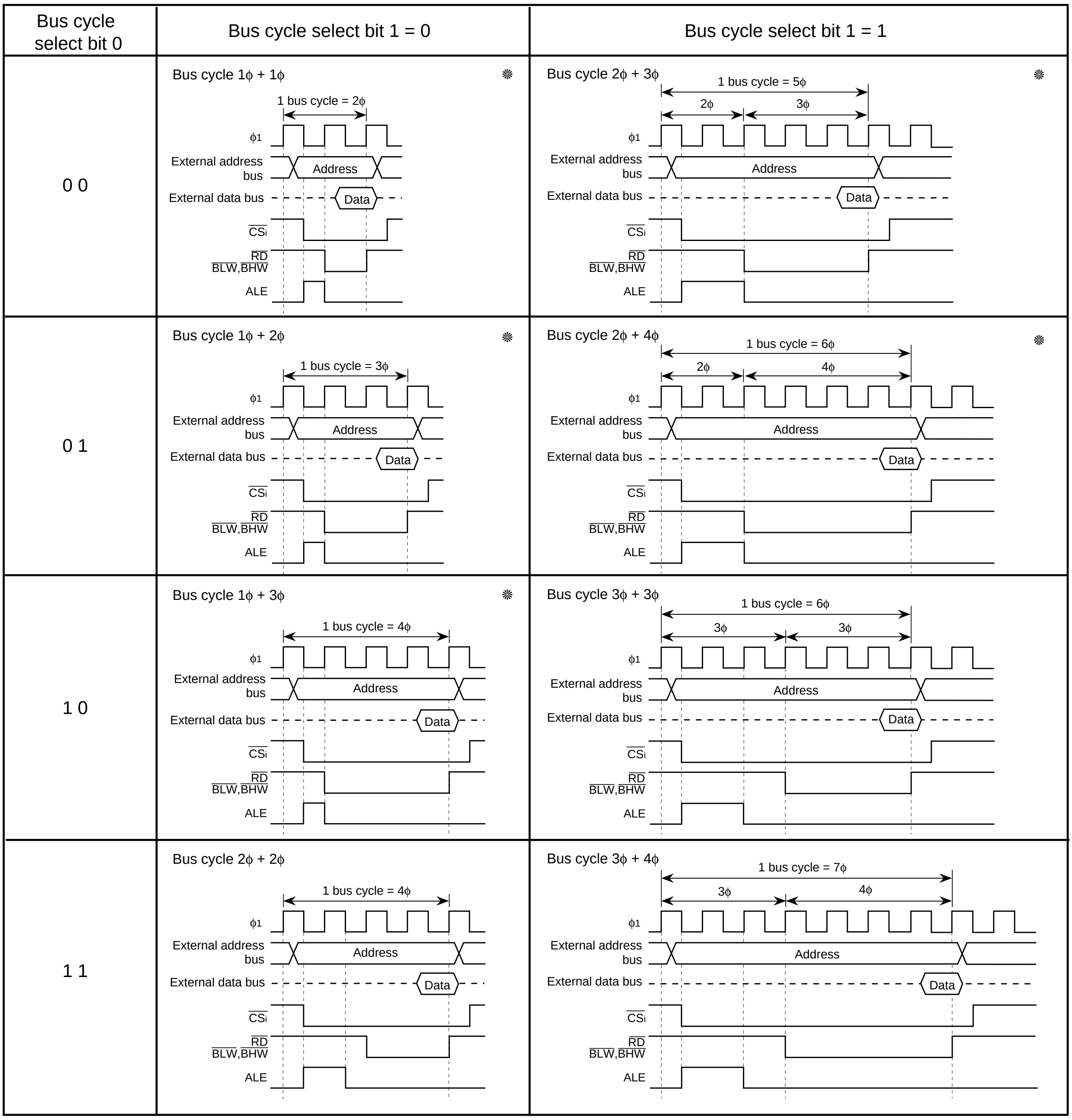
● Number of bus cycles

Figure 17 shows the bus cycle waveform at access to the internal area. Bit 7 of the processor mode register 1 (address $5F_{16}$) selects the number of bus cycles for the internal ROM: 3ϕ or 2ϕ. (This bit 7 is the internal ROM bus cycle select bit.) The internal RAM, SFRs (internal peripheral devices' control registers) are always accessed with 1 bus cycle = 2ϕ. Figure 18 shows the bus cycle waveform at access to the external area. The bus cycle select bits 0, 1 (See the note in Figure 18.) select the number of the bus cycles for each $\overline{CS_i}$ area from 8 types of numbers.



Note: When reprogramming the internal flash memory in the CPU reprogramming mode, select the bus cycle = 3ϕ.

Fig. 17 Bus cycle waveform at access to internal area



Notes 1: The bus cycle type is determined by the following bits:

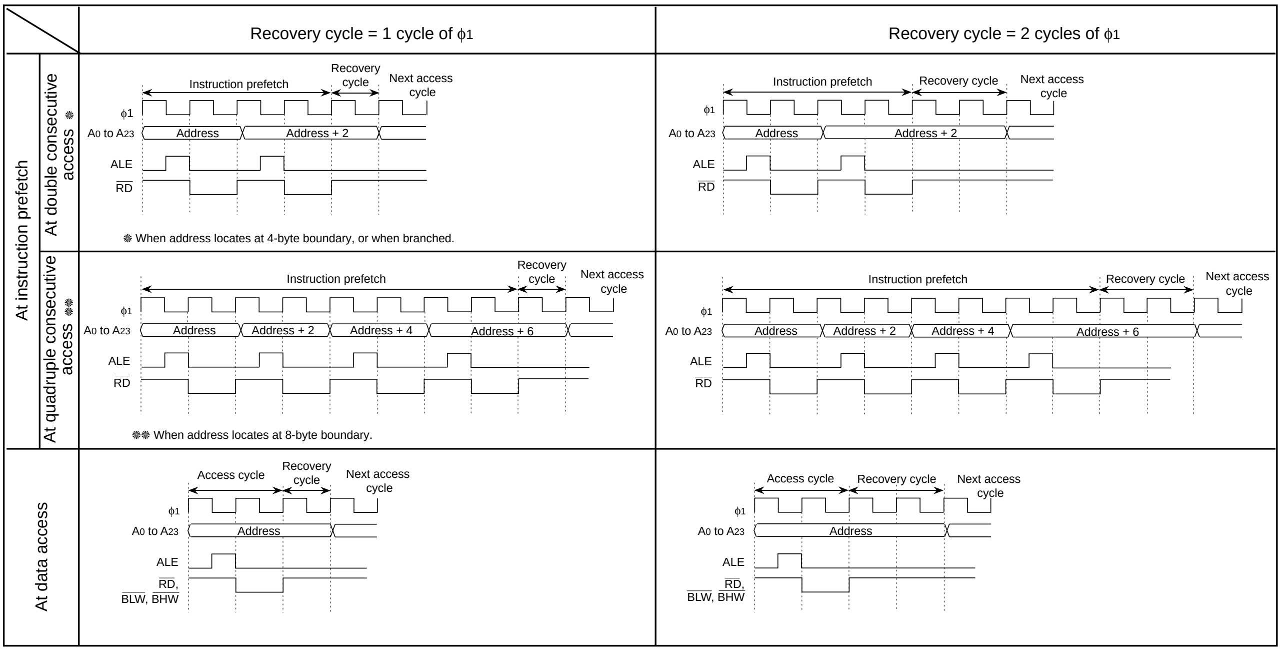
- Areas out of area $\overline{\text{CS}_\text{i}}$: external bus cycle select bit 0 (bits 2 and 3 at address $5E_{16}$)
  external bus cycle select bit 1 (bit 0 at address $5F_{16}$)
- Area $\overline{\text{CS}_\text{i}}$: area $\overline{\text{CS}_\text{i}}$ bus cycle select bit 0 (bits 0 and 1 at addresses 80_{16}, 82_{16}, 84_{16}, 86_{16})
  area $\overline{\text{CS}_\text{i}}$ bus cycle select bit 1 (bit 3 at addresses 81_{16}, 83_{16}, 85_{16}, 87_{16})

2: ❋ indicates the bus cycle, where the burst ROM access specification is enabled.

Fig. 18 Bus cycle types at access to external area

● Recovery cycle

A recovery cycle which is equivalent to 1 or 2 cycles of ϕ_1 can be inserted after each area $\overline{CS_i}$'s access cycle. Whether the recovery cycle is inserted or not is determined by the recovery cycle insert select bit of each $\overline{CS_i}$ control register L (bit 6 at addresses 80$_{16}$, 82$_{16}$, 84$_{16}$, 86$_{16}$). Also, the number of the recovery cycles is selected by the recovery-cycle-insert-number select bit of the processor mode register 1 (bit 6 at address 5F$_{16}$). Figure 19 shows a waveform example when a recovery cycle is inserted.


Recovery cycle = 1 cycle of ϕ1
Recovery cycle = 2 cycles of ϕ1
At instruction prefetch
At double consecutive access ✻
At quadruple consecutive access ✻✻
At data access
Instruction prefetch
Recovery cycle
Next access cycle
Access cycle
ϕ1
A0 to A23
ALE
RD
RD, BLW, BHW
Address
Address + 2
Address + 4
Address + 6
✻ When address locates at 4-byte boundary, or when branched.
✻✻ When address locates at 8-byte boundary.

Notes 1: The recovery cycle insert is specified by the recovery cycle insert select bit and the recovery-cycle-insert-number select bit (bits 4 and 6 at address 5F$_{16}$). Recovery cycle insertion is valid only at access to area $\overline{CS_i}$.
2: The above is applied when 1 bus cycle = 2ϕ.

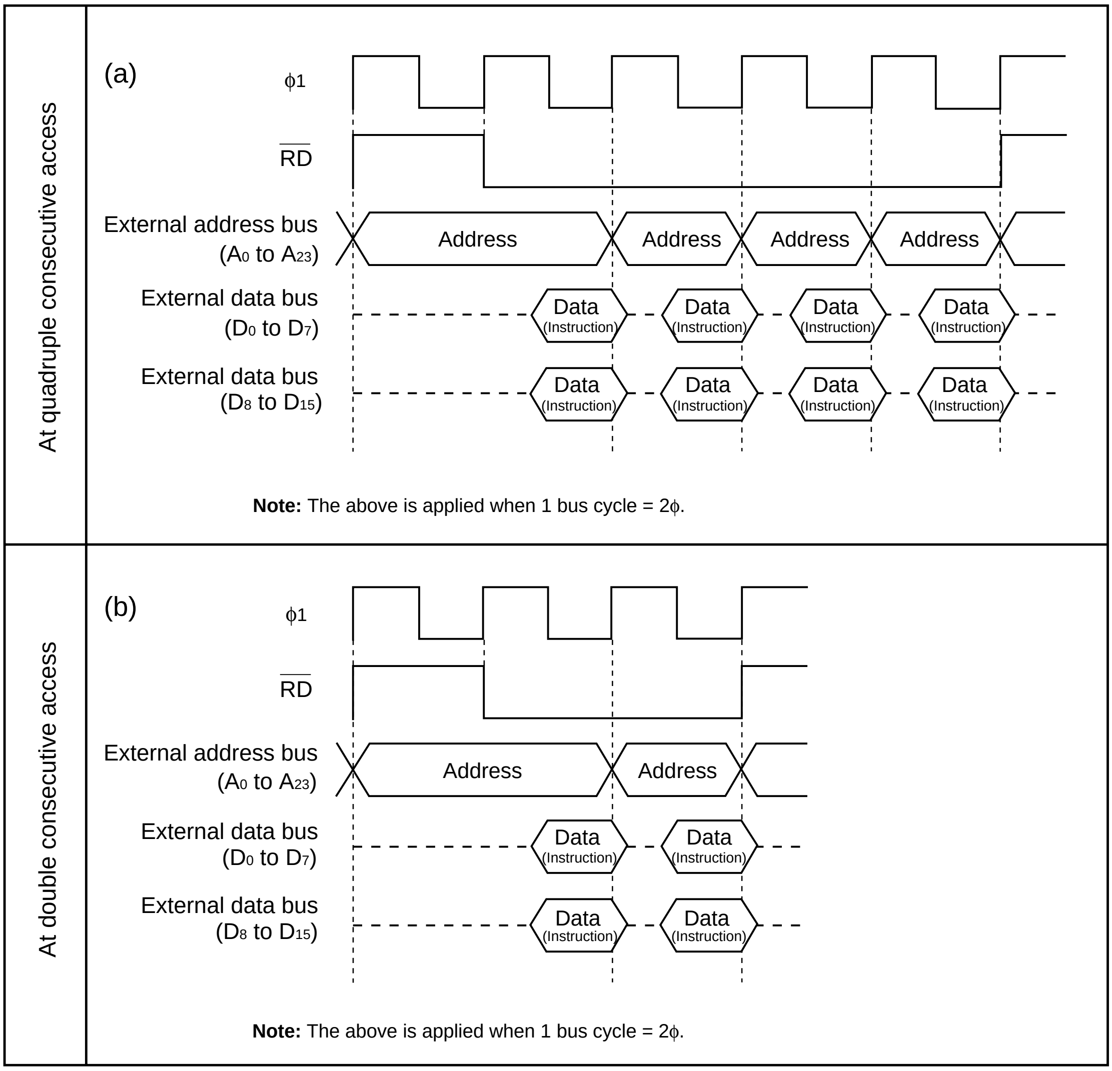
Fig. 19 Waveform example when recovery cycle is inserted

● Burst ROM access

When ROM supporting the burst ROM access has been allocated to area $\overline{CS_i}$, the burst ROM access can be specified. The burst ROM access is specified by each burst ROM access select bit of the $\overline{CS_i}$ control register L (bit 5 at addresses 80_{16}, 82_{16}, 84_{16}, 86_{16}). The burst ROM access is valid only when the external data bus width = 16 bits with an instruction prefetched. In the other cases, the normal access is performed regardless of the contents of the burst ROM access select bit. The burst ROM access can be specified only in the case of ❋ in Figure 18.

Figure 20 shows a waveform example at burst ROM access.

When an instruction is prefetched from the burst ROM, 8 bytes are fetched starting from an 8-byte boundary (the low-order 3 bits of address, A_2, A_1, A_0 = "000") in waveform (a). When branched, regardless of the 8-byte boundary of the branch destination address, access starting from the 4-byte boundary (the low-order 2 bits of address, A_1, A_0 = "00") is performed in waveform (b). Once the 8-byte boundary has been selected, instructions will be prefetched in waveform (a) until a branch.


At quadruple consecutive access
(a)
φ1
RD
External address bus (A0 to A23)
Address
Address
Address
Address
External data bus (D0 to D7)
Data (Instruction)
External data bus (D8 to D15)
Data (Instruction)
At double consecutive access
(b)

Note: The above is applied when 1 bus cycle = 2φ.

Note: The above is applied when 1 bus cycle = 2φ.

Notes 1: The burst ROM access is selected by the burst ROM access select bit (bit 5 at addresses 80_{16}, 82_{16}, 84_{16}, 86_{16}).
2: The burst ROM access can be selected only in the case of ❋ in Figure 18.

Fig. 20 Waveform example at burst ROM access

● Address output selection

As shown in Figure 21, the unnecessary state change of address output pins (A_0 to A_{23}) can be avoided, without outputting an address at access to the internal area.

When the address output select bit of the particular function select register 1 (bit 4 at address 63_{16}) is set to “1”, an address is output only at access to the external area. Also, at access to the internal area, the address at the preceding access to the external area is retained. The address output start timing in this case is the half cycle of ϕ1 later than that at the normal access (when the address output select bit = “0”). For the bit structure of the particular function select register 1, refer to the section on the standby function.

Also, at the normal access, an address is output at both of the access to the internal and external areas.

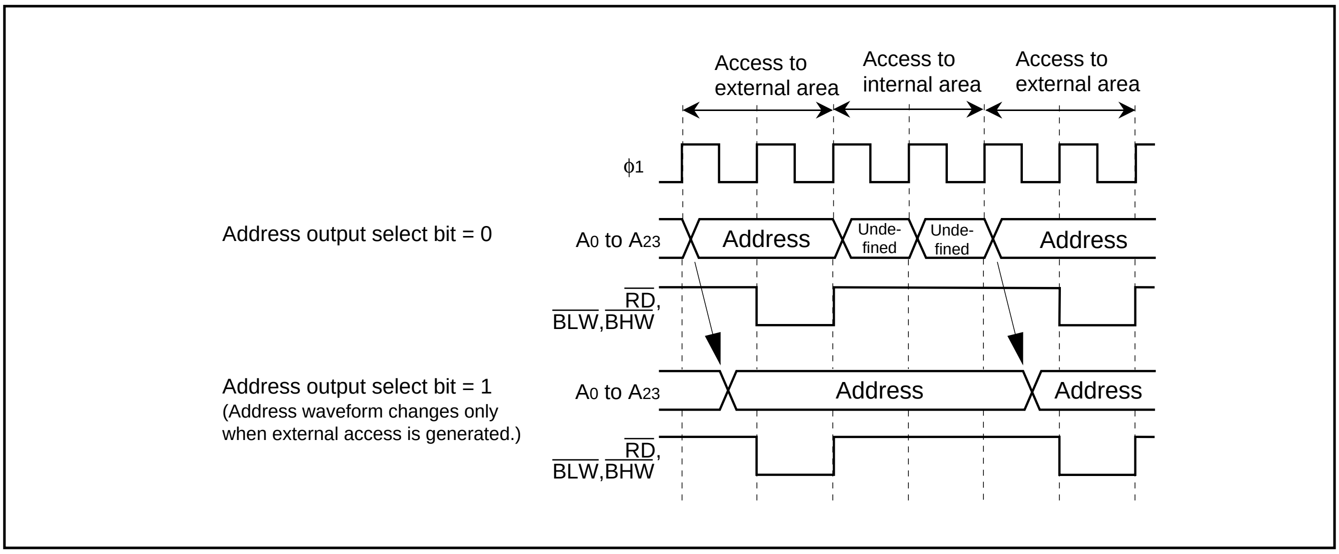
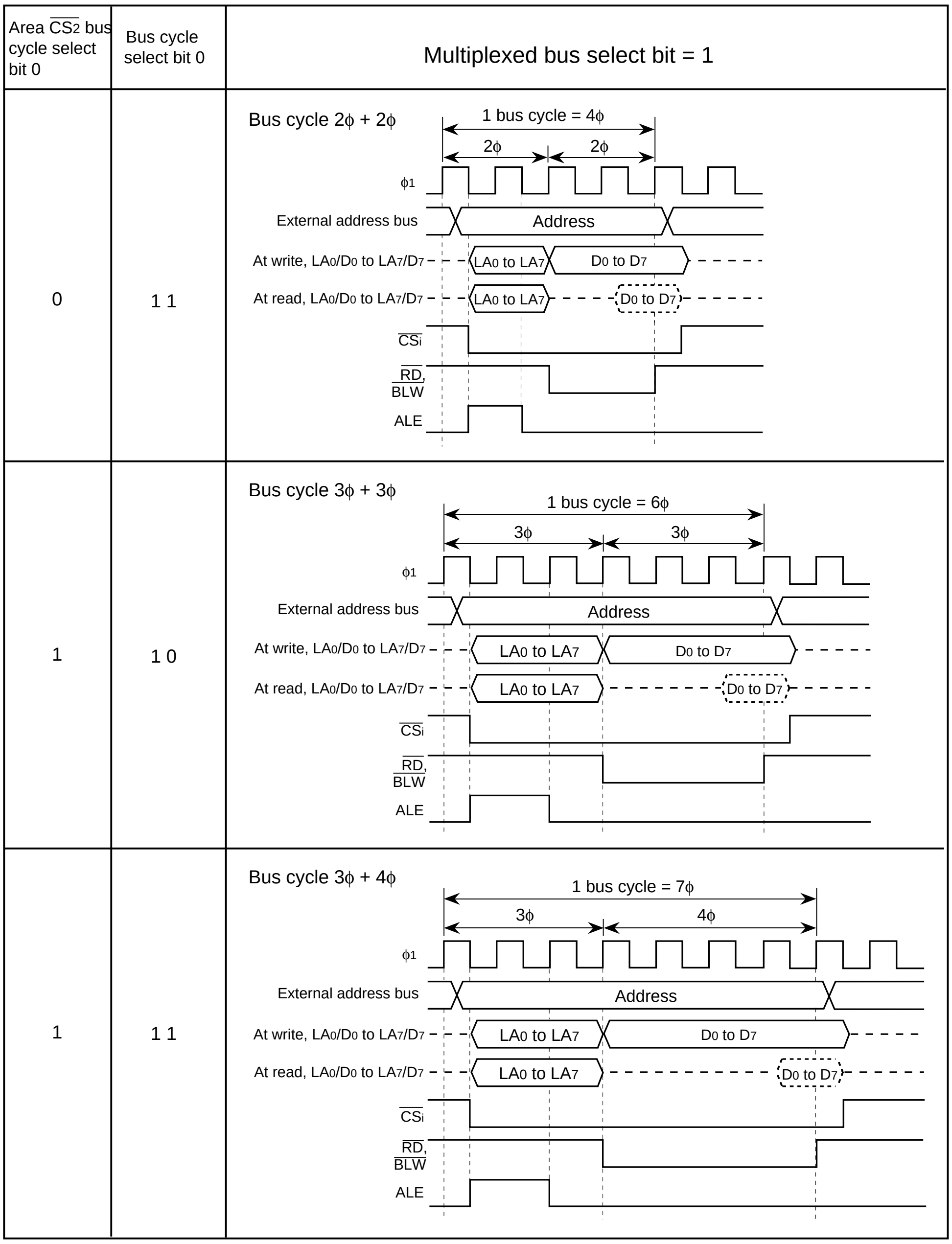


Fig. 21 Waveform example depending on address output function selection

● Area multiplication

When area $\overline{\text{CS}_2}$'s external data bus width = 8 bits with the multiplexed bus select bit of the $\overline{\text{CS}_2}$ control register H (bit 5 at address 85_{16}) = "1", the external bus type can be changed to the multiplexed bus type only at access to area $\overline{\text{CS}_2}$. In this case, the low-order 8 bits of an address (LA_0 to LA_7) are output, and the low-order 8 bits of data (D_0 to D_7) are input/output with the time-sharing method, respectively.

Figure 22 shows a waveform example of area multiplication for each bus cycle. Do not select the area multiplication function for a bus cycle not shown in Figure 22.



Notes 1: The number of bus cycles is determined by the following bits:
- Area $\overline{\text{CS}_2}$ bus cycle select bit 0 (bits 0 and 1 at address 84_{16})
- Area $\overline{\text{CS}_2}$ bus cycle select bit 1 (bit 3 at address 85_{16})

Area multiplication is specified by the multiplexed bus select bit (bit 5 at address 85_{16}).

2: Do not select the area multiplication function for a bus cycle not shown in Figure 22.

Fig. 22 Waveform example of area multiplication for each bus cycle

PROCESSOR MODES

Any of the three processor modes (single-chip mode, memory expansion mode, microprocessor mode) can be selected with the following:

- Processor mode bits of the processor mode register 0 (bits 1 and 0 at address $5E_{16}$; Figure 24)
- Voltage level applied to pin MD0

Table 3 lists the selection method of a processor mode.

The memory map which the CPU can access depends on the selected processor mode. Figure 23 shows the memory maps in three processor modes.

Also, the functions of ports P0 to P4, P10, P11 depend on the selected processor mode. For details, see Tables 5 and 6.

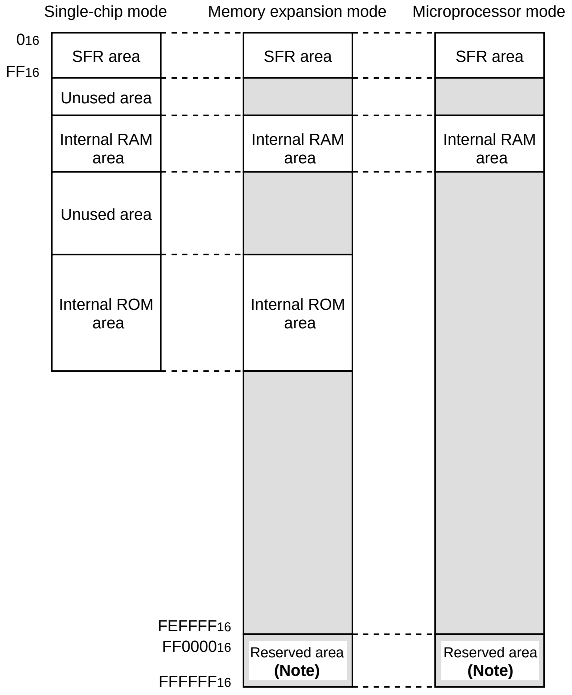
Figures 24 to 26 show the bit configurations of the processor mode registers 0, 1, and port function control register.

In the single-chip mode, ports P0 to P4, P10, P11 function as I/O ports. (While the internal peripheral devices are used, these ports function as these devices' I/O pins.) In this mode, only the internal area (SFRs, internal RAM, internal ROM) is accessible.

In the memory expansion and microprocessor modes, external devices assigned in the external memory area can be connected via buses. Therefore, ports P0 to P4, P10, P11 function as I/O pins for the address bus, data bus, bus control signals. (Some port functions are selectable.) Table 4 lists each bus control signal's function.

In the memory expansion mode, all of the internal area (SFRs, internal RAM, internal ROM) and external area are accessible. In the microprocessor mode, the internal area except for the internal ROM (in other words, SFRs and internal RAM) and the external area are accessible.

Note that, when the external devices are located to an area where the internal area and external area overlap, only the internal area can be read/written; the external area cannot be read/written.



SFR area : Internal peripheral devices' control registers are allocated here.

▨ External area : Access to this area enables the access to the devices which are connected with the external.

Note: Do not access this area (bank FF_{16}).

Fig. 23 Memory maps in three processor modes

Table 3. Selection method of processor mode

MD0	Processor mode bits (Note 1)	Processor mode	Description
V_{SS}	00	Single-chip mode	After reset is removed, the single-chip mode is selected. By changing the processor mode bits' contents by software, the memory expansion mode or microprocessor mode can be selected.
	01	Memory expansion mode	
	10	Microprocessor mode	
V_{CC}	10 **(Note 2)**	Microprocessor mode	After reset is removed, the microprocessor mode is selected.

Notes 1: Processor mode bits = bits 0 and 1 of the processor mode register 0 (address $5E_{16}$)

2: While the Vcc level voltage is applied to pin MD0, the processor mode bits are fixed to "10".

Table 4. Each bus control signal's function

Signal	I/O	Function	Remarks
$\overline{RD}$	Output	Read signal. Outputs "L" at read from the external area.	
$\overline{BLW}$ $\overline{BHW}$	Output	Write signal. Outputs "L" at write to the external area.	For operation differences between $\overline{BLW}$ and $\overline{BHW}$ depending on the external data bus width, see Table 5.
ALE	Output	Address latch enable signal. Outputs "H" level pulse in the period just before signals $\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$ become "L". This is used to latch an address in an external circuit.	In order to latch an address with signal ALE, do as follows: • While ALE = "H", be sure to open a latch, so the address will pass it. • While ALE = "L", be sure to hold the address.
ϕ_1	Output	Internal standard clock's output. Outputs system clock (f_{sys}).	
$\overline{RDY}$	Input	Ready signal. The "L" level period of the last ϕ_1 in the access cycle for the external area (in other words, "L" level period of $\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$) will be extended while "L" level voltage is applied to this pin.	
$\overline{HOLD}$	Input	Hold request signal. Appliance of "L" level voltage will generate a hold request; appliance of "H" level voltage will request to terminate the hold state.	Acceptance and termination of a hold request is performed at completion of the bus cycle while the BIU operates. In the hold state, A_0–A_{23}, D_0–D_{15}, $\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$, ALE, $\overline{CS_0}$–$\overline{CS_3}$ enter the floating state. At termination of the hold state, simultaneously with the timing when $\overline{HLDA}$ becomes "H" level, the above floating state is terminated. Then, bus access will be restarted 1 cycle of ϕ_1 after. In the hold state, also, the CPU operates with access to the internal area. If the CPU accesses the external area, in the hold state, the CPU stops its operation.
$\overline{HDLA}$	Output	Hold acknowledge signal. Outputs "L" in the hold state.	
$\overline{CS_0}$–$\overline{CS_3}$	Output	Chip select signal. Outputs "L" in access to the specified chip select area.	For details, refer to the section on the chip select wait controller.
BYTE	Input	Input signal to select the external data bus width. When this pin's level = Vss, 16-bit width will be selected; and when Vcc, 8-bit width will be selected.	When BYTE = Vss level, by the register setting, each chip select area ($\overline{CS_1}$ to $\overline{CS_3}$) can have the 8-bit data bus, independently. For details, refer to the section on the chip select wait controller.

Table 5. Relationship between processor modes, memory area, and port function (1)

		Single-chip mode	Memory expansion mode	Microprocessor mode
Mode **(Note 1)**	Pin MD0	V_{SS} level voltage is applied	V_{SS} level voltage is applied	V_{CC} level voltage is applied
	Processor mode bits **(Note 2)**	00	01	10
Memory area	SFR area	SFR area	SFR area	SFR area
	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area
	Internal ROM area	Internal ROM area	Internal ROM area	External memory area
	Other area	(Do not access.)	External memory area	External memory area
Port pins $P10_0$ to $P10_7$		I/O port pins $P10_0$ to $P10_7$	Low-order address (A_0 to A_7) is output.	Low-order address (A_0 to A_7) is output.
Port pins $P11_0$ to $P11_7$		I/O port pins $P11_0$ to $P11_7$	Middle-order address (A_8 to A_{15}) is output. I/O port pins $P11_0$ to $P11_7$ **(Note 3)**	Middle-order address (A_8 to A_{15}) is output. I/O port pins $P11_0$ to $P11_7$ **(Note 3)**
Port pins $P0_0$ to $P0_7$		I/O port pins $P0_0$ to $P0_7$	High-order address (A_{16} to A_{23}) is output. I/O port pins $P0_0$ to $P0_7$ **(Note 3)**	High-order address (A_{16} to A_{23}) is output. I/O port pins $P0_0$ to $P0_7$ **(Note 3)**
Port pins $P1_0$ to $P1_7$	External data bus width = 16 bits	I/O port pins $P1_0$ to $P1_7$	Low-order data (D_0 to D_7, data at even address) is input/output.	Low-order data (D_0 to D_7, data at even address) is input/output.
	External data bus width = 8 bits		Low-order data (D_0 to D_7, data at even/odd address) is input/output. Low-order address (LA_0 to LA_7) is output. Low-order data (D_0 to D_7, data at even/odd address) is input/output **(Note 4)**.	Low-order data (D_0 to D_7, data at even/odd address) is input/output. Low-order address (LA_0 to LA_7) is output. Low-order data (D_0 to D_7, data at even/odd address) is input/output **(Note 4)**.
Port pins $P2_0$ to $P2_7$	External data bus width = 16 bits	I/O port pins $P2_0$ to $P2_7$	High-order data (D_8 to D_{15}, data at odd address) is input/output.	High-order data (D_8 to D_{15}, data at odd address) is input/output.
	External data bus width = 8 bits		I/O port pins $P2_0$ to $P2_7$ **(Note 5)**	I/O port pins $P2_0$ to $P2_7$ **(Note 5)**
Port pin $P3_0$		I/O port pin $P3_0$	I/O port pin $P3_0$ Ready signal $\overline{RDY}$ is input **(Note 6).**	Ready signal $\overline{RDY}$ is input. I/O port pin $P3_0$ **(Note 6)**
Port pin $P3_1$		I/O port pin $P3_1$	Read signal $\overline{RD}$ is output.	Read signal $\overline{RD}$ is output
Port pin $P3_2$	External data bus width = 16 bits	I/O port pin $P3_2$	Write signal $\overline{BLW}$ (write to even address) is output.	Write signal $\overline{BLW}$ (write to even address) is output.
	External data bus width = 8 bits		Write signal $\overline{BLW}$ (write to even/odd address) is output.	Write signal $\overline{BLW}$ (write to even/odd address) is output.
Port pin $P3_3$	External data bus width = 16 bits	I/O port pin $P3_3$	Write signal $\overline{BHW}$ (write to odd address) is output.	Write signal $\overline{BHW}$ (write to odd address) is output.
	External data bus width = 8 bits		I/O port pin $P3_3$ **(Note 5)**	I/O port pin $P3_3$ **(Note 5)**

Table 6. Relationship between processor modes, memory area, and port function (2)

	Single-chip mode	Memory expansion mode	Microprocessor mode
Port pin $P4_0$	I/O port pin $P4_0$	I/O port pin $P4_0$	Address latch enable signal ALE is output.
		Address latch enable signal ALE is output **(Note 6)**.	I/O port pin $P4_0$ **(Note 6)**
Port pin $P4_1$	I/O port pin $P4_1$	I/O port pin $P4_1$	Clock ϕ_1 is output.
	Clock ϕ_1 is output **(Note 6).**	Clock ϕ_1 is output **(Note 6)**.	I/O port pin $P4_1$ **(Note 6)**
Port pin $P4_2$	I/O port pin $P4_2$	I/O port pin $P4_2$	Hold acknowledge signal $\overline{HLDA}$ is output.
		Hold acknowledge signal $\overline{HLDA}$ is output **(Note 6)**.	I/O port pin $P4_2$ **(Note 6)**
Port pin $P4_3$	I/O port pin $P4_3$	I/O port pin $P4_3$	Hold request signal Signal $\overline{HOLD}$ is input.
		Hold request signal $\overline{HOLD}$ is input **(Note 6)**.	I/O port pin $P4_3$ **(Note 6)**
Port pin $P4_4$	I/O port pin $P4_4$	I/O port pin $P4_4$	Chip select signal $\overline{CS_0}$ is output.
		Chip select signal $\overline{CS_0}$ is output **(Note 7)**.	
Port pins $P4_5$ to $P4_7$	I/O port pins $P4_5$ to $P4_7$	I/O port pins $P4_5$ to $P4_7$	I/O port pin $P4_5$ to $P4_7$
		Chip select signals $\overline{CS_1}$ to $\overline{CS_3}$ are output **(Note 8)**.	Chip select signals $\overline{CS_1}$ to $\overline{CS_3}$ are output **(Note 8)**.

Notes 1: For details of the processor mode setting, see Table 3.

2: Processor mode bits = bits 0 and 1 of the processor mode register 0 (address $5E_{16}$).

3: The middle-order/high-order address output pins in the memory expansion or microprocessor mode can be switched to I/O port pins by the address/port switch select bits of the port function control register (bits 2 to 0 at address 92_{16}).

4: When the external data bus width for the chip select area, $\overline{CS_2}$, has been set to 8 bits, only in the access to area $\overline{CS_2}$, by the multiplexed bus select bit of the $\overline{CS_2}$ control register H (bit 5 at address 85_{16}), a multiplexed bus which performs the following operations with the time-sharing method is realized:

- Output of address LA_0 to LA_7
- Input/Output of data D_0 to D_7

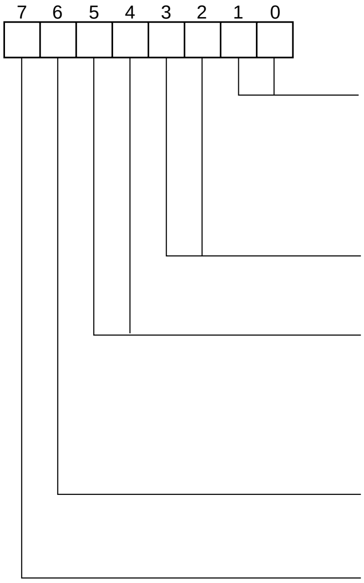
5: When one of areas $\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ is accessed under the following conditions, pins D_8 to D_{15} enter the floating state, and pin $\overline{BHW}$ outputs "H" level. (They do not become I/O port pins.)

- Pin BYTE is at Vss level.
- One of bit 2s at addresses 82_{16}, 84_{16}, 86_{16} (the external data bus width select bit of the $\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ control register L) is set to "1" (external data bus width = 8 bits).

6: In the memory expansion mode, by the corresponding select bits of the processor mode register 0 and 1 (addresses $5E_{16}$, $5F_{16}$), port pins $P3_0$, $P4_0$ to $P4_3$ can operate as pins for $\overline{RDY}$ input, ALE output, ϕ_1 output, $\overline{HLDA}$ output, $\overline{HOLD}$ input, respectively.
In the microprocessor mode, by the above select bits, the above pins ($\overline{RDY}$, ALE, ϕ_1, $\overline{HLDA}$, $\overline{HOLD}$) can operate as port pins $P3_0$, $P4_0$ to $P4_3$, respectively.
In the single-chip mode, port pin $P4_1$ can operate as the ϕ_1 output pin by the above select bits.

7: In the memory expansion mode, port pin $P4_4$ can operate as the $\overline{CS_0}$ output pin by the $\overline{CS_0}$ output select bit of the $\overline{CS_0}$ control register L (bit 7 at address 80_{16}).

8: In the memory expansion and microprocessor modes, port pins $P4_5$ to $P4_7$ can operate as the $\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ output pins by the $\overline{CS_i}$ output select bits (i = 1 to 3) (bit 7s at addresses 82_{16}, 84_{16}, 86_{16}).



Processor mode register 0 — Address $5E_{16}$

Processor mode bits **(Note 1)**
0 0 : Single-chip mode
0 1 : Memory expansion mode
1 0 : Microprocessor mode
1 1 : Do not select.

External bus cycle select bit 0 **(Note 2)**
See Figure 18.

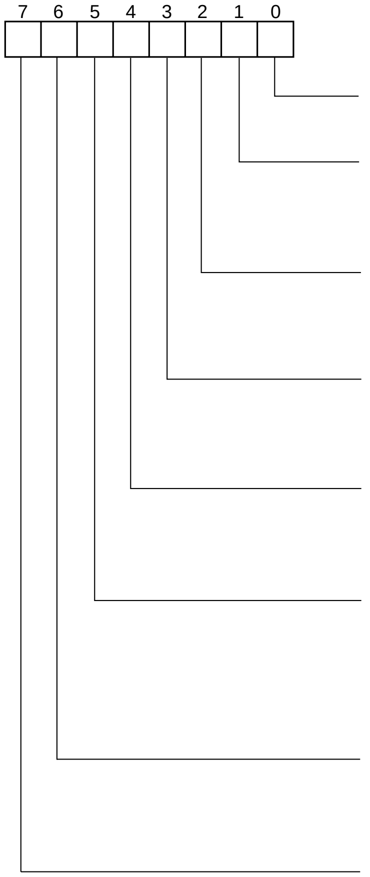
Interrupt priority detection time select bits
0 0 : 7 cycles of f_{sys}
0 1 : 4 cycles of f_{sys}
1 0 : 2 cycles of f_{sys}
1 1 : Do not select.

Software reset bit
By a write of “1” to this bit, the microcomputer will be reset, and then, restarted.

Clock ϕ_1 output select bit **(Note 3)**
0 : ϕ_1 output is disabled. ($P4_1$ functions as an programmable I/O port pin.)
1 : ϕ_1 output is enabled. ($P4_1$ functions as the clock ϕ1 output pin.)

Notes 1: While V_{SS} level voltage is applied to pin MD0, this bit’s state is cleared to “0” at reset. While V_{CC} level voltage is applied to pin MD0, on the other hand, this bit’s state is set to “1” at reset. (Fixed to “1”.)
2: These bits are valid to the external area except for chip select area (area $\overline{CS_i}$). The bus cycle of area $\overline{CS_i}$ is selected by the corresponding area $\overline{CS_i}$ bus cycle select bits 0, 1.
3: While V_{SS} level voltage is applied to pin MD0, this bit’s state is cleared to “0” at reset. While V_{CC} level voltage is applied to pin MD0, on the other hand, this bit’s state is set to “1” at reset.

Fig. 24 Bit configuration of processor mode register 0



Processor mode register 1 Address $5F_{16}$

External bus cycle select bit 1 **(Note 1)**
See Figure 18.

Direct page register switch bit **(Note 2)**
0 : Only DPR0 is used.
1 : DPR0 to DPR3 are used.

$\overline{\text{RDY}}$ input select bit **(Notes 3 to 5)**
0 : $\overline{\text{RDY}}$ input is disabled. ($P3_0$ functions as a programmable I/O port pin.)
1 : $\overline{\text{RDY}}$ input is enabled. ($P3_0$ functions as pin $\overline{\text{RDY}}$.)

ALE output select bit **(Notes 3 and 4)**
0 : ALE output is disabled. ($P4_0$ functions as a programmable I/O port pin.)
1 : ALE output is enabled. ($P4_0$ functions as pin ALE.)

Recovery cycle insert select bit **(Notes 3 and 4)**
0 : No recovery cycle is inserted at access to the external area.
1 : Recovery cycle is inserted at access to the external area.

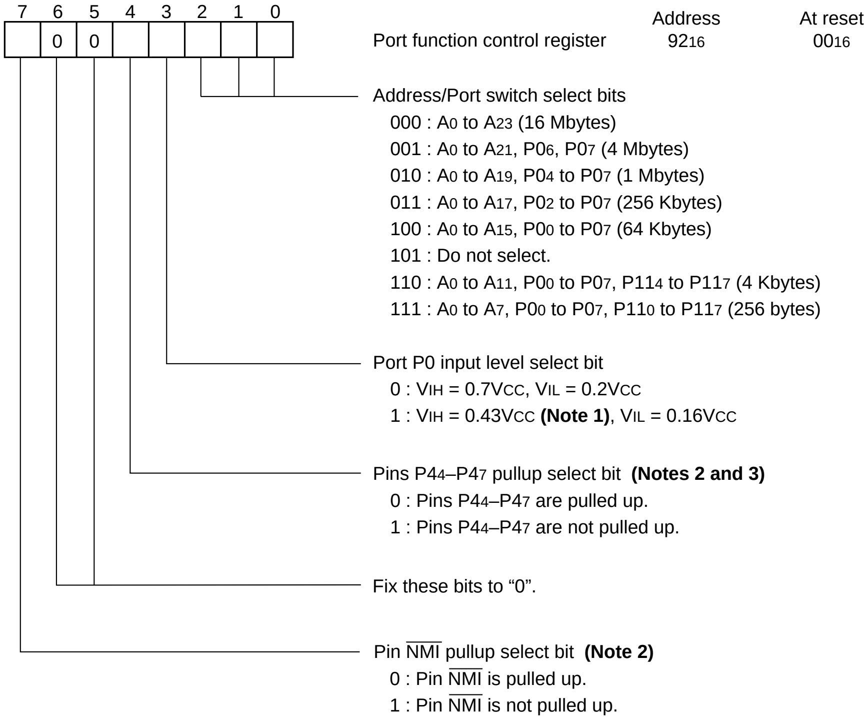
$\overline{\text{HOLD}}$ input, $\overline{\text{HLDA}}$ output select bit **(Notes 3 to 5)**
0 : $\overline{\text{HOLD}}$ input and $\overline{\text{HLDA}}$ output are disabled.
($P4_3$ and $P4_2$ function as programmable I/O port pins.)
1 : $\overline{\text{HOLD}}$ input and $\overline{\text{HLDA}}$ output are enabled.
($P4_3$ and $P4_2$ function as pins $\overline{\text{HOLD}}$ and $\overline{\text{HLDA}}$, respectively.)

Recovery-cycle-insert number select bit **(Note 6)**
0 : 1 cycle
1 : 2 cycles

Internal ROM bus cycle select bit **(Note 7)**
0 : 3ϕ
1 : 2ϕ

Notes 1: This bit is valid to the external area except for chip select areas (area $\overline{\text{CS}}$i), and the bus cycle of area $\overline{\text{CS}}$i is independent of this bit's contents.
The bus cycle of area $\overline{\text{CS}}$i is selected by the corresponding area $\overline{\text{CS}}$i bus cycle select bits 0, 1 (bits 0, 1 at addresses 80_{16}, 82_{16}, 84_{16}, 86_{16}; bit 3 at addresses 81_{16}, 83_{16}, 85_{16}, 87_{16}).
2: After reset, this bit's contents can be switched only once. During the software execution, be sure not to switch this bit's contents.
3: In the single-chip mode, these bits' functions are disabled regardless of these bits' contents.
4: While V_{SS} level voltage is applied to pin MD0, each of these bits is "0" at reset. While V_{CC} level voltage is applied to pin MD0, on the other hand, each of these bits is "1" at reset.
5: In the memory expansion or microprocessor mode, if this bit's contents is switched from "1" to "0", this bit will be cleared to "0". After this clearance, this bit cannot return to "1". If it is necessary to set this bit to "1", be sure to reset the microcomputer.
6: The program which switches this bit's contents must be assigned to the internal area.
7: In the microprocessor mode, this bit is invalid.
When the internal flash memory is reprogrammed in the CPU reprogramming mode, be sure to clear this bit to "0".

Fig. 25 Bit configuration of processor mode register 1



	Address	At reset
Port function control register	92_{16}	00_{16}

Address/Port switch select bits
000 : A_0 to A_{23} (16 Mbytes)
001 : A_0 to A_{21}, $P0_6$, $P0_7$ (4 Mbytes)
010 : A_0 to A_{19}, $P0_4$ to $P0_7$ (1 Mbytes)
011 : A_0 to A_{17}, $P0_2$ to $P0_7$ (256 Kbytes)
100 : A_0 to A_{15}, $P0_0$ to $P0_7$ (64 Kbytes)
101 : Do not select.
110 : A_0 to A_{11}, $P0_0$ to $P0_7$, $P11_4$ to $P11_7$ (4 Kbytes)
111 : A_0 to A_7, $P0_0$ to $P0_7$, $P11_0$ to $P11_7$ (256 bytes)

Port P0 input level select bit
0 : $V_{IH} = 0.7V_{CC}$, $V_{IL} = 0.2V_{CC}$
1 : $V_{IH} = 0.43V_{CC}$ **(Note 1)**, $V_{IL} = 0.16V_{CC}$

Pins $P4_4$–$P4_7$ pullup select bit **(Notes 2 and 3)**
0 : Pins $P4_4$–$P4_7$ are pulled up.
1 : Pins $P4_4$–$P4_7$ are not pulled up.

Fix these bits to “0”.

Pin $\overline{\text{NMI}}$ pullup select bit **(Note 2)**
0 : Pin $\overline{\text{NMI}}$ is pulled up.
1 : Pin $\overline{\text{NMI}}$ is not pulled up.

Notes 1: For the M37902FxM (power source voltage = 3.3 V±0.3 V), $V_{IH} = 0.5V_{CC}$.
2: When MD1 = V_{CC} and MD0 = V_{CC} (flash memory parallel I/O mode), pins $P4_4$ to $P4_7$ and $\overline{\text{NMI}}$ are not pulled up, regardless of these bits’ contents.
3: When MD1 = V_{SS} and MD0 = V_{CC} (microprocessor mode), pin $\overline{\text{CS}_0}$ ($P4_4$) is not pulled up, regardless of the bit’s contents.

Fig. 26 Bit configuration of port function control register

Chip select wait controller

By the control of the chip select wait controller (CSWC), the chip select function for the maximum of 4 blocks can be set at the bus access to the external area.

Also, by the setting of the CSWC, port pins $P4_4$ to $P4_7$ can operate as chip select output pins ($\overline{CS_0}$ to $\overline{CS_3}$).

Figure 27 shows a chip select output waveform example.

This chip select function determines the following items of the chip select area: start address, address's block size, wait number, external data bus width, $\overline{RDY}$ control validity, burst ROM specification, recovery cycle insertion validity, and area multiplication validity.

For the external area except for areas $\overline{CS_0}$ to $\overline{CS_3}$, the processor mode registers 0, 1 determine the above items. After reset is removed, when the microcomputer starts it's operation in the microprocessor mode, area $\overline{CS_0}$ is automatically selected.

Table 7 lists the function of areas $\overline{CS_0}$ to $\overline{CS_3}$.

Figure 28 shows the bit configuration of the $\overline{CS_0}$/$\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ control register Ls. These registers determine the following items of a device to be connected: wait number, external data bus width (**Note:** The external data bus width of area $\overline{CS_0}$ is determined by pin BYTE's level.), $\overline{RDY}$ control validity, burst ROM access specification, recovery cycle insertion validity, and output validity of $\overline{CS_0}$ to $\overline{CS_3}$.

Figure 29 shows the bit configuration of the $\overline{CS_0}$/$\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ control register Hs. These registers determine block size, etc. of an external area to be connected. For areas $\overline{CS_0}$ to $\overline{CS_2}$, by selecting mode 1 with the area $\overline{CS_k}$ setting mode select bit, an chip select area can be set to the external area in bank 0.

Figures 30 shows the bit configuration of the area $\overline{CS_0}$/$\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ start address registers. For details of these addresses' setting, see Figures 31 to 33.

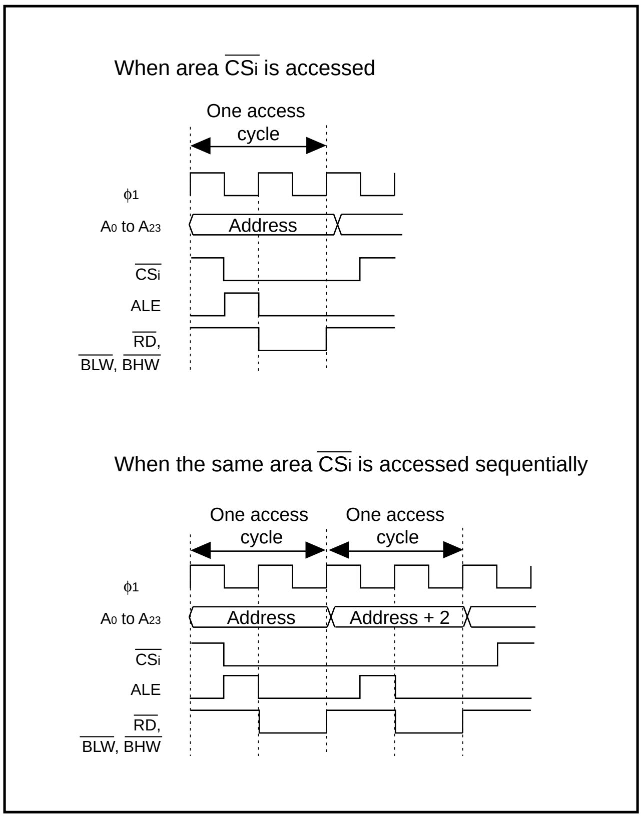

When area $\overline{CS_i}$ is accessed
One access cycle
ϕ_1
A_0 to A_{23}
Address
$\overline{CS_i}$
ALE
$\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$
When the same area $\overline{CS_i}$ is accessed sequentially
One access cycle
One access cycle
ϕ_1
A_0 to A_{23}
Address
Address + 2
$\overline{CS_i}$
ALE
$\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$

Fig. 27 Chip select output waveform example

Table 7. Function of areas $\overline{\text{CS}_0}$ to $\overline{\text{CS}_3}$

	$\overline{\text{CS}_0}$		$\overline{\text{CS}_1}$, $\overline{\text{CS}_2}$		$\overline{\text{CS}_3}$	External area except for $\overline{\text{CS}_0}$ to $\overline{\text{CS}_3}$
	Mode 0	Mode 1	Mode 0	Mode 1		
Space where start address can be set	Banks 2_{16} to FE_{16}	Bank 0_{16}	Banks 2_{16} to FE_{16}	Bank 0_{16}	Banks 2_{16} to FE_{16}	
Block size	128 Kbytes, 256 Kbytes, 512 Kbytes, 1 Mbytes, 2 Mbytes, 4 Mbytes, or 8 Mbytes	128 Kbytes, 256 Kbytes, 512 Kbytes, 1 Mbytes, 2 Mbytes, 4 Mbytes, or 8 Mbytes	128 Kbytes, 256 Kbytes, 512 Kbytes, 1 Mbytes, 2 Mbytes, 4 Mbytes, or 8 Mbytes	4 Kbytes or 8 Kbytes	128 Kbytes, 256 Kbytes, 512 Kbytes, 1 Mbytes, 2 Mbytes, 4 Mbytes, or 8 Mbytes	
Bus cycle	Bus cycle: •1φ + 1φ •1φ + 2φ •1φ + 3φ •2φ + 2φ •2φ + 3φ •2φ + 4φ •3φ + 3φ •3φ + 4φ (Selected by bits 0, 1 at address 80_{16} and bit 3 at address 81_{16}.)		Bus cycle: •1φ + 1φ •1φ + 2φ •1φ + 3φ •2φ + 2φ •2φ + 3φ •2φ + 4φ •3φ + 3φ •3φ + 4φ (Selected by bits 0, 1 at addresses 82_{16}, 84_{16} and bit 3 at addresses 83_{16}, 85_{16}.)		Bus cycle: •1φ + 1φ •1φ + 2φ •1φ + 3φ •2φ + 2φ •2φ + 3φ •2φ + 4φ •3φ + 3φ •3φ + 4φ (Selected by bits 0, 1 at address 86_{16} and bit 3 at address 87_{16}.)	Bus cycle: •1φ + 1φ •1φ + 2φ •1φ + 3φ •2φ + 2φ •2φ + 3φ •2φ + 4φ •3φ + 3φ •3φ + 4φ (Selected by bits 2, 3 at address 5E_{16} and bit 0 at address 5F_{16}.)
External data bus width	Determined by pin BYTE’s level.		When BYTE = V_{SS} level, 8-bit width or 16-bit width can be selected arbitrary **(Note 1)**. (Selected by bit 2 at addresses 82_{16}, 84_{16}.)		When BYTE = V_{SS} level, 8-bit width or 16-bit width can be selected arbitrary **(Note 1)**. (Selected by bit 2 at address 86_{16}.)	Determined by pin BYTE’s level
$\overline{\text{RDY}}$ control	Valid (Selected by bit 2 at address 5F_{16} and bit 3 at address 80_{16}.)		Valid (Selected by bit 2 at address 5F_{16} and bit 3 at addresses 82_{16}, 84_{16}.)		Valid (Selected by bit 2 at address 5F_{16} and bit 3 at address 86_{16}.)	Valid (Selected by bit 2 at address 5F_{16}.)
Burst ROM access **(Notes 2, 3)**	Available.		Available.		Available.	Not available.
Recovery cycle insertion	Available.		Available.		Available.	Available.
Area multiplexed bus access **(Note 3)**	Not available.		$\overline{\text{CS}_1}$: Not available. $\overline{\text{CS}_2}$: Available. **(Note 4)**		Not available.	Not available.
Address output selection **(Note 5)**	Available.		Available.		Available.	Available.

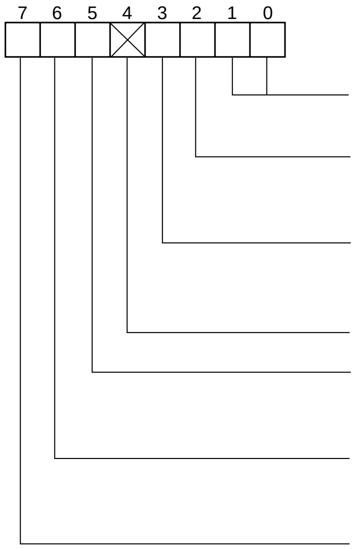
Notes 1: When BYTE = Vcc level, the external data bus width is fixed to 8 bits.

2: Burst ROM access is valid only when the external data bus width is 16 bits at instruction prefetch.

3: Burst ROM access and area multiplexed bus access cannot be used at the same time.

4: Valid only when area $\overline{\text{CS}_2}$ is accessed with the 8-bit external data bus width.

5: Selected by the address output select bit (bit 4 at address 63_{16}). The address output selection for each area is not available.



	Address	At reset
$\overline{CS_0}$ control register L	80_{16}	42_{16}

Area $\overline{CS_0}$ bus cycle select bit 0
See Figure 18.

External data bus width select bit **(Note 1)**
0 : 16-bit width
1 : 8-bit width

$\overline{RDY}$ control bit **(Note 2)**
0 : $\overline{RDY}$ control is valid.
1 : $\overline{RDY}$ control is invalid.

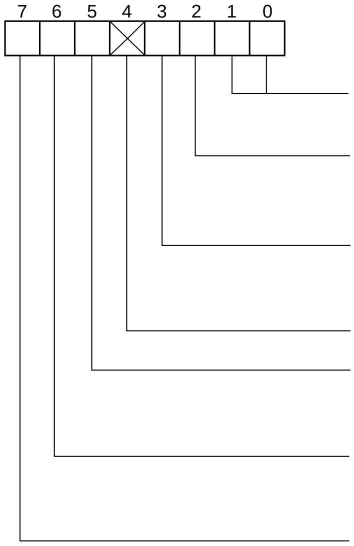
“0” at read.

Burst ROM access select bit **(Note 3)**
0 : Normal access
1 : Burst ROM access

Recovery cycle insert select bit
0 : No recovery cycle is inserted at access to area $\overline{CS_0}$.
1 : Recovery cycle is inserted at access to area $\overline{CS_0}$.

$\overline{CS_0}$ output select bit **(Notes 4, 5)**
0 : $\overline{CS_0}$ output is disabled. ($P4_4$ functions as a programmble I/O port pin.)
1 : $\overline{CS_0}$ output is enabled. ($P4_4$ functions as pin $\overline{CS_0}$.)

Notes 1: While V_{SS} level voltage is applied to pin BYTE, this bit’s state is cleared to “0” at reset. While V_{CC} level voltage is applied to pin BYTE, on the other hand, this bit’s state is set to “1” at reset.
2: This bit is valid when the $\overline{RDY}$ input select bit (bit 2 at address $5F_{16}$) = “1”.
3: While V_{CC} level voltage is applied to pin BYTE, the normal access is selected regardless of this bit’s contents.
4: In the single-chip mode, this bit’s contents are invalid. ($\overline{CS_0}$ output is disabled.)
5: While V_{SS} level voltage is applied to pin MD0, this bit’s state is cleared to “0” at reset. While V_{CC} level voltage is applied to pin MD0, on the other hand, this bit’s state is set to “1” at reset. (Fixed to “1”.)



	Address	At reset
$\overline{CS_1}$ control register L	82_{16}	42_{16}
$\overline{CS_2}$ control register L	84_{16}	42_{16}
$\overline{CS_3}$ control register L	86_{16}	42_{16}

Area $\overline{CSj}$ bus cycle select bit 0 (j = 1 to 3)
See Figure 18.

External data bus width select bit
0 : 16-bit width
1 : 8-bit width **(Note 1)**

$\overline{RDY}$ control bit **(Note 2)**
0 : $\overline{RDY}$ control is valid.
1 : $\overline{RDY}$ control is invalid.

“0” at reading.

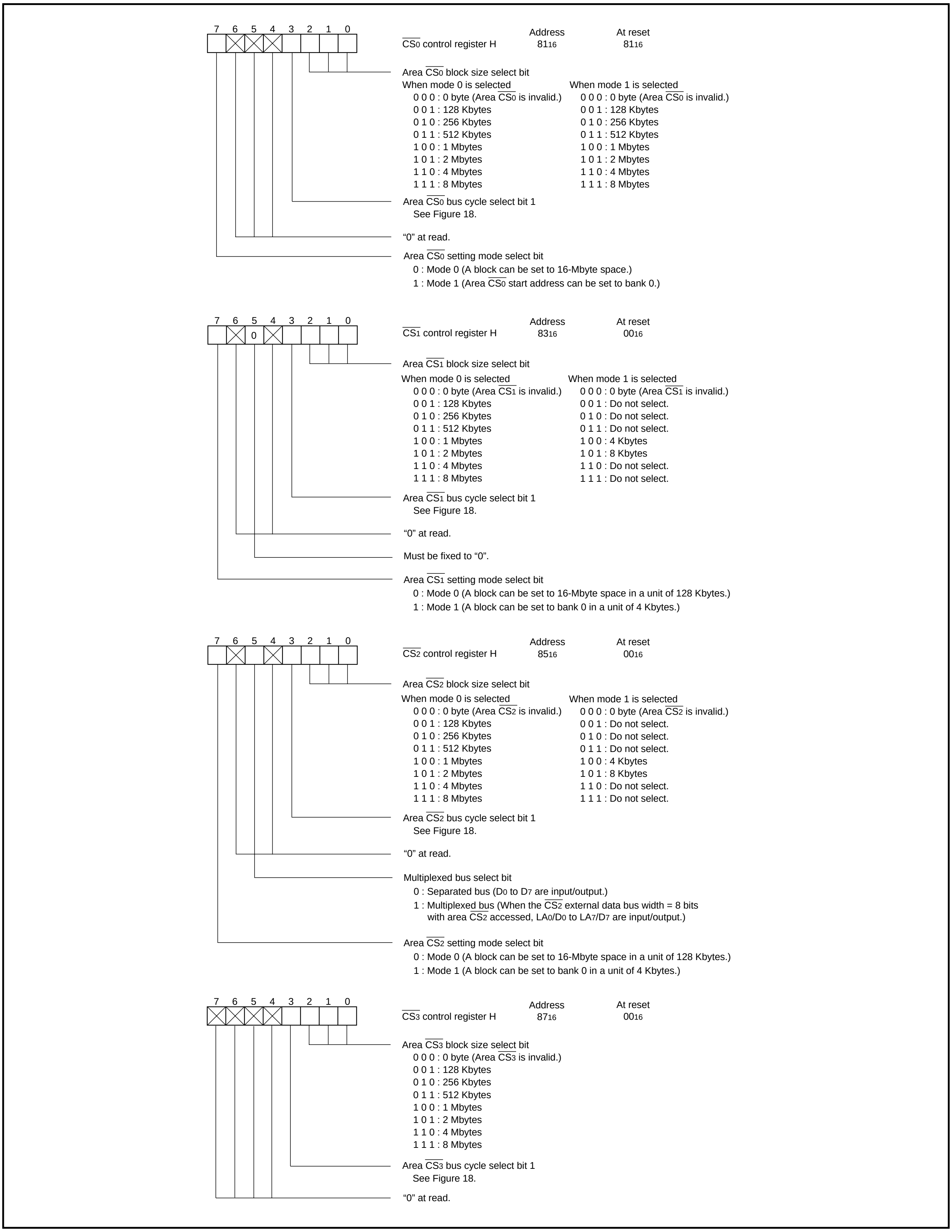
Burst ROM access select bit **(Note 3)**
0 : Normal access
1 : Burst ROM access

Recovery cycle insert select bit
0 : No recovery cycle is inserted at access to area $\overline{CSj}$.
1 : Recovery cycle is inserted at access to area $\overline{CSj}$.

$\overline{CSj}$ output select bit (j = 1 to 3) **(Note 4)**
0 : $\overline{CSj}$ output is disabled. ($P4_5$ to $P4_7$ function as programmable I/O port pins.)
1 : $\overline{CSj}$ output is enabled. ($P4_5$ to $P4_7$ function as pin $\overline{CSj}$.)

Notes 1: While V_{CC} level voltage is applied to pin BYTE, this bit is fixed to “1” (8-bit width).
2: This bit is valid when the $\overline{RDY}$ input select bit (bit 2 at address $5F_{16}$) = “1”.
3: When only the external data bus width select bit (bit 2) = “1” or while V_{CC} level voltage is applied to pin BYTE, the normal access is selected regardless of this bit’s contents.
4: In the single-chip mode, this bit’s contents are invalid. ($\overline{CS_0}$ output is disabled.)

Fig. 28 Bit configuration of $\overline{CS_0}$/$\overline{CS_1}$/$\overline{CS_2}$/$\overline{CS_3}$ control register Ls



$\overline{CS_0}$ control register H — Address 81_{16} — At reset 81_{16}

Area $\overline{CS_0}$ block size select bit

When mode 0 is selected	When mode 1 is selected
0 0 0 : 0 byte (Area $\overline{CS_0}$ is invalid.)	0 0 0 : 0 byte (Area $\overline{CS_0}$ is invalid.)
0 0 1 : 128 Kbytes	0 0 1 : 128 Kbytes
0 1 0 : 256 Kbytes	0 1 0 : 256 Kbytes
0 1 1 : 512 Kbytes	0 1 1 : 512 Kbytes
1 0 0 : 1 Mbytes	1 0 0 : 1 Mbytes
1 0 1 : 2 Mbytes	1 0 1 : 2 Mbytes
1 1 0 : 4 Mbytes	1 1 0 : 4 Mbytes
1 1 1 : 8 Mbytes	1 1 1 : 8 Mbytes

Area $\overline{CS_0}$ bus cycle select bit 1
See Figure 18.

“0” at read.

Area $\overline{CS_0}$ setting mode select bit
0 : Mode 0 (A block can be set to 16-Mbyte space.)
1 : Mode 1 (Area $\overline{CS_0}$ start address can be set to bank 0.)

$\overline{CS_1}$ control register H — Address 83_{16} — At reset 00_{16}

Area $\overline{CS_1}$ block size select bit

When mode 0 is selected	When mode 1 is selected
0 0 0 : 0 byte (Area $\overline{CS_1}$ is invalid.)	0 0 0 : 0 byte (Area $\overline{CS_1}$ is invalid.)
0 0 1 : 128 Kbytes	0 0 1 : Do not select.
0 1 0 : 256 Kbytes	0 1 0 : Do not select.
0 1 1 : 512 Kbytes	0 1 1 : Do not select.
1 0 0 : 1 Mbytes	1 0 0 : 4 Kbytes
1 0 1 : 2 Mbytes	1 0 1 : 8 Kbytes
1 1 0 : 4 Mbytes	1 1 0 : Do not select.
1 1 1 : 8 Mbytes	1 1 1 : Do not select.

Area $\overline{CS_1}$ bus cycle select bit 1
See Figure 18.

“0” at read.

Must be fixed to “0”.

Area $\overline{CS_1}$ setting mode select bit
0 : Mode 0 (A block can be set to 16-Mbyte space in a unit of 128 Kbytes.)
1 : Mode 1 (A block can be set to bank 0 in a unit of 4 Kbytes.)

$\overline{CS_2}$ control register H — Address 85_{16} — At reset 00_{16}

Area $\overline{CS_2}$ block size select bit

When mode 0 is selected	When mode 1 is selected
0 0 0 : 0 byte (Area $\overline{CS_2}$ is invalid.)	0 0 0 : 0 byte (Area $\overline{CS_2}$ is invalid.)
0 0 1 : 128 Kbytes	0 0 1 : Do not select.
0 1 0 : 256 Kbytes	0 1 0 : Do not select.
0 1 1 : 512 Kbytes	0 1 1 : Do not select.
1 0 0 : 1 Mbytes	1 0 0 : 4 Kbytes
1 0 1 : 2 Mbytes	1 0 1 : 8 Kbytes
1 1 0 : 4 Mbytes	1 1 0 : Do not select.
1 1 1 : 8 Mbytes	1 1 1 : Do not select.

Area $\overline{CS_2}$ bus cycle select bit 1
See Figure 18.

“0” at read.

Multiplexed bus select bit
0 : Separated bus (D_0 to D_7 are input/output.)
1 : Multiplexed bus (When the $\overline{CS_2}$ external data bus width = 8 bits
with area $\overline{CS_2}$ accessed, LA_0/D_0 to LA_7/D_7 are input/output.)

Area $\overline{CS_2}$ setting mode select bit
0 : Mode 0 (A block can be set to 16-Mbyte space in a unit of 128 Kbytes.)
1 : Mode 1 (A block can be set to bank 0 in a unit of 4 Kbytes.)

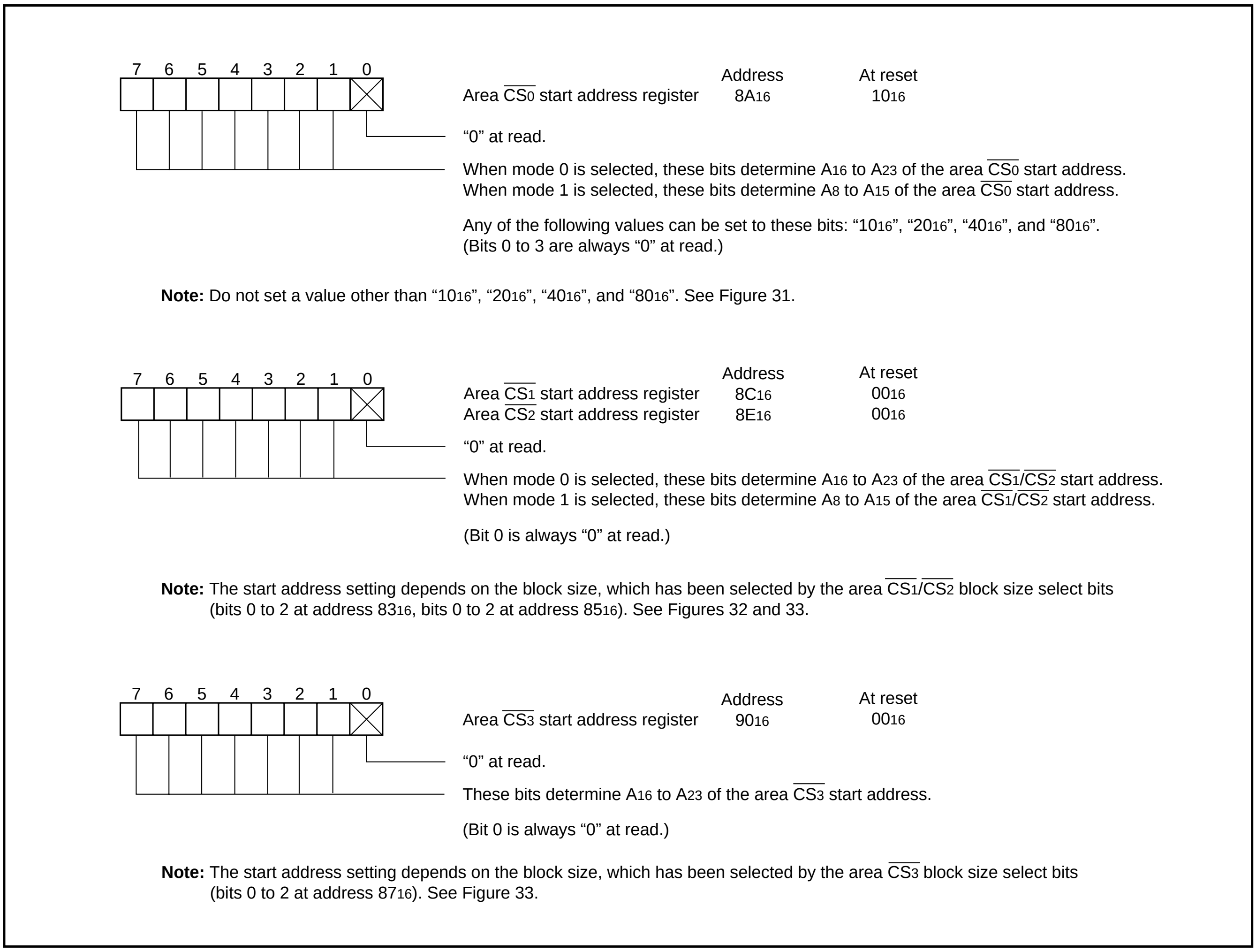
$\overline{CS_3}$ control register H — Address 87_{16} — At reset 00_{16}

Area $\overline{CS_3}$ block size select bit
0 0 0 : 0 byte (Area $\overline{CS_3}$ is invalid.)
0 0 1 : 128 Kbytes
0 1 0 : 256 Kbytes
0 1 1 : 512 Kbytes
1 0 0 : 1 Mbytes
1 0 1 : 2 Mbytes
1 1 0 : 4 Mbytes
1 1 1 : 8 Mbytes

Area $\overline{CS_3}$ bus cycle select bit 1
See Figure 18.

“0” at read.

Fig. 29 Bit configuration of $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ control register Hs



	Address	At reset
Area $\overline{CS_0}$ start address register	$8A_{16}$	10_{16}

"0" at read.

When mode 0 is selected, these bits determine A_{16} to A_{23} of the area $\overline{CS_0}$ start address.
When mode 1 is selected, these bits determine A_8 to A_{15} of the area $\overline{CS_0}$ start address.

Any of the following values can be set to these bits: "10_{16}", "20_{16}", "40_{16}", and "80_{16}".
(Bits 0 to 3 are always "0" at read.)

Note: Do not set a value other than "10_{16}", "20_{16}", "40_{16}", and "80_{16}". See Figure 31.

	Address	At reset
Area $\overline{CS_1}$ start address register	$8C_{16}$	00_{16}
Area $\overline{CS_2}$ start address register	$8E_{16}$	00_{16}

"0" at read.

When mode 0 is selected, these bits determine A_{16} to A_{23} of the area $\overline{CS_1}/\overline{CS_2}$ start address.
When mode 1 is selected, these bits determine A_8 to A_{15} of the area $\overline{CS_1}/\overline{CS_2}$ start address.

(Bit 0 is always "0" at read.)

Note: The start address setting depends on the block size, which has been selected by the area $\overline{CS_1}/\overline{CS_2}$ block size select bits (bits 0 to 2 at address 83_{16}, bits 0 to 2 at address 85_{16}). See Figures 32 and 33.

	Address	At reset
Area $\overline{CS_3}$ start address register	90_{16}	00_{16}

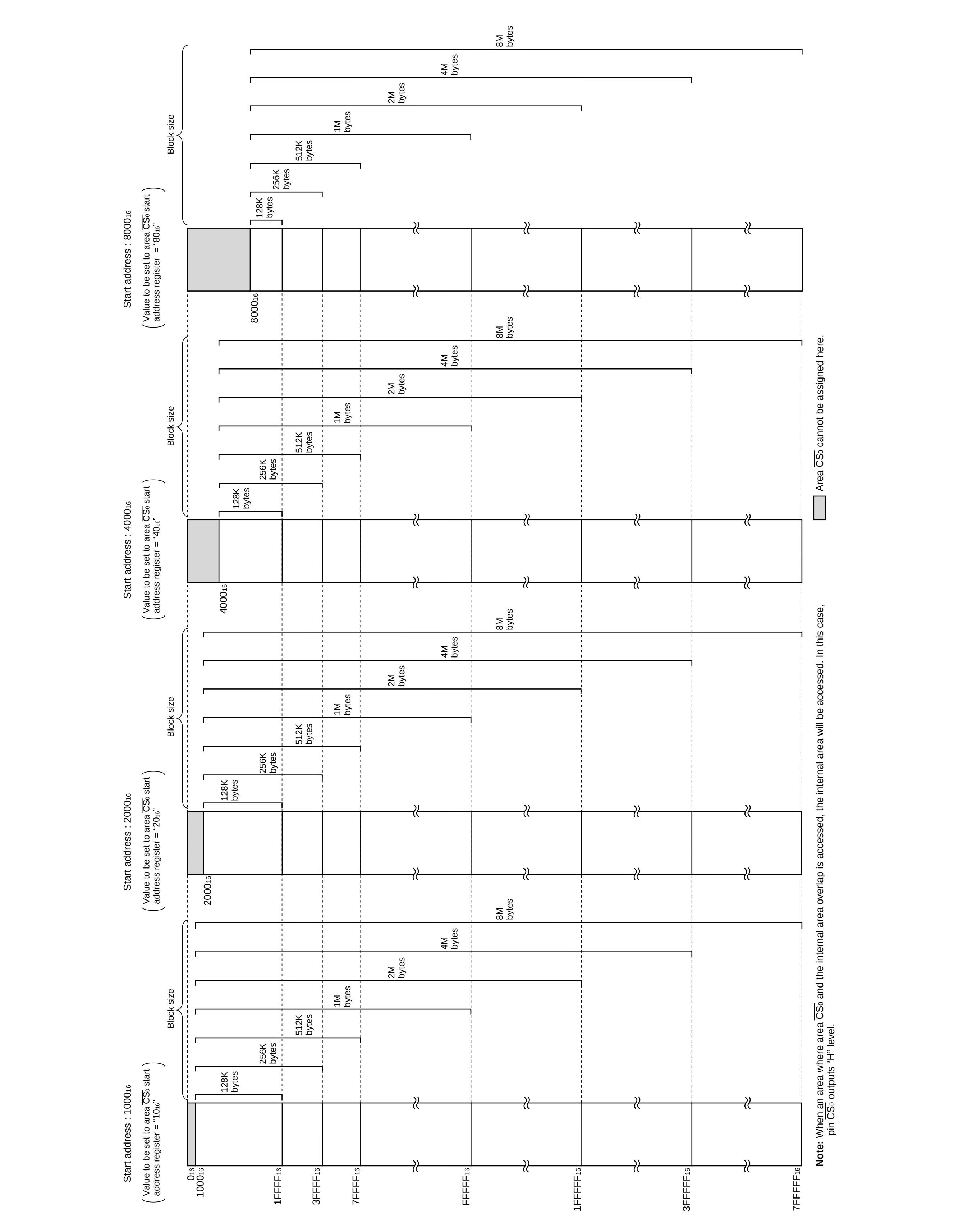
"0" at read.

These bits determine A_{16} to A_{23} of the area $\overline{CS_3}$ start address.

(Bit 0 is always "0" at read.)

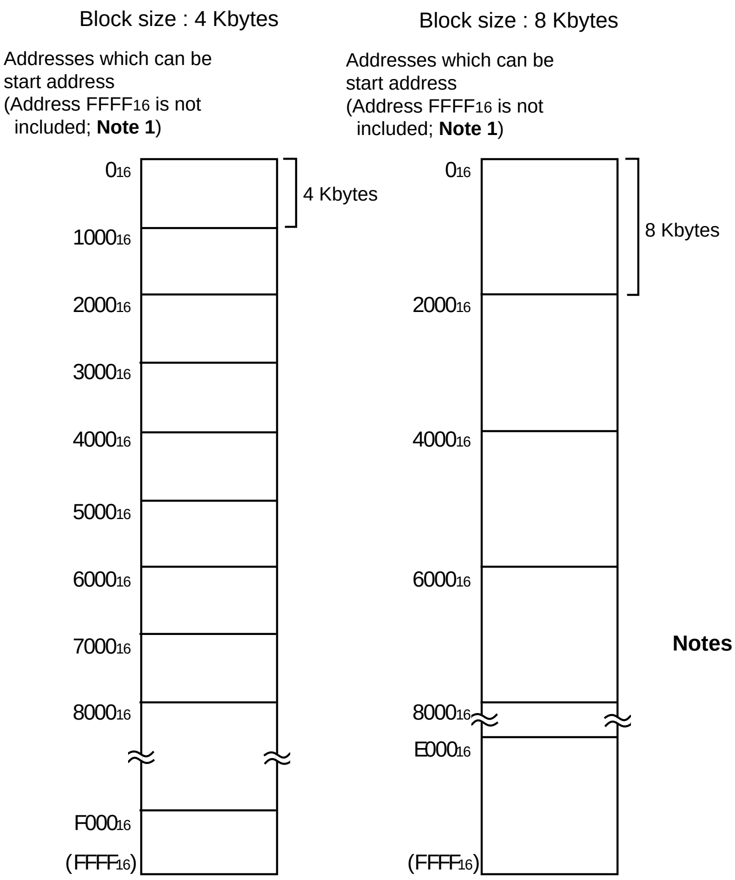
Note: The start address setting depends on the block size, which has been selected by the area $\overline{CS_3}$ block size select bits (bits 0 to 2 at address 87_{16}). See Figure 33.

Fig. 30 Bit configuration of area $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ start address registers


Start address : 1000₁₆
(Value to be set to area $\overline{CS_0}$ start address register = "10₁₆")
Start address : 2000₁₆
(Value to be set to area $\overline{CS_0}$ start address register = "20₁₆")
Start address : 4000₁₆
(Value to be set to area $\overline{CS_0}$ start address register = "40₁₆")
Start address : 8000₁₆
(Value to be set to area $\overline{CS_0}$ start address register = "80₁₆")
Block size
128K bytes
256K bytes
512K bytes
1M bytes
2M bytes
4M bytes
8M bytes
0₁₆
1000₁₆
2000₁₆
4000₁₆
8000₁₆
1FFFF₁₆
3FFFF₁₆
7FFFF₁₆
FFFFF₁₆
1FFFFF₁₆
3FFFFF₁₆
7FFFFF₁₆
Area $\overline{CS_0}$ cannot be assigned here.

Note: When an area where area $\overline{CS_0}$ and the internal area overlap is accessed, the internal area will be accessed. In this case, pin $\overline{CS_0}$ outputs "H" level.

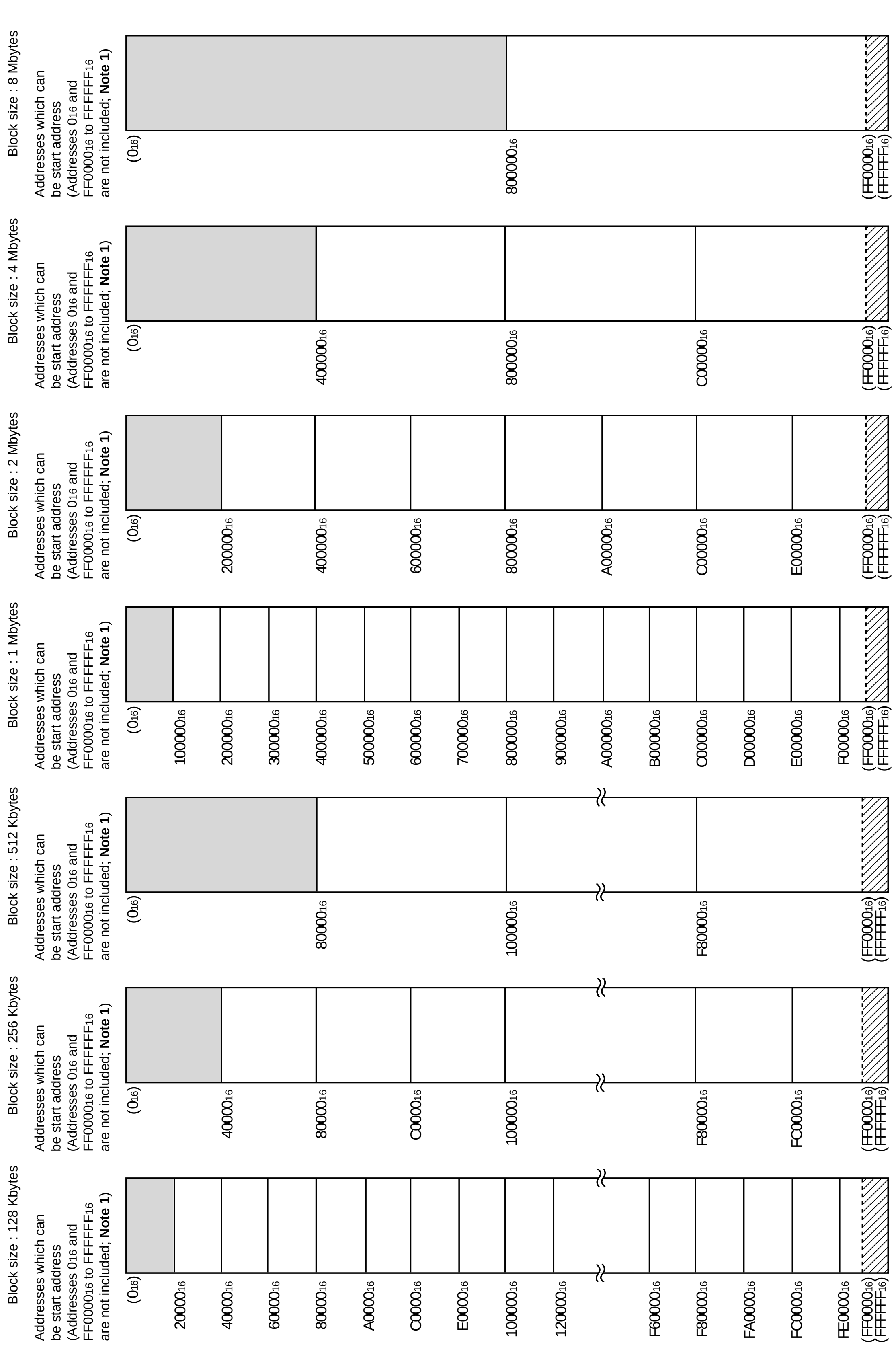
Fig. 31 Area $\overline{CS_0}$ (mode 1)



Notes 1: Only A_8 to A_{15} of one of these addresses can be set to the area $\overline{CS_1}/\overline{CS_2}$ start address register. Do not set another address not shown here.

2: When an area where area $\overline{CS_1}/\overline{CS_2}$ and the internal area overlap is accessed, the internal area will be accessed. In this case, pin $\overline{CS_1}/\overline{CS_2}$ outputs “H” level.

Fig. 32 Area $\overline{CS_1}/\overline{CS_2}$ (mode 1)


Block size : 128 Kbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
20000₁₆
40000₁₆
60000₁₆
80000₁₆
A0000₁₆
C0000₁₆
E0000₁₆
100000₁₆
120000₁₆
F60000₁₆
F80000₁₆
FA0000₁₆
FC0000₁₆
FE0000₁₆
(FF0000₁₆)
(FFFFFF₁₆)
Block size : 256 Kbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
40000₁₆
80000₁₆
C0000₁₆
100000₁₆
F80000₁₆
FC0000₁₆
(FF0000₁₆)
(FFFFFF₁₆)
Block size : 512 Kbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
80000₁₆
100000₁₆
F80000₁₆
(FF0000₁₆)
(FFFFFF₁₆)
Block size : 1 Mbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
100000₁₆
200000₁₆
300000₁₆
400000₁₆
500000₁₆
600000₁₆
700000₁₆
800000₁₆
900000₁₆
A00000₁₆
B00000₁₆
C00000₁₆
D00000₁₆
E00000₁₆
F00000₁₆
(FF0000₁₆)
(FFFFFF₁₆)
Block size : 2 Mbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
200000₁₆
400000₁₆
600000₁₆
800000₁₆
A00000₁₆
C00000₁₆
E00000₁₆
(FF0000₁₆)
(FFFFFF₁₆)
Block size : 4 Mbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
400000₁₆
800000₁₆
C00000₁₆
(FF0000₁₆)
(FFFFFF₁₆)
Block size : 8 Mbytes
Addresses which can be start address (Addresses 0₁₆ and FF0000₁₆ to FFFFFF₁₆ are not included; Note 1)
(0₁₆)
800000₁₆
(FF0000₁₆)
(FFFFFF₁₆)

▒ : Area $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ cannot be assigned here.

▨ : Reserved area. Do not access this area.

Notes 1: Only A_{16} to A_{23} of one of these addresses can be set to the area $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ start address register. Do not set another address not shown here.

2: When an area where area $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ and the internal area overlap is accessed, the internal area will be accessed. In this case, pin $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ outputs "H" level.

Fig. 33 Area $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}$ (mode 0) and area $\overline{CS_3}$

INTERRUPTS

Table 8 shows the interrupt sources and the corresponding interrupt vector addresses. Reset is also handled as a type of interrupt in this section, too.

$\overline{\text{DBC}}$ and BRK instruction are interrupts used only for debugging. Therefore, do not use these interrupts.

Interrupts other than reset, watchdog timer, zero divide, $\overline{\text{NMI}}$, and address matching detection all have interrupt control registers. Table 9 shows the addresses of the interrupt control registers and Figure 35 shows the bit configuration of the interrupt control register.

The interrupt request bit is automatically cleared by the hardware during reset or when processing an interrupt. Also, interrupt request bits other than watchdog timer and $\overline{\text{NMI}}$ can be cleared by software.

An $\overline{\text{NMI}}$ interrupt request is a non-maskable interrupt by an external input and is accepted at the falling edge of an input to pin $\overline{\text{NMI}}$. Also, pin $\overline{\text{NMI}}$ has the pullup function. For more details, refer to the section on input/output pins.

An $\overline{\text{INT}_i}$ (i = 0 to 4) interrupt request is generated by an external input.

$\overline{\text{INT}_0}$ to $\overline{\text{INT}_2}$ are external interrupts; whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level/edge select bit. Furthermore, the polarity of the interrupt input can be selected with the polarity select bit.

For $\overline{\text{INT}_3}$ and $\overline{\text{INT}_4}$, the interrupt signal's polarity can be change by the polarity select bit. (This is valid only in the edge sense.)

By pins $\overline{\text{INT}_2}$ to $\overline{\text{INT}_4}$ select bits (bits 4 to 6 at address 94_{16}; see Figure 40.), pin position of $\overline{\text{INT}_2}$ to $\overline{\text{INT}_4}$ can be changed.

When using the following pins as external interrupt input pins, clear the direction registers of the corresponding multiplexed ports to "0": pins $P6_2/\overline{\text{INT}_0}$, $P6_3/\overline{\text{INT}_1}$, $P6_4(P7_7)/\overline{\text{INT}_2}$, $P8_0(P7_4)/\overline{\text{INT}_3}$, and $P8_4(P7_5)/\overline{\text{INT}_4}$.

Furthermore, the $\overline{\text{INT}_3}$ interrupt can function as the key input interrupt. For details, refer to the section on the key input interrupt.

When the external interrupt input read register (address 95_{16}) is read out, the status of pins $\overline{\text{INT}_0}$ to $\overline{\text{INT}_4}$ and $\overline{\text{NMI}}$ can directly be read.

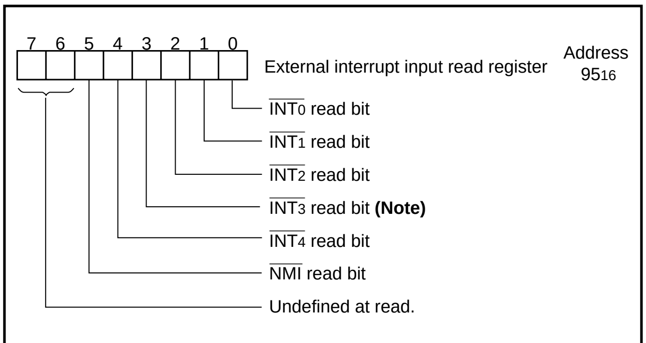
Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupt requests are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 36.

The hardware priority is fixed as the following:
reset > $\overline{\text{NMI}}$ > watchdog timer > other interrupts

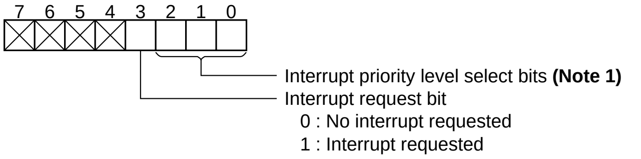
Table 8. Interrupt sources and interrupt vector addresses

Interrupts	Vector addresses	
Address matching detection interrupt	$00FFCA_{16}$	$00FFCB_{16}$
$\overline{\text{INT}_4}$ external interrupt	$00FFD0_{16}$	$00FFD1_{16}$
$\overline{\text{INT}_3}$ external interrupt	$00FFD2_{16}$	$00FFD3_{16}$
A-D conversion	$00FFD4_{16}$	$00FFD5_{16}$
UART1 transmit	$00FFD6_{16}$	$00FFD7_{16}$
UART1 receive	$00FFD8_{16}$	$00FFD9_{16}$
UART0 transmit	$00FFDA_{16}$	$00FFDB_{16}$
UART0 receive	$00FFDC_{16}$	$00FFDD_{16}$
Timer B2	$00FFDE_{16}$	$00FFDF_{16}$
Timer B1	$00FFE0_{16}$	$00FFE1_{16}$
Timer B0	$00FFE2_{16}$	$00FFE3_{16}$
Timer A4	$00FFE4_{16}$	$00FFE5_{16}$
Timer A3	$00FFE6_{16}$	$00FFE7_{16}$
Timer A2	$00FFE8_{16}$	$00FFE9_{16}$
Timer A1	$00FFEA_{16}$	$00FFEB_{16}$
Timer A0	$00FFEC_{16}$	$00FFED_{16}$
$\overline{\text{INT}_2}$ external interrupt	$00FFEE_{16}$	$00FFEF_{16}$
$\overline{\text{INT}_1}$ external interrupt	$00FFF0_{16}$	$00FFF1_{16}$
$\overline{\text{INT}_0}$ external interrupt	$00FFF2_{16}$	$00FFF3_{16}$
$\overline{\text{NMI}}$ external interrupt	$00FFF4_{16}$	$00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$	$00FFF7_{16}$
$\overline{\text{DBC}}$ (Do not select.)	$00FFF8_{16}$	$00FFF9_{16}$
Break instruction (Do not select.)	$00FFFA_{16}$	$00FFFB_{16}$
Zero divide	$00FFFC_{16}$	$00FFFD_{16}$
Reset	$00FFFE_{16}$	$00FFFF_{16}$


7 6 5 4 3 2 1 0
External interrupt input read register
Address
95_{16}
$\overline{\text{INT}_0}$ read bit
$\overline{\text{INT}_1}$ read bit
$\overline{\text{INT}_2}$ read bit
$\overline{\text{INT}_3}$ read bit **(Note)**
$\overline{\text{INT}_4}$ read bit
$\overline{\text{NMI}}$ read bit
Undefined at read.

Note: When the key input interrupt select bit (bit 0 at address 94_{16}) = "1", the status of pin $\overline{\text{INT}_3}$ cannot be read out.

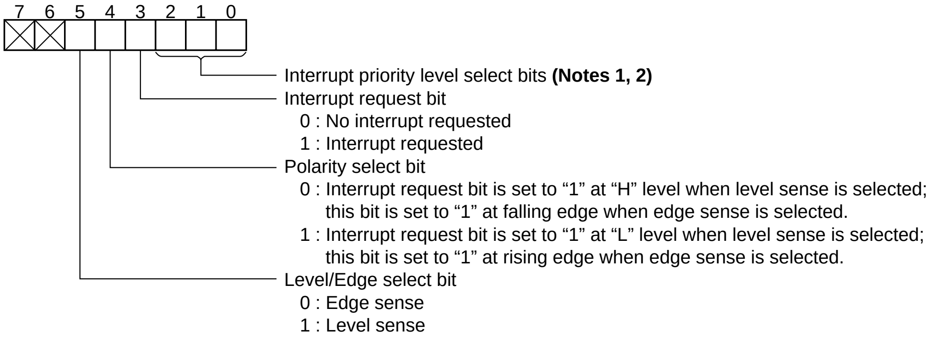
Fig. 34 Bit configuration of external interrupt input read register



Interrupt priority level select bits **(Note 1)**

Interrupt request bit
0 : No interrupt requested
1 : Interrupt requested

Interrupt control register bit configuration for A-D converter, UART0, UART1, timer A0 to timer A4, and timer B0 to timer B2.



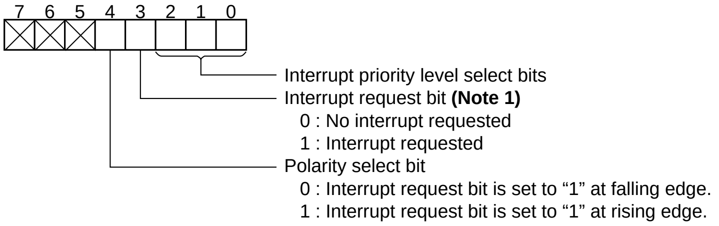
Interrupt priority level select bits **(Notes 1, 2)**

Interrupt request bit
0 : No interrupt requested
1 : Interrupt requested

Polarity select bit
0 : Interrupt request bit is set to “1” at “H” level when level sense is selected; this bit is set to “1” at falling edge when edge sense is selected.
1 : Interrupt request bit is set to “1” at “L” level when level sense is selected; this bit is set to “1” at rising edge when edge sense is selected.

Level/Edge select bit
0 : Edge sense
1 : Level sense

Interrupt control register bit configuration for $\overline{INT_0}$– $\overline{INT_2}$



Interrupt priority level select bits

Interrupt request bit **(Note 1)**
0 : No interrupt requested
1 : Interrupt requested

Polarity select bit
0 : Interrupt request bit is set to “1” at falling edge.
1 : Interrupt request bit is set to “1” at rising edge.

Interrupt control register bit configuration for $\overline{INT_3}$ and $\overline{INT_4}$

Notes 1: Use the **MOVM (MOVMB)** instruction or the **STA (STAB, STAD)** instruction for writing to this bit.
2: Interrupt request bits of $\overline{INT_0}$ to $\overline{INT_2}$ are invalid when the level sense is selected.

Fig. 35 Bit configuration of interrupt control register

Table 9. Addresses of interrupt control registers

Interrupt control registers	Addresses
$\overline{\text{INT}_3}$ interrupt control register	$00006E_{16}$
$\overline{\text{INT}_4}$ interrupt control register	$00006F_{16}$
A-D interrupt control register	000070_{16}
UART0 transmit interrupt control register	000071_{16}
UART0 receive interrupt control register	000072_{16}
UART1 transmit interrupt control register	000073_{16}
UART1 receive interrupt control register	000074_{16}
Timer A0 interrupt control register	000075_{16}
Timer A1 interrupt control register	000076_{16}
Timer A2 interrupt control register	000077_{16}
Timer A3 interrupt control register	000078_{16}
Timer A4 interrupt control register	000079_{16}
Timer B0 interrupt control register	$00007A_{16}$
Timer B1 interrupt control register	$00007B_{16}$
Timer B2 interrupt control register	$00007C_{16}$
$\overline{\text{INT}_0}$ interrupt control register	$00007D_{16}$
$\overline{\text{INT}_1}$ interrupt control register	$00007E_{16}$
$\overline{\text{INT}_2}$ interrupt control register	$00007F_{16}$

Interrupts caused by the address matching detection and when dividing by zero are software interrupts and are not included in Figure 36.

Other interrupts previously mentioned are A-D converter, UART, etc. interrupts. The priority of these interrupts can be changed by changing the priority level in the corresponding interrupt control register by software.

Figure 37 shows a diagram of the interrupt priority detection circuit When an interrupt is caused, each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is "0". The request is not accepted if flag I is "1". The reset, $\overline{\text{NMI}}$, and watchdog timer interrupts are not affected by the interrupt disable flag I.

When an interrupt is accepted, the contents of the processor status register (PS) is saved to the stack and the interrupt disable flag I is set to "1".

Furthermore, the interrupt request bit of the accepted interrupt is cleared to "0" and the processor interrupt priority level (IPL) in the processor status register (PS) is replaced by the priority level of the accepted interrupt.

Therefore, multi-level priority interrupts are possible by resetting the interrupt disable flag I to "0" and enable further interrupts.

For reset, watchdog timer, zero divide, $\overline{\text{NMI}}$, and address match detection interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 10.

The interrupt request bit and the interrupt priority level of each interrupt source are sampled and latched at each operation code fetch cycle while f_{sys} is "H". However, no sampling pulse is generated until the cycles whose number is selected by software has passed, even if the next operation code fetch cycle is generated. The detection of an interrupt which has the highest priority is performed during that time.

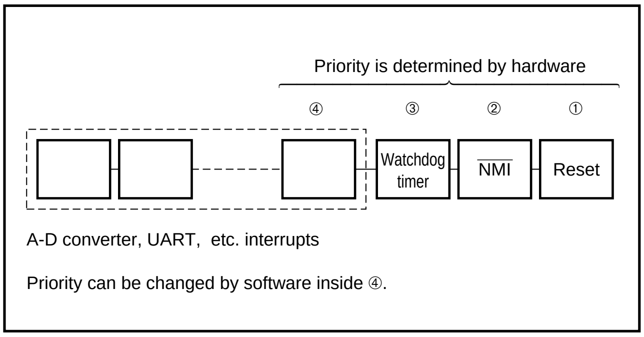

Priority is determined by hardware
④
③
②
①
Watchdog timer
NMI
Reset
A-D converter, UART, etc. interrupts
Priority can be changed by software inside ④.

Fig. 36 Interrupt priority

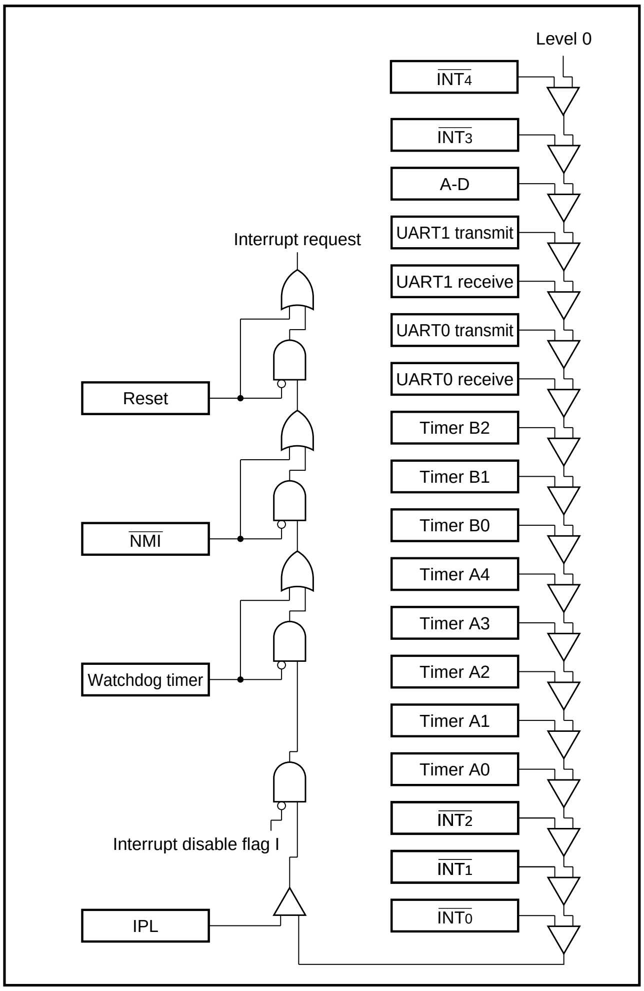


Fig. 37 Interrupt priority detection

As shown in Figure 38, there are three different interrupt priority detection time from which one is selected by software. After the selected time has elapsed, the highest priority is determined and is processed after the currently executing instruction has been completed.
The time is selected with bits 4 and 5 of the processor mode register 0 (address $5E_{16}$) shown in Figure 24. Table 11 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register 0 is initialized to "00_{16}." Therefore, the longest time is automatically set, however, the shortest time must be selected by software.

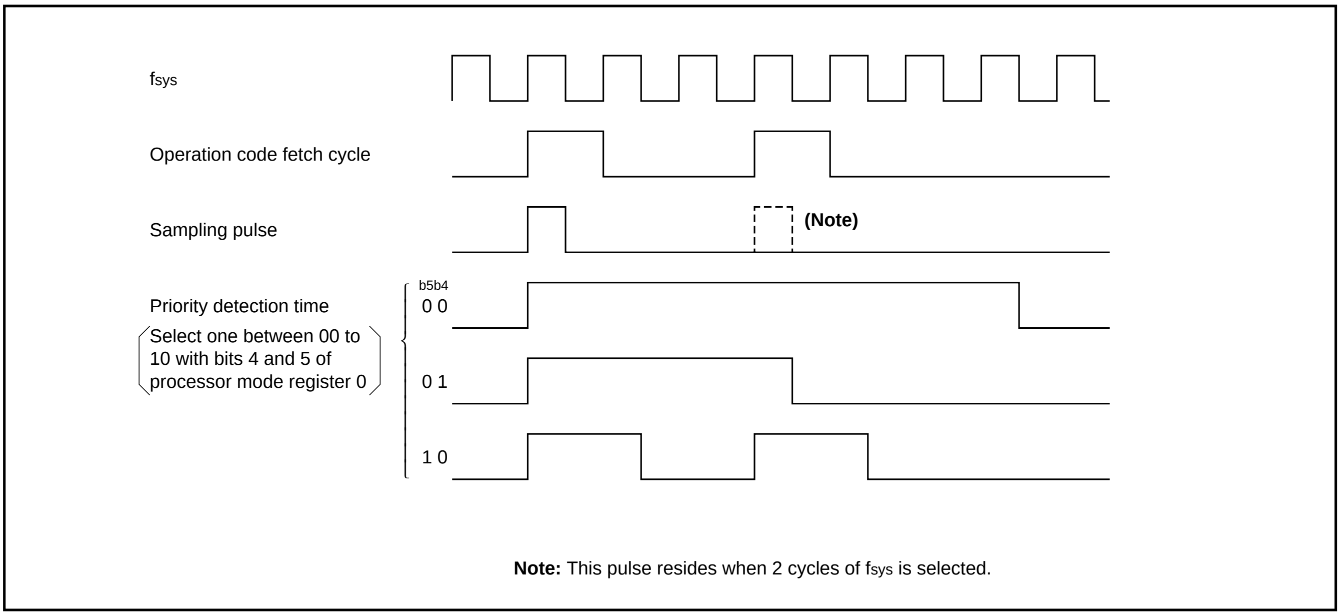
Table 10. Value loaded in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
Watchdog timer	7
$\overline{\text{NMI}}$	7
Zero divide	Not change value of IPL.
Address matching detection	Not change value of IPL.

Table 11. Relationship between interrupt priority detection time select bit and number of cycles

Priority detection time select bit		Number of cycles **(Note)**
Bit 5	Bit 4	
0	0	7 cycles of f_{sys}
0	1	4 cycles of f_{sys}
1	0	2 cycles of f_{sys}

Note: For system clock f_{sys}, refer to the section on the clock generating circuit.


fsys
Operation code fetch cycle
Sampling pulse
(Note)
Priority detection time
Select one between 00 to 10 with bits 4 and 5 of processor mode register 0
b5b4
0 0
0 1
1 0

Note: This pulse resides when 2 cycles of f_{sys} is selected.

Fig. 38 Interrupt priority detection time

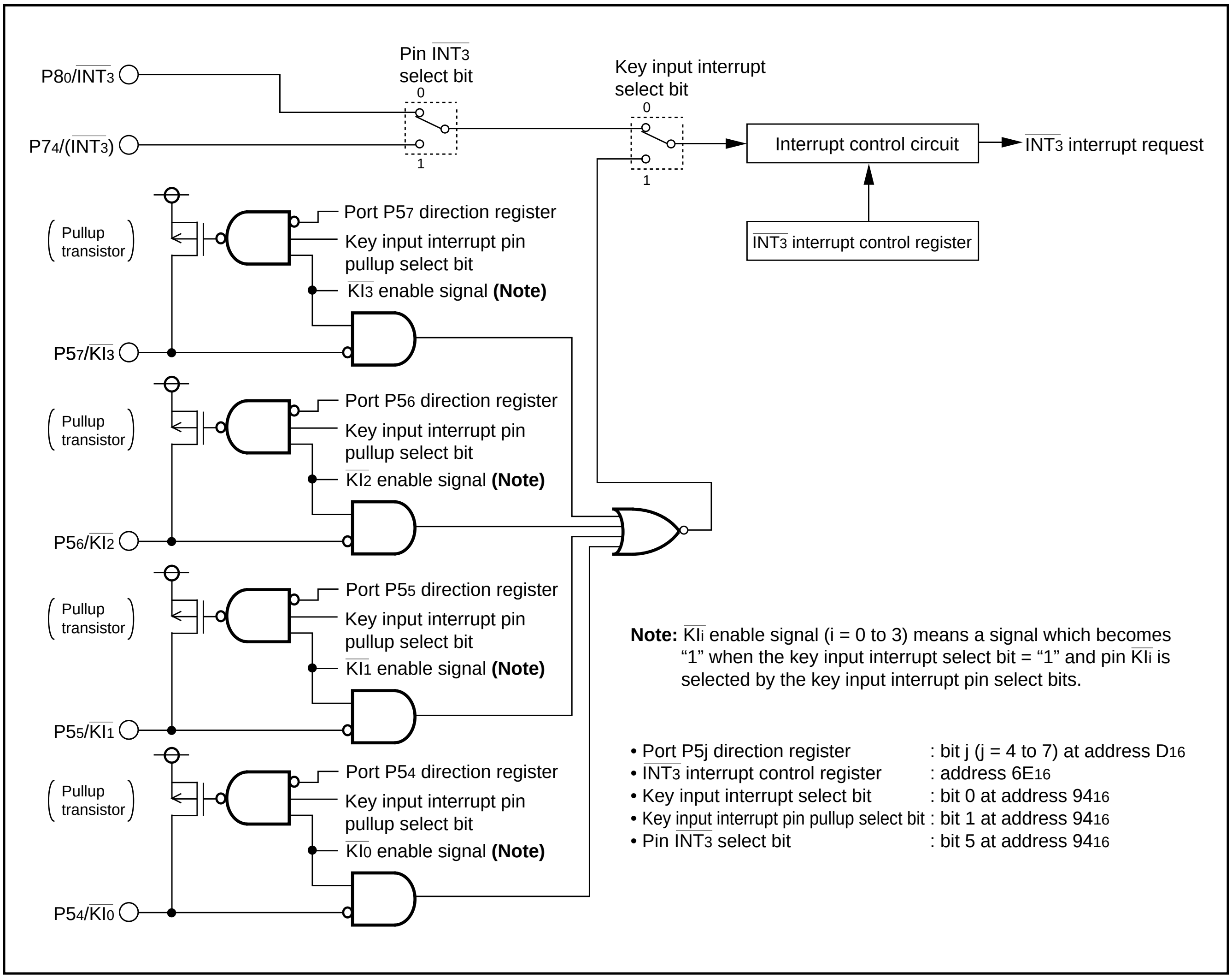
Key Input Interrupt

The $\overline{INT_3}$ interrupt can function as the key input interrupt by setting bits 1 to 3 of the external interrupt input control register (address 94_{16}). The key input interrupt uses inputs $\overline{KI_0}$ to $\overline{KI_3}$. Figure 39 shows the block diagram of the $\overline{INT_3}$/key input interrupt input circuit, and Figure 40 shows the bit configuration of the external interrupt input control register.

When bit 0 of the external interrupt input control register (key input interrupt select bit)= “0”, a signal from pin $\overline{INT_3}$ is connected to the $\overline{INT_3}$ interrupt control circuit, and $\overline{INT_3}$ external interrupt is normally performed. When bit 0 = “1”, signals from pins $\overline{KI_0}$ to $\overline{KI_3}$, which correspond to ports $P5_4$ to $P5_7$ pins, are inverted, and then, the logical sum of these signals is connected to the $\overline{INT_3}$ interrupt control register. In this case, the external interrupt which uses pins $\overline{KI_0}$ to $\overline{KI_3}$ is performed.

Bits 2 and 3 of the external interrupt input control register are the key input interrupt pin select bits. By setting these bits, the combination of key input interrupt pins can be selected. The interrupt vector addresses and interrupt control register of the key input interrupt are common to those of the $\overline{INT_3}$ interrupt. Additionally, pullup resistors (transistors) can be added to pins $\overline{KI_0}$ to $\overline{KI_4}$ by setting as follows:

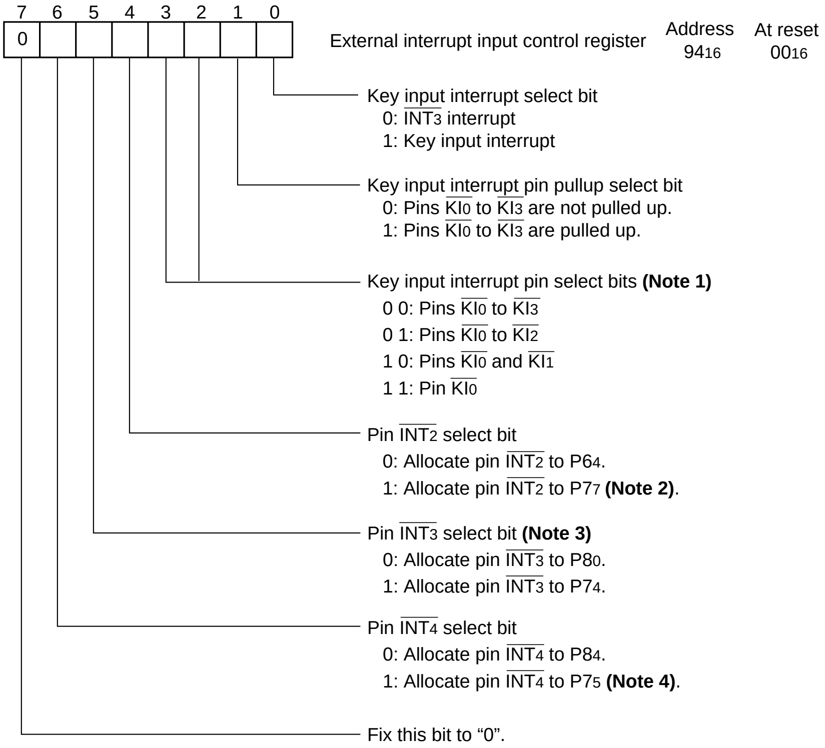
- Set bit 1 of the external interrupt input control register to “1”.
- Next, select the key input interrupt pins by bits 2 and 3 of the external interrupt input control register.
- Then, clear the contents of the port direction register which corresponds to the selected pins to “0”.


P8₀/INT3
P7₄/(INT3)
Pin INT3 select bit
0
1
Key input interrupt select bit
0
1
Interrupt control circuit
INT3 interrupt request
INT3 interrupt control register
Pullup transistor
Port P5₇ direction register
Key input interrupt pin pullup select bit
KI3 enable signal (Note)
P5₇/KI3
Pullup transistor
Port P5₆ direction register
Key input interrupt pin pullup select bit
KI2 enable signal (Note)
P5₆/KI2
Pullup transistor
Port P5₅ direction register
Key input interrupt pin pullup select bit
KI1 enable signal (Note)
P5₅/KI1
Pullup transistor
Port P5₄ direction register
Key input interrupt pin pullup select bit
KI0 enable signal (Note)
P5₄/KI0

Note: $\overline{KI_i}$ enable signal (i = 0 to 3) means a signal which becomes “1” when the key input interrupt select bit = “1” and pin $\overline{KI_i}$ is selected by the key input interrupt pin select bits.

• Port P5j direction register	: bit j (j = 4 to 7) at address D_{16}
• $\overline{INT_3}$ interrupt control register	: address $6E_{16}$
• Key input interrupt select bit	: bit 0 at address 94_{16}
• Key input interrupt pin pullup select bit	: bit 1 at address 94_{16}
• Pin $\overline{INT_3}$ select bit	: bit 5 at address 94_{16}

Fig. 39 Block diagram of $\overline{INT_3}$/key input interrupt input circuit



External interrupt input control register

Address	At reset
94_{16}	00_{16}

Key input interrupt select bit
0: $\overline{INT_3}$ interrupt
1: Key input interrupt

Key input interrupt pin pullup select bit
0: Pins $\overline{KI_0}$ to $\overline{KI_3}$ are not pulled up.
1: Pins $\overline{KI_0}$ to $\overline{KI_3}$ are pulled up.

Key input interrupt pin select bits **(Note 1)**
0 0: Pins $\overline{KI_0}$ to $\overline{KI_3}$
0 1: Pins $\overline{KI_0}$ to $\overline{KI_2}$
1 0: Pins $\overline{KI_0}$ and $\overline{KI_1}$
1 1: Pin $\overline{KI_0}$

Pin $\overline{INT_2}$ select bit
0: Allocate pin $\overline{INT_2}$ to $P6_4$.
1: Allocate pin $\overline{INT_2}$ to $P7_7$ **(Note 2)**.

Pin $\overline{INT_3}$ select bit **(Note 3)**
0: Allocate pin $\overline{INT_3}$ to $P8_0$.
1: Allocate pin $\overline{INT_3}$ to $P7_4$.

Pin $\overline{INT_4}$ select bit
0: Allocate pin $\overline{INT_4}$ to $P8_4$.
1: Allocate pin $\overline{INT_4}$ to $P7_5$ **(Note 4)**.

Fix this bit to “0”.

Notes 1: When using pin $\overline{KI_i}$, do not select timer A’s output pins and pulse output pins which are multiplexed with pin $\overline{KI_i}$.
2: When pin $\overline{INT_2}$ is allocated to $P7_7$, do not use pin $AN_7/\overline{AD_{TRG}}$. Additionally, clear the D-A_1 output enable bit (bit 1 at address 96_{16}) to “0” (output disabled).
3: When pin $\overline{INT_3}$ is allocated to $P8_0$, clear the D-A_2 output enable bit (bit 2 at address 96_{16}) to “0” (output disabled). When pin $\overline{INT_3}$ is allocated to $P7_4$, do not use pin AN_4.
4: When pin $\overline{INT_4}$ is allocated to $P7_5$, do not use pin AN_5.

Fig. 40 Bit configuration of external interrupt input control register

TIMER

There are eight 16-bit timers. They are divided by type into timer A(5) and timer B(3).

The timer I/O pins are multiplexed with I/O pins for port P5 and P6. To use these pins as timer input pins, the port direction register bit corresponding to the pin must be cleared to “0” to specify input mode.

TIMER A

Figure 41 shows a block diagram of timer A.

Timer A has four modes: timer mode, event counter mode, one-shot pulse mode, and pulse width modulation mode. The mode is selected with bits 0 and 1 of the timer Ai mode register (i = 0 to 4). Each of these modes is described below.

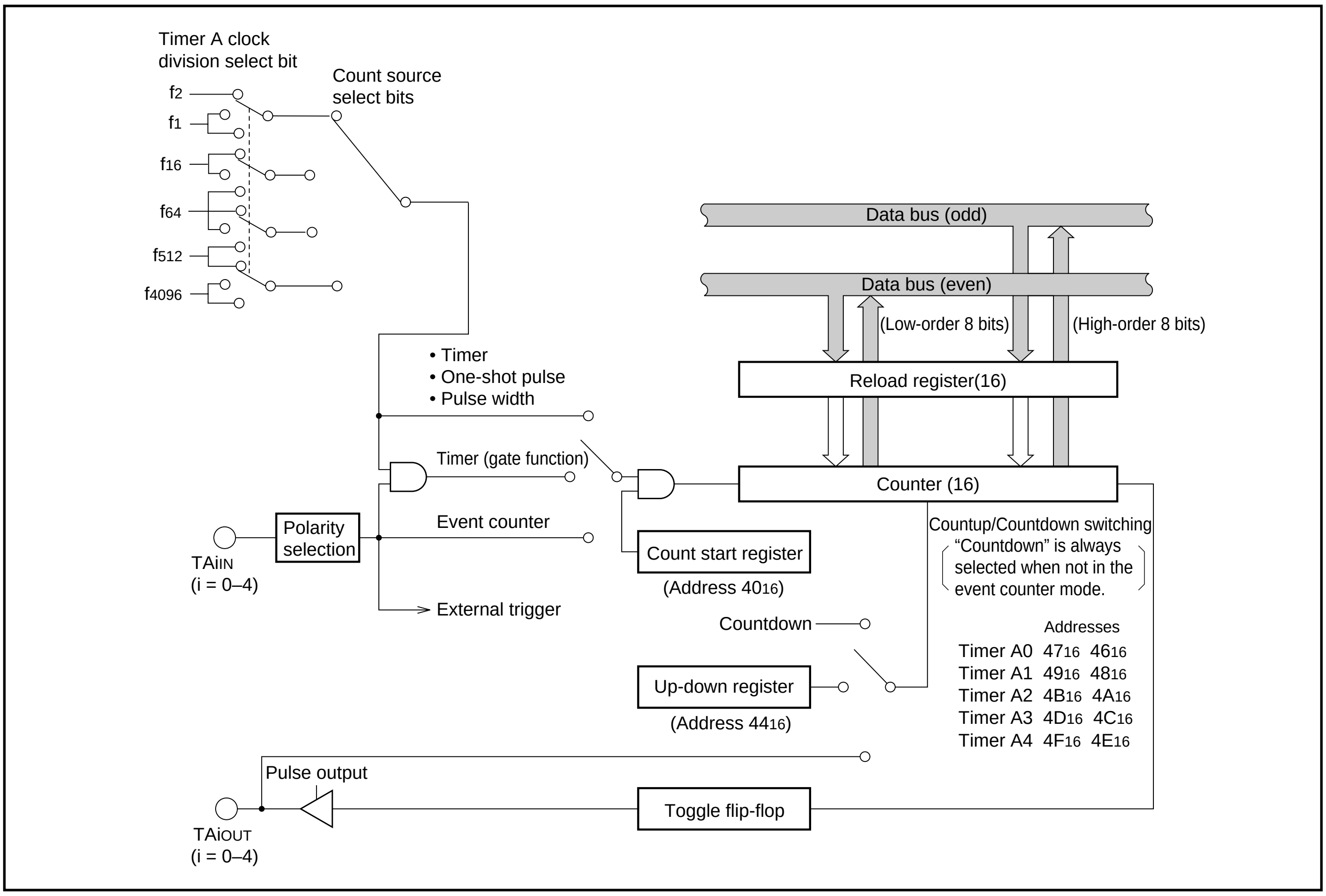
Figure 42 shows the bit configuration of the timer A clock division select register. Timers A0 to A4 use the count source which has been selected by bits 0 and 1 of this register.

(1) Timer mode [00]

Figure 43 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1 and 5 of the timer Ai mode register must be “0” in timer mode. The timer A’s count source is selected by bits 6 and 7 of the timer Ai mode register and the contents of the timer A clock division select register. (See Table 12.)

The counting of the selected clock starts when the count start bit is “1” and stops when it is “0”.

Figure 44 shows the bit configuration of the count start bit. The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.



	Addresses	
Timer A0	47_{16}	46_{16}
Timer A1	49_{16}	48_{16}
Timer A2	$4B_{16}$	$4A_{16}$
Timer A3	$4D_{16}$	$4C_{16}$
Timer A4	$4F_{16}$	$4E_{16}$

Fig. 41 Block diagram of timer A

When bit 2 of the timer Ai mode register is “1”, the output is generated from TAi_{OUT} pin. The output is toggled each time the contents of the counter reaches to 0000_{16}. When the contents of the count start bit is “0”, “L” is output from TAi_{OUT} pin.

When bit 2 is “0”, TAi_{OUT} can be used as a normal port pin. When bit 4 is “0”, TAi_{IN} can be used as a normal port pin.

When bit 4 is “1”, counting is performed only while the input signal from the TAi_{IN} pin is “H” or “L” as shown in Figure 45. Therefore, this can be used to measure the pulse width of the TAi_{IN} input signal. Whether to count while the input signal is “H” or while it is “L” is determined by bit 3. If bit 3 is “1”, counting is performed while the TAi_{IN} pin input signal is “H” and if bit 3 is “0”, counting is performed while it is “L”.

Note that, the duration of “H” or “L” on the TAi_{IN} pin must be 2 or more cycles of the timer count source.

When data is written to timer Ai register with timer Ai halted, the same data is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The new data is reloaded from the reload register to the counter at the next reload time and counting continues. The contents of the counter can be read at any time.

When the value set in the timer Ai register is n, the timer frequency division ratio is 1/(n+1).

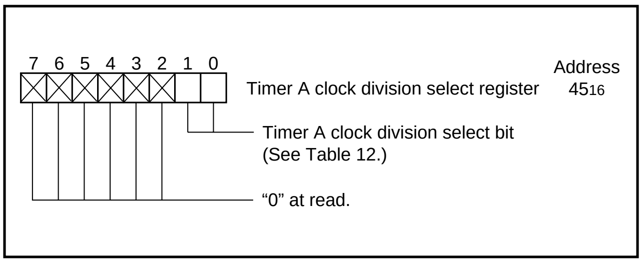
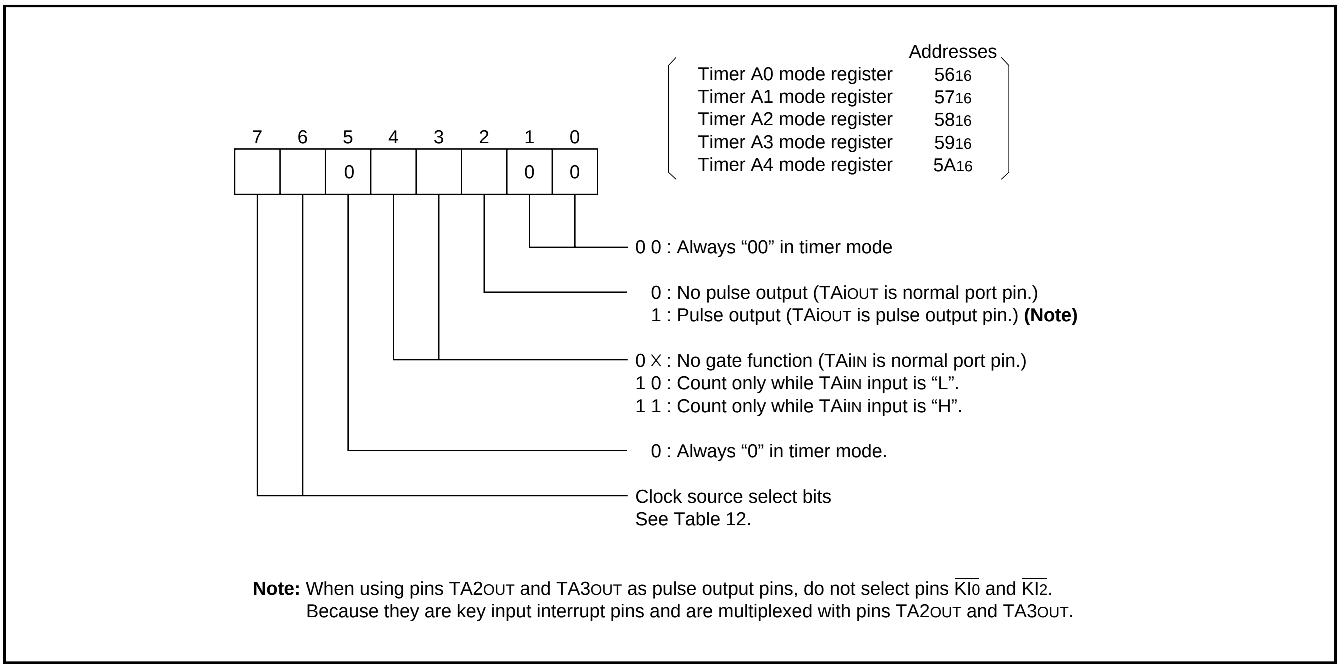

7 6 5 4 3 2 1 0
Address
Timer A clock division select register
45_{16}
Timer A clock division select bit
(See Table 12.)
“0” at read.

Fig. 42 Bit configuration of timer A clock division select register

Table 12. Relationship between timer A clock division select bits, clock source select bits, and count source

Clock source select bits (bits 7 and 6 at addresses 56_{16} to $5A_{16}$)	Timer A clock division select bits (bits 1 and 0 at address 45_{16})			
	00	01	10	11
0 0	f_2	f_1	f_1	Do not select.
0 1	f_{16}	f_{16}	f_{64}	
1 0	f_{64}	f_{64}	f_{512}	
1 1	f_{512}	f_{4096}	f_{4096}	

Note: Timers A0 to A4 use the same clock, which is selected by the timer A clock division select bits.


Addresses
Timer A0 mode register 56_{16}
Timer A1 mode register 57_{16}
Timer A2 mode register 58_{16}
Timer A3 mode register 59_{16}
Timer A4 mode register $5A_{16}$
7 6 5 4 3 2 1 0
0 0 0
0 0 : Always “00” in timer mode
0 : No pulse output (TAi_{OUT} is normal port pin.)
1 : Pulse output (TAi_{OUT} is pulse output pin.) (Note)
0 X : No gate function (TAi_{IN} is normal port pin.)
1 0 : Count only while TAi_{IN} input is “L”.
1 1 : Count only while TAi_{IN} input is “H”.
0 : Always “0” in timer mode.
Clock source select bits
See Table 12.

Note: When using pins $TA2_{OUT}$ and $TA3_{OUT}$ as pulse output pins, do not select pins $\overline{KI_0}$ and $\overline{KI_2}$. Because they are key input interrupt pins and are multiplexed with pins $TA2_{OUT}$ and $TA3_{OUT}$.

Fig. 43 Bit configuration of timer Ai mode register during timer mode

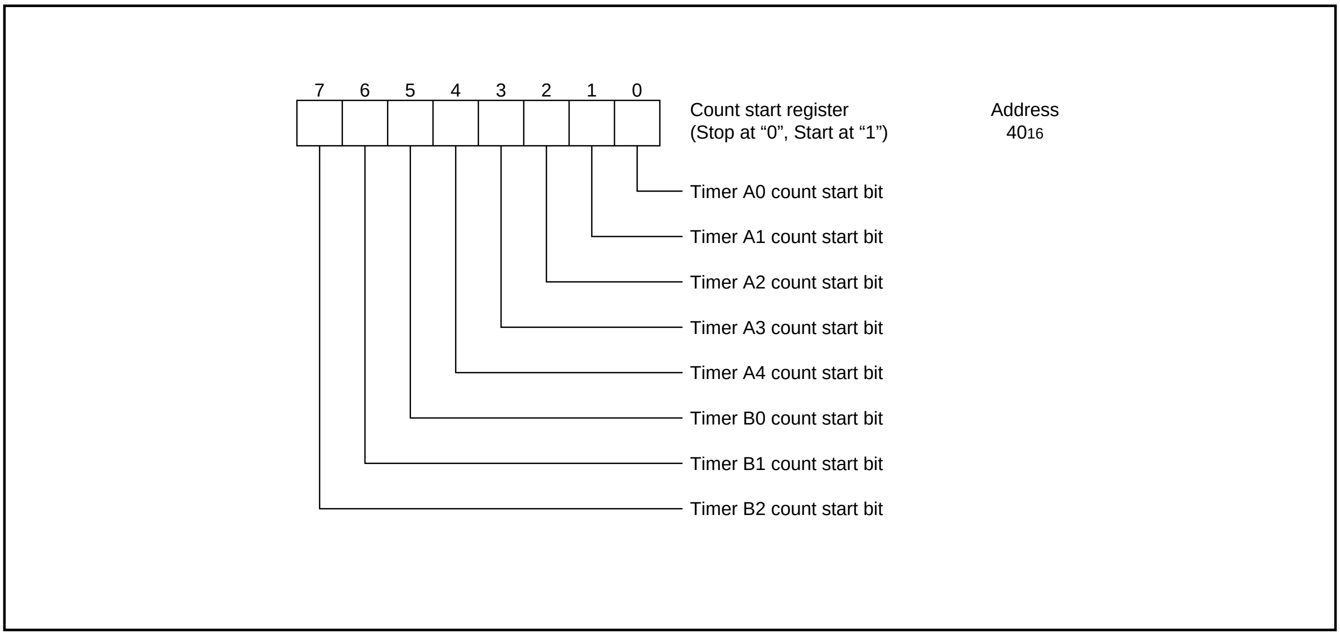


Fig. 44 Bit configuration of count start register

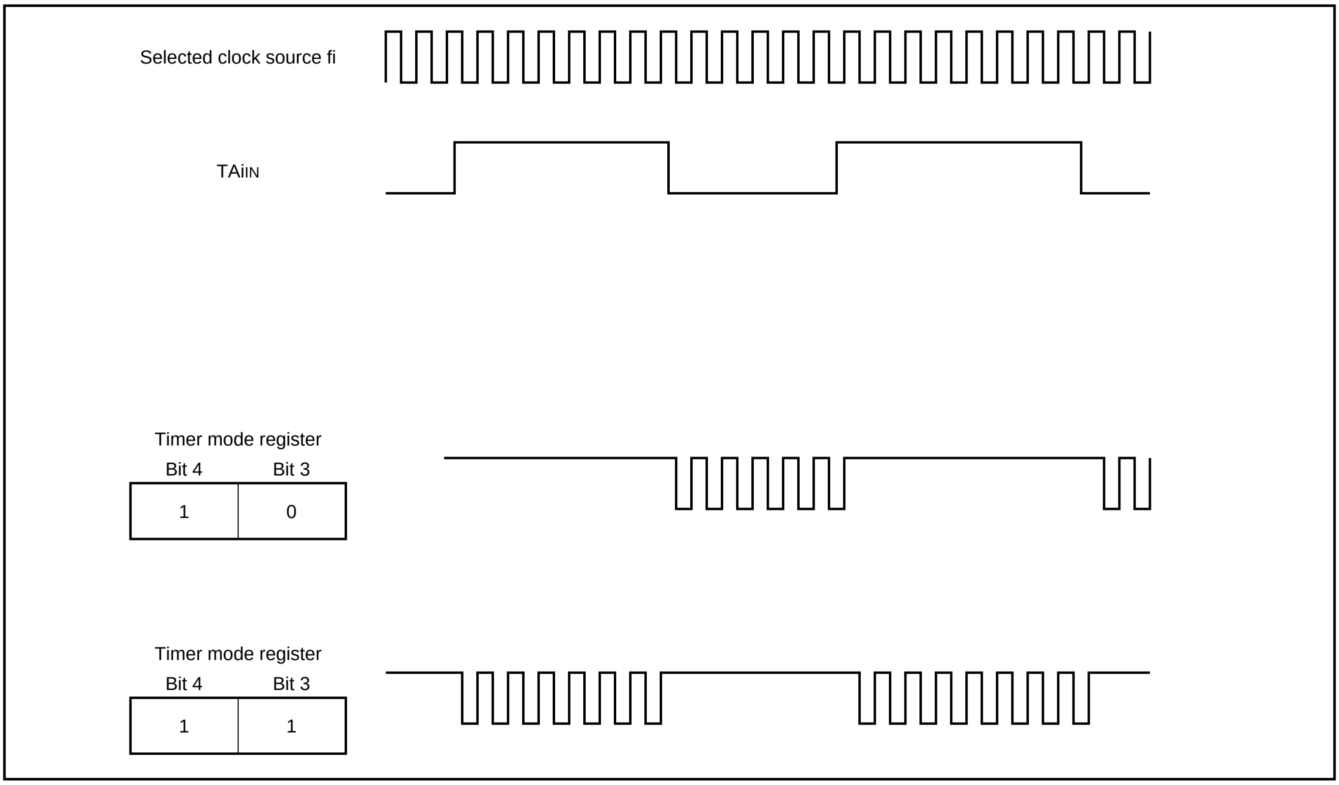


Fig. 45 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 46 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, bit 0 of the timer Ai mode register must be “1” and bits 1 and 5 must be “0”.

The input signal from the TAiIN pin is counted when the count start bit shown in Figure 44 is “1” and counting is stopped when it is “0”. Count is performed at the fall of the input signal when bit 3 is “0” and at the rise of the signal when it is “1”.

In event counter mode, whether to increment or decrement the count can be selected with the up-down bit or the input signal from the TAiOUT pin.

When bit 4 of the timer Ai mode register is “0”, the up-down bit is used to determine whether to increment or decrement the count (decrement when the bit is “0” and increment when it is “1”). Figure 47 shows the bit configuration of the up-down register.

When bit 4 of the timer Ai mode register is “1”, the input signal from the TAiOUT pin is used to determine whether to increment or decrement the count. However, note that bit 2 must be “0” if bit 4 is “1.” It is because if bit 2 is “1”, TAiOUT pin becomes an output pin to output pulses.

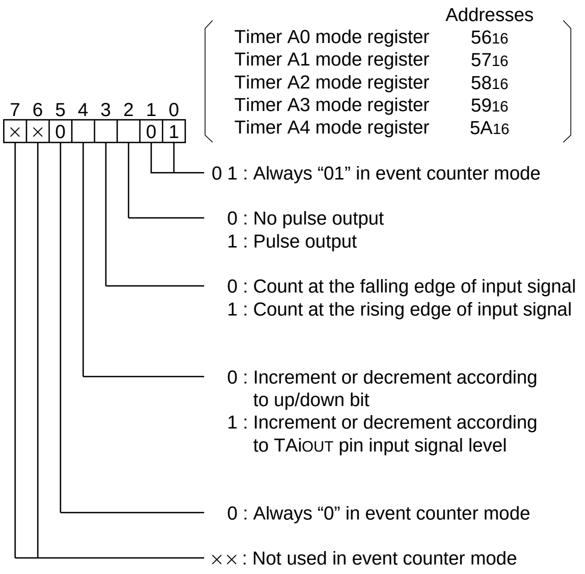
The count is decremented when the input signal from the TAiOUT pin is “L” and incremented when it is “H”. Determine the level of the input signal from the TAiOUT pin before a valid edge is input to the TAiIN pin.

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set when the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

When bit 2 is “1,” each time the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$(increment count), the waveform’s polarity is reversed and is output from TAiOUT pin.

If bit 2 is “0”, TAiOUT pin can be used as a normal port pin.

However, if bit 4 is “1” and the TAiOUT pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 must be “0” unless the output from the TAiOUT pin is to be used to select the count direction.


Addresses
Timer A0 mode register 56₁₆
Timer A1 mode register 5716
Timer A2 mode register 5816
Timer A3 mode register 5916
Timer A4 mode register 5A16
7 6 5 4 3 2 1 0
× × 0 0 1
0 1 : Always “01” in event counter mode
0 : No pulse output
1 : Pulse output
0 : Count at the falling edge of input signal
1 : Count at the rising edge of input signal
0 : Increment or decrement according to up/down bit
1 : Increment or decrement according to TAiOUT pin input signal level
0 : Always “0” in event counter mode
× × : Not used in event counter mode

Note: When using pins TA2OUT and TA3OUT as pulse output pins, do not select pins $\overline{KI_0}$ and $\overline{KI_2}$. Because they are key input interrupt pins and are multiplexed with pins TA2OUT and TA3OUT.

Fig. 46 Bit configuration of timer Ai mode register during event counter mode

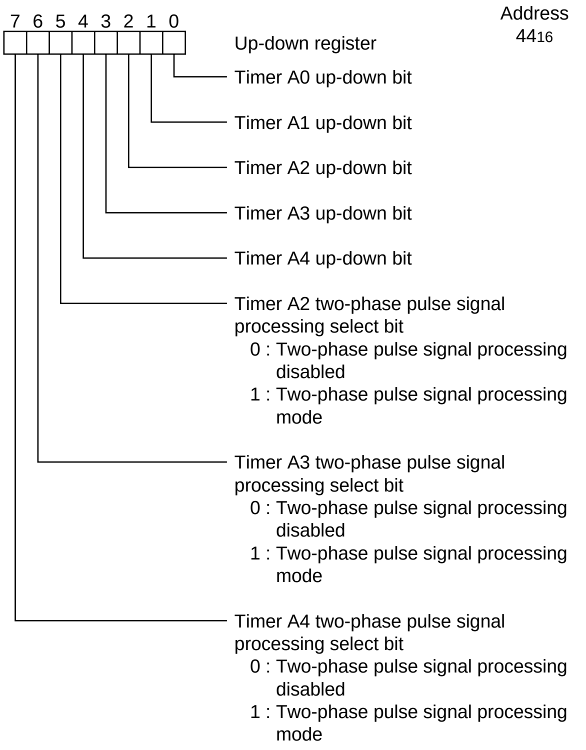

7 6 5 4 3 2 1 0
Up-down register
Address
4416
Timer A0 up-down bit
Timer A1 up-down bit
Timer A2 up-down bit
Timer A3 up-down bit
Timer A4 up-down bit
Timer A2 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Timer A3 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Timer A4 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode

Fig. 47 Bit configuration of up-down register

Data write and data read are performed in the same way as for timer mode. That is, when data is written to timer Ai halted, it is also written to the reload register and the counter. When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time. The counter can be read at any time.

In event counter mode, whether to increment or decrement the counter can also be determined by supplying two kinds of pulses of which phases differ by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timers A2 and A3, and the other uses timer A4. In both processing operations, two pulses described above are input to the TAj$_{OUT}$ (j = 2 to 4) pin and TAj$_{IN}$ pin respectively.

When timers A2 and A3 are used, as shown in Figure 48, the count is incremented when a rising edge is input to the TAk$_{IN}$ pin after the level of TAk$_{OUT}$(k=2, 3) pin changes from “L” to “H”, and when the falling edge is input, the count is decremented.

For timer A4, as shown in Figure 49, when a phase-related pulse with a rising edge input to the TA4$_{IN}$ pin is input after the level of TA4$_{OUT}$ pin changes from “L” to “H”, the count is incremented at the respective rising edge and falling edge of the TA4$_{OUT}$ pin and TA4$_{IN}$ pin.

When a phase-related pulse with a falling edge input to the TA4$_{OUT}$ pin is input after the level of TA4$_{IN}$ pin changes from “H” to “L”, the count is decremented at the respective rising edge and falling edge of the TA4$_{IN}$ pin and TA4$_{OUT}$ pin. When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to “1” and bits 1, 2, 3, and 5 must be “0”. Bits 6 and 7 are ignored. (See Figure 50.) Note that bits 5, 6, and 7 of the up-down register (address 44$_{16}$) are the two-phase pulse signal processing select bits for timers A2, A3 and A4 respectively. Each timer operates in normal event counter mode when the corresponding bit is “0” and performs two-phase pulse signal processing when it is “1”.

Count is started by setting the count start bit to “1”. Data write and read are performed in the same way as for normal event counter mode. Note that the direction register of the input port must be set to input mode because two kinds of pulse signals, described above, are input. Also, there can be no pulse output in this mode.

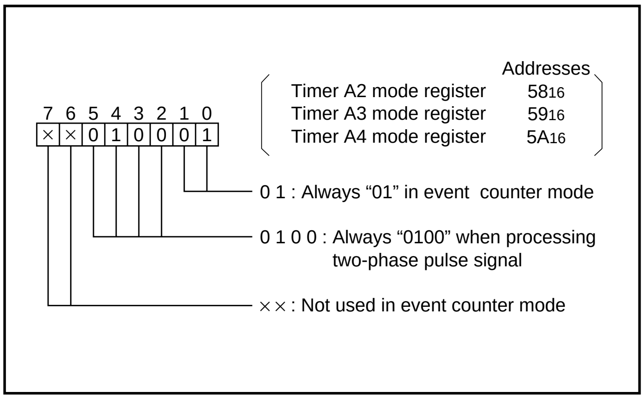

Addresses
Timer A2 mode register 58$_{16}$
Timer A3 mode register 59$_{16}$
Timer A4 mode register 5A$_{16}$
7 6 5 4 3 2 1 0
× × 0 1 0 0 0 1
0 1 : Always “01” in event counter mode
0 1 0 0 : Always “0100” when processing two-phase pulse signal
× × : Not used in event counter mode

Fig. 50 Bit configuration of timer Aj mode register when performing two-phase pulse signal processing in event counter mode

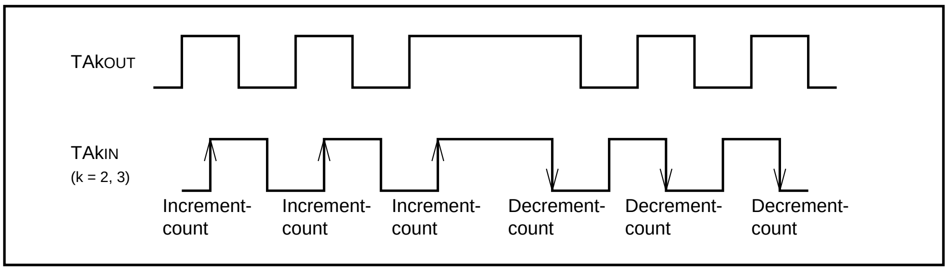


Fig. 48 Two-phase pulse processing operation of timers A2 and A3

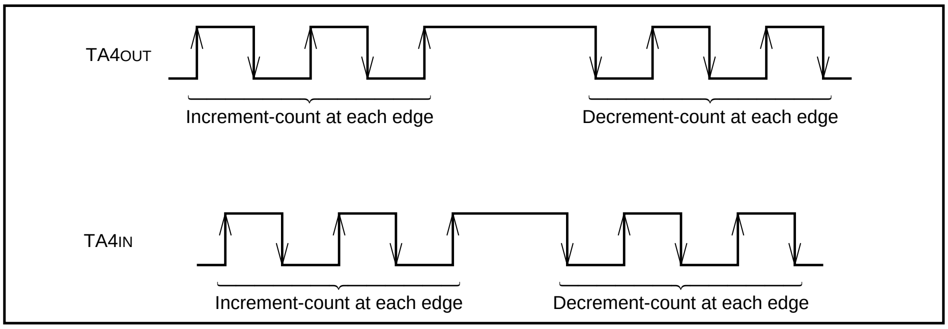

TA4$_{OUT}$
Increment-count at each edge
Decrement-count at each edge
TA4$_{IN}$
Increment-count at each edge
Decrement-count at each edge

Fig. 49 Two-phase pulse processing operation of timer A4

(3) One-shot pulse mode [10]

Figure 51 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be “0” and bit 1 and bit 2 must be “1”.
The trigger is enabled when the count start bit is “1”. The trigger can be generated by software or it can be input from the TAiIN pin. Software trigger is selected when bit 4 is “0” and the input signal from the TAiIN pin is used as the trigger when it is “1“.
Bit 3 is used to determine whether to trigger at the fall of the trigger signal or at the rise. The trigger is at the fall of the trigger signal when bit 3 is “0” and at the rise of the trigger signal when it is “1”.
Software trigger is generated by setting “1” to a bit in the one-shot start register. Each bit corresponds to each timer.
Figure 52 shows the bit configuration of the one-shot start register.
As shown in Figure 53, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7 and the contents of the timer A clock division select register. (Set Table 12.)
If the contents of the counter is not 0000_{16}, the TAiOUT pin goes “H” when a trigger signal is received. The count direction is decrement. When the counter reaches 0001_{16}, the TAiOUT pin goes “L” and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received.
The output pulse width is

$$\frac{1}{\text{pulse frequency of the selected clock}} \times \text{(counter's value at the time of trigger).}$$

If the count start flag is “0”, TAiOUT goes “L”. Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start bit.
As shown in Figure 54, a trigger signal can be received before the operation for the previous trigger signal is completed. In this case, the contents of the reload register is transferred to the counter by the trigger and then that value is decremented.
Except when retriggering while operating, the contents of the reload register are not transferred to the counter by triggering.
When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.
Data write is performed in the same way as for timer mode.
When data is written in timer Ai halted, it is also written to the reload register and the counter.
When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.
Undefined data is read when timer Ai is read.

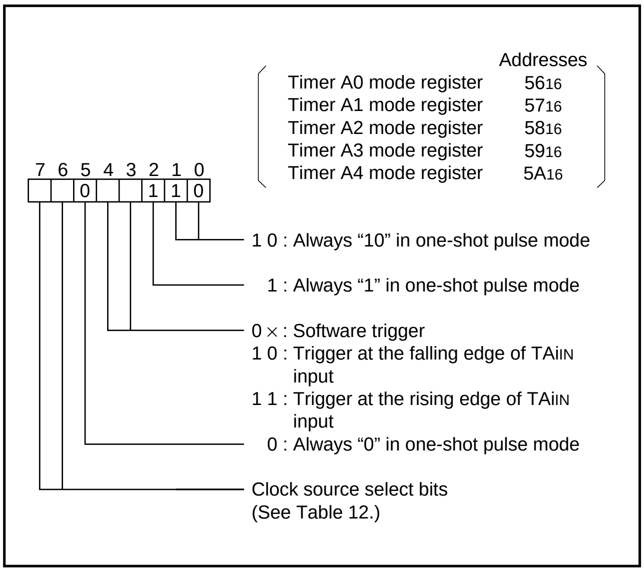

Addresses
Timer A0 mode register 56₁₆
Timer A1 mode register 57₁₆
Timer A2 mode register 58₁₆
Timer A3 mode register 59₁₆
Timer A4 mode register 5A₁₆
7 6 5 4 3 2 1 0
0 1 1 0
1 0 : Always “10” in one-shot pulse mode
1 : Always “1” in one-shot pulse mode
0 × : Software trigger
1 0 : Trigger at the falling edge of TAiIN input
1 1 : Trigger at the rising edge of TAiIN input
0 : Always “0” in one-shot pulse mode
Clock source select bits
(See Table 12.)

Fig. 51 Bit configuration of timer Ai mode register during one-shot pulse mode

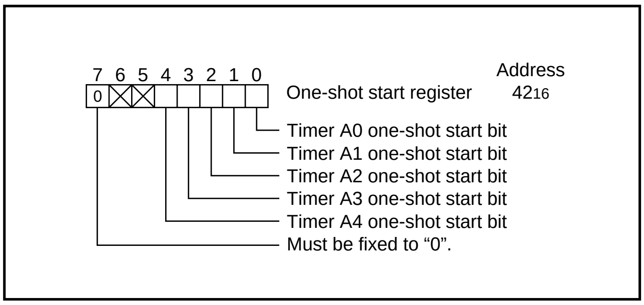

Address
7 6 5 4 3 2 1 0
0
One-shot start register 42₁₆
Timer A0 one-shot start bit
Timer A1 one-shot start bit
Timer A2 one-shot start bit
Timer A3 one-shot start bit
Timer A4 one-shot start bit
Must be fixed to “0”.

Fig. 52 Bit configuration of one-shot start register

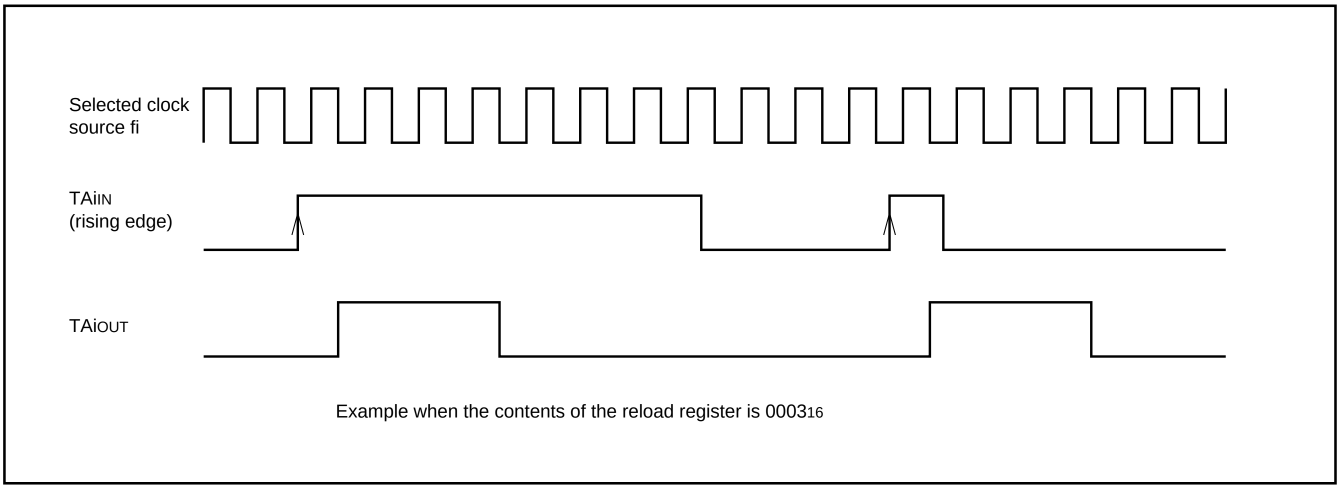


Fig. 53 Pulse output example when external rising edge is selected

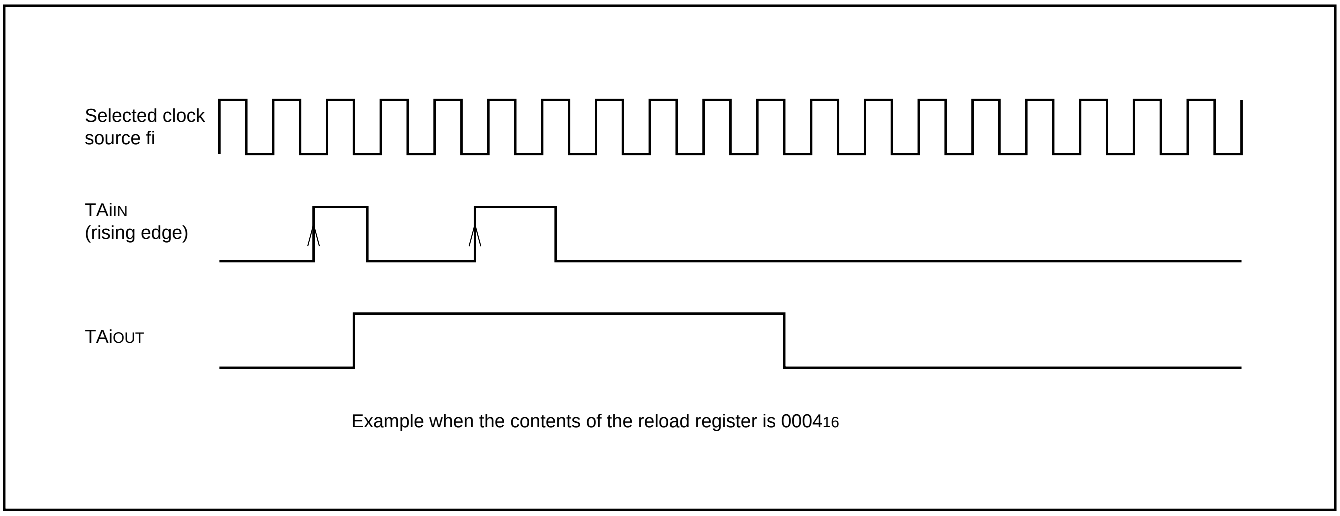


Fig. 54 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode [11]

Figure 55 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to “1”.

Bit 5 is used to determine whether to perform 16-bit length pulse width modulator or 8-bit length pulse width modulator. 16-bit length pulse width modulator is selected when bit 5 is “0” and 8-bit length pulse width modulator is selected when it is “1”. The 16-bit length pulse width modulator is described first.

The pulse width modulator can be started with a software trigger or with an input signal from a TAi_{IN} pin (external trigger).

The software trigger mode is selected when bit 4 is “0”.

Pulse width modulator is started and a pulse is output from TAi_{OUT} when the count start bit is set to “1”.

The external trigger mode is selected when bit 4 is “1”.

Pulse width modulation starts when a trigger signal is input from the TAi_{IN} pin when the count start bit is “1”. Whether to trigger at the fall or rise of the trigger signal is determined by bit 3. The trigger is at the fall of the trigger signal when bit 3 is “0” and at the rise when it is “1”.

When data is written to timer Ai with the pulse width modulator halted, it is written to the reload register and the counter.

Then when the count start bit is set to “1” and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 56 is output continuously.

Once modulation is started, triggers are not accepted. If the value in the reload register is m, the duration “H” of pulse is

$$\frac{1}{\text{selected clock frequency}} \times m$$

and the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (2^{16} - 1).$$

An interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set at each fall of the output pulse.

The width of the output pulse is changed by updating timer data. The update can be performed at any time. The output pulse width is changed at the rise of the pulse after data is written to the timer.

The contents of the reload register are transferred to the counter just before the rise of the next pulse so that the pulse width is changed from the next output pulse.

Undefined data is read when timer Ai is read.

The 8-bit length pulse width modulator is described next.

The 8-bit length pulse width modulator is selected when the timer Ai mode register bit 5 is “1”.

The reload register and the counter are both divided into 8-bit halves.

The low-order 8 bits function as a prescaler and the high-order 8 bits function as the 8-bit length pulse width modulator. The prescaler counts the clock selected by bits 6, 7, and the contents of the timer A clock division select register. (See Table 12.) A pulse is generated when the counter reaches 0000_{16} as shown in Figure 57. At the same time, the contents of the reload register is transferred to the counter and count is continued.

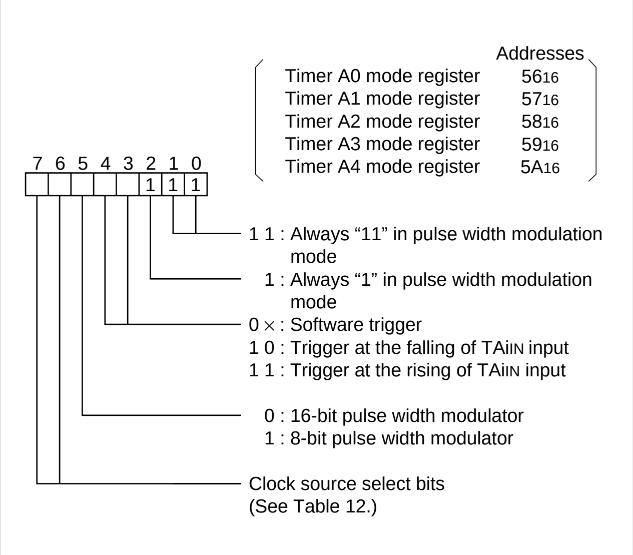

Addresses
Timer A0 mode register 56₁₆
Timer A1 mode register 57₁₆
Timer A2 mode register 58₁₆
Timer A3 mode register 59₁₆
Timer A4 mode register 5A₁₆
7 6 5 4 3 2 1 0
1 1 1
1 1 : Always “11” in pulse width modulation mode
1 : Always “1” in pulse width modulation mode
0 × : Software trigger
1 0 : Trigger at the falling of TAiIN input
1 1 : Trigger at the rising of TAiIN input
0 : 16-bit pulse width modulator
1 : 8-bit pulse width modulator
Clock source select bits
(See Table 12.)

Fig. 55 Bit configuration of timer Ai mode register during pulse width modulation mode

Therefore, if the low-order 8 bits of the reload register are n, the period of the generated pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1).$$

The high-order 8 bits function as an 8-bit length pulse width modulator using this pulse as input. The operation is the same as for 16-bit length pulse width modulator except that the length is 8 bits. If the high-order 8 bits of the reload register are m, the duration "H" of pulse is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times m.$$

And the output pulse period is

$$\frac{1}{\text{selected clock frequency}} \times (n+1) \times (2^8–1).$$

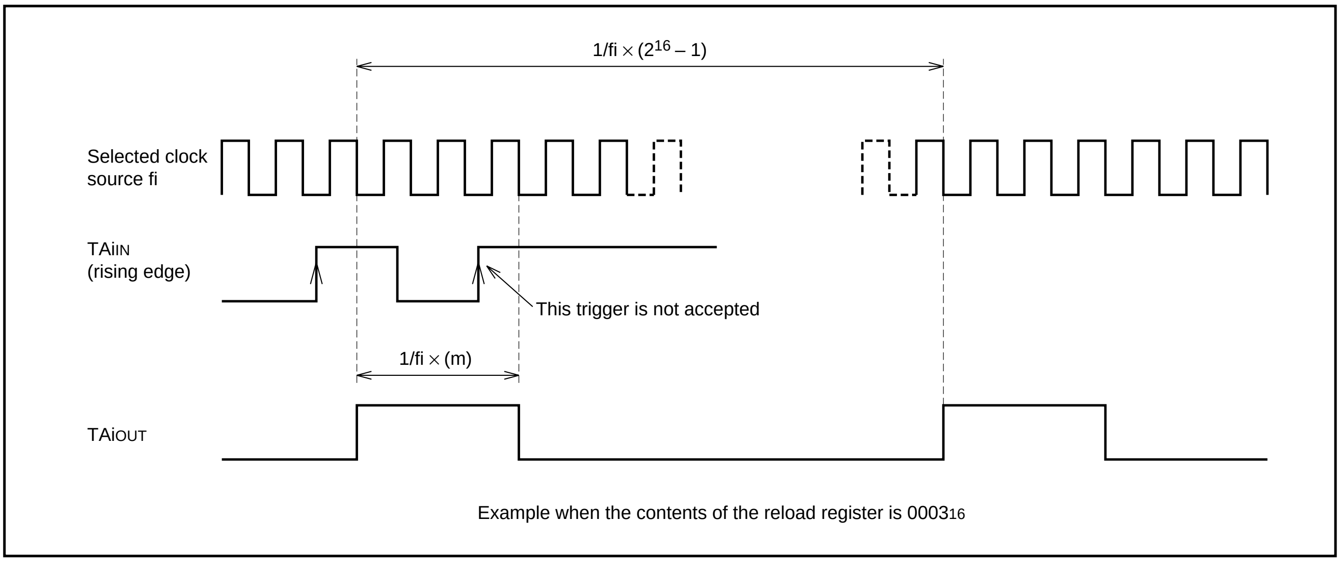

1/fi × (2^16 – 1)
Selected clock source fi
TAiIN (rising edge)
This trigger is not accepted
1/fi × (m)
TAiOUT
Example when the contents of the reload register is 0003₁₆

Fig. 56 16-bit length pulse width modulator output pulse example

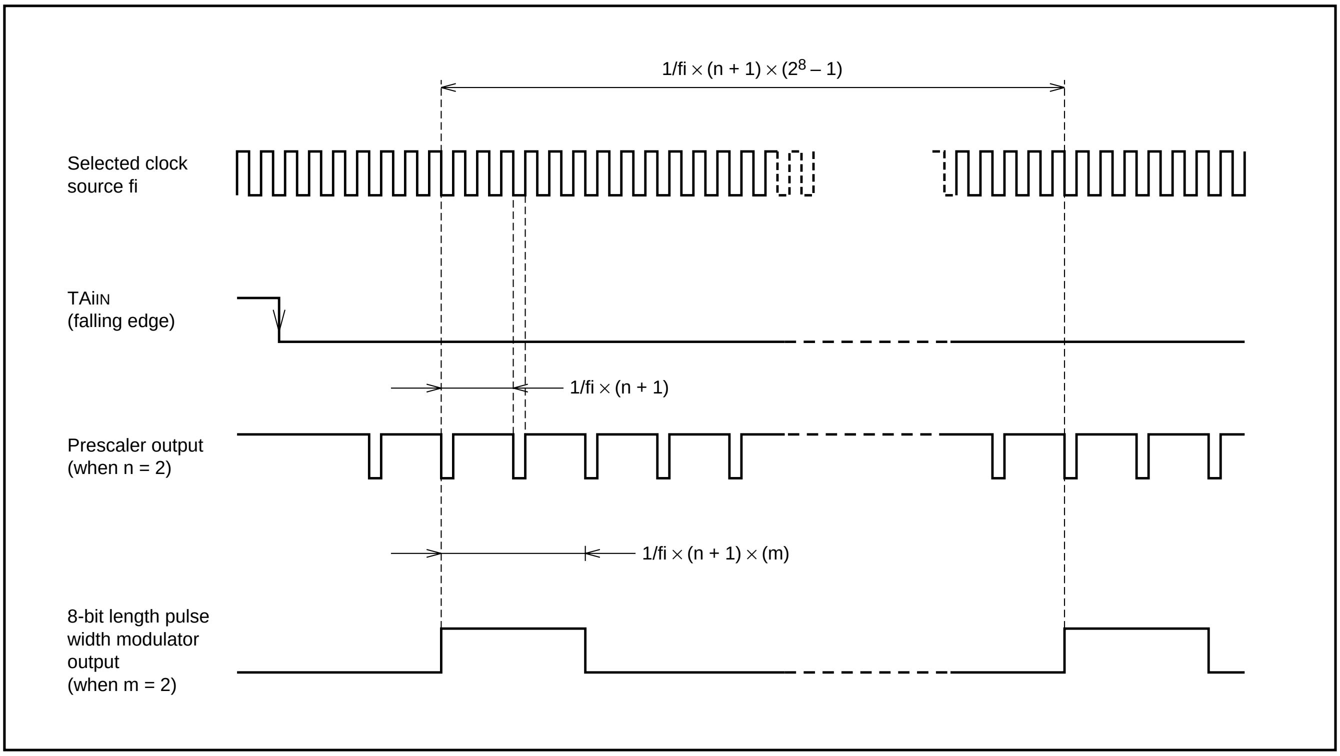

1/fi × (n + 1) × (2^8 – 1)
Selected clock source fi
TAiIN (falling edge)
1/fi × (n + 1)
Prescaler output (when n = 2)
1/fi × (n + 1) × (m)
8-bit length pulse width modulator output (when m = 2)

Fig. 57 8-bit length pulse width modulator output pulse example

TIMER B

Figure 58 shows a block diagram of timer B.

Timer B has three modes: timer mode, event counter mode, and pulse period measurement/pulse width measurement mode. The mode is selected with bits 0 and 1 of the timer Bi mode register (i=0 to 2). Each of these modes is described below.

(1) Timer mode [00]

Figure 59 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0 and 1 of the timer Bi mode register must always be "0" in timer mode.

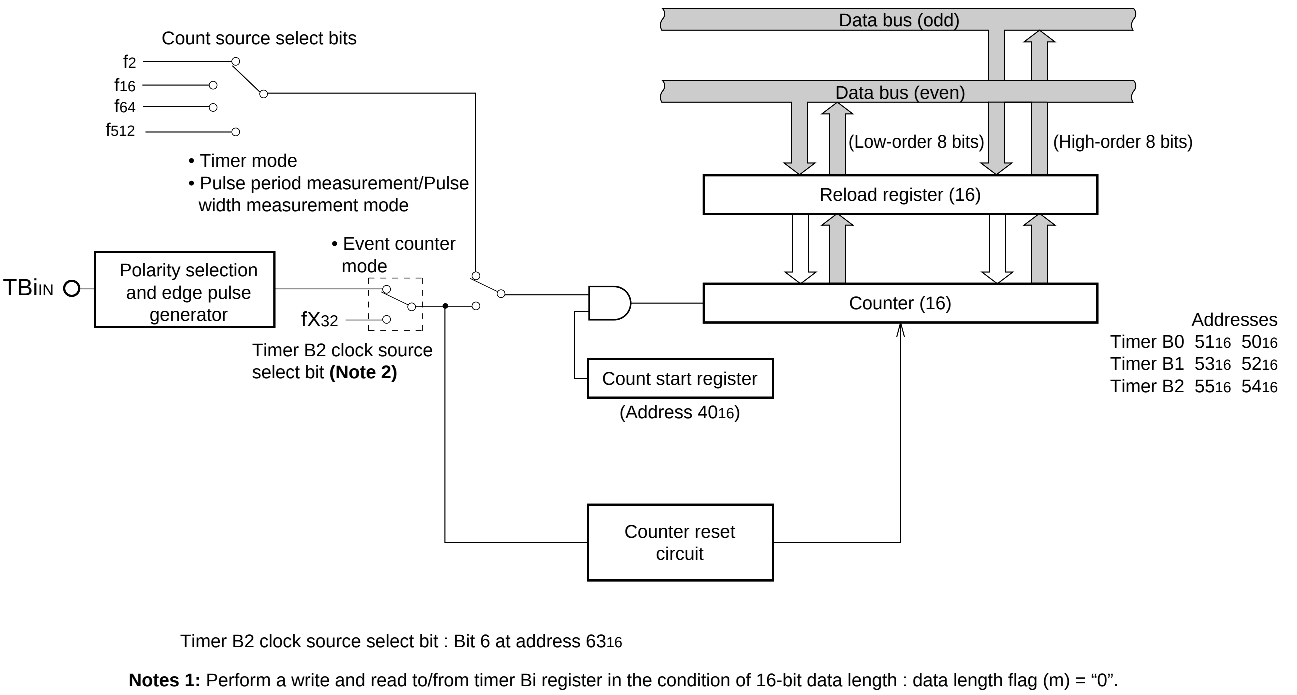
Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start bit is "1" and stops when "0".

As shown in Figure 44, the timer Bi count start bit is at the same address as the timer Ai count start bit. The count is decremented, an interrupt occurs, and the interrupt request bit in the timer Bi interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

Timer Bi does not have a pulse output function or a gate function like timer A.

When data is written to timer Bi halted, it is written to the reload register and the counter. When data is written to timer Bi which is busy, the data is written to the reload register, but not to the counter. The new data is reloaded from the reload register to the counter at the next reload time and counting continues.

The contents of the counter can be read at any time.



Timer B2 clock source select bit : Bit 6 at address 63_{16}

Notes 1: Perform a write and read to/from timer Bi register in the condition of 16-bit data length : data length flag (m) = "0".
2: Only for timer B2, a count source in the event counter mode can be selected.

Fig. 58 Block diagram of timer B

(2) Event counter mode [01]

Figure 60 shows the bit configuration of the timer Bi mode register during event counter mode. In event counter mode, bit 0 in the timer Bi mode register must be “1” and bit 1 must be “0”.

The input signal from the TBi$_{IN}$ pin is counted when the count start bit is “1” and counting is stopped when it is “0”.

Count is performed at the fall of the input signal when bits 2, and 3 are “0” and at the rise of the input signal when bit 3 is “0” and bit 2 is “1”.

When bit 3 is “1” and bit 2 is “0”, count is performed at the rise and fall of the input signal.

Data write, data read and timer interrupt are performed in the same way as for timer mode.

Only for timer B2, when the timer B2 clock source select bit of the particular function select register 1 (bit 6 at address 63_{16}) = “1” in the event counter mode, fX_{32} can be selected. (When this bit is “0”, an input signal from pin TB2$_{IN}$ becomes the count source as described above.) For the bit configuration of the particular function select register 1, refer to the section on the standby function.

Note: fX_{32} = f(X$_{IN}$)/32

(3) Pulse period measurement/Pulse width measurement mode [10]

Figure 61 shows the bit configuration of the timer Bi mode register during pulse period measurement/pulse width measurement mode. In pulse period measurement/pulse width measurement mode, bit 0 must be “0” and bit 1 must be “1”. Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start bit is “1” and counting stops when it is “0”.

The pulse period measurement mode is selected when bit 3 is “0”. In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBi$_{IN}$ pin to the next fall or at the rise of the input signal to the next rise; the result is stored in the reload register. In this case, the reload register acts as a buffer register.

When bit 2 is “0”, the clock is counted from the fall of the input signal to the next fall. When bit 2 is “1“, the clock is counted from the rise of the input signal to the next rise.

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 62, when the fall of the input signal from TBi$_{IN}$ pin is detected, the contents of the counter is transferred to the reload register. Next, the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

After the contents of the counter is transferred to the reload register, an interrupt request signal is generated and the interrupt request bit in the timer Bi interrupt control register is set. However, no interrupt request signal is generated when the contents of the counter is transferred first to the reload register after the count start bit is set to “1”.

When bit 3 is “1”, the pulse width measurement mode is selected. Pulse width measurement mode is the same as the pulse period measurement mode except that the clock is counted from the fall of the TBi$_{IN}$ pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 63.

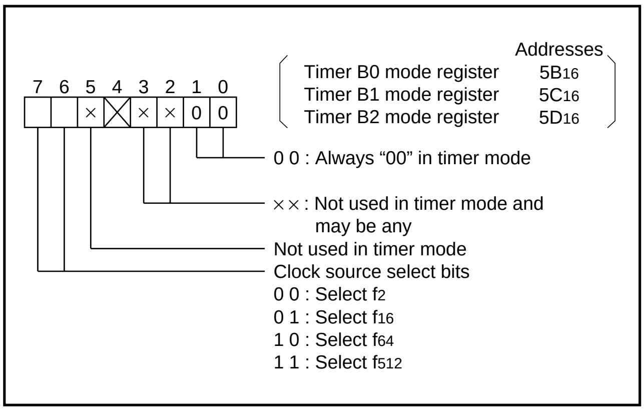

7 6 5 4 3 2 1 0
× × × 0 0
Addresses
Timer B0 mode register 5B16
Timer B1 mode register 5C16
Timer B2 mode register 5D16
0 0 : Always “00” in timer mode
× × : Not used in timer mode and may be any
Not used in timer mode
Clock source select bits
0 0 : Select f2
0 1 : Select f16
1 0 : Select f64
1 1 : Select f512

Fig. 59 Bit configuration of timer Bi mode register during timer mode

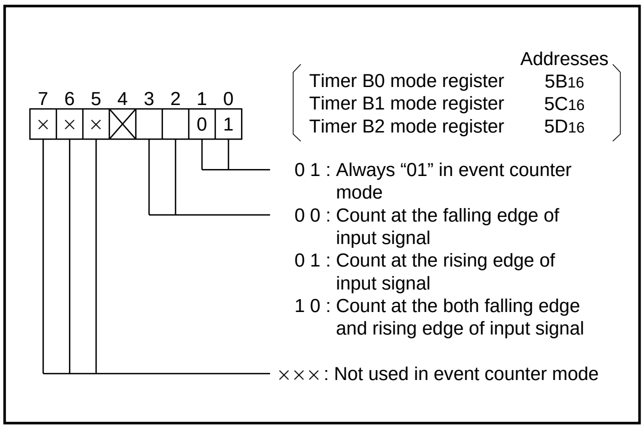

7 6 5 4 3 2 1 0
× × × 0 1
Addresses
Timer B0 mode register 5B16
Timer B1 mode register 5C16
Timer B2 mode register 5D16
0 1 : Always “01” in event counter mode
0 0 : Count at the falling edge of input signal
0 1 : Count at the rising edge of input signal
1 0 : Count at the both falling edge and rising edge of input signal
× × × : Not used in event counter mode

Fig. 60 Bit configuration of timer Bi mode register during event counter mode

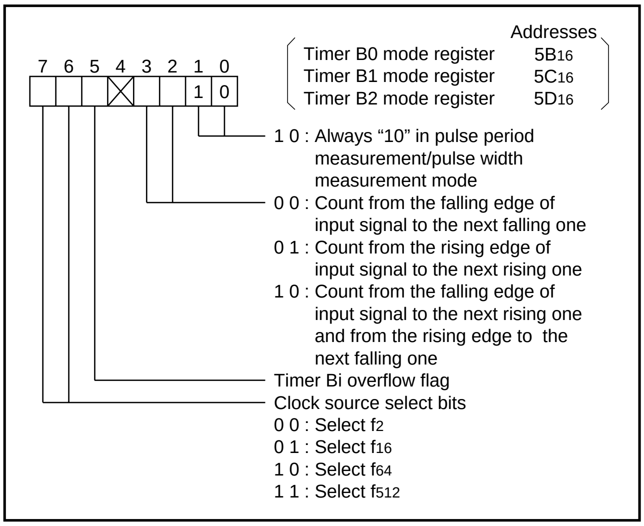

7 6 5 4 3 2 1 0
1 0
Addresses
Timer B0 mode register 5B16
Timer B1 mode register 5C16
Timer B2 mode register 5D16
1 0 : Always “10” in pulse period measurement/pulse width measurement mode
0 0 : Count from the falling edge of input signal to the next falling one
0 1 : Count from the rising edge of input signal to the next rising one
1 0 : Count from the falling edge of input signal to the next rising one and from the rising edge to the next falling one
Timer Bi overflow flag
Clock source select bits
0 0 : Select f2
0 1 : Select f16
1 0 : Select f64
1 1 : Select f512

Fig. 61 Bit configuration of timer Bi mode register during pulse period measurement/pulse width measurement mode

When timer Bi is read, the contents of the reload register is read. Note that in this mode, the interval between the fall of the TBi_{IN} pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.

Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}, which indicates that a pulse width or pulse period is longer than that which can be measured by a 16-bit length.

This flag is cleared by writing data to the corresponding timer Bi mode register. This flag is set to "1"at reset.

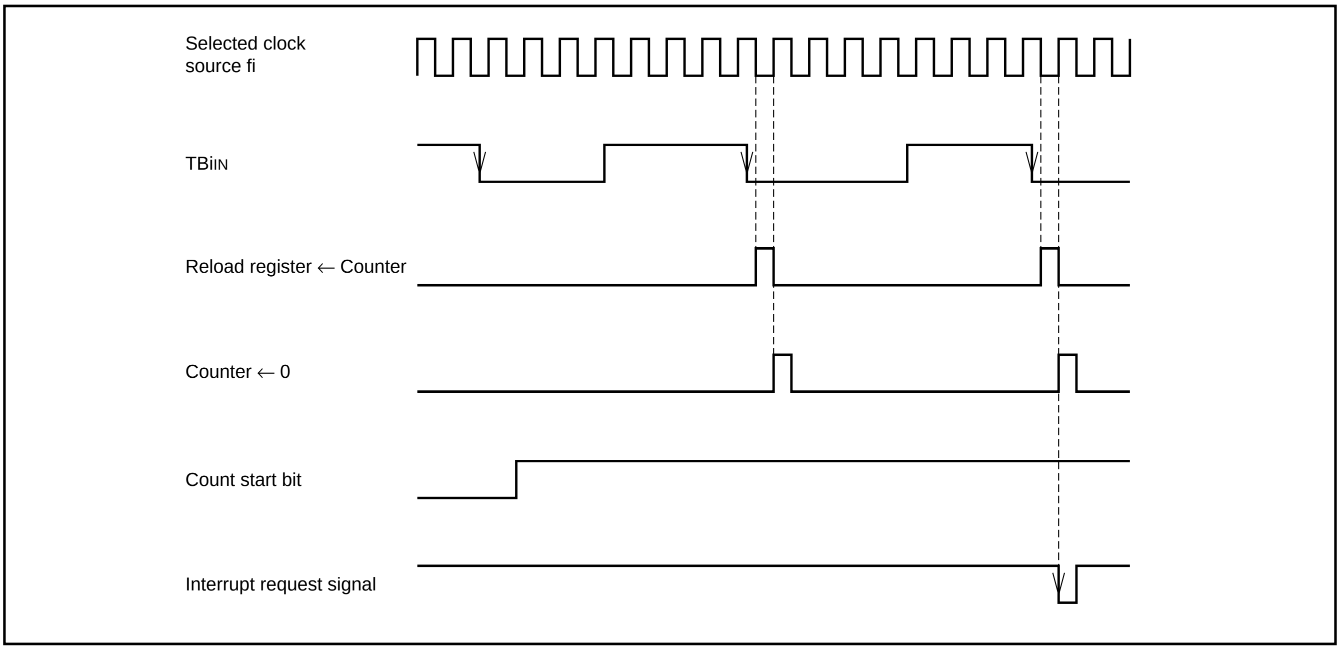


Fig. 62 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

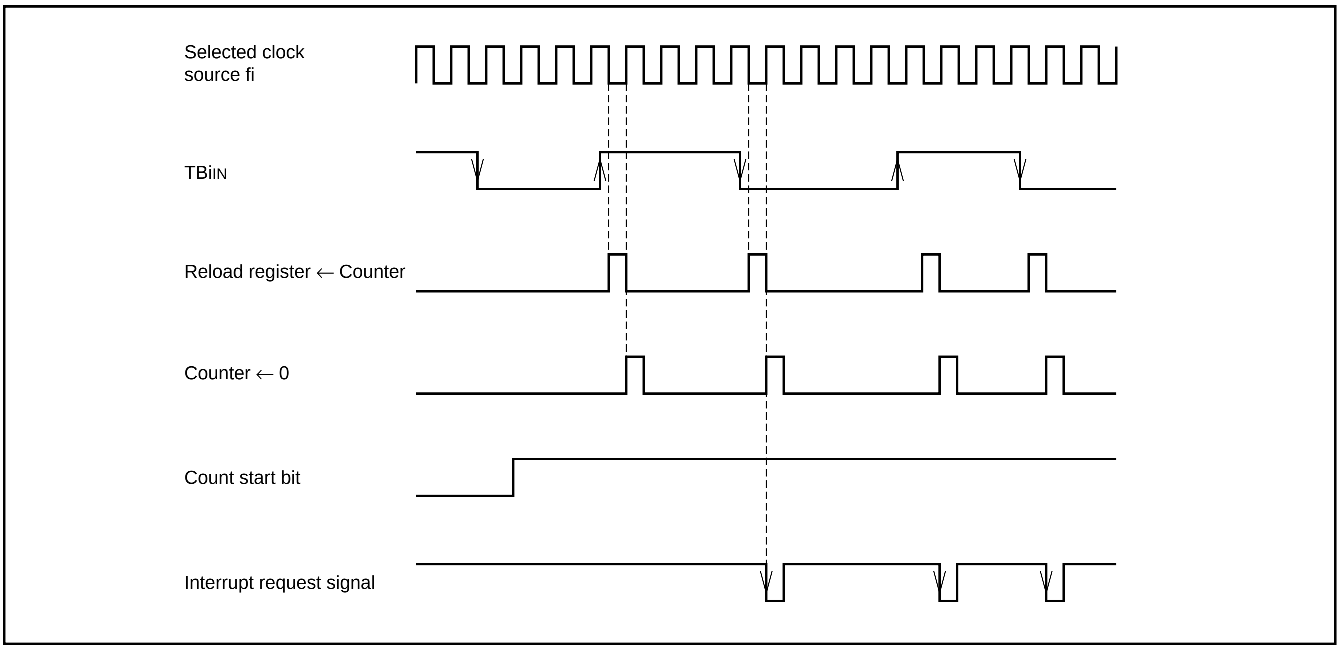


Fig. 63 Pulse width measurement mode operation

SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 64 shows a block diagram of the serial I/O ports.
Bits 0 to 2 of the UARTi(i = 0,1) transmit/receive mode register shown in Figure 65 are used to determine whether to use port P8 as a programmable I/O port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port which uses start and stop bits.
Figures 66 and 67 show the block diagrams of the receiver/transmitter .
Figure 68 shows the bit configuration of the UARTi transmit/receive control register.
Each communication method is described below.

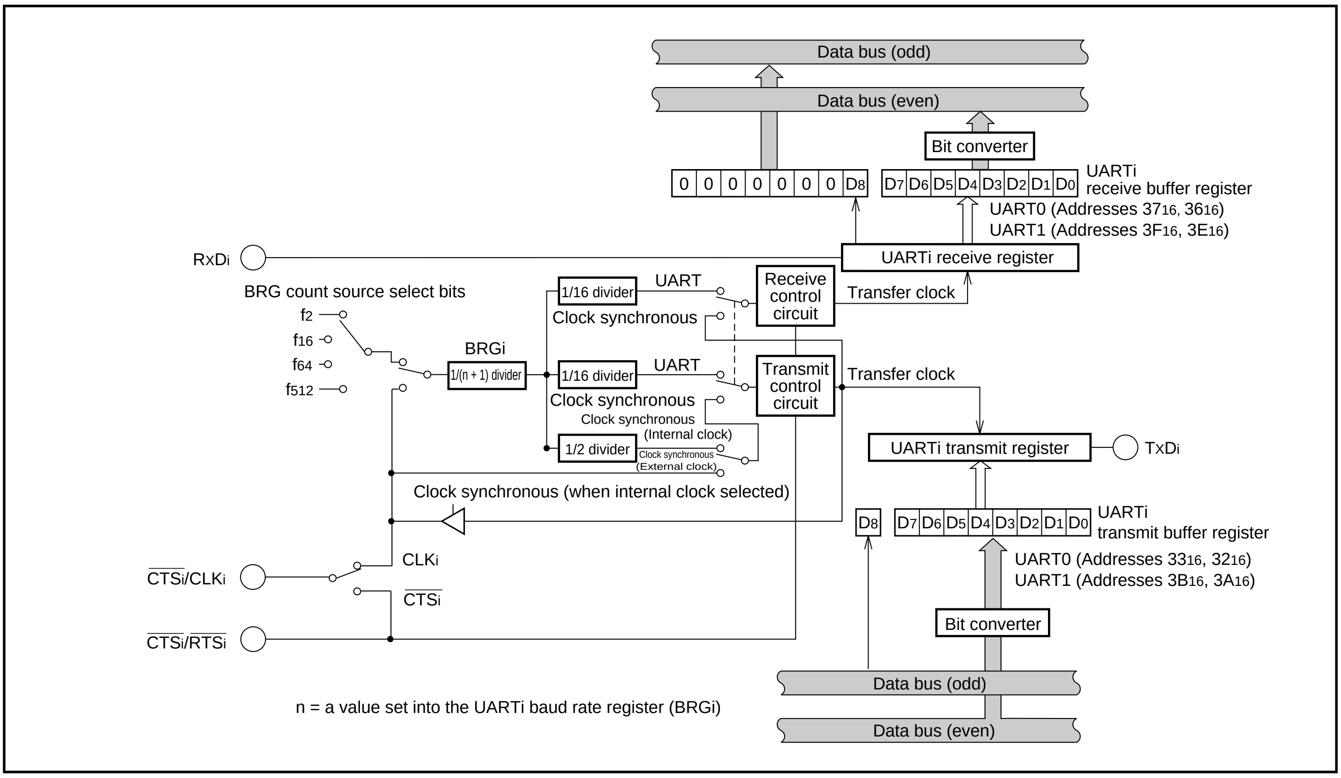
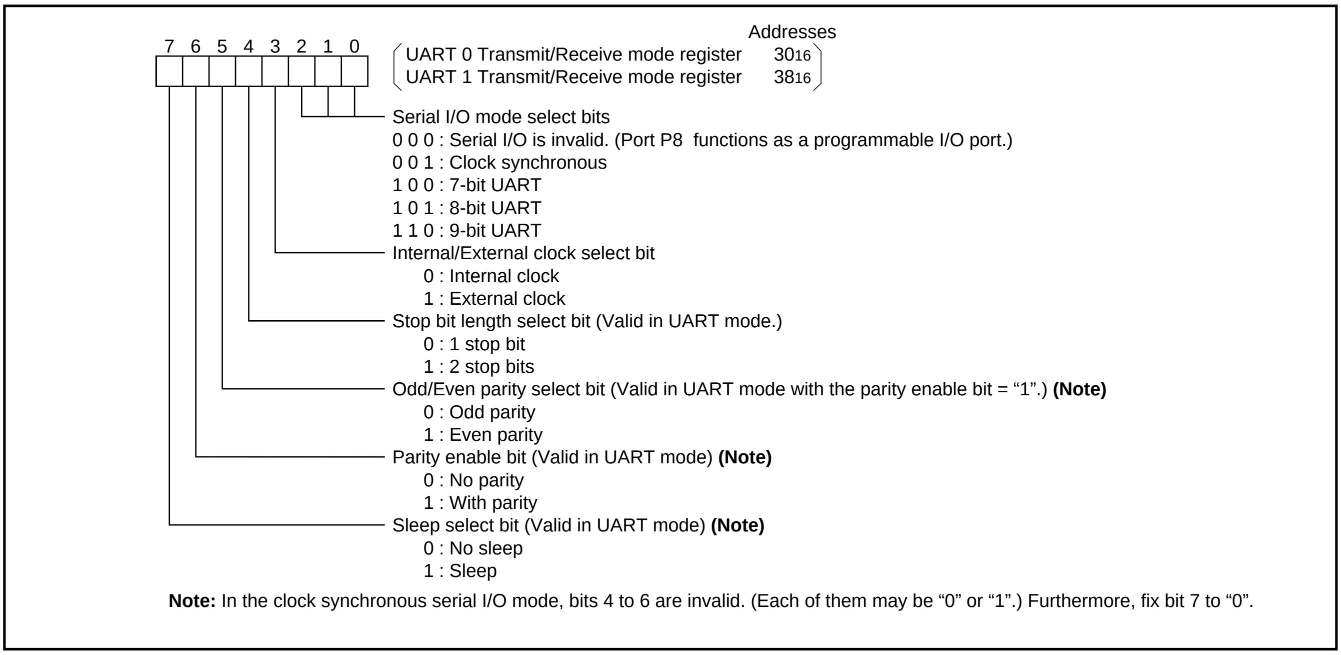


Fig. 64 Block diagram of serial I/O port



	Addresses
UART 0 Transmit/Receive mode register	30_{16}
UART 1 Transmit/Receive mode register	38_{16}

Serial I/O mode select bits
0 0 0 : Serial I/O is invalid. (Port P8 functions as a programmable I/O port.)
0 0 1 : Clock synchronous
1 0 0 : 7-bit UART
1 0 1 : 8-bit UART
1 1 0 : 9-bit UART

Internal/External clock select bit
0 : Internal clock
1 : External clock

Stop bit length select bit (Valid in UART mode.)
0 : 1 stop bit
1 : 2 stop bits

Odd/Even parity select bit (Valid in UART mode with the parity enable bit = “1”.) **(Note)**
0 : Odd parity
1 : Even parity

Parity enable bit (Valid in UART mode) **(Note)**
0 : No parity
1 : With parity

Sleep select bit (Valid in UART mode) **(Note)**
0 : No sleep
1 : Sleep

Note: In the clock synchronous serial I/O mode, bits 4 to 6 are invalid. (Each of them may be “0” or “1”.) Furthermore, fix bit 7 to “0”.

Fig. 65 Bit configuration of UARTi transmit/receive mode register

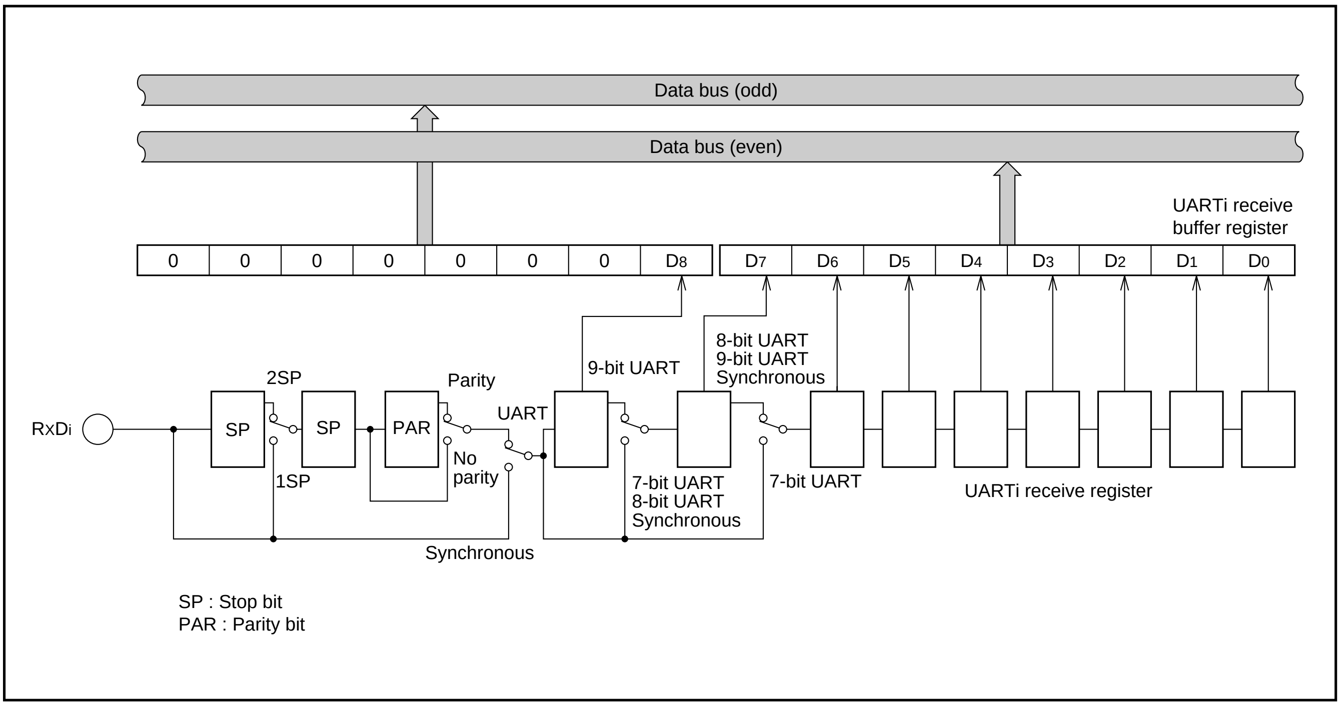


Fig. 66 Block diagram of receiver

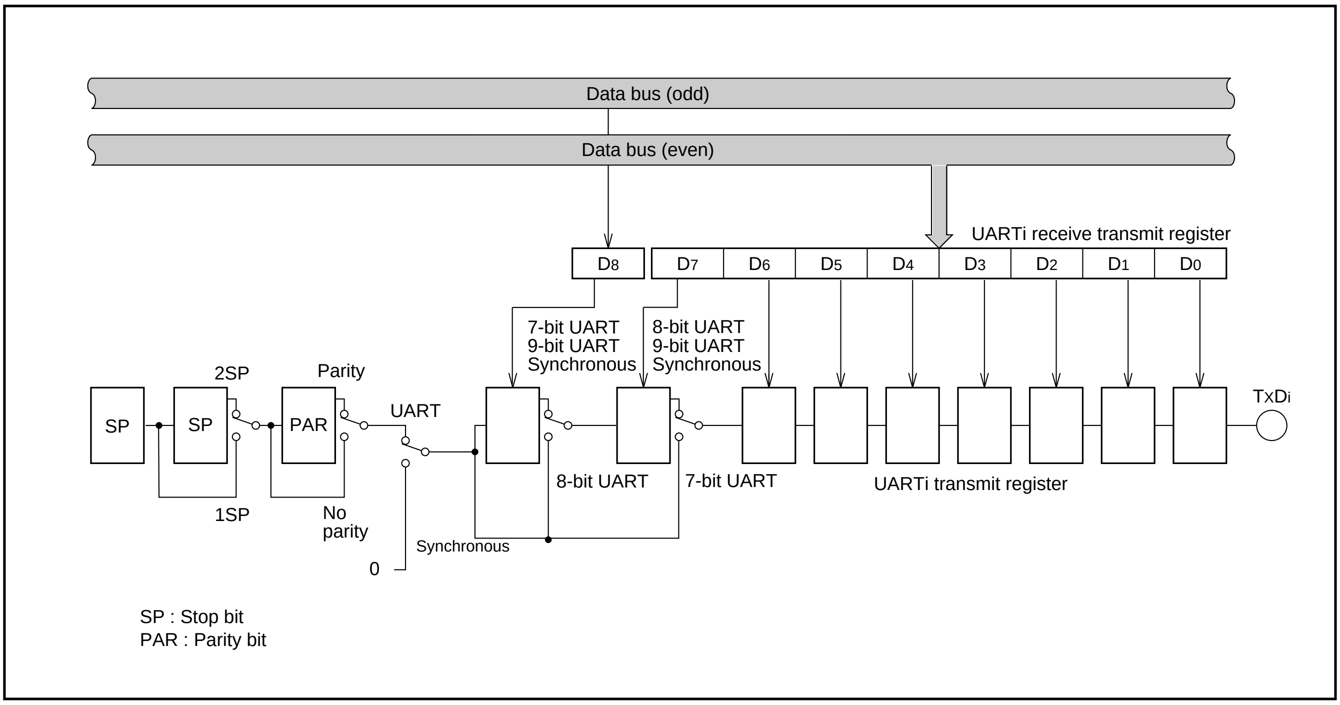
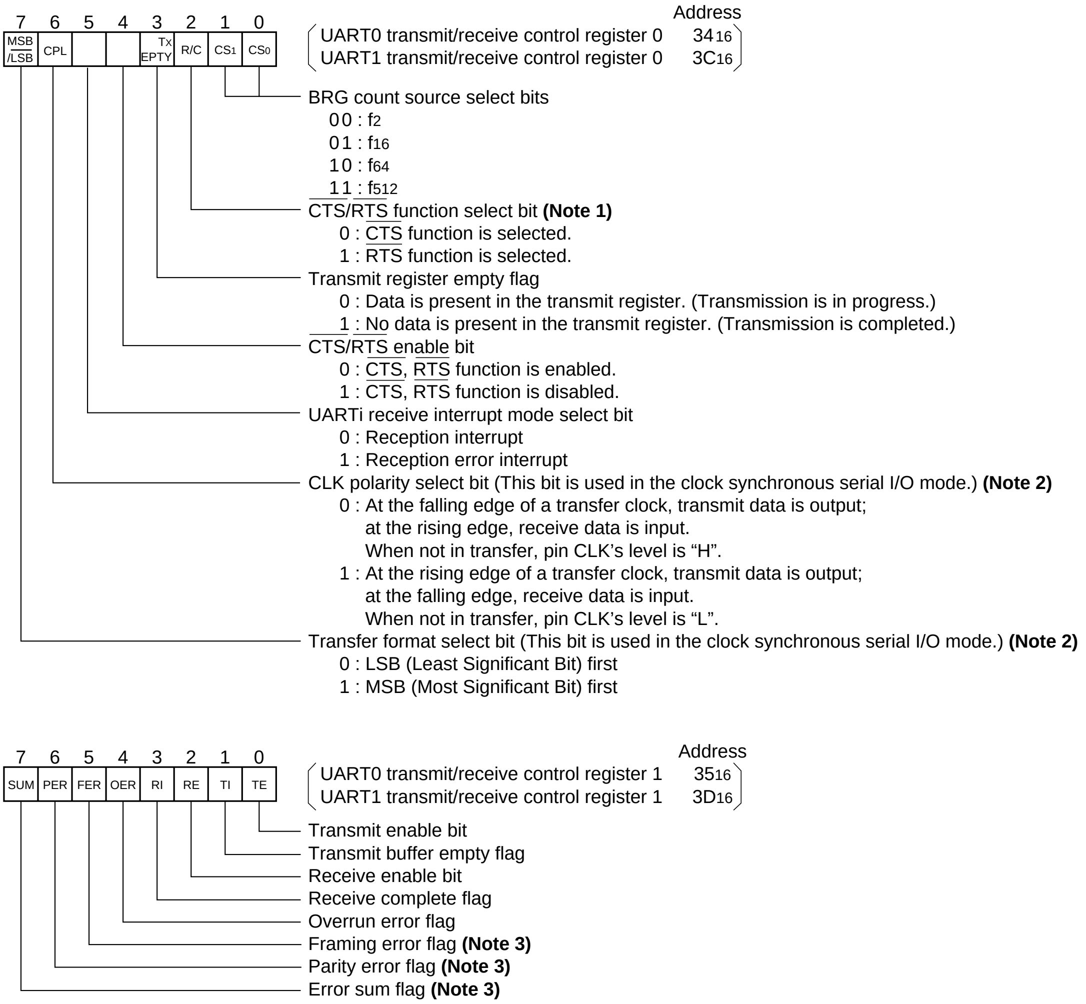


Fig. 67 Block diagram of transmitter



	Address
UART0 transmit/receive control register 0	34_{16}
UART1 transmit/receive control register 0	$3C_{16}$

BRG count source select bits
00 : f_2
01 : f_{16}
10 : f_{64}
11 : f_{512}

$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit **(Note 1)**
0 : $\overline{\text{CTS}}$ function is selected.
1 : $\overline{\text{RTS}}$ function is selected.

Transmit register empty flag
0 : Data is present in the transmit register. (Transmission is in progress.)
1 : No data is present in the transmit register. (Transmission is completed.)

$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ enable bit
0 : $\overline{\text{CTS}}$, $\overline{\text{RTS}}$ function is enabled.
1 : $\overline{\text{CTS}}$, $\overline{\text{RTS}}$ function is disabled.

UARTi receive interrupt mode select bit
0 : Reception interrupt
1 : Reception error interrupt

CLK polarity select bit (This bit is used in the clock synchronous serial I/O mode.) **(Note 2)**
0 : At the falling edge of a transfer clock, transmit data is output;
at the rising edge, receive data is input.
When not in transfer, pin CLK's level is "H".
1 : At the rising edge of a transfer clock, transmit data is output;
at the falling edge, receive data is input.
When not in transfer, pin CLK's level is "L".

Transfer format select bit (This bit is used in the clock synchronous serial I/O mode.) **(Note 2)**
0 : LSB (Least Significant Bit) first
1 : MSB (Most Significant Bit) first

	Address
UART0 transmit/receive control register 1	35_{16}
UART1 transmit/receive control register 1	$3D_{16}$

Transmit enable bit
Transmit buffer empty flag
Receive enable bit
Receive complete flag
Overrun error flag
Framing error flag **(Note 3)**
Parity error flag **(Note 3)**
Error sum flag **(Note 3)**

Notes 1: Valid when the $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ enable bit (bit 4) = "0".
2: Fix these bits to "0" in UART mode or when serial I/O is invalid.
3: Valid in UART mode.

Fig. 68 Bit configuration of UARTi transmit/receive control register

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 69 will be described. (The transmission side will be denoted by subscript j and the receiving side will be denoted by subscript k.)

Bit 0 of the UARTj transmit/receive mode register and UARTk transmit/receive mode register must be set to “1” and bits 1 and 2 must be “0”. The length of the transmission data is fixed at 8 bits.

Bit 3 of the UARTj transmit/receive mode register of the clock sending side is cleared to “0” to select the internal clock. Bit 3 of the UARTk transmit/receive mode register of the clock receiving side is set to “1” to select the external clock. Bits 4, 5 and 6 are ignored in clock synchronous mode. Bit 7 must always be “0”.

The clock source is selected by bit 0 (CS_0) and bit 1 (CS_1) of the clock-sending-side UARTj transmit/receive control register 0. As shown in Figure 64, the selected clock is divided by (n+1), then by 2, is passed through a transmission control circuit, and is output as transmission clock CLKj. Therefore, when the selected clock is fi,

$$\text{Bit Rate} = f_i / \{(n + 1) \times 2\}$$

On the clock receiving side, the CS_0 and CS_1 bits of the UARTk transmit/receive control register 0 are ignored because an external clock is selected.

Both of UART0 and UART1 can use $\overline{CTS}$ and $\overline{RTS}$ functions.

Bit 4 of the UARTi transmit/receive control register 0 is used to determine whether to use $\overline{CTS}$ or $\overline{RTS}$ signal. Bit 4 must be “0” when $\overline{CTS}$ or $\overline{RTS}$ signal is used. Bit 4 must be “1” when $\overline{CTS}$ and $\overline{RTS}$ signals are not used. When $\overline{CTS}$ and $\overline{RTS}$ signals are not used, $\overline{CTS}$/$\overline{RTS}$ pin can be used as a normal port pin.

When using pin $\overline{CTS}$/$\overline{RTS}$, :

- If bit 2 of the UARTi transmit/receive control register 0 is cleared to “0”, $\overline{CTS}$ input is selected.
- If bit 2 is set to “1”, $\overline{RTS}$ output is selected.

The case using $\overline{CTS}$ and $\overline{RTS}$ signals are explained below. As shown in Figure 76, bits 2 and 3 of the serial I/O pin control register can determine whether port pins $P8_3$ and $P8_7$ are used as pins TxDi or as port pins. When bits 2 and 3 are “0”, $P8_3$ and $P8_7$ function as pins TxDi; when bits 2 and 3 are “1”, $P8_3$ and $P8_7$ function as port pins. Therefore, in the input-only system where pins TxDi are not used, pins TxDi can function as port pins.

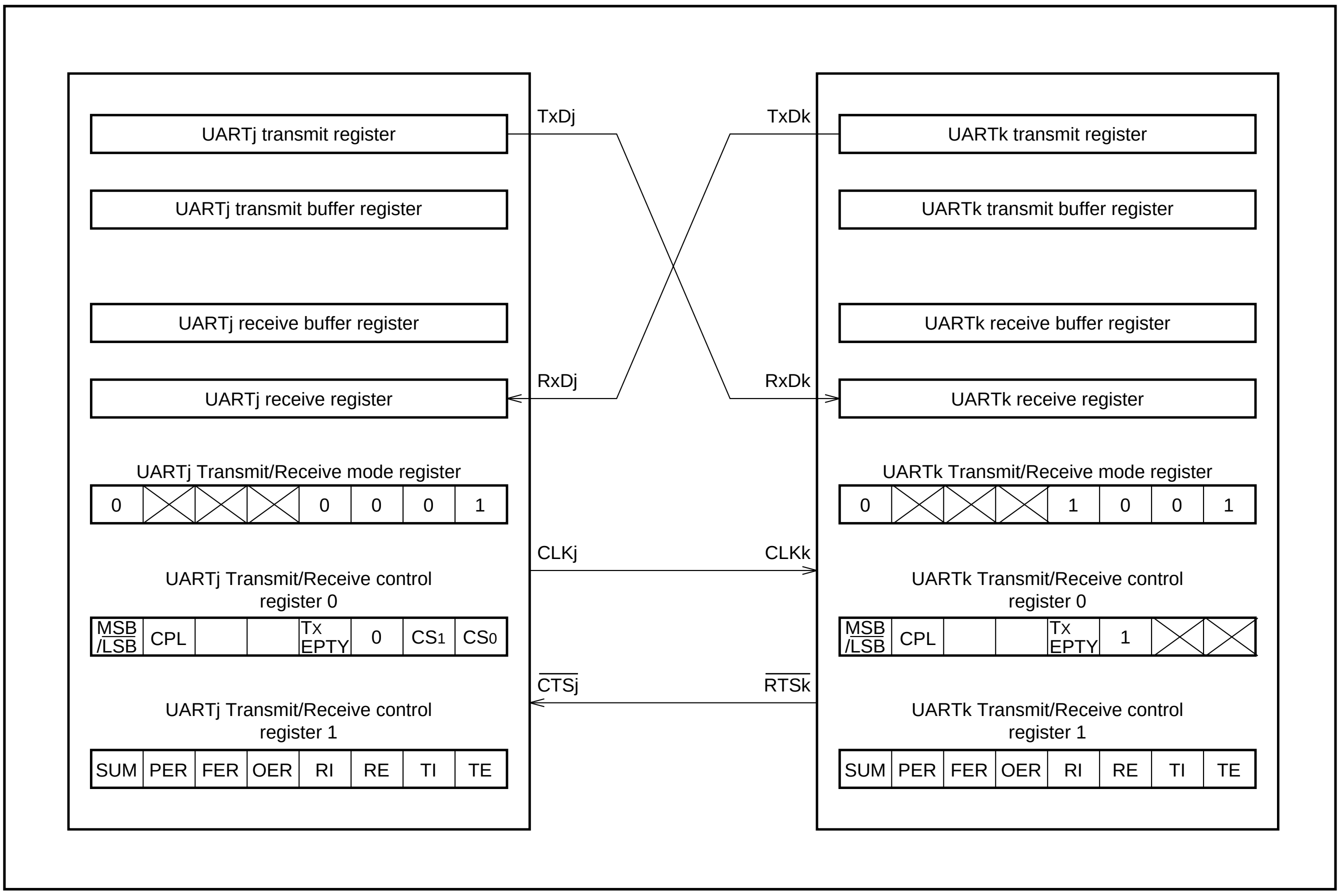

UARTj transmit register
UARTj transmit buffer register
UARTj receive buffer register
UARTj receive register
TxDj
RxDj
TxDk
RxDk
UARTk transmit register
UARTk transmit buffer register
UARTk receive buffer register
UARTk receive register
UARTj Transmit/Receive mode register
0 | ✕ | ✕ | ✕ | 0 | 0 | 0 | 1
UARTk Transmit/Receive mode register
0 | ✕ | ✕ | ✕ | 1 | 0 | 0 | 1
CLKj
CLKk
UARTj Transmit/Receive control register 0
MSB/LSB | CPL | | | Tx EPTY | 0 | CS1 | CS0
UARTk Transmit/Receive control register 0
MSB/LSB | CPL | | | Tx EPTY | 1 | ✕ | ✕
CTSj
RTSk
UARTj Transmit/Receive control register 1
SUM | PER | FER | OER | RI | RE | TI | TE
UARTk Transmit/Receive control register 1
SUM | PER | FER | OER | RI | RE | TI | TE

Fig. 69 Clock synchronous serial communication

Transmission

Transmission is started when bit 0 (TEj flag: transmit enable bit) of UARTj transmit/receive control register 1 is “1”, bit 1 (TIj flag) of one is “0”, and $\overline{\text{CTSj}}$ input is “L”. The TIj flag indicates whether the transmit buffer register is empty or not. It is cleared to “0” when data is written in the transmit buffer register ; it is set to “1” when the contents of the transmit buffer register is transferred to the transmit register and the transmit buffer register becomes empty.

When all of the transmit conditions are satisfied, the transmit data in the transmit buffer register are transferred to the transmit register, and transmission starts. As shown in Figure 70, data is output from TxDj pin each time when transmission clock CLKj changes from “H” to “L”. (In the clock synchronous serial I/O mode, the polarity of a transfer clock can be changed. For details, refer to the section on the selection of the transfer clock polarity.) The data is output from the least significant bit.

When the transmit register becomes empty after the contents has been transmitted, data is transferred automatically from the transmit buffer register to the transmit register if the next transmission start condition is satisfied. The next transmission is performed succeedingly. Once transmission has started, the TEj flag, TIj flag, and CTSj signals are ignored until data transmission completes. Therefore, transmission is not interrupt when $\overline{\text{CTSj}}$ input is changed to “H” during transmission.

The transmission start condition indicated by TEj flag, TIj flag, and $\overline{\text{CTSj}}$ is checked while the TENDj signal (shown in Figure 70) is “H”. Therefore, data can be transmitted continuously if the next transmission data is written in the transmit buffer register and TIj flag is cleared to “0” before theTENDj signal goes “H”.

Bit 3 (TxEPTYj flag) of UARTj transmit/receive control register 0 changes to “1” at the next cycle just after the TENDj signal goes “H” and changes to “0” when transmission starts. Therefore, this flag can be used to determine whether data transmission has completed.

When the TIj flag changes from “0” to “1”, the interrupt request bit in the UARTj transmit interrupt control register is set to “1”.

Receive

When bit 2 of the UARTk transmit/receive control register 1 is set to “1”, reception becomes enabled. In this case, when the CLKk signal is input, the receive operation starts simultaneously with this signal. The $\overline{\text{RTSk}}$ output is “H” when the REK flag is “0”. When the REK flag is set to “1”, the $\overline{\text{RTSk}}$ output becomes “L”. This informs the transmitter side that reception becomes enabled. When the receive operation starts, the $\overline{\text{RTSk}}$ output automatically becomes “H”.

When the receive operation starts, the receiver takes data from pin RxDk each time when the transmit clock (CLKj) turns from “L” to “H”. Simultaneously with reception, the contents of the receiver register is shifted bit by bit.

(Note that, in the clock synchronous serial communication, the polarity of a transfer clock can be inverted. For details, refer to the section on the polarity of the transfer clock.) When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and bit 3 (RIk flag) of UARTk transmit/receive control register 1 is set to “1”. In other words, the setting “1” to the RIk flag indicates that the receive buffer register contains the received data. At this time, if the low-order byte of the UARTk receive buffer register is read out, the $\overline{\text{RTSk}}$ output turns back to “L”. This indicates that the next data reception becomes enabled. Bit 4 (OERk flag) of UARTk transmit/receive control register 1 is set to “1” when the next data is transferred from the receive register to the receive buffer register while RIk flag is “1”, and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. (In other words, this indicates that an overrun error has occurred.) RIk flag is automatically cleared to “0” when the low-order byte of the receive buffer register is read or when the REk flag is cleared to “0”. The OERk flag is cleared when the REk flag is cleared. Bit 5 (FERk flag), bit 6 (PERk flag), and bit 7 (SUMk flag) are ignored in clock synchronous mode.

As shown in Figure 64, with clock synchronous serial communication, data cannot be received unless the transmitter is operating because the receive clock is created from the transmission clock. Therefore, the transmitter must be operating even when there is no need to sent data from UARTk to UARTj.

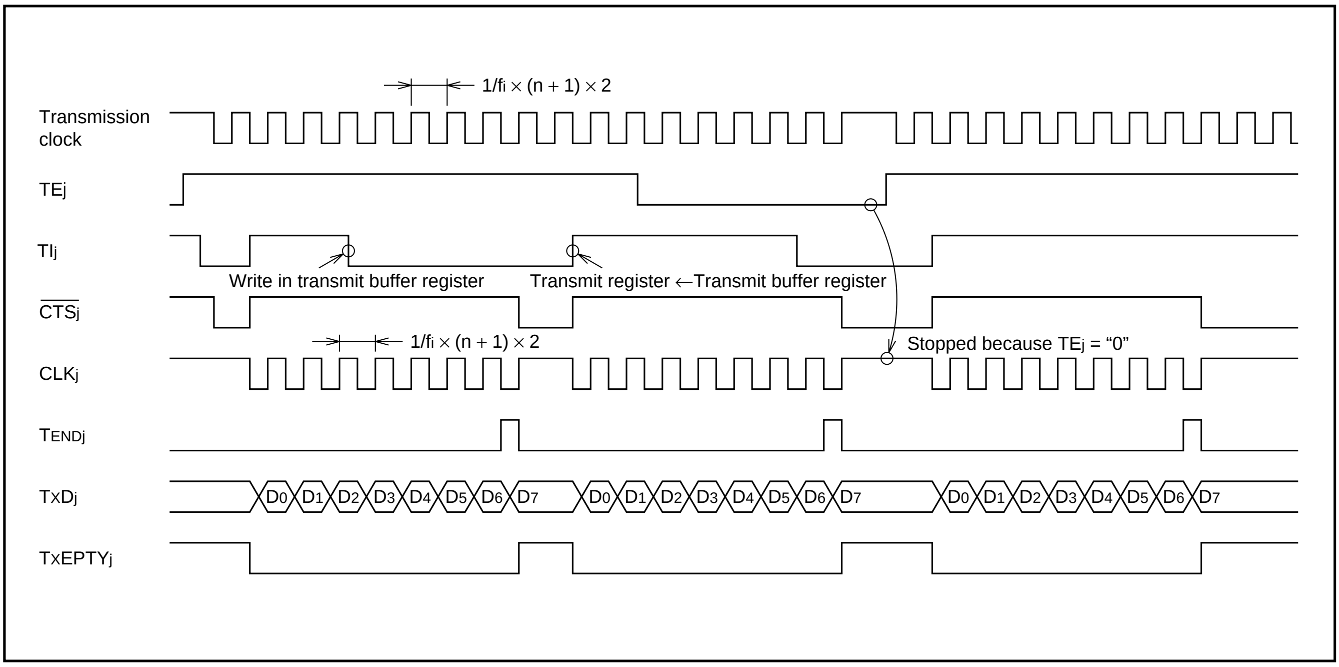


Fig. 70 Clock synchronous serial I/O timing

Interrupt request at completion of reception

When the RIk flag changes from "0" to "1", in other words, when the receive operation is completed, the interrupt request bit of the UARTk receive interrupt control register can be set to "1".

The timing when this interrupt request bit is to be set to "1" can be selected from the following:

- Each reception
- When an error occurs at reception

If bit 5 of the UARTk transmit/receive control register 0 (UART receive interrupt mode select bit) is cleared to "0", the interrupt request bit is set to "1" at each reception. If bit 5 is set to "1", the interrupt request bit is set to "1" only when an error occurs. (In the clock synchronous serial communication, only when an overrun error occurs, the interrupt request bit is set to "1".)

Polarity of transfer clock

In the clock synchronous serial communication, by bit 6 of the UARTj transmit/receive control register 0 (CPL), the polarity of a transfer clock can be selected.

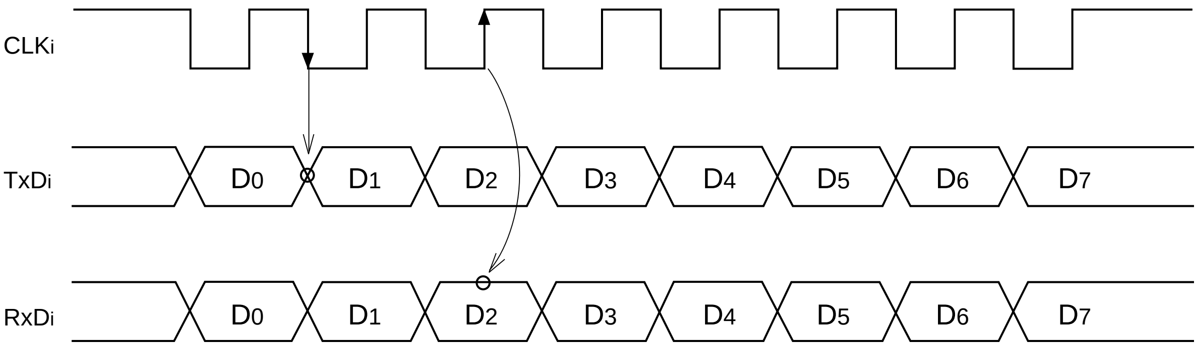
As shown in Figure 71, when bit 6 = "0", the polarity is as follows:

- In transmission, transmit data is output at the falling edge of CLKj.
- In reception, receive data is input at the rising edge of CLKk.
- When not in transfer, CLKi is at "H" level.

When bit 6 = "1", the polarity is as follows:

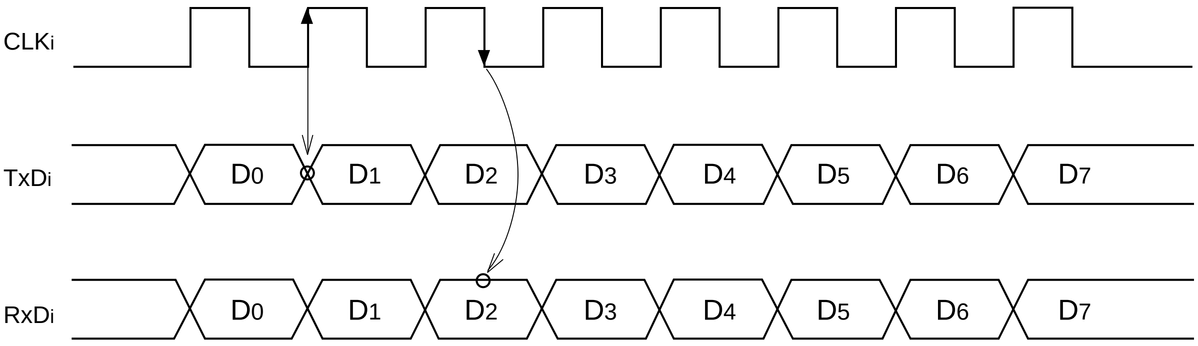
- In transmission, transmit data is output at the rising edge of CLKj.
- In reception, receive data is input at the rising edge of CLKk.
- When not in transfer, CLKi is at "L" level.

■ CLK polarity select bit = 0



❋ Transmit data is output to pin TxDi at the falling edge of transfer clock, and receive data is input from pin RxDi at the rising edge of transfer clock.
When not in transfer, pin CLKi's level is "H".

■ CLK polarity select bit = 1



❋ Transmit data is output to pin TxDi at the rising edge of transfer clock, and receive data is input from pin RxDi at the falling edge of transfer clock.
When not in transfer, pin CLKi's level is "L".

Fig. 71 Polarity of transfer clock

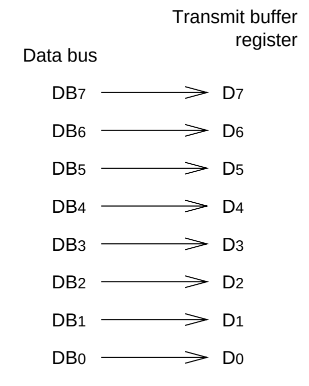
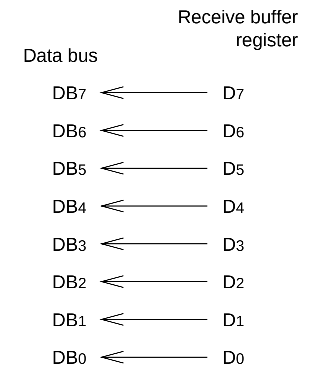
Selection of transfer format

In clock synchronous serial communication, transfer format can be selected by bit 7 of the transmit/receive control register 0. When bit 7 is “0”, transfer format is LSB first. When bit 7 is “1”, transfer format is MSB first.

This function is realized by changing connection relation between the transmit buffer register and the receive buffer register when writing transmit data to the transmit buffer register or reading receive data from the receive buffer register. Accordingly, the transmitter’s operation is the same in both transfer formats.

Figure 72 shows the connection relation.

Bit 7 in transmit/receive control register 0	Write to transmit buffer register	Read from receive buffer register

0
(LSB first)
	 Transmit buffer register
	Data bus
	DB7 → D7
	DB6 → D6
	DB5 → D5
	DB4 → D4
	DB3 → D3
	DB2 → D2
	DB1 → D1
	DB0 → D0
		 Receive buffer register
		Data bus
		DB7 ← D7
		DB6 ← D6
		DB5 ← D5
		DB4 ← D4
		DB3 ← D3
		DB2 ← D2
		DB1 ← D1
		DB0 ← D0

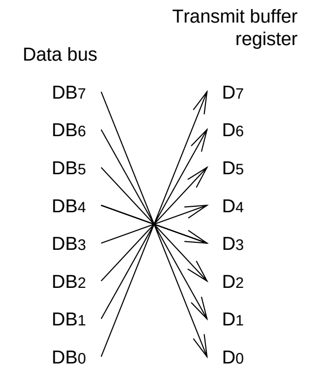
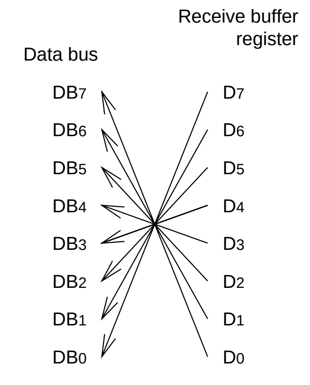
1 (MSB first)	 Transmit buffer register Data bus DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 D7 D6 D5 D4 D3 D2 D1 D0	 Receive buffer register Data bus DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 D7 D6 D5 D4 D3 D2 D1 D0

Fig. 72 Connection relation between transmit buffer register, receive buffer register, and data bus

Precautions for clock synchronous serial communication

When using pin $\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$, be sure to clear the D-A2 output enable bit (bit 2 at address 96_{16}) to “0” (output disabled). Also, in the clock synchronous serial communication, the separate function for $\overline{\text{CTS}_i}$/$\overline{\text{RTS}_i}$ cannot be selected. Furthermore, when an internal clock is selected, $\overline{\text{RTS}}$ output is undefined. Therefore, do not use the $\overline{\text{RTS}}$ function.

Before transmit operation is performed, be sure to clear bits 2 and 3 of the serial I/O pin control register (address AC_{16}) to “00”.

ASYNCHRONOUS SERIAL COMMUNICATION

Asynchronous serial communication can be performed using 7-, 8-, or 9-bit length data. The operation is the same for all data lengths. The following is the description for 8-bit asynchronous communication.

With 8-bit asynchronous communication, bit 0 of UARTi transmit/receive mode register is “1”, bit 1 is “0”, and bit 2 is “1”.

Bit 3 is used to select an internal clock or an external clock. If bit 3 is “0”, an internal clock is selected and if bit 3 is “1”, then external clock is selected. If an internal clock is selected, bit 0 (CS_0) and bit 1 (CS_1) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLKi pin can be used as a normal I/O pin. The selected internal or external clock is divided by (n+1), then by 16, and is passed through a control circuit to create the UART transmission clock or UART receive clock.

Therefore, the transmission speed can be changed by changing the contents (n) of the bit rate generator. If the selected clock is an internal clock Pfi or an external clock f_{EXT},

Bit 4 is the stop bit length select bit to select 1 stop bit or 2 stop bits.

Bit 5 is a select bit of odd parity or even parity.

In the odd parity mode, the parity bit is adjusted so that the sum of 1s in the data and parity bit is always odd.

In the even parity mode, the parity bit is adjusted so that the sum of the 1s in the data and parity bit is always even.

Bit 6 is the parity bit select bit which indicates whether to add parity bit or not.

Bits 4 to 6 must be set or reset according to the data format used in the communicating devices.

Bit 7 is the sleep select bit. The sleep mode is described later.

Figure 76 shows the bit configuration of the serial I/O pin control register. By bits 0 and 1 of the serial I/O pin control register ($\overline{CTSi}$/$\overline{RTSi}$ separate select bits), the function of the $\overline{CTS}$/$\overline{RTS}$ pin can be separated into two functions, and each function can be assigned to two different pins. When bits 0 and 1 = “11”, the above separation is performed. When bits 0 and 1 = “00”, no separation is performed.

Table 13 lists the selection methods of the $\overline{CTS}$/$\overline{RTS}$ function.

Bit Rate = (fi or f_{EXT}) / {(n+1)×16}

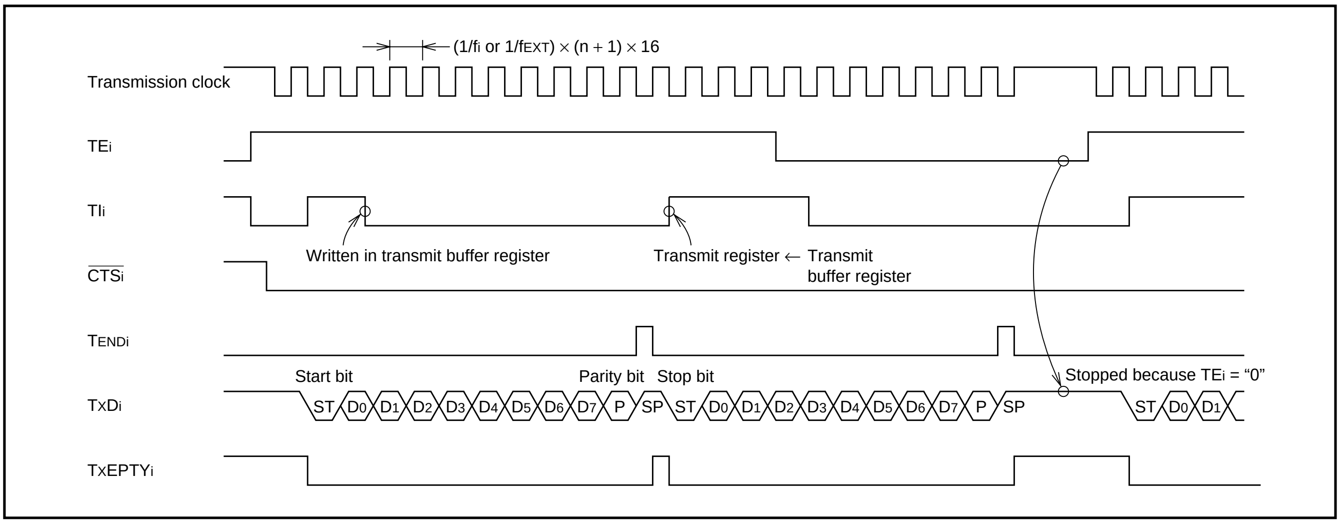

(1/fi or 1/fEXT) × (n + 1) × 16
Transmission clock
TEi
TIi
CTSi
TENDi
TxDi
TxEPTYi
Written in transmit buffer register
Transmit register ← Transmit buffer register
Start bit
Parity bit
Stop bit
Stopped because TEi = “0”
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1

Fig. 73 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit selected

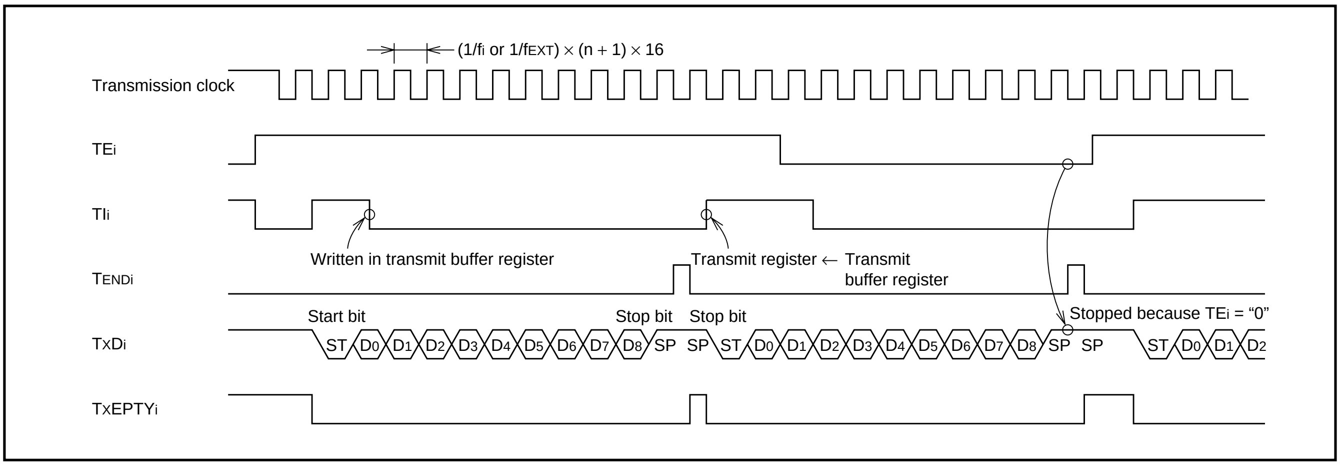

(1/fi or 1/fEXT) × (n + 1) × 16
Transmission clock
TEi
TIi
TENDi
TxDi
TxEPTYi
Written in transmit buffer register
Transmit register ← Transmit buffer register
Start bit
Stop bit
Stop bit
Stopped because TEi = “0”
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2

Fig. 74 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits selected

Transmission

Transmission is started when bit 0 (TEi flag transmit enable flag) of UARTi transmit/receive control register 1 is "1", bit 1 (TIi flag) is "0", and $\overline{\text{CTSi}}$ input (in other words, transmit enable signal input from receiver) is "L." The TIi flag indicates whether the transmit buffer is empty or not. It is cleared to "0" when data is written in the transmit buffer; it is set to "1" when the contents of the transmit buffer register is transferred to the transmit register.

When all of the transmission conditions are satisfied, transmit data is transferred to the transmit register, and transmit operation starts.

As shown in Figures 73 and 74, data is output from the TxDi pin with the stop bit or parity bit specified by bits 4 to 6 of UARTi transmit/receive mode register. The data is output from the least significant bit.

When the transmit register becomes empty after the contents has been transmitted, data is transferred automatically from the transmit buffer register to the transmit register if the next transmit start condition is satisfied. Then, the next transmission is performed succeedingly.

Once transmission has started, the TEi flag, TIi flag, and $\overline{\text{CTSi}}$ signal are ignored until data transmission is completed.

Therefore, transmission does not stop until it completes event if, during transmission, the TEi flag is cleared to "0" or $\overline{\text{CTSi}}$ input is set to "1".

The transmission start condition indicated by TEi flag, TIi flag, and $\overline{\text{CTSi}}$ is checked while the TENDi signal shown in Figure 73 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmit buffer register and TIi flag is cleared to "0" before the TENDi signal goes "H".

Bit 3 (TxEPTYi flag) of UARTi transmit/receive control register 0 changes to "1" at the next cycle just after the TENDi signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmission is completed.

When the TIi flag changes from "0" to "1", the interrupt request bit of the UARTi transmit interrupt control register is set to "1".

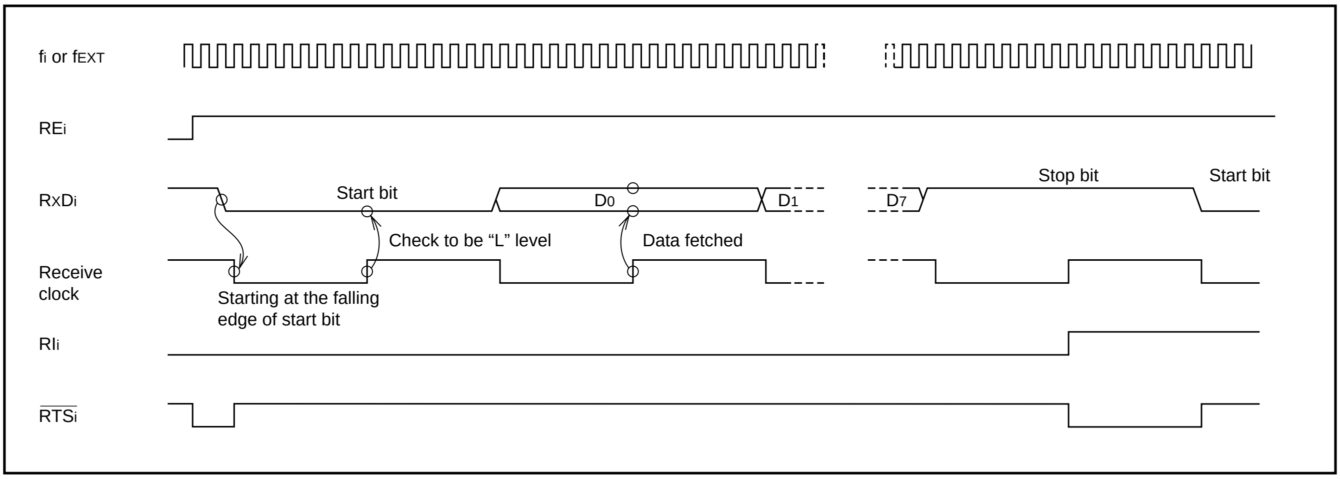


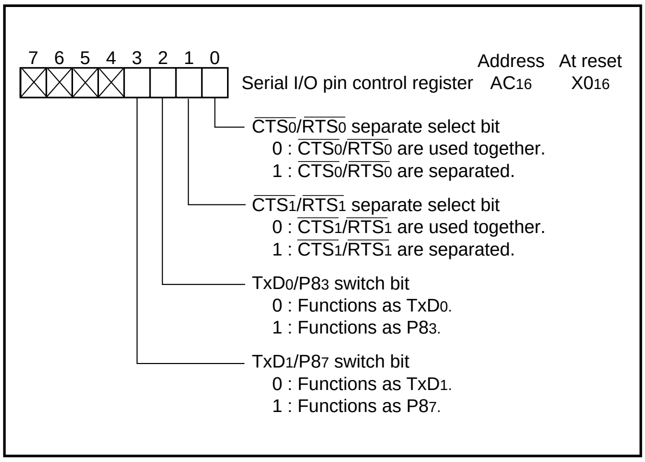
Fig. 75 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit selected

Table 13. Selection methods of $\overline{\mathrm{CTS}}$/$\overline{\mathrm{RTS}}$ function

$\overline{\mathrm{CTS}}$/$\overline{\mathrm{RTS}}$ enable bit	$\overline{\mathrm{CTSi}}$/$\overline{\mathrm{RTSi}}$ separate select bit	$\overline{\mathrm{CTS}}$/$\overline{\mathrm{RTS}}$ function select bit	Functions			
			Pin $P8_0$/$\overline{\mathrm{CTS}_0}$/$\overline{\mathrm{RTS}_0}$ **(Note 1)**	Pin $P8_1$/$\overline{\mathrm{CTS}_0}$/$\mathrm{CLK}_0$	Pin $P8_4$/$\overline{\mathrm{CTS}_1}$/$\overline{\mathrm{RTS}_1}$	Pin $P8_5$/$\overline{\mathrm{CTS}_1}$/$\mathrm{CLK}_1$
0	0	0	$\overline{\mathrm{CTS}_0}$	$P8_1$ or CLK_0	$\overline{\mathrm{CTS}_1}$	$P8_5$ or CLK_1
		1	$\overline{\mathrm{RTS}_0}$	$P8_1$ or CLK_0	$\overline{\mathrm{RTS}_1}$	$P8_5$ or CLK_1
	1	×	$\overline{\mathrm{RTS}_0}$	$\overline{\mathrm{CTS}_0}$ **(Notes 2 and 3)**	$\overline{\mathrm{RTS}_1}$	$\overline{\mathrm{CTS}_1}$ **(Notes 2 and 3)**
1	×	×	$P8_0$	$P8_1$ or CLK_0	$P8_4$	$P8_5$ or CLK_1

×: It may be “0” or “1”.

Notes 1: When using the $\overline{\mathrm{CTS}_0}$/$\overline{\mathrm{RTS}_0}$ pin, be sure to clear the D-A_2 output enable bit (bit 2 at address 96_{16}) to “0”.
2: When using the $\overline{\mathrm{CTS}}$ function, be sure to clear the corresponding bit of the port P8 direction register to “0”.
3: When $\overline{\mathrm{CTSi}}$ and $\overline{\mathrm{RTSi}}$ has been separated, the CLKi pin cannot be used. Therefore, in the clock synchronous serial communication, $\overline{\mathrm{CTSi}}$ and $\overline{\mathrm{RTSi}}$ cannot be separated. Also, when $\overline{\mathrm{CTSi}}$ and $\overline{\mathrm{RTSi}}$ are separated in UART mode, be sure to select an internal clock.



Serial I/O pin control register AC_{16} $X0_{16}$

$\overline{\mathrm{CTS}_0}$/$\overline{\mathrm{RTS}_0}$ separate select bit
0 : $\overline{\mathrm{CTS}_0}$/$\overline{\mathrm{RTS}_0}$ are used together.
1 : $\overline{\mathrm{CTS}_0}$/$\overline{\mathrm{RTS}_0}$ are separated.

$\overline{\mathrm{CTS}_1}$/$\overline{\mathrm{RTS}_1}$ separate select bit
0 : $\overline{\mathrm{CTS}_1}$/$\overline{\mathrm{RTS}_1}$ are used together.
1 : $\overline{\mathrm{CTS}_1}$/$\overline{\mathrm{RTS}_1}$ are separated.

TxD_0/$P8_3$ switch bit
0 : Functions as TxD_0.
1 : Functions as $P8_3$.

TxD_1/$P8_7$ switch bit
0 : Functions as TxD_1.
1 : Functions as $P8_7$.

Fig. 76 Bit configuration of serial I/O pin control register

Receive

Receive is enabled when bit 2 (REi flag) of UARTi transmit/receive control register 1 is set to “1.” As shown in Figure 75, the frequency divider circuit (1/16) at the receiving side begin to work when a start bit arrives and the data is received.

If $\overline{\mathrm{RTSi}}$ output is selected by setting bit 2 of UARTi transmit/receive control register 0 to “1”, the $\overline{\mathrm{RTSi}}$ output is “H” when the REi flag is “0”. When the REi flag changes to “1”, the $\overline{\mathrm{RTSi}}$ output goes “L” to inform the receiver that reception has become enabled. When the receive operation starts, the $\overline{\mathrm{RTSi}}$ output automatically becomes “H”.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 66. At this point, the contents of the receive register is transferred to the receive buffer register and bit 3 (RIi flag) of UARTi transmit/receive control register 1 is set to “1.” In other words, the RIi flag indicates that the receive buffer register contains data when it is set to “1.” At this time, when the low-order byte of the UARTk receive buffer register is read out, $\overline{\mathrm{RTSi}}$ output goes back to “L” to indicate that the register is ready to receive the next data.

Bit 4 (OERi flag) of UARTi transmit/receive control register 1 is set to “1” when the next data is transferred from the receive register to the receive buffer register while the RIi flag is “1”, in other words, when an overrun error occurs. If the OERi flag is “1”, it indicates that the next data has been transferred to the receive buffer register before the contents of the receive buffer register has been read.

Bit 5 (FERi flag) is set to “1” when the number of stop bits is less than required (framing error).

Bit 6 (PERi flag) is set to “1” when a parity error occurs.

Bit 7 (SUMi flag) is set to “1” when either the OERi flag, FERi flag, or the PERi flag is set to “1.” Therefore, the SUMi flag can be used to determine whether there is an error.

The setting of the RIi flag, OERi flag, FERi flag, and the PERi flag is performed while transferring the contents of the receive register to the receive buffer register.

The FERi, PERi, and SUMi flags are cleared to “0” when reading the low-order byte of the receive buffer register or when writing “0” to the REi flag.
The OERi flag is cleared to “0” when writing “0” to the REi flag.

Interrupt request at completion of reception

When the RIk flag changes from “0” to “1”, in other words, when the receive operation is completed, the interrupt request bit of the UARTk receive interrupt control register can be set to “1”.
The timing when this interrupt request bit is to be set to “1” can be selected from the following:

- Each reception
- When an error occurs at reception

If bit 5 of the UARTk transmit/receive control register 0 (UART receive interrupt mode select bit) is cleared to “0”, the interrupt request bit is set to “1” at each reception. If bit 5 is set to “1”, the interrupt request bit is set to “1” only when an error occurs. (In the clock asynchronous serial communication, when an overrun error, framing error, or parity error occurs, the interrupt request bit is set to “1”.)

Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.
The microcomputer enters the sleep mode when bit 7 of UARTi transmit/receive mode register is set to “1.”
The operation of the sleep mode for an 8-bit asynchronous communication is described below.
When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asynchronous communication) of the received data is “0”. Also the RIi, OERi, FERi, PERi, and the SUMi flags are unchanged. Therefore, the interrupt request bit of the UARTi receive interrupt control register is also unchanged. Normal receive operation takes place when bit 7 of the received data is “1”.
The following is an example of how the sleep mode can be used.
The main microcomputer first sends data: bit 7 is “1” and bits 0 to 6 are set to the address of the subordinate microcomputer to be communicated with. Then all subordinate microcomputers receive this data. Each subordinate microcomputer checks the received data, clears the sleep bit to “0” if bits 0 to 6 are its own address and sets the sleep bit to “1” if not. Next, the main microcomputer sends data with bit 7 cleared. Then the microcomputer which cleared the sleep bit will receive the data, but the microcomputers which set the sleep bit to “1” will not. In this way, the main microcomputer is able to communicate only with the designated microcomputer.

Precautions for clock asynchronous (UART) serial communication

When using pin $\overline{\mathrm{CTS}}_0/\overline{\mathrm{RTS}}_0$, be sure to clear the D-A$_2$ output enable bit (bit 2 at address 96_{16}) to “0” (output disabled). Also, when $\overline{\mathrm{CTS}}$i and $\overline{\mathrm{RTS}}$i are separated, pin CLKi cannot be used. Therefore, when $\overline{\mathrm{CTS}}$i and $\overline{\mathrm{RTS}}$i are separated in UART mode, be sure to select an internal clock.
Before transmit operation is performed, be sure to clear bits 2 and 3 of the serial I/O pin control register (address AC_{16}) to “00”.

A-D CONVERTER

The A-D converter is a 10-bit successive approximation converter. Figure 77 shows the block diagram of the A-D converter, Figure 78 shows the bit configuration of the A-D control register 0 (address $1E_{16}$), and Figure 79 shows the bit configuration of the A-D control register 1 (address $1F_{16}$).

A-D conversion accuracy

Bit 3 of A-D control register 1 is used to select whether to regard the conversion result as 10-bit or as 8-bit data. The conversion result is regarded as 10-bit data when bit 3 is "1" and as 8-bit data when bit 3 is "0".

When the conversion result is used as 10-bit data, the low-order 8 bits of the conversion result is stored in the even address of the corresponding A-D register and the high-order 2 bits are stored in bits 0 and 1 at the odd address of the corresponding A-D register. Bits 2 to 7 of the A-D register odd address are "000000_2" when read.

When the conversion result is used as 8-bit data, the conversion result are stored in even address of the corresponding A-D register. In this case, the value at the A-D register's odd address is "00_{16}" when read.

A-D conversion frequency

An operation clock (ϕ_{AD}) of an A-D converter can be selected with bit 7 of the A-D control register 0 and bit 4 of the A-D control register 1. When bit 4 of the A-D control register 1 is "0", ϕ_{AD} becomes $f_2/4$ when bit 7 of the A-D control register 0 is "0", ϕ_{AD} becomes $f_2/2$ when bit 7 of the A-D control register 0 is "1".

When bit 4 of the A-D control register 1 is "1", ϕ_{AD} becomes f_2 when bit 7 of the A-D control register 0 is "0", ϕ_{AD} becomes f_1 when bit 7 of the A-D control register 0 is "1". Note that $\phi_{AD} = f_1$ (in other words, the fastest speed) can be selected only in the 8-bit mode.

ϕ_{AD} during A-D conversion must be 250 kHz or more because the comparator uses a capacity coupling amplifier.

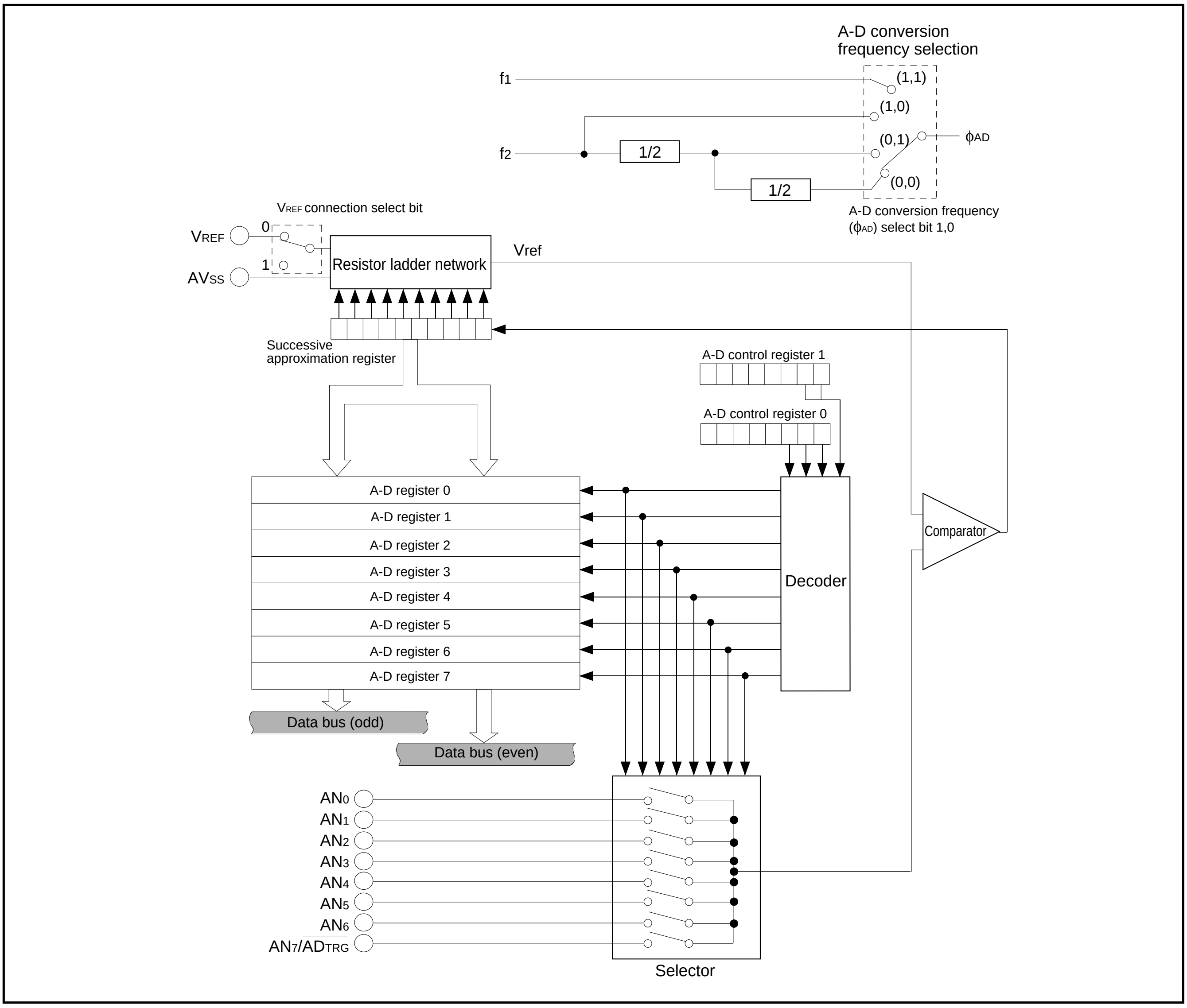


Fig. 77 Block diagram of A-D converter

Trigger

A-D conversion can be started by an internal trigger or by an external trigger.

An internal trigger is selected when bit 5 of A-D control register 0 is “0” and an external trigger is selected when it is “1”. When trigger is selected, A-D conversion is started when bit 6 (A-D conversion start bit) is set to “1.”

When an external trigger is selected, the polarity of a trigger input can be selected by bit 5 of the A-D control register 1. When bit 5 = “0”, a falling edge is selected, and when bit 5 = “1”, a rising edge is selected.

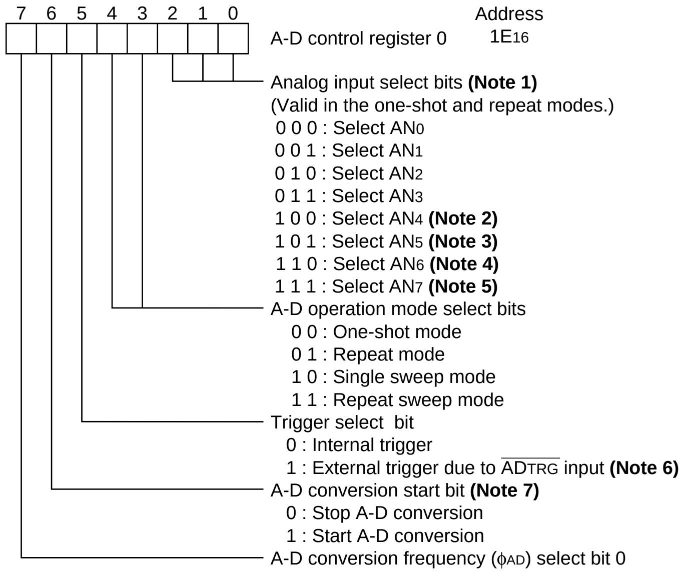
A-D conversion starts when the A-D conversion start bit is “1” and the $\overline{\text{AD}_{\text{TRG}}}$ input changes from “H” to “L” (or “L” to “H.”) In this case, the pins that can be used for A-D conversion are AN$_0$ to AN$_6$ because the $\overline{\text{AD}_{\text{TRG}}}$ pin is multiplexed with an analog voltage input pin, AN$_7$. If an external trigger is selected, even when the A-D conversion is completed, the A-D conversion start bit keeps “1”. Also, a retrigger can be available even when A-D conversion is in progress.

V$_{\text{REF}}$ connection

Whether to connect the reference voltage input (V$_{\text{REF}}$) with the resistor ladder network or not depends on bit 6 of the A-D control register 1. The V$_{\text{REF}}$ pin is connected when bit 6 is “0” and is disconnected when bit 6 is “1” (High impedance state).

When A-D conversion is not performed, current from the V$_{\text{REF}}$ pin to the resistor ladder network can be cut off by disconnecting resistor ladder network from the V$_{\text{REF}}$ pin.

Before starting A-D conversion, wait for 1 μs or more after clearing bit 6 to “0”.



A-D control register 0 — Address $1E_{16}$

Analog input select bits **(Note 1)**
(Valid in the one-shot and repeat modes.)
0 0 0 : Select AN$_0$
0 0 1 : Select AN$_1$
0 1 0 : Select AN$_2$
0 1 1 : Select AN$_3$
1 0 0 : Select AN$_4$ **(Note 2)**
1 0 1 : Select AN$_5$ **(Note 3)**
1 1 0 : Select AN$_6$ **(Note 4)**
1 1 1 : Select AN$_7$ **(Note 5)**

A-D operation mode select bits
0 0 : One-shot mode
0 1 : Repeat mode
1 0 : Single sweep mode
1 1 : Repeat sweep mode

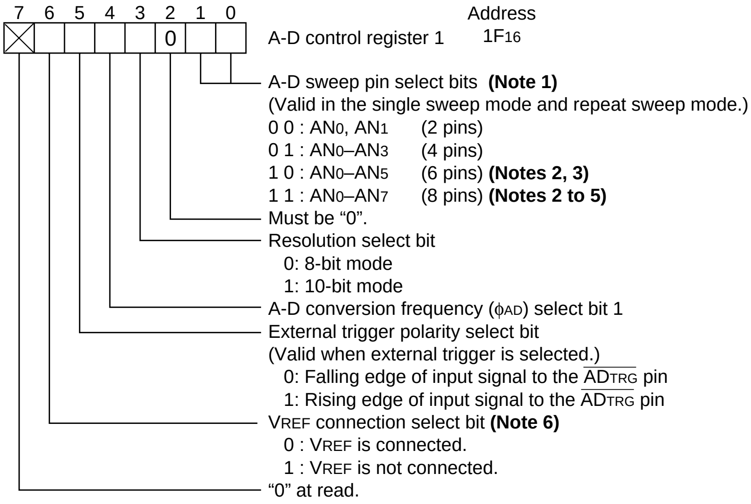
Trigger select bit
0 : Internal trigger
1 : External trigger due to $\overline{\text{AD}_{\text{TRG}}}$ input **(Note 6)**

A-D conversion start bit **(Note 7)**
0 : Stop A-D conversion
1 : Start A-D conversion

A-D conversion frequency (ϕ_{AD}) select bit 0

Notes 1: Ignored in the single sweep and repeat sweep modes. (Each of these bits may be “0” or “1”.)
2: When using the AN$_4$ pin, be sure to clear the $\overline{\text{INT}_3}$ pin select bit (bit 5 at address 94_{16}) to “0”.
3: When using the AN$_5$ pin, be sure to clear the $\overline{\text{INT}_4}$ pin select bit (bit 6 at address 94_{16}) to “0”.
4: When using the AN$_6$ pin, be sure to clear the D-A$_0$ output enable bit (bit 0 at address 96_{16}) to “0” (output disabled).
5: When using the AN$_7$ pin, be sure to clear both of the $\overline{\text{INT}_2}$ pin select bit (bit 4 at address 94_{16}) and the D-A$_1$ output enable bit (bit 1 at address 96_{16}) to “0”.
6: When using an external trigger, be sure to clear the $\overline{\text{INT}_2}$ pin select bit (bit 4 at address 94_{16}) and D-A$_1$ output enable bit (bit 1 at address 96_{16}) to “0”.
7: For writing to this bit, use the MOVM (MOVMB) instruction, or the STA (STAB, STAD) instruction.
8: Rewriting to each bit of the A-D control register 0 (except for bit 6) must be performed while A-D conversion is stopped.

Fig. 78 Bit configuration of A-D control register 0



A-D sweep pin select bits **(Note 1)**
(Valid in the single sweep mode and repeat sweep mode.)

0 0 : AN$_0$, AN$_1$	(2 pins)	
0 1 : AN$_0$–AN$_3$	(4 pins)	
1 0 : AN$_0$–AN$_5$	(6 pins)	**(Notes 2, 3)**
1 1 : AN$_0$–AN$_7$	(8 pins)	**(Notes 2 to 5)**

Must be “0”.

Resolution select bit
0: 8-bit mode
1: 10-bit mode

A-D conversion frequency (ϕ_{AD}) select bit 1

External trigger polarity select bit
(Valid when external trigger is selected.)
0: Falling edge of input signal to the $\overline{AD_{TRG}}$ pin
1: Rising edge of input signal to the $\overline{AD_{TRG}}$ pin

V$_{REF}$ connection select bit **(Note 6)**
0 : V$_{REF}$ is connected.
1 : V$_{REF}$ is not connected.

“0” at read.

A-D conversion frequency (ϕ_{AD}) select bit		ϕ_{AD}
Bit 1	Bit 0	
0	0	f$_2$/4
0	1	f$_2$/2
1	0	f$_2$
1	1	f$_1$ (Selectable only in 8-bit mode)

Notes 1: Ignored in the one-shot and repeat modes. (Each of these bits may be “0” or “1”.)
2: When using the AN$_4$ pin, be sure to clear the $\overline{INT_3}$ pin select bit (bit 5 at address 94$_{16}$) to “0”.
3: When using the AN$_5$ pin, be sure to clear the $\overline{INT_4}$ pin select bit (bit 6 at address 94$_{16}$) to “0”.
4: When using the AN$_6$ pin, be sure to clear the D-A$_0$ output enable bit (bit 0 at address 96$_{16}$) to “0” (output disabled).
5: When using the AN$_7$ pin, be sure to clear both of the $\overline{INT_2}$ pin select bit (bit 4 at address 94$_{16}$) and the D-A$_1$ output enable bit (bit 1 at address 96$_{16}$) to “0”. When an external trigger is selected, the AN$_7$ pin cannot be used as an analog input pin.
6: Once this bit is cleared from “1” to “0”, it is necessary to wait for 1 µs or more before the A-D or D-A conversion starts.
7: Rewriting to each bit of the A-D control register 1 must be performed while A-D conversion is stopped.

Fig. 79 Bit configuration of A-D control register 1

Operation mode

The operation mode is selected by bits 3 and 4 of A-D control register 0. The available operation modes are one-shot, repeat, single sweep, and repeat sweep. Analog input port pins are multiplexed with port P7 pins. Therefore, bits which correspond to pins for A-D conversion must be “0” (input mode).

(1) One-shot mode

One-shot mode is selected when bits 3 and 4 of A-D control register 0 are “0” is “0”. The A-D conversion pins are selected with bits 0 to 2 of A-D control register 0. A-D conversion can be started by a software trigger or by an external trigger.

When an internal trigger is selected, A-D conversion is started when bit 6 (A-D conversion start bit) is set to “1.”

When bit 3 of the A-D control register 1 is “1”, A-D conversion ends after 59 ϕ_{AD} cycles, and the interrupt request bit of the A-D interrupt control register is set to “1.” At the same time, A-D control register 0 bit 6 (A-D conversion start bit) is cleared to “0” and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start bit is “1” and a valid edge is input to the $\overline{\mathrm{AD_{TRG}}}$ pin, This operation is the same as that for internal trigger except that the A-D conversion start bit is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(2) Repeat mode

Repeat mode is selected when bit 3 of A-D control register 0 is “1” and bit 4 is “0”.

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated.

No interrupt request is generated in this mode. Furthermore, if an external trigger is selected, the A-D conversion start bit is not cleared.

The contents of the A-D register can be read at any time.

(3) Single sweep mode

Single sweep mode is selected when bit 3 of A-D control register 0 is “0” and bit 4 is “1”.

In the single sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D control register 1 (address $1F_{16}$). Two pins, four pins, six pins, or eight pins can be selected as analog input pins, depending on the contents of these bits.

A-D conversion is performed only for selected input pins. After A-D conversion is performed for input of AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion can be started with an internal trigger or with an external trigger input. An internal trigger is selected when bit 5 of the A-D control register 0 (address $1E_{16}$) is “0” and an external trigger is selected when it is “1”.

When an internal trigger is selected, A-D conversion is started when A-D control register 0 bit 6. (A-D conversion start bit) is set to “1.” When A-D conversion of all selected pins end, the interrupt request bit of the A-D conversion interrupt control register is set to “1.” At the same time, A-D conversion start bit is cleared to “0” and A-D conversion stops.

When an external trigger is selected, A-D conversion starts when the A-D conversion start bit is “1” and a valid edge is input to the $\overline{\mathrm{AD_{TRG}}}$ pin. In this case, the A-D conversion result which is stored in the A-D register 7 becomes invalid.

The operation by external trigger is the same as that by an internal trigger except that the A-D conversion start bit is not cleared to “0” after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode

Repeat sweep mode is selected when bit 3 of A-D control register 0 is “1” and bit 4 is “1”.

The difference from the single sweep mode is that A-D conversion does not stop after conversion for all selected pins, but repeats again from the AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, if an internal trigger is selected, the A-D convension start bit is not cleared. The A-D register can be read at any time.

Precautions for A-D conversion interrupt function

Clear the interrupt request bit of the A-D interrupt control register (bit 3 at address 70_{16}) before using the A-D interrupt. It is because the interrupt request bit is undefined just after reset.

D-A CONVERTER

Three independent D-A converters are included in this microcomputer, and each D-A converter adopts an 8-bit R-2R method. Figure 80 shows the block diagram of the D-A converter, Figure 81 shows the bit configuration of the A-D control register 1, and Figure 82 shows the bit configuration of the D-A control register.

D-A conversion is performed by writing a value to the corresponding D-A register i. Whether to output the analog voltage or not is determined by bits 0 to 2 of the D-A control register. When any of bits 0 to 2 = "1", the corresponding pin (D-A$_0$ to D-A$_2$) outputs the analog voltage.

This analog voltage (V) is determined according to value n. ("n" = decimal number. This has been set in the D-A register.)

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

V_{REF} : Reference voltage

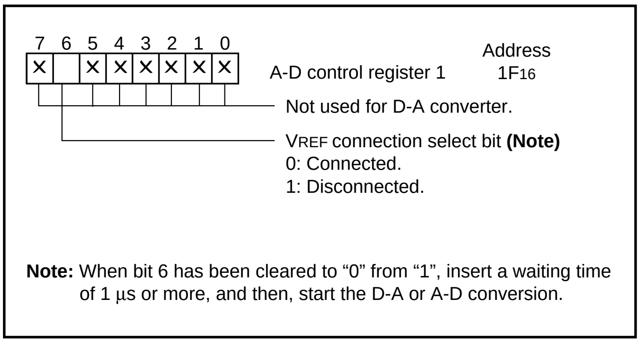
The contents of the corresponding D-A output enable bit and D-A register are cleared to "0" at reset. Whether to connect the reference voltage input (VREF) with the ladder network or not depends on bit 6 of the A-D control register 1. Pin VREF is connected with the ladder network when bit 6 = "0" and is disconnected when bit 6 = "1" (high impedance state). When not performing the A-D or D-A conversion, current from pin VREF to the ladder network can be cut off by disconnecting ladder network from pin VREF.

Before starting A-D or D-A conversion, be sure to clear bit 6 to "0", and then, insert a waiting time of 1 µs or more.

An external buffer is necessary when connecting a low impedance load with the D-A converter. It is because that a D-A output pin doesn't include a buffer.

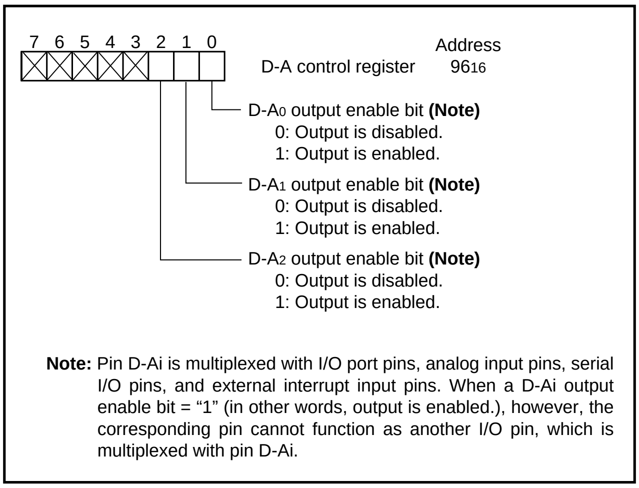
Pin D-Ai is multiplexed with I/O port pins, analog input pins, serial I/O pins, and external interrupt input pins. When a D-Ai output enable bit = "1" (in other words, output is enabled.), however, the corresponding pin cannot function as another I/O pin, which is multiplexed with pin D-Ai.

Also, when not using the D-A converter, be sure to clear the contents of the corresponding D-A output enable bit and D-A register to "0".



Note: When bit 6 has been cleared to "0" from "1", insert a waiting time of 1 µs or more, and then, start the D-A or A-D conversion.

Fig. 81 Bit configuration of A-D control register 1



Note: Pin D-Ai is multiplexed with I/O port pins, analog input pins, serial I/O pins, and external interrupt input pins. When a D-Ai output enable bit = "1" (in other words, output is enabled.), however, the corresponding pin cannot function as another I/O pin, which is multiplexed with pin D-Ai.

Fig. 82 Bit configuration of D-A control register

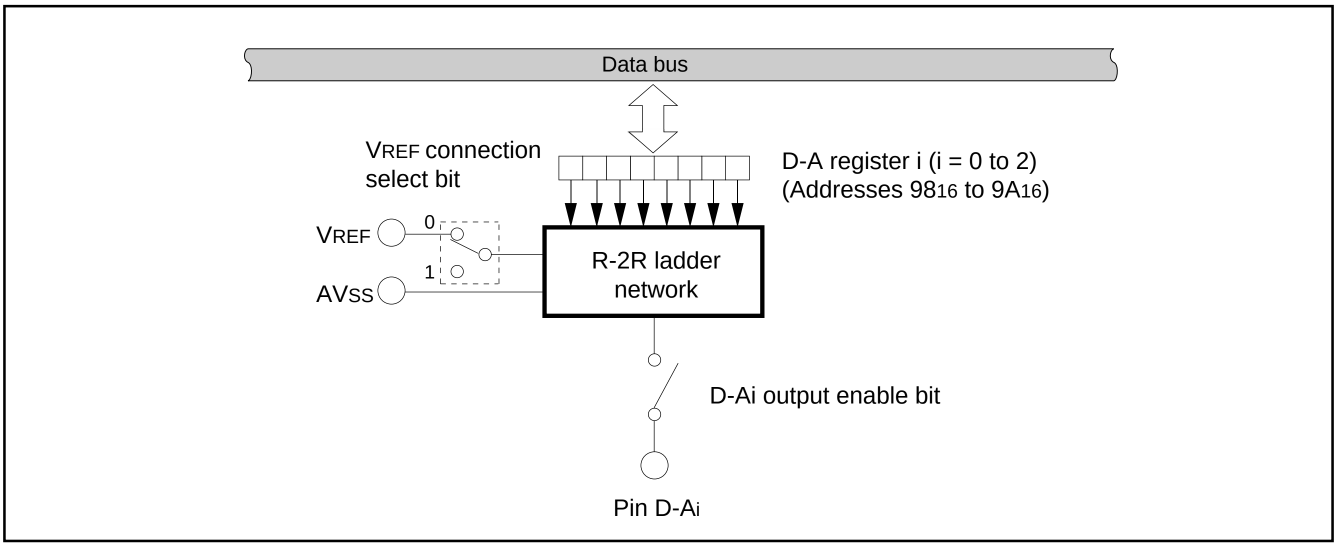


Fig. 80 Block diagram of D-A converter

REAL-TIME OUTPUT

The real-time output function enables to change the output level of several pins simultaneously with a specified timer’s counting.

Whether to use the real-time output function is decided by the waveform output select bits of the 8-bit real-time output control register (bits 0 and 1 at address A0$_{16}$). (See Figure 83.) Also, the real-time output controlled by the pulse output mode select bit of the real-time output control register (bit 2 at address A0$_{16}$) and is used in one of the following ways:

- 4 bits × 2 channels
- 6 bits × 1 channel + 2 bits × 1 channels

(1) Pulse mode 0

When the pulse output mode select bit is cleared to “0”, the microcomputer enters pulse output port is controlled by 2 groups of 4 bits. Figures 84 and 85 show the bit configuration of the pulse output data register 0/1 (address A2$_{16}$/A4$_{16}$) and real-time output structure in pulse mode 0, respectively.

When the waveform output select bits are set to “01” (bit 1 = “0” and bit 0 = “1”), RTP0$_3$ to RTP0$_0$ become pulse output port pins, in other words, RTP0 is selected.

When the waveform output select bits are set to “10” (bit 1 = “1” and bit 0 = “0”), RTP1$_3$ to RTP1$_0$ become pulse output port pins, in other words, RTP1 is selected.

When the waveform output select bits are set to “11” (bit 1 = “1” and bit 0 = “1”), two groups consisting of RTP1$_3$ to RTP1$_0$ and RTP0$_3$ to RTP0$_0$ become pulse output port pins, in other words, RTP1 and RTP0 are selected.

When the waveform output select bits are set to “00” (bit 1 = bit 0 = “0”), port P5 pins become normal programmable I/O port pins.

The contents of the pulse output data register 1 (high-order 4 bits at address A4$_{16}$), which corresponds to RTP1$_3$ to RTP1$_0$, is output to these ports each time when the contents of timer A1 counter becomes “0000$_{16}$”. The contents of the pulse output data register 0 (low-order 4 bits at address A2$_{16}$), which corresponds to RTP0$_3$ to RTP0$_0$, is output to these ports each time when the contents of timer A0 counter becomes “0000$_{16}$”.

When “0” is written to a specified bit of the pulse output data register, a low-level signal is output to a pulse output port if the counter contents of the timer which corresponds to the bit becomes “0000$_{16}$”: when “1” is written to the bit, a high-level signal is output to a pulse output port which corresponds to the bit at the same timing.

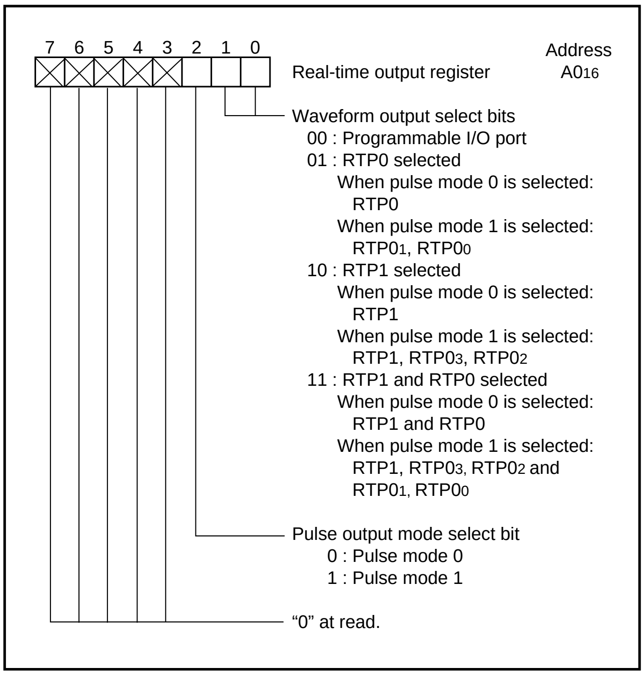

7 6 5 4 3 2 1 0
Real-time output register
Address
A0$_{16}$
Waveform output select bits
00 : Programmable I/O port
01 : RTP0 selected
When pulse mode 0 is selected:
RTP0
When pulse mode 1 is selected:
RTP0$_1$, RTP0$_0$
10 : RTP1 selected
When pulse mode 0 is selected:
RTP1
When pulse mode 1 is selected:
RTP1, RTP0$_3$, RTP0$_2$
11 : RTP1 and RTP0 selected
When pulse mode 0 is selected:
RTP1 and RTP0
When pulse mode 1 is selected:
RTP1, RTP0$_3$, RTP0$_2$ and
RTP0$_1$, RTP0$_0$
Pulse output mode select bit
0 : Pulse mode 0
1 : Pulse mode 1
“0” at read.

Fig. 83 Bit configuration real-time output control register

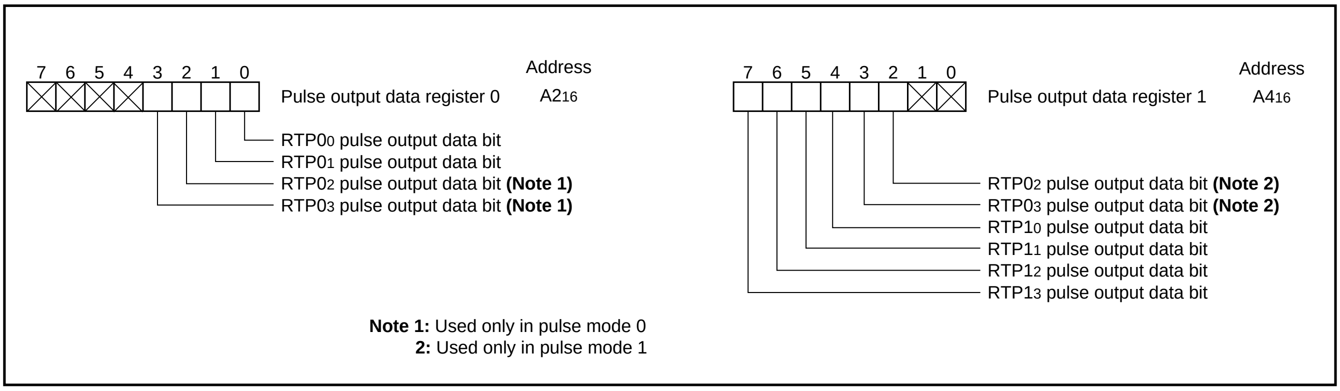

7 6 5 4 3 2 1 0
Pulse output data register 0
Address
A2$_{16}$
RTP0$_0$ pulse output data bit
RTP0$_1$ pulse output data bit
RTP0$_2$ pulse output data bit (Note 1)
RTP0$_3$ pulse output data bit (Note 1)
7 6 5 4 3 2 1 0
Pulse output data register 1
Address
A4$_{16}$
RTP0$_2$ pulse output data bit (Note 2)
RTP0$_3$ pulse output data bit (Note 2)
RTP1$_0$ pulse output data bit
RTP1$_1$ pulse output data bit
RTP1$_2$ pulse output data bit
RTP1$_3$ pulse output data bit
Note 1: Used only in pulse mode 0
2: Used only in pulse mode 1

Fig. 84 Bit configuration of pulse output data register

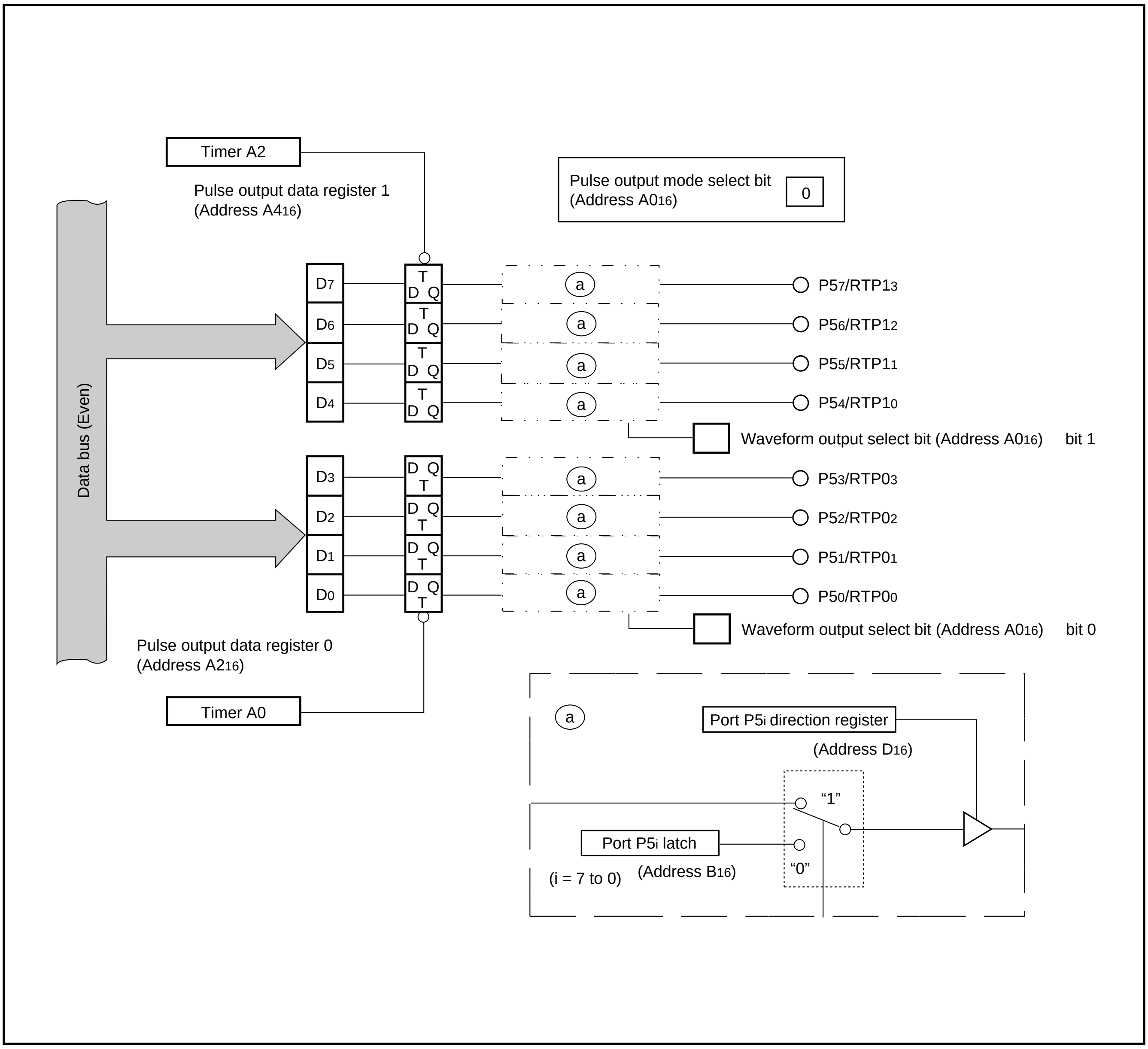


Fig. 85 Real-time output structure in pulse mode 0

(2) Pulse mode 1

When the pulse output mode select bit is set to “1”, the microcomputer enters pulse mode 1, and a pulse output port pins are separately controlled (6 bits and 2 bits).

Figures 86 shows the real-time output structure in pulse mode 1.

When the waveform output select bits are set to “01” (bit 1 = “0” and bit 0 = “1”), RTP13 to RTP10, RTP03, and RTP02 become programmable I/O port pins. Simultaneously, RTP01 and RTP00 become pulse output port pins.

When the waveform output select bits are set to “10” (bit 1 = “1” and bit 0 = “0”), RTP13 to RTP10, RTP03, and RTP02 become pulse output port pins. At this time, RTP01 and RTP00 become programmable I/O port pins.

When the waveform output select bits are set to “11” (bit 1 = bit 0 = “1”), pulse output port pins are divided into two groups; one consists of RTP13 to RTP10, RTP03, RTP02 and the other consists of RTP01 and RTP00.

When the waveform output select bits are set to “00” (bit 1 = bit 0 = “0”), port P5 pins become normal programmable I/O port pins.

RTP13 to RTP10, RTP03, and RTP02 are controlled by timer A2. Also, RTP01 and RTP00 are controlled by timer A0.

The contents of the pulse output data register 1 (high-order 6 bits at address A416), which corresponds to RTP13 to RTP10, RTP03, and RTP02, are output to this port each time when the contents of timer A2 counter becomes “000016”. The contents of the pulse output data register 0 (low-order 2 bits at address A216), which corresponds to RTP01 and RTP00, are output to this port each time when the contents of timer A0 counter become “000016”.

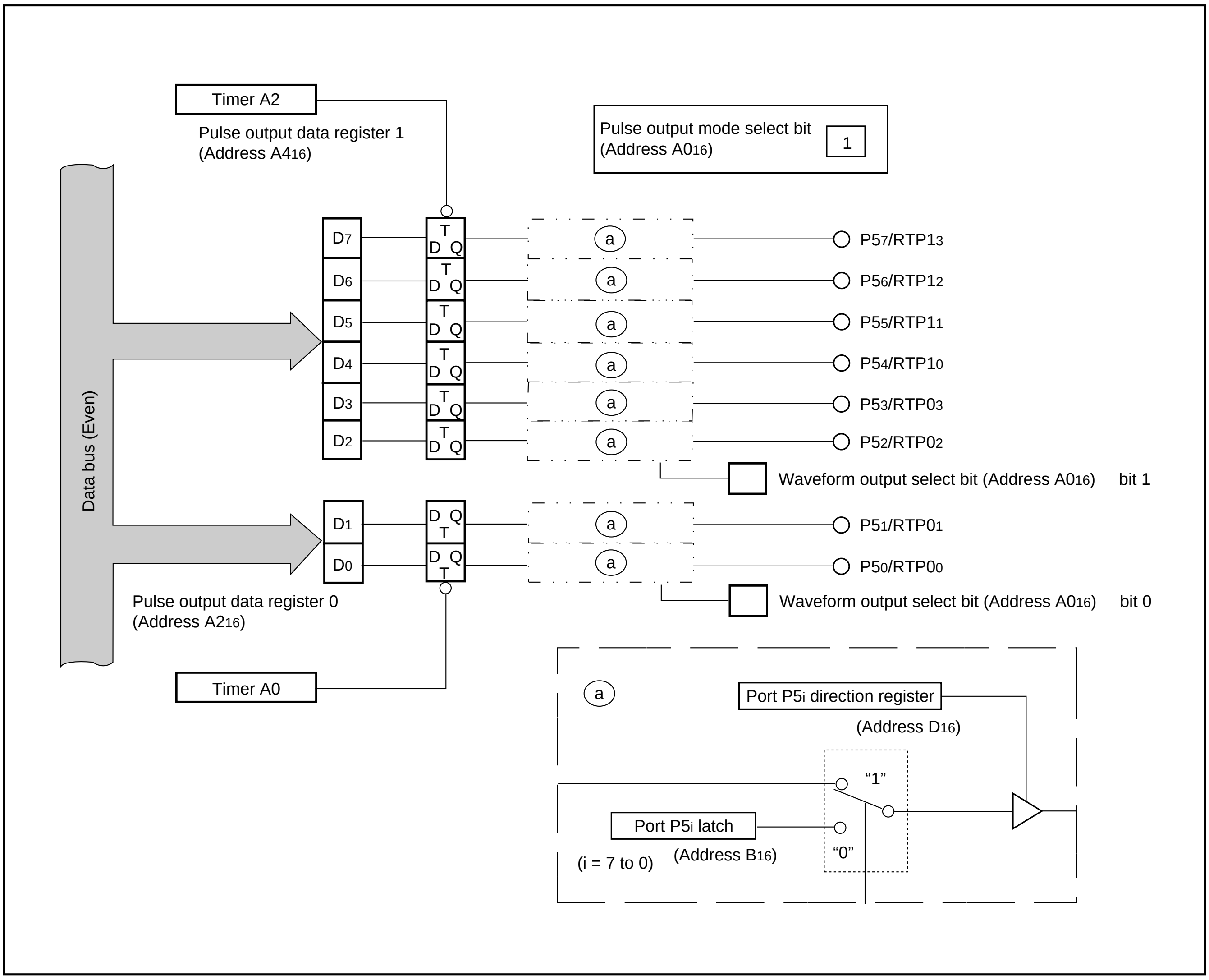


Fig. 86 Real-time output structure in pulse mode 1

Table 14 lists the port P5/RTP pin output when all of the port P5 direction registers are set to the output mode.

Precautions for real-time output function

After reset, the port P5 direction register is set to the input mode, and port P5i (i = 0 to 7) pins function as normal I/O port pins. When using these pins as real-time output port pins, set the corresponding bits of the port P5 direction register to the output mode. Additionally, by reading the real-time output port's value from the port P5 register, output level of pins can be read out.

Table 14 Port P5/RTP pin output

Real-time output control register (Address A016)			Store address for port P5/RTP pin output data							
bit 2	bit 1	bit 0	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	0	0	0B	0B	0B	0B	0B	0B	0B	0B
	0	1	0B	0B	0B	0B	A2	A2	A2	A2
	1	0	A4	A4	A4	A4	0B	0B	0B	0B
	1	1	A4	A4	A4	A4	A2	A2	A2	A2
1	0	0	0B	0B	0B	0B	0B	0B	0B	0B
	0	1	0B	0B	0B	0B	0B	0B	A2	A2
	1	0	A4	A4	A4	A4	A4	A4	0B	0B
	1	1	A4	A4	A4	A4	A4	A4	A2	A2

Address $0B_{16}$: Port P5

Address $A2_{16}$: Pulse output data register 0

Address $A4_{16}$: Pulse output data register 1

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software runaway and others. Figure 87 shows the block diagram of the watchdog timer.

The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts clock Wf_{32}, which is obtained by dividing the peripheral devices' clock f_2 by 16; or clock Wf_{512}, which is obtained by doing it by 256. Bit 0 of the watchdog timer frequency select register (watchdog timer frequency select bit) shown in Figure 88 selects which clock is to be counted.

Wf_{512} is selected when this bit 0 is "0", and Wf_{32} is selected when bit 0 is "1". Bit 0 is cleared to "0" after reset.

FFF_{16} is set in the watchdog timer when "L" level voltage is applied to pin $\overline{\text{RESET}}$, STP instruction is executed, data is written to the watchdog timer register (address 60_{16}), or the most significant bit of the watchdog timer becomes "0".

After FFF_{16} is set in the watchdog timer, when the watchdog timer counts Wf_{32} or Wf_{512} by 2048 counts, the most significant bit of watchdog timer becomes "0", the watchdog timer interrupt request bit is set to "1", and FFF_{16} is set again in the watchdog timer.

In program coding, make sure that data is written in the watchdog timer before the most significant bit of the watchdog timer becomes "0". If this routine is not executed owing to unexpected program execution or others, the most significant bit of the watchdog timer becomes "0" and an interrupt is generated.

The microcomputer can generate a reset pulse by writing "1" to bit 6 (software reset bit) of processor mode register 0 in an interrupt routine and can be restarted.

The watchdog timer can also be used to return from the **STP** state, where a clock has stopped its operation owing to the **STP** instruction execution. For details, refer to the sections on the clock generating circuit and standby function.

The watchdog timer stops its operation in the following cases, and at this time, input to the watchdog timer is disabled:

- When the external area is accessed in the hold state
- In the wait mode
- In the stop mode

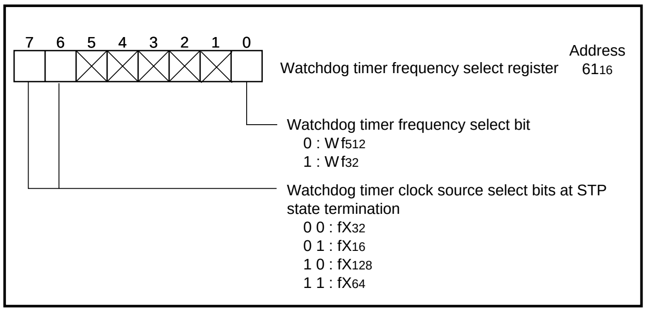
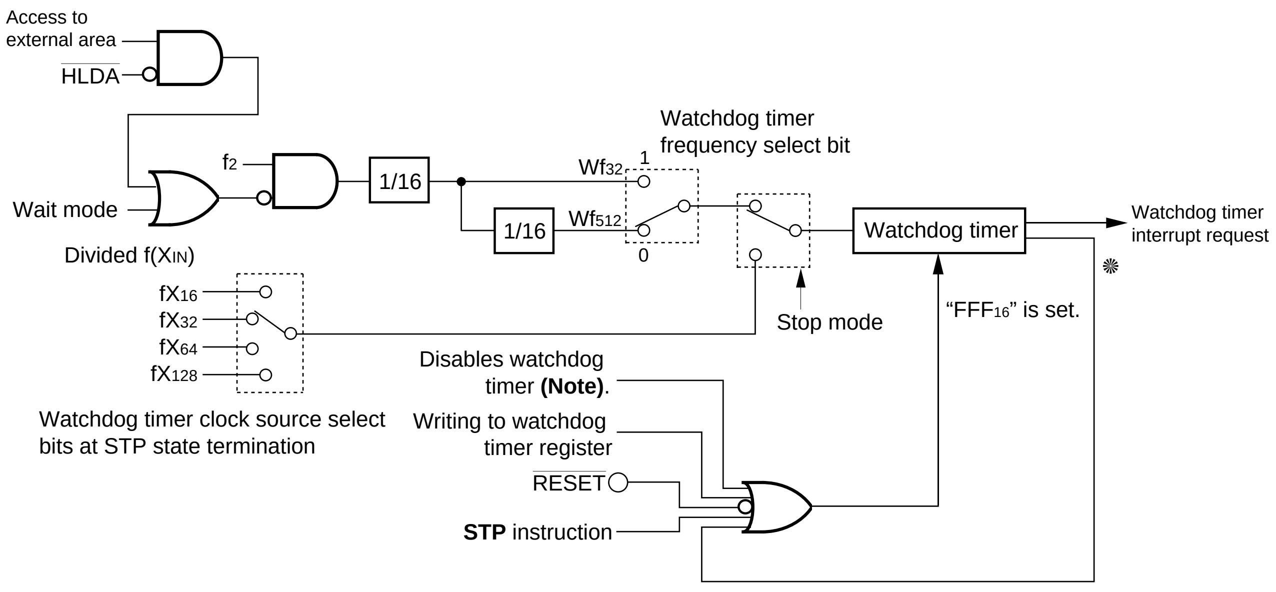

7 6 5 4 3 2 1 0
Watchdog timer frequency select register
Address
61_{16}
Watchdog timer frequency select bit
0 : Wf_{512}
1 : Wf_{32}
Watchdog timer clock source select bits at STP state termination
0 0 : fX_{32}
0 1 : fX_{16}
1 0 : fX_{128}
1 1 : fX_{64}

Fig. 88 Bit configuration of watchdog timer frequency select register


Access to external area
$\overline{\text{HLDA}}$
Wait mode
f_2
1/16
1/16
Wf_{32}
Wf_{512}
1
0
Watchdog timer frequency select bit
Watchdog timer
Watchdog timer interrupt request
❋
Divided f(X_{IN})
fX_{16}
fX_{32}
fX_{64}
fX_{128}
Watchdog timer clock source select bits at STP state termination
Stop mode
"FFF_{16}" is set.
Disables watchdog timer **(Note)**.
Writing to watchdog timer register
$\overline{\text{RESET}}$
STP instruction

- Watchdog timer register: address 60_{16}
- Watchdog timer frequency select register: bit 0 at address 61_{16}
- Watchdog timer clock source select bits at STP state termination: bits 6, 7 at address 61_{16}

❋ When the most significant bit of the watchdog timer becomes "0", this signal will be generated.

Note: During the stop mode and until the stop mode is terminated, setting for disabling the watchdog timer is ignored.

Fig. 87 Block diagram of watchdog timer

How to disable watchdog timer

When not using the watchdog timer, it can be disabled. When the watchdog timer is disabled, it's operation stops and no watchdog timer interrupt has been generated.
Setting for disabling the watchdog timer is possible by writing "79_{16}" and "50_{16}" to the particular function select register 2 (address 64_{16}) sequentially with the following instructions:

- MOVMB/STAB instruction, or
- MOVM/STA instruction (m = 1)

If any method other than above has been adopted in order to access (in other words, read/write) the particular function select register 2, the watchdog timer will not be disabled until reset operation is performed. (Also, reset is the only one method to remove the setting for disabling the watchdog timer.)
Moreover, this setting for disabling the watchdog timer is ignored at return from the STP mode, and the watchdog timer operates. (For details, refer to the section on the standby function.)

INPUT/OUTPUT PINS

Ports P0 to P8, P10, P11 all have the direction register, and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding bit of direction register is “1”, and an input pin when it is “0”.

When a pin is programmed for output, the data is written to its port latch and it is output to the output pin. When a pin is programmed for output, the contents of the port latch is read instead of the value of the pin. Accordingly, a previously output value can be read correctly even when the output “H” voltage is lowered or the output “L” voltage is raised owing to an external load, etc.

A pin programmed as an input pin is in the flooting state, and the value input to the pin can be read. When a pin is programmed as an input pin, the data is written only in the port latch and the pin remains floating.

Each of Figures 89 and 90 shows the block diagram for each port pin and pin $\overline{NMI}$. Figure 91 shows the bit configuration of the port function control register.

Bit 3 of the port function control register serves as the port P0 input level select bit, which selects the V_{IH}/V_{IL} level under the condition that port P0 is used as an input port.

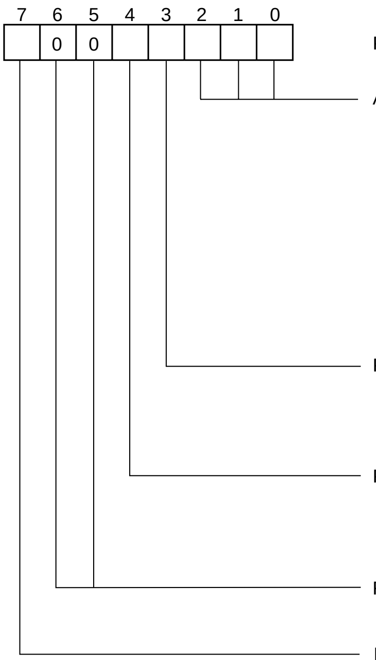
Bit 4 of the port function control register serves as the P4$_4$–P4$_7$ pullup connection select bit. This bit determines whether port pins P4$_4$–P4$_7$, which are multiplexed with chip select pins, are to be pulled up or not. At reset, this bit 4 = “0” and P$_4$–P4$_7$ are pulled up. The pullup function is valid only when the corresponding port is used an input port.

Bit 7 of the port function control register serves as the $\overline{NMI}$ pullup connection select bit. At reset, this bit 7 = “0” and pin $\overline{NMI}$ is pulled up. The pullup function is valid only when the corresponding port is used as an input port.

When using port pins P5$_4$–P5$_7$ as the key input interrupt input pins ($\overline{KI_0}$ to $\overline{KI_3}$), the pullup function can be selected, also. For details, refer to the section on interrupts.

When using a port pin as an internal peripheral device’s input pin, clear the corresponding port direction register’s bit to “0”. When using a port pin as an internal peripheral device’s output pin, the port direction register’s bit may be “0” or “1”.

In the memory expansion or microprocessor mode, port pins of P0 to P4, P10, P11 become I/O pins, and the their functions as I/O port pins are invalid. Note that, however, some port pins can function as port pins by the special setting. For details, refer to the section on the processor modes.



Port function control register — Address 92_{16} — At reset 00_{16}

Address/Port switch bits
000 : A_0 to A_{23} (16 Mbytes)
001 : A_0 to A_{21}, P0$_6$, P0$_7$ (4 Mbytes)
010 : A_0 to A_{19}, P0$_4$ to P0$_7$ (1 Mbytes)
011 : A_0 to A_{17}, P0$_2$ to P0$_7$ (256 Kbytes)
100 : A_0 to A_{15}, P0$_0$ to P0$_7$ (64 Kbytes)
101 : Do not select.
110 : A_0 to A_{11}, P0$_0$ to P0$_7$, P11$_4$ to P11$_7$ (4 Kbytes)
111 : A_0 to A_7, P0$_0$ to P0$_7$, P11$_0$ to P11$_7$ (256 bytes)

Port P0 input level select bit
0 : V_{IH} = 0.7V_{CC}, V_{IL} = 0.2V_{CC}
1 : V_{IH} = 0.43V_{CC} **(Note 1)**, V_{IL} = 0.16V_{CC}

Pins P4$_4$–P4$_7$ pullup connection select bit **(Notes 2 and 3)**
0 : Pins P4$_4$–P4$_7$ are pulled up.
1 : Pins P4$_4$–P4$_7$ are not pulled up.

Fix these bits to “0”.

Pin $\overline{NMI}$ pullup connection select bit **(Note 2)**
0 : Pin $\overline{NMI}$ is pulled up.
1 : Pin $\overline{NMI}$ is not pulled up.

Notes 1: For the M37902FxM (power source voltage = 3.3 V±0.3 V), V_{IH} = 0.5V_{CC}.
2: When MD1 = V_{CC} and MD0 = V_{CC} (flash memory parallel I/O mode), pins P4$_4$ to P4$_7$ and $\overline{NMI}$ are not pulled up, regardless of these bits’ contents.
3: When MD1 = V_{SS} and MD0 = V_{CC} (microprocessor mode), pin $\overline{CS_0}$ (P4$_4$) is not pulled up, regardless of the bit’s contents.

Fig. 91 Bit configuration of port function control register

[Inside dotted-line not included]
P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$,
P3$_1$ to P3$_3$, P10$_0$ to P10$_7$,
P11$_0$ to P11$_7$

[Inside dotted-line included]
P3$_0$/$\overline{\text{RDY}}$, P4$_3$/$\overline{\text{HOLD}}$,
P6$_1$/TA4$_{\text{IN}}$,
P6$_2$/$\overline{\text{INT}}_0$, P6$_3$/$\overline{\text{INT}}_1$, P6$_4$/$\overline{\text{INT}}_2$,
P6$_5$/TB0$_{\text{IN}}$, P6$_6$/TB1$_{\text{IN}}$, P6$_7$/TB2$_{\text{IN}}$,
P8$_2$/R$_{\text{X}}$D$_0$, P8$_6$/R$_{\text{X}}$D$_1$

[Inside dotted-line not included]
P4$_0$/ALE, P4$_1$/ϕ_1, P4$_2$/$\overline{\text{HLDA}}$,
P8$_3$/T$_{\text{X}}$D$_0$, P8$_7$/T$_{\text{X}}$D$_1$

[Inside dotted-line included]
P6$_0$/TA4$_{\text{OUT}}$

[Shaded area included]
P4$_4$/$\overline{\text{CS}}_0$, P4$_5$/$\overline{\text{CS}}_1$,
P4$_6$/$\overline{\text{CS}}_2$, P4$_7$/$\overline{\text{CS}}_3$

[Shaded area not included]
P5$_1$/TA0$_{\text{IN}}$/RTP0$_1$,
P5$_3$/TA1$_{\text{IN}}$/RTP0$_3$

[Shaded area included]
P5$_5$/TA2$_{\text{IN}}$/RTP1$_1$/$\overline{\text{KI}}_1$,
P5$_7$/TA3$_{\text{IN}}$/RTP1$_3$/$\overline{\text{KI}}_3$

[Shaded area not included]
P5$_0$/TA0$_{\text{OUT}}$/RTP0$_0$,
P5$_2$/TA1$_{\text{OUT}}$/RTP0$_2$

[Shaded area included]
P5$_4$/TA2$_{\text{OUT}}$/RTP1$_0$/$\overline{\text{KI}}_0$,
P5$_6$/TA3$_{\text{OUT}}$/RTP1$_2$/$\overline{\text{KI}}_2$

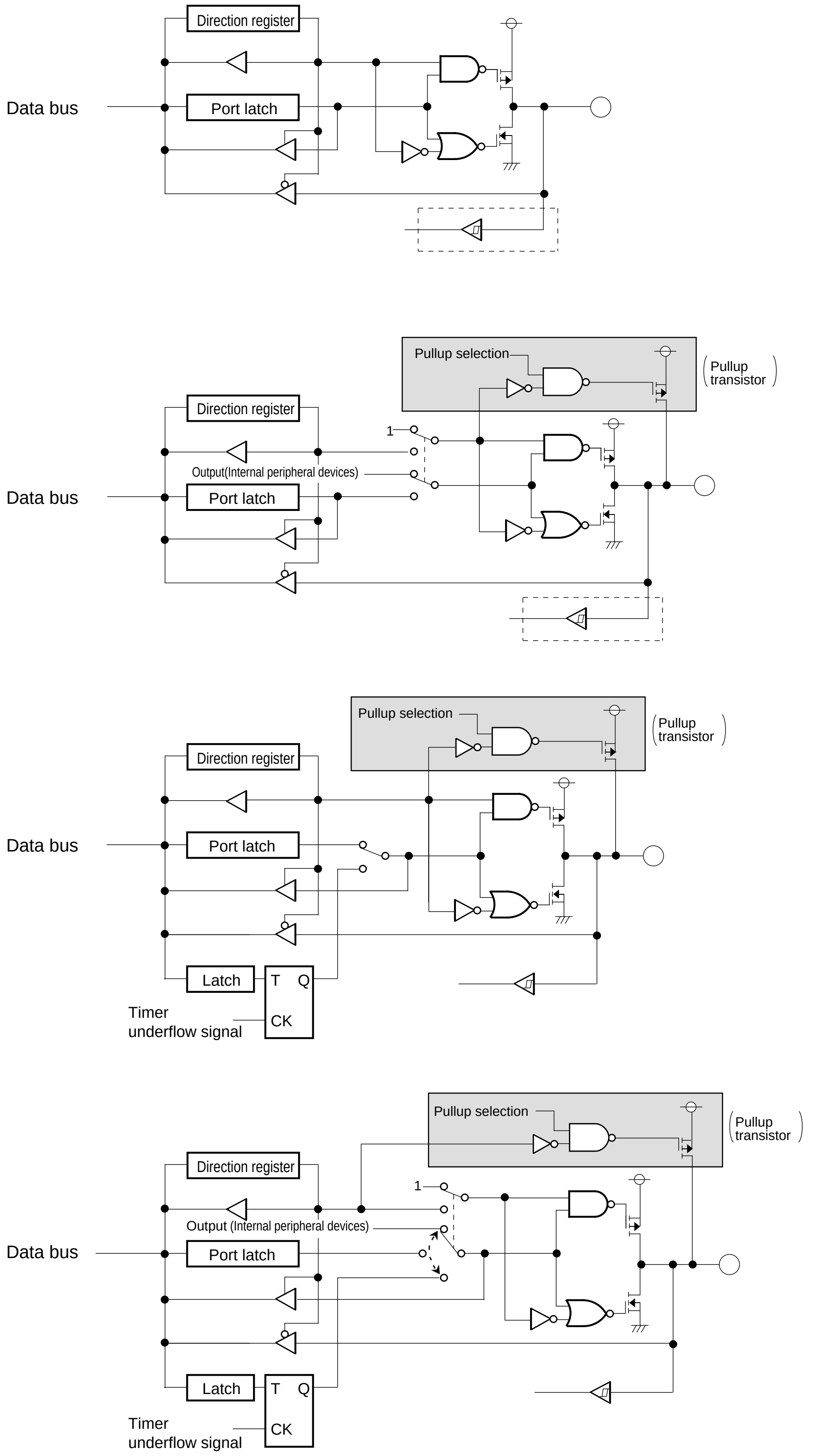


Fig. 89 Block diagram for each port pin and pin $\overline{\text{NMI}}$ (1)

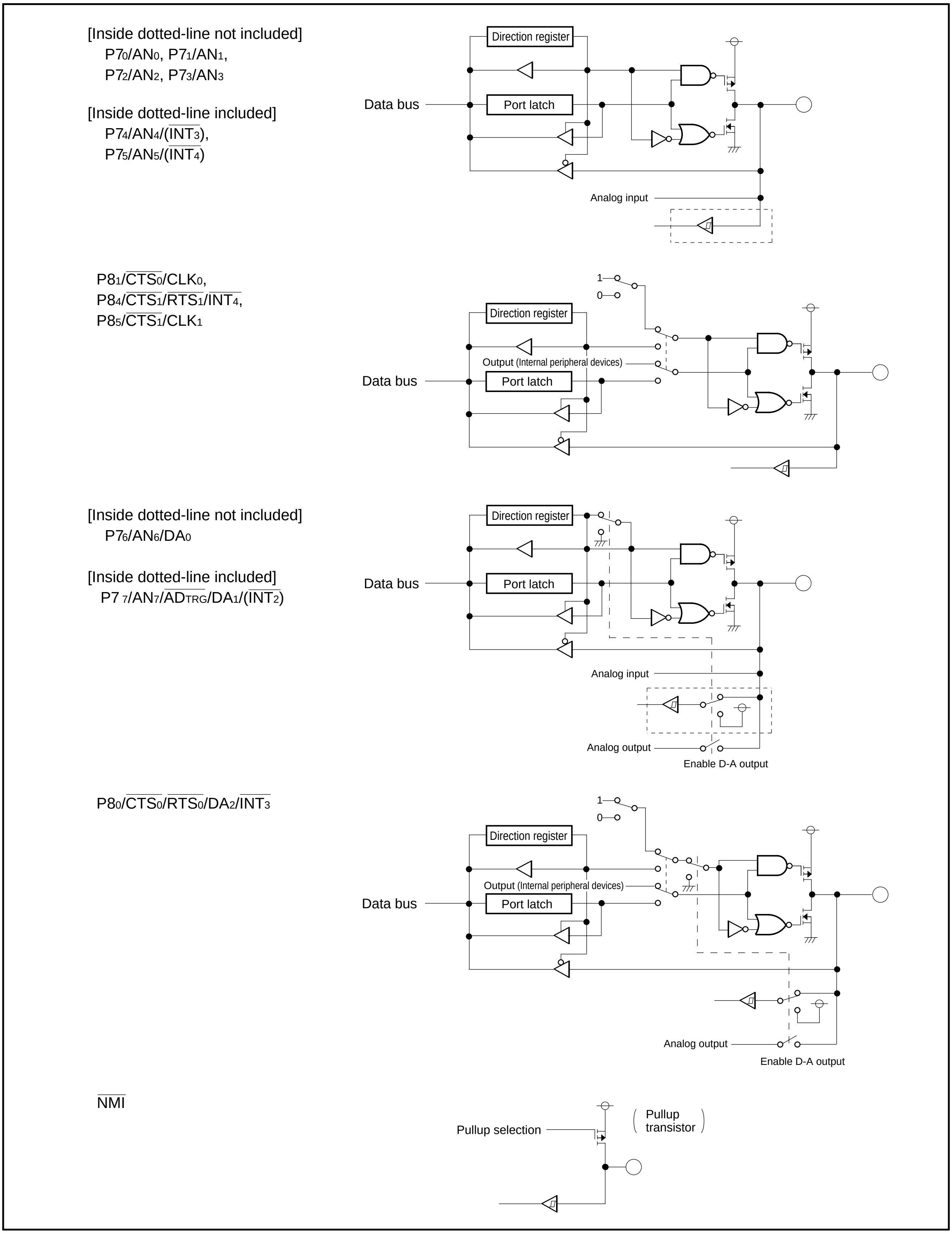

[Inside dotted-line not included]
P7$_0$/AN$_0$, P7$_1$/AN$_1$,
P7$_2$/AN$_2$, P7$_3$/AN$_3$
[Inside dotted-line included]
P7$_4$/AN$_4$/($\overline{\text{INT}_3}$),
P7$_5$/AN$_5$/($\overline{\text{INT}_4}$)
Direction register
Data bus
Port latch
Analog input
P8$_1$/$\overline{\text{CTS}_0}$/CLK$_0$,
P8$_4$/$\overline{\text{CTS}_1}$/$\overline{\text{RTS}_1}$/$\overline{\text{INT}_4}$,
P8$_5$/$\overline{\text{CTS}_1}$/CLK$_1$
1
0
Direction register
Output (Internal peripheral devices)
Data bus
Port latch
[Inside dotted-line not included]
P7$_6$/AN$_6$/DA$_0$
[Inside dotted-line included]
P7 $_7$/AN$_7$/$\overline{\text{AD}_{\text{TRG}}}$/DA$_1$/($\overline{\text{INT}_2}$)
Direction register
Data bus
Port latch
Analog input
Analog output
Enable D-A output
P8$_0$/$\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$/DA$_2$/$\overline{\text{INT}_3}$
1
0
Direction register
Output (Internal peripheral devices)
Data bus
Port latch
Analog output
Enable D-A output
$\overline{\text{NMI}}$
Pullup selection
(Pullup transistor)

Fig. 90 Block diagram for each port pin and pin $\overline{\text{NMI}}$ (2)

RESET CIRCUIT

While the power source voltage satisfies the recommended operating condition, reset state is removed if pin $\overline{\text{RESET}}$'s level returns from the stabilized "L" level to the "H" level. As a result, program execution starts from the reset vector address. This reset vector address is expressed as shown below:

- A_{23} to A_{16} = 00_{16}
- A_{15} to A_8 = Contents at address $FFFF_{16}$
- A_7 to A_0 = Contents at address $FFFE_{16}$

Figures 92 and 93 show the microcomputer internal register's status at reset, and Figure 94 shows an operation example of the reset circuit. Apply "L" level voltage to pin $\overline{\text{RESET}}$ for a period (2 µs or more) under the following conditions:

- Pin Vcc's level satisfies the recommended operating condition.
- Oscillator's operation has been stabilized.

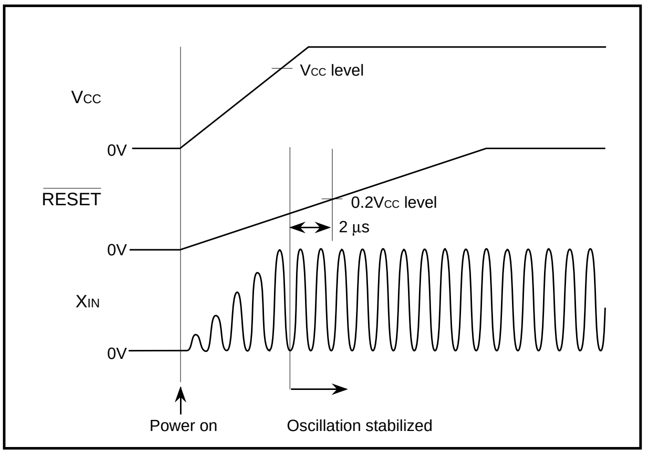


Fig. 94 Operation example of reset circuit (Note that proper evaluation is necessary in the system development stage.)

	Address								
$\overline{CS_0}$ control register L	(80_{16})···	(Note 2)	1	0	☒	0	(Note 3)	1	0
$\overline{CS_0}$ control register H	(81_{16})···	1	☒	☒	☒	0	0	0	1
$\overline{CS_1}$ control register L	(82_{16})···	0	1	0	☒	0	(Note 3)	1	0
$\overline{CS_1}$ control register H	(83_{16})···	0	☒	0	☒	0	0	0	0
$\overline{CS_2}$ control register L	(84_{16})···	0	1	0	☒	0	(Note 3)	1	0
$\overline{CS_2}$ control register H	(85_{16})···	0	☒	0	☒	0	0	0	0
$\overline{CS_3}$ control register L	(86_{16})···	0	1	0	☒	0	(Note 3)	1	0
$\overline{CS_3}$ control register H	(87_{16})···	☒	☒	☒	☒	0	0	0	0
Area $\overline{CS_0}$ start address register	($8A_{16}$)···	0	0	0	1	0	0	0	0
Area $\overline{CS_1}$ start address register	($8C_{16}$)···	0	0	0	0	0	0	0	0
Area $\overline{CS_2}$ start address register	($8E_{16}$)···	0	0	0	0	0	0	0	0
Area $\overline{CS_3}$ start address register	(90_{16})···	0	0	0	0	0	0	0	0
Port function control register	(92_{16})···	0	0	0	0	0	0	0	0
External interrupt input control register	(94_{16})···	0	0	0	0	0	0	0	0
D-A control register	(96_{16})···	☒	☒	☒	☒	☒	0	0	0

	Address	
D-A register 0	(98_{16})···	00_{16}
D-A register 1	(99_{16})···	00_{16}
D-A register 2	($9A_{16}$)···	00_{16}
Flash memory control register	($9E_{16}$)···	☒ ☒ 0 0 0 0 0 1
Real-time output control register	($A0_{16}$)···	☒ ☒ ☒ ☒ ☒ 0 0 0
Serial I/O pin control register	(AC_{16})···	☒ ☒ ☒ ☒ 0 0 0 0
Clock control register	(BC_{16})···	0 0 0 0 0 1 1 1
Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
Program bank register PG		00_{16}
Program counter PC_H		Contents at address $FFFF_{16}$
Program counter PC_L		Contents at address $FFFE_{16}$
Direct page registers DPR0 to DPR3		0000_{16}
Data bank register DT		00_{16}
Stack pointer		FFF_{16}

Notes 1: The contents of the other registers and RAM are undefined at reset and must be initialized by software.
2: While Vss level voltage is applied to pin MD0, this bit is "0". While Vcc level voltage is applied to pin MD0, on the other hand, this bit is "1".
3: While Vss level voltage is applied to pin BYTE, these bits are "0". While Vcc level voltage is applied to pin BYTE, on the other hand, these bits are "1".

Fig. 93 Microcomputer internal register's status at reset (2)

	Address								
Port P0 direction register	(04_{16})···	00_{16}							
Port P1 direction register	(05_{16})···	00_{16}							
Port P2 direction register	(08_{16})···	00_{16}							
Port P3 direction register	(09_{16})···	☒	☒	☒	☒	0	0	0	0
Port P4 direction register	$(0C_{16})$···	00_{16}							
Port P5 direction register	$(0D_{16})$···	00_{16}							
Port P6 direction register	(10_{16})···	00_{16}							
Port P7 direction register	(11_{16})···	00_{16}							
Port P8 direction register	(14_{16})···	00_{16}							
Port P10 direction register	(18_{16})···	00_{16}							
Port P10 direction register	(19_{16})···	00_{16}							
A-D control register 0	$(1E_{16})$···	0	0	0	0	0	?	?	?
A-D control register 1	$(1F_{16})$···	☒	0	0	0	0	0	1	1
UART 0 Transmit/Receive mode register	(30_{16})···	00_{16}							
UART 1 Transmit/Receive mode register	(38_{16})···	00_{16}							
UART 0 Transmit/Receive control register 0	(34_{16})···	0	0	0	0	1	0	0	0
UART 1 Transmit/Receive control register 0	$(3C_{16})$···	0	0	0	0	1	0	0	0
UART 0 Transmit/Receive control register 1	(35_{16})···	0	0	0	0	0	0	1	0
UART 1 Transmit/Receive control register 1	$(3D_{16})$···	0	0	0	0	0	0	1	0
Count start register	(40_{16})···	00_{16}							
One-shot start register	(42_{16})···	0	☒	☒	0	0	0	0	0
Up-down register	(44_{16})···	00_{16}							
Timer A clock frequency select register	(45_{16})···	☒	☒	☒	☒	☒	☒	0	0
Timer A0 mode register	(56_{16})···	00_{16}							
Timer A1 mode register	(57_{16})···	00_{16}							
Timer A2 mode register	(58_{16})···	00_{16}							
Timer A3 mode register	(59_{16})···	00_{16}							
Timer A4 mode register	$(5A_{16})$···	00_{16}							
Timer B0 mode register	$(5B_{16})$···	0	0	?	☒	0	0	0	0

	Address								
Timer B1 mode register	$(5C_{16})$···	0	0	?	☒	0	0	0	0
Timer B2 mode register	$(5D_{16})$···	0	0	?	☒	0	0	0	0
Processor mode register 0	$(5E_{16})$···	(Note 2)	0	0	0	1	0	(Note 2)	0
Processor mode register 1	$(5F_{16})$···	0	0	**(Note 2)**				0	0
Watchdog timer	(60_{16})···	FFF_{16}							
Watchdog timer frequency select register	(61_{16})···	0	0	☒	☒	☒	☒	☒	0
Particular function select register 0	(62_{16})···	0	☒	0	☒	☒	☒	0	0
Particular function select register 1	(63_{16})···	☒	0	☒	0	0	0	**(Note 3)**	
Debug control register 0	(66_{16})···	1	(Note 3)						
Debug control register 1	(67_{16})···	0	0	0	(Note 3)	0	0	0	(Note 3)
$\overline{\text{INT}_3}$ interrupt control register	$(6E_{16})$···	☒	☒	☒	0	0	0	0	0
$\overline{\text{INT}_4}$ interrupt control register	$(6F_{16})$···	☒	☒	☒	0	0	0	0	0
A-D conversion interrupt control register	(70_{16})···	☒	☒	☒	☒	?	0	0	0
UART 0 transmit interrupt control register	(71_{16})···	☒	☒	☒	☒	0	0	0	0
UART 0 receive interrupt control register	(72_{16})···	☒	☒	☒	☒	0	0	0	0
UART 1 transmit interrupt control register	(73_{16})···	☒	☒	☒	☒	0	0	0	0
UART 1 receive interrupt control register	(74_{16})···	☒	☒	☒	☒	0	0	0	0
Timer A0 interrupt control register	(75_{16})···	☒	☒	☒	☒	0	0	0	0
Timer A1 interrupt control register	(76_{16})···	☒	☒	☒	☒	0	0	0	0
Timer A2 interrupt control register	(77_{16})···	☒	☒	☒	☒	0	0	0	0
Timer A3 interrupt control register	(78_{16})···	☒	☒	☒	☒	0	0	0	0
Timer A4 interrupt control register	(79_{16})···	☒	☒	☒	☒	0	0	0	0
Timer B0 interrupt control register	$(7A_{16})$···	☒	☒	☒	☒	0	0	0	0
Timer B1 interrupt control register	$(7B_{16})$···	☒	☒	☒	☒	0	0	0	0
Timer B2 interrupt control register	$(7C_{16})$···	☒	☒	☒	☒	0	0	0	0
$\overline{\text{INT}_0}$ interrupt control register	$(7D_{16})$···	☒	☒	0	0	0	0	0	0
$\overline{\text{INT}_1}$ interrupt control register	$(7E_{16})$···	☒	☒	0	0	0	0	0	0
$\overline{\text{INT}_2}$ interrupt control register	$(7F_{16})$···	☒	☒	0	0	0	0	0	0

Notes 1: The contents of the other registers and RAM are undefined at reset and must be initialized by software.
2: While Vss level voltage is applied to pin MD0, these bits are “0”. While Vcc level voltage is applied to pin MD0, on the other hand, these bits are “1”.
3: At power-on reset, these bits are clear to “0”. At hardware or software reset, on the other hand, these bits retain the value just before reset.

Fig. 92 Microcomputer internal register’s status at reset (1)

OSCILLATION CIRCUIT

An oscillation circuit locates between pins X_{IN} and X_{OUT}, and Figure 95 shows a circuit example with a oscillator (an external ceramic resonator or quartz crystal oscillator). The constants such as capacitance etc. depend on a oscillator. Therefore, for these constants, adopt the oscillator manufacturer's recommended values.

Figure 96 shows a circuit example with an external clock source. When an external clock is input, be sure to leave pin X_{OUT} open. Also, in this case, when the external clock input select bit (bit 1 of the particular function select register 0; See Figure 100.) is set to "1", the oscillation circuit stops it's operation, and the current dissipation is reduced. Moreover, this bit has another function, which selects the return condition from the stop mode. For details, refer to the section on the standby function.

On the other hand, the PLL (Phase Locked Loop) frequency multiplier (hereafter, referred as PLL circuit.) is included, also. This PLL circuit uses an clock input from pin X_{IN} and generates a multiplied clock. When using the PLL circuit, be sure to connect pin V_{CONT} with an external filter circuit. (See Figure 97.) When not using the PLL circuit, be sure to leave pin V_{CONT} open.

When not using the PLL circuit, be sure to clear the PLL circuit operation enable bit (bit 1 of the clock control register; See Figure 99.), so that the PLL circuit will stop it's operation.

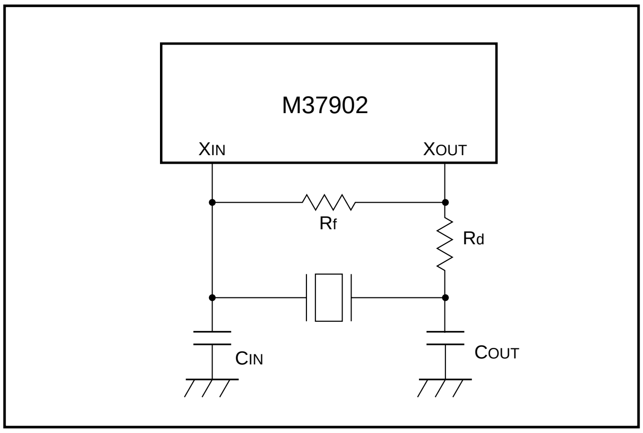


Fig. 95 Circuit example with external ceramic resonator or quartz crystal oscillator

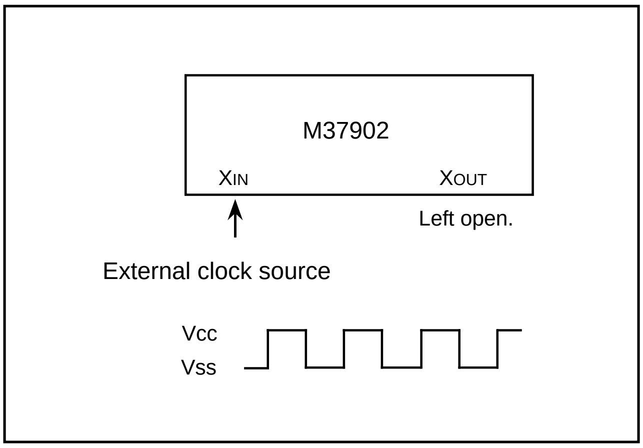
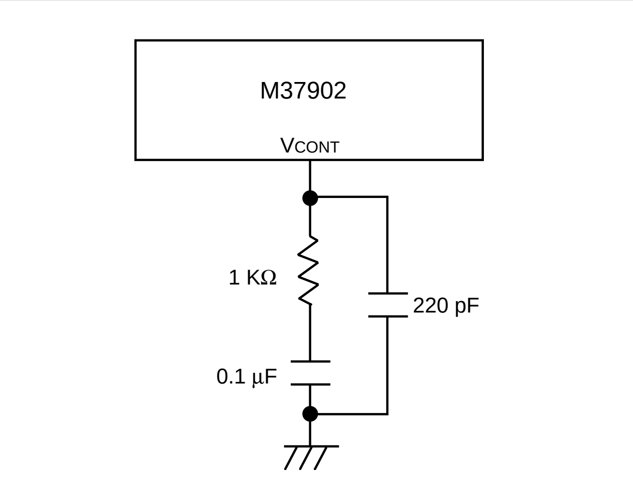


Fig. 96 Circuit example with external clock source



Note: Make the wiring length as short as possible, and shield it with the GND line which surrounds this circuit. Also, for the clock supply to pin X_{IN}, see Figures 95 and 96.

Fig. 97 Circuit example with pin V_{CONT} and PLL circuit

CLOCK GENERATING CIRCUIT

Figure 98 shows the block diagram of the clock generating circuit. The clock generating circuit consists of the clock oscillation circuit, PLL frequency multiplier (PLL circuit), system clock switch circuit, peripheral devices' clock switch circuit, clock divider, standby control circuit, etc. As control registers for the clock generating circuit, also, the clock control register (address BC_{16}), particular function select register 0 (address 62_{16}) are provided. (See Figures 99 and 100.)

As shown in Figure 98, clocks used in the CPU, BIU, peripheral devices, watchdog timer (in other words, clocks ϕ_{CPU}, ϕ_{BIU}, f_1 to f_{4096}, Wf_{32}, Wf_{512}) are made from system clock f_{sys}. System clock f_{sys} can be selected between fX_{IN} (in other words, a clock input from pin X_{IN}) and f_{PLL} (in other words, an output clock generated by the PLL circuit). By setting the clock ϕ_1 output select bit (bit 7 of the processor mode register 0) to "1", also, system clock f_{sys} can be output from port pin $P4_1$, as clock ϕ_1.

The PLL circuit's operation, system clock (f_{sys}) selection, and divide ratio selection for peripheral devices' clocks (f_1 to f_{4096}) are controlled by the clock control register. The following describes about these control.

Bit 1 of the clock control register (the PLL circuit operation enable bit) selects the PLL circuit's operation (stopped/active). When this bit is set to "1", pin V_{CONT} will becomes valid, and the PLL circuit will operate. At reset, the PLL circuit operation enable bit becomes "1". (In this case, the PLL circuit operates.) When not using the PLL circuit, be sure to clear the PLL circuit operation enable bit to "0" (stopped). At the **STP** instruction execution or while the flash memory parallel I/O mode is set, the PLL circuit stops its operation, and pin V_{CONT} is invalid, regardless of this bit 1's status.

Bits 2 and 3 of the clock control register (the PLL multiplication ratio select bits) select the ratio of f_{PLL}/fX_{IN}. The PLL multiplication ratio must be set so that the frequency of the PLL output clock (f_{PLL}) must be in the range from 10 MHz to 26 MHz. At reset, the PLL multiplication ratio select bits become "0,1" (× 2). The change of the multiplication ratio must be performed while input clock fX_{IN} is set as system clock. (In this case, bit 5 of the clock control register = "0".) After that, be sure to wait that the operation-stabilizing time of the PLL circuit has passed, and switch the system clock to the PLL output clock (f_{PLL}). (In other words, set bit 5 to "1".) Note that, after reset, the PLL multiplication ratio select bits are allowed to be changed only once.

Bit 5 of the clock control register is the system clock select bit, and fX_{IN} is selected as the system clock when bit 5 = "0". On the other hand, when bit 5 = "1", the PLL output clock (f_{PLL}) is selected. At reset, the system clock select bit becomes "0". When selecting f_{PLL}, be sure that the PLL circuit's operation has been stabilized properly, and then, set the system clock select bit to "1". Also, when the PLL circuit operation enable bit is cleared to "0" (the PLL circuit is stopped.), the system clock select bit will automatically be cleared to "0". Note that a value of "1" cannot be written to the system clock select bit while the PLL circuit operation enable bit ="0".

Table 15 lists the f_{sys} selection.

Bits 6 and 7 of the clock control register are the peripheral devices' clock select bits 0, 1, and these bits select the multiplication ratio of (f_1 to f_{4096})/(f_{sys}).

Table 16 lists the internal peripheral devices' operation clock frequency. At reset, these bits become "0, 0".

Table 15. f_{sys} selection

System clock select bit (Bit 5)	PLL circuit operation enable bit (Bit 1)	PLL multiplication ratio select bits (Bits 3, 2) **(Note)**	System clock f_{sys}	
			Clock source	Frequency **(Note)**
0	——	——	fX_{IN}	$f(X_{IN})$
1	1	01 (× 2)	f_{PLL}	$f(X_{IN})$ × 2
		10 (× 3)	f_{PLL}	$f(X_{IN})$ × 3
		11 (× 4)	f_{PLL}	$f(X_{IN})$ × 4

Note: The PLL multiplication ratio must be set so that the frequency of the PLL output clock (f_{PLL}) must be in the range from 10 MHz to 26 MHz. $f(X_{IN})$ means the frequency of the input clock from pin X_{IN} (fX_{IN}). After reset, the PLL multiplication ratio select bits are allowed to be changed only once.

Table 16. Internal peripheral devices' operation clock frequency

Internal peripheral devices' operation clock	Peripheral devices' clock select bits 1, 0 (bits 7, 6)			
	0 0	0 1 **(Note)**	1 0	1 1
f_1	f_{sys}	f_{sys}	$f_{sys}/2$	Do not select.
f_2	$f_{sys}/2$	f_{sys}	$f_{sys}/4$	
f_{16}	$f_{sys}/16$	$f_{sys}/8$	$f_{sys}/32$	
f_{64}	$f_{sys}/64$	$f_{sys}/32$	$f_{sys}/128$	
f_{512}	$f_{sys}/512$	$f_{sys}/256$	$f_{sys}/1024$	
f_{4096}	$f_{sys}/4096$	$f_{sys}/2048$	$f_{sys}/8192$	

Note: When selecting the peripheral devices' clock select bits 1, 0 = "01_2", be sure that system clock f_{sys} does not exceed 13 MHz.

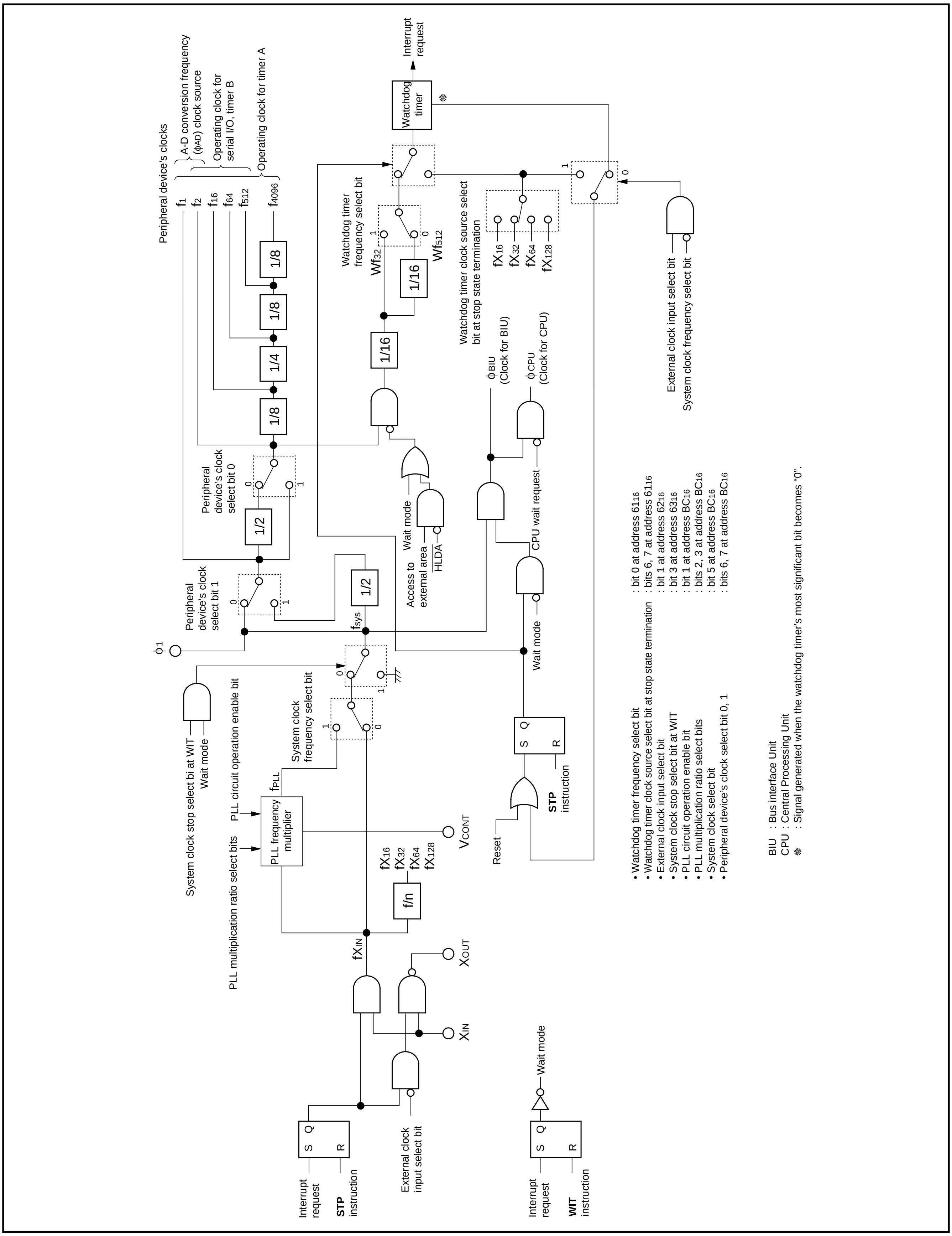
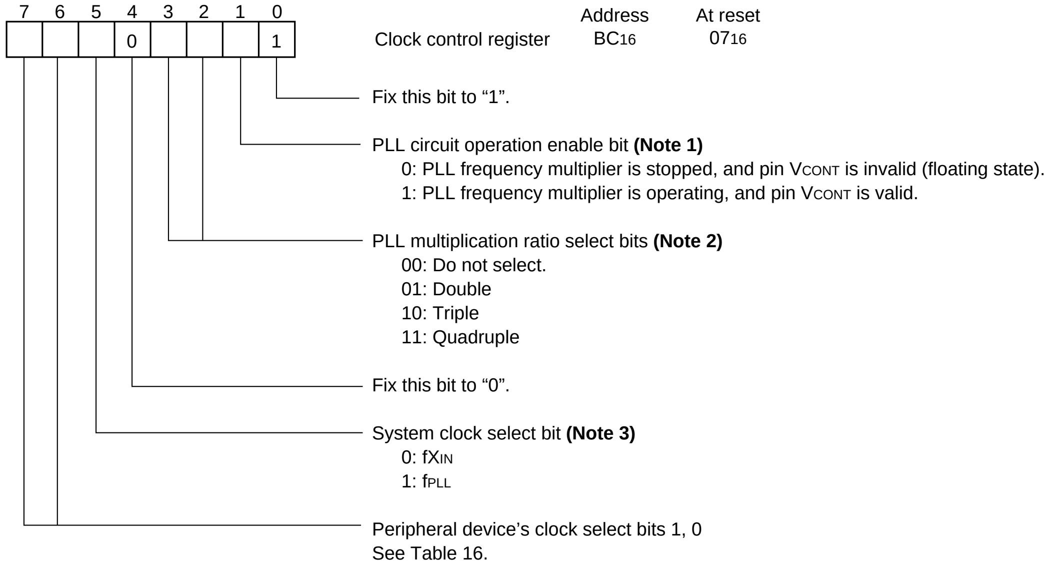

Peripheral device's clocks
f1
f2
f16
f64
f512
f4096
A-D conversion frequency (φAD) clock source
Operating clock for serial I/O, timer B
Operating clock for timer A
1/8
1/4
1/8
1/8
1/2
1/2
1/16
1/16
Peripheral device's clock select bit 0
Peripheral device's clock select bit 1
φ1
fsys
Watchdog timer frequency select bit
Wf32
Wf512
Watchdog timer
Interrupt request
Watchdog timer clock source select bit at stop state termination
fX16
fX32
fX64
fX128
External clock input select bit
System clock frequency select bit
φBIU (Clock for BIU)
φCPU (Clock for CPU)
Wait mode
Access to external area
HLDA
CPU wait request
Wait mode
S Q R
STP instruction
Reset
System clock stop select bi at WIT
Wait mode
PLL circuit operation enable bit
PLL multiplication ratio select bits
PLL frequency multiplier
fPLL
System clock frequency select bit
VCONT
f/n
fX16
fX32
fX64
fX128
fXIN
XOUT
XIN
External clock input select bit
Interrupt request
STP instruction
Interrupt request
WIT instruction
Wait mode
0
1
• Watchdog timer frequency select bit : bit 0 at address 61₁₆
• Watchdog timer clock source select bit at stop state termination : bits 6, 7 at address 61₁₆
• External clock input select bit : bit 1 at address 62₁₆
• System clock stop select bit at WIT : bit 3 at address 63₁₆
• PLL circuit operation enable bit : bit 1 at address BC₁₆
• PLL multiplication ratio select bits : bits 2, 3 at address BC₁₆
• System clock select bit : bit 5 at address BC₁₆
• Peripheral device's clock select bit 0, 1 : bits 6, 7 at address BC₁₆
BIU : Bus interface Unit
CPU : Central Processing Unit
✹ : Signal generated when the watchdog timer's most significant bit becomes "0".

Fig. 98 Block diagram of clock generating circuit



Fix this bit to “1”.

PLL circuit operation enable bit **(Note 1)**
0: PLL frequency multiplier is stopped, and pin V_{CONT} is invalid (floating state).
1: PLL frequency multiplier is operating, and pin V_{CONT} is valid.

PLL multiplication ratio select bits **(Note 2)**
00: Do not select.
01: Double
10: Triple
11: Quadruple

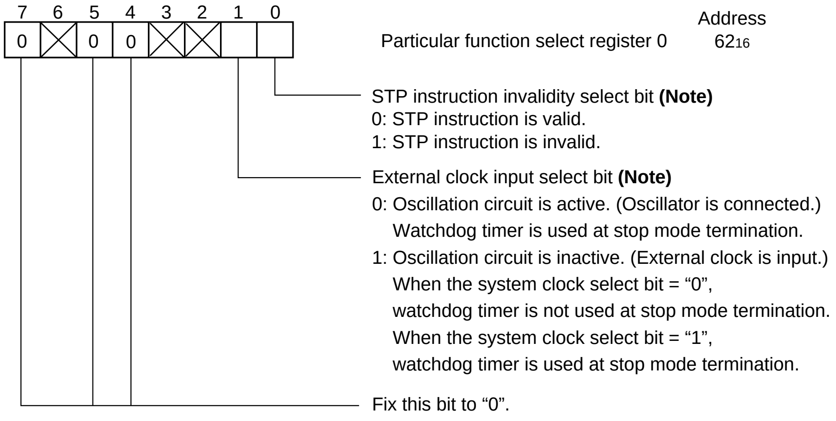
Fix this bit to “0”.

System clock select bit **(Note 3)**
0: f_{XIN}
1: f_{PLL}

Peripheral device’s clock select bits 1, 0
See Table 16.

Notes 1: When not using the PLL frequency multiplier, clear this bit to “0”. In the stop mode or in the flash memory parallel I/O mode, the PLL circuit stops it’s operation regardless of this bit’s contents; at this time, pin V_{CONT} is invalid.
2: When rewriting this bit, be sure to clear bit 5 to “0” simultaneously. Also, after this bit is rewritten, insert a waiting time of 2 ms, and then set bit 5 to “1”.
3: When the PLL circuit operation enable bit (bit 1) has been cleared to “0”, this bit will also be cleared to “0”. Also, bit 1 = “0”, nothing can be written to this bit. (Fixed to be “0”.)

Fig. 99 Bit configuration of clock control register



STP instruction invalidity select bit **(Note)**
0: STP instruction is valid.
1: STP instruction is invalid.

External clock input select bit **(Note)**
0: Oscillation circuit is active. (Oscillator is connected.)
Watchdog timer is used at stop mode termination.
1: Oscillation circuit is inactive. (External clock is input.)
When the system clock select bit = “0”,
watchdog timer is not used at stop mode termination.
When the system clock select bit = “1”,
watchdog timer is used at stop mode termination.

Fix this bit to “0”.

Note: Writing to these bits requires the following procedure:
• Write “55_{16}” to this register. (The bit status does not change only by this writing.)
• Succeedingly, write “0” or “1” to each bit.
Also, use the MOVM (MOVMB) instruction or STA (STAB, STAD) instruction

Fig. 100 Bit configuration of particular function select register 0

STANDBY FUNCTION

The standby function provides the stop (hereafter called STP) and the wait (hereafter called WIT) mode. These modes are used to save the power dissipation of the system, by stopping oscillation or system clock in the case that the CPU needs not be operating.

The microcomputer enters the STP or WIT mode by executing the STP or WIT instruction, and either mode is terminated by acceptance of an interrupt request or reset.

To terminate the STP or WIT mode by an interrupt request, the interrupt to be used for termination of the STP or WIT mode must be enabled in advance to execution of the STP or WIT instruction. The interrupt priority level of this interrupt is required to be higher than the processor interrupt priority level (IPL) of the routine where the STP or WIT instruction will be executed.

Figures 100 to 102 show the bit configurations of the particular function select registers 0, 1, and watchdog timer frequency select register respectively. Setting the STP instruction invalidity select bit (bit 0 of the particular function select register 0) to “1” invalidates the STP instruction, and the STP instruction will be ignored. Since the above bit is cleared to “0” after reset is removed, however, the STP instruction is valid.

The STP- or the WIT-instruction-execution status bit (bit 0 or 1 of the particular function select register 1) is set to “1” by the execution of the STP or the WIT instruction, and so, after the STP or WIT mode has been terminated, each bit will indicate that the STP or WIT instruction has been executed. Accordingly, each of these bits must be cleared to “0” by software at termination of the STP or the WIT mode.

Table 17 explains the microcomputer’s operation in the STP and WIT modes.

The external bus fixation function can also be provided. This function enables the user to specify the states of the external bus and the bus control signals in the memory expansion and the microprocessor mode in the STP or WIT mode. For more information, refer to the section on the power saving function.

STP mode

The execution of the STP instruction stops the oscillation circuit and PLL circuit. It also stops input clock fX_{IN}, system clock f_{sys}, ϕ_{BIU}, ϕ_{CPU}, and peripheral devices’ clocks f_1 to f_{4096}, Wf_{32} and Wf_{512} in the “L” state, and divide clocks fX_{16} to fX_{128} in the “H” state. In the watchdog timer, “FFF$_{16}$” is automatically set. As shown in Figure 98, any one of divide clocks fX_{16} to fX_{128}, which is selected by the watchdog timer clock source select bits at STP termination (bits 6 and 7 of the watchdog timer frequency select register), becomes the watchdog timer’s clock source.

In the STP mode, the A-D converter and watchdog timer, which uses peripheral devices’ clocks f_1 to f_{4096}, Wf_{32} and Wf_{512}, are stopped. At this time, timers A and B operate only in the event counter mode, and serial I/O communication is active while an external clock is selected.

The STP mode is terminated by acceptance of an interrupt request or reset, and the oscillation circuit and PLL circuit restart their operations. Input clock fX_{IN}, system clock f_{sys}, and peripheral devices’ clocks f_1 to f_{4096}, Wf_{32} and Wf_{512} are also supplied.

When the STP mode is terminated by reset, supply of ϕ_{BIU} and ϕ_{CPU} starts immediately after the oscillation circuit and PLL circuit restart their operations. Therefore, the reset input must be raised “H” after the operation-stabilizing time for these circuits has passed.

The following two modes are available in order to terminate the STP mode by an interrupt:

(1) The watchdog timer is used in order to measure the period from the operation restart of the oscillation circuit and PLL circuit until the supply start of ϕ_{BIU} and ϕ_{CPU}.

(2) The supply of ϕ_{BIU} and ϕ_{CPU} is started immediately after the operation restart of the oscillation circuit and PLL circuit.

When the external clock input select bit (bit 1 of the particular function select register 0) = “0” or the system clock select bit (bit 5 of the clock control register) = “1”, the watchdog timer will start counting

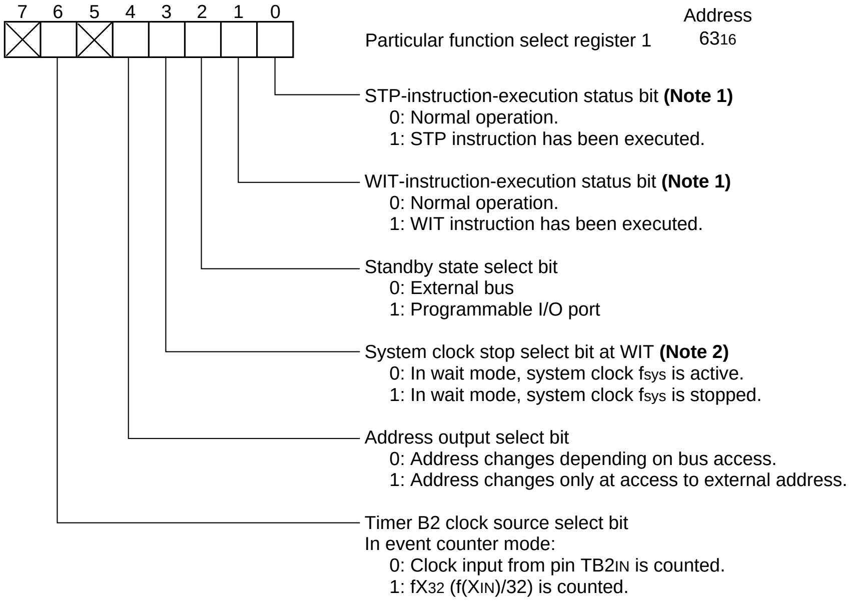
Table 17. Microcomputer’s operation in STP and WIT modes

Mode	System clock stop select bit at WIT	Operations of function while WIT, STP modes					
		Oscillation circuit	PLL circuit	f_{sys}, ϕ_1, f_1 to f_{4096}	Wf_{32}, Wf_{512}	ϕ_{BIU}, ϕ_{CPU}	Peripheral devices using f_1 to f_{4096}, Wf_{32}, Wf_{512}
STP	—	Stopped	Stopped	Stopped (“L”)	Stopped (“L”)	Stopped (“L”)	Timers A, B: Operation is enabled only in the event counter mode. Serial I/O: Operation is enabled only while an external clock is selected. A-D converter: Stopped. (Watchdog timer: Stopped.)
WIT	“0”	Active **(Note 1)**	Active **(Note 2)**	Active	Stopped (“L”)	Stopped (“L”)	Timers A, B, Serial I/O, A-D converter: Operation is enabled. (Watchdog timer: Stopped.)
	“1”	Active **(Note 1)**	Active **(Note 2)**	Stopped (“L”)	Stopped (“L”)	Stopped (“L”)	Timers A, B: Operation is enabled only in the event counter mode. Serial I/O: Operation is enabled only while an external clock is selected. A-D converter: Stopped. (Watchdog timer: Stopped.)

Notes 1: When the external clock input select bit = “1”, the oscillation circuit stops. Also, clock input from pin X_{IN} is available.
2: When the PLL operation enable bit = “0”, the PLL circuit stops.

down with one of the above divide clocks, fX_{16} to fX_{128}, after the oscillation circuit and PLL circuit have been restarted their operations owing to an interrupt. The most significant bit of the watchdog timer reaching “0”, supply of ϕ_{BIU} and ϕ_{CPU} restarts.

On the other hand, when the external clock input select bit = “1 ” and the system clock select bit = “0”, supply of ϕ_{BIU} and ϕ_{CPU} will restart immediately after the oscillation circuit has been restarted their operations owing to an interrupt. (In actual fact, after the selected one of the above divide clocks, fX_{16} to fX_{128}, has been changed from “H” to “L”, this supply will restart.)



Particular function select register 1 — Address 63_{16}

STP-instruction-execution status bit **(Note 1)**
0: Normal operation.
1: STP instruction has been executed.

WIT-instruction-execution status bit **(Note 1)**
0: Normal operation.
1: WIT instruction has been executed.

Standby state select bit
0: External bus
1: Programmable I/O port

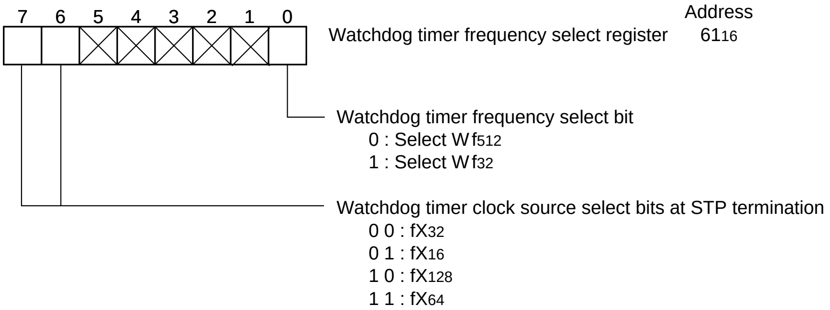
System clock stop select bit at WIT **(Note 2)**
0: In wait mode, system clock f_{sys} is active.
1: In wait mode, system clock f_{sys} is stopped.

Address output select bit
0: Address changes depending on bus access.
1: Address changes only at access to external address.

Timer B2 clock source select bit
In event counter mode:
0: Clock input from pin $TB2_{IN}$ is counted.
1: fX_{32} ($f(X_{IN})/32$) is counted.

Notes 1: At power-on reset, this bit becomes “0”. At hardware reset or software reset, this bit retains the value just before reset. Even when “1” is written, the bit status will not change.
2: Setting this bit to “1” must be performed just before execution of the WIT instruction. Also, after the wait state is terminated, this bit must be cleared to “0” immediately.

Fig. 101 Bit configuration of particular function select register 1



Watchdog timer frequency select register — Address 61_{16}

Watchdog timer frequency select bit
0 : Select Wf_{512}
1 : Select Wf_{32}

Watchdog timer clock source select bits at STP termination
0 0 : fX_{32}
0 1 : fX_{16}
1 0 : fX_{128}
1 1 : fX_{64}

Fig. 102 Bit configuration of watchdog timer frequency select register

WIT mode

When the WIT instruction is executed with the system clock stop select bit at WIT (bit 3 of the particular function select register 1 in Figure 101) being “0”, ϕ_{BIU}, ϕ_{CPU}, and divide clocks Wf_{32} and Wf_{512} are stopped in the “L“ state. However, the oscillation circuit, PLL circuit, input clock f_{XIN}, system clock f_{sys}, ϕ_1, and peripheral devices’ clock f_1 to f_{4096} remain operating. Therefore, BIU and CPU are stopped, whereas timers A and B, serial I/O, and the A-D converter, which use the peripheral devices’ clocks f_1 to f_{4096}, are still operating. Note that the watchdog timer is stopped.

On the other hand, when the WIT instruction is executed with the system clock stop select bit at WIT being “1”, the oscillation circuit, PLL circuit, and input clock f_{XIN} are operating, while system clock f_{sys}, ϕ_{BIU}, ϕ_{CPU}, and peripheral devices’ clocks stop operating. As a result, the A-D converter and watchdog timer, which use peripheral devices’ clocks f_1 to f_{4096}, Wf_{32} and Wf_{512}, are stopped. At this time, timers A and B operate only in the event counter mode, and serial I/O communication is active only while an external clock is selected. If the internal peripheral devices are not used in the WIT mode, the latter is better because the current dissipation is more saved. Note that the system clock stop select bit at WIT is to be set to “1” immediately before execution of the WIT instruction and cleared to “0” immediately after the WIT mode is terminated.

The WIT state is terminated by acceptance of an interrupt request, and then, supply of ϕ_{BIU} and ϕ_{CPU} will restart. Since the oscillation circuit, PLL circuit, and clock input f_{XIN} are operating in the WIT mode, an interrupt processing can be executed just after the WIT mode termination.

POWER SAVING FUNCTION

The following functions can save the power dissipation of the whole system.

(1) External bus fixation in standby state

By setting the standby state select bit (bit 2 of the particular function select register 1) to “1”, in the stop or wait mode, the I/O pins of the external buses and bus control signals can be switched to programmable I/O port pins. By setting these pins’ state with the corresponding port registers and port direction registers, unnecessary current will not flow between the microcomputer and external devices. As a result, in the stop or wait mode, the power dissipation of the whole system can be lowered. Table 18 lists the correspondence between the external buses, bus control signals, and programmable I/O port pins.

This function is valid only in the stop or wait mode. At termination of the stop or wait mode, the original function of external buses and bus control signals become valid.

Table 18. Correspondence between external buses, bus control signals, and programmable I/O port pins

External buses, Bus control signals	Standby state select bit	
	0	1
A_0 to A_7, A_8 to A_{15}, A_{16} to A_{23}	A_0 to A_7, A_8 to A_{15}, A_{16} to A_{23}	$P10_0$ to $P10_7$, $P11_0$ to $P11_7$, $P0_0$ to $P0_7$
D_0 to D_7, D_8 to D_{15}	D_0 to D_7, D_8 to D_{15} **(Note)**	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$
$\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$	$\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$ **(Note)**	$P3_1$, $P3_2$, $P3_3$
$\overline{CS_0}$	$\overline{CS_0}$	$P9_0$

Note: When the external data bus width = 8 bits (BYTE = V_{CC} level), this becomes a programmable I/O port pin, regardless of the standby state select bit’s contents.

(2) Stop of system clock in wait mode

In the wait mode, if the internal peripheral devices need not to operate, the system clock stop select bit at WIT (bit 3 of the particular function select register 1) = “1”, both of system clock f_{sys} and peripheral devices’ clock stop their operations, and the power dissipation can be saved.

For details, refer to the section on the standby function.

(3) Stop of oscillation circuit

When an externally-generated-stable clock is input to pin X_{IN}, the power dissipation can be saved if both of the following conditions are met:

- the external clock input select bit (bit 1 of the particular function select register 0) = “1”.
- the oscillation driver between pins X_{IN} and X_{OUT} stops its operation.

At this time, the output level at pin X_{OUT} is fixed to “H”. When not using a PLL output clock, also, the supply of ϕ_{BIU} and ϕ_{CPU} restarts their operations just after the microcomputers returns from the stop mode, owing to an interrupt request occurrence. Therefore, an instruction can be executed just after the termination of the stop mode. For details, refer to the section on the clock generating circuit and standby function.

(4) Disconnection from pin V_{REF}

When not using the A-D converter and D-A converter, by setting the V_{REF} connection select bit (bit 6 of the A-D control register 1) to “1”, the resistor ladder network of the A-D converter will be disconnected from the reference voltage input pin (V_{REF}). In this case, no current flows from pin V_{REF} to the resistor ladder network, and the power dissipation can be saved. Note that, after the V_{REF} connection select bit changes from “1” (V_{REF} disconnected) to “0” (V_{REF} connected), be sure that a waiting time of 1 µs of more has passed before the A-D conversion starts. For details, refer to the sections on the A-D converter and D-A converter.

(5) Address output selection

In the memory expansion mode or microprocessor mode, when the address output select bit (bit 4 of the particular function select register 1) becomes “1”, the unnecessary change of address pins’ state will be avoided, without output of an address at access to the internal area.

For details, refer to the section on the BIU.

DEBUG FUNCTION

When the CPU fetches an instruction code, an interrupt request will be generated if a selected condition is satisfied, as a resultant of comparison between a specified address and the start address where the instruction code is stored (the contents of PG and PC). The decision whether this condition is satisfied or not is called address matching detection, and the interrupt generated by this detection is called an address matching detection interrupt. (For interrupt vector addresses, refer to the section on interrupts.)

In the address matching detection, a non-maskable interrupt routine is proceeded without execution of the original instruction which has been allocated to the target address.

The debug function provides the following two modes:

- the address matching detection mode, which is used to avoid the area where program exists or modify a program.
- the out-of-address-area detection mode, which is used to detect a program runaway.

Figures 103 shows the block diagram of the debug function. Figures 104 and 105 show the bit configurations of the debug control registers 0, 1, and address compare registers 0,1, respectively.

The detect condition select bits of the debug control register 0 can select one condition between the following 4 conditions. When the selected address condition is satisfied, an address matching detection interrupt request will be generated:

(1) Address matching detection 0
The contents of PG and PC match with the address which has been set in the address compare register 0.

(2) Address matching detection 1
The contents of PG and PC match with the address which has been set in the address compare register 1.

(3) Address matching detection 2
The contents of PG and PC match with the address which has been set in either of the address compare register 0 or address compare register 1.

(4) Out-of-address-area detection
The contents of PG and PC are less than the address which has been set in the address compare register 0 or larger than the address which has been set in the address compare register 1.

By setting the detect enable bit of the debug control register 0 to “1”, an address matching detection interrupt request will be generated if any one of the above address conditions is satisfied. Clearing the detect enable bit to “0” generates no interrupt request even if any of the above address conditions is satisfied.

The address compare register access enable bit of the debug control register 1 must be set to “1” by the instruction just before the access operation (read/write). Then, this bit must be cleared to “0” (disabled) by the next instruction. While this bit = “0”, the address compare registers 0, 1 cannot be accessed.

The address-matching-detection 2 decision bit of the debug control register 1 decides, whether the address which has been set in the address compare register 0 or 1 matches with the contents of PG, PC, when the address matching detection 2 is selected. The contents of this bit is invalid when address matching detection 0 or 1 is selected.

In order to use the debug function to avoid the area where program exists or modify a program, perform the necessary processing within an address matching interrupt routine. As a result, the contents of PG, PC, PS at acceptance of an address matching detection interrupt request (i.e. the address at which an address matching detection condition is satisfied) have been pushed on to the stack. If a return destination address after the interrupt processing is to be altered, rewrite the contents of the stack, and then return by the RTI instruction.

To use the debug function to detect a program runaway, set an address area where no program exists into the address compare registers 0 and 1 by using the out-of-address-area detection. When the CPU fetches instruction codes from this address area and executes them, an address matching detection interrupt request will be generated.

The above debug function cannot be evaluated by a debugger, so that the debug function must not be used while a debugger is running.

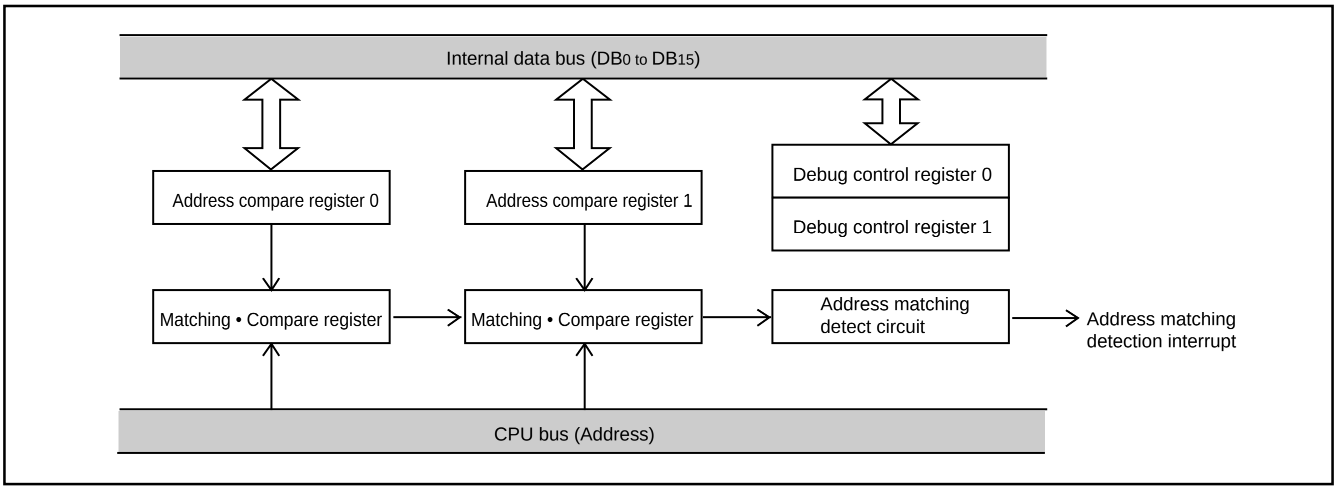
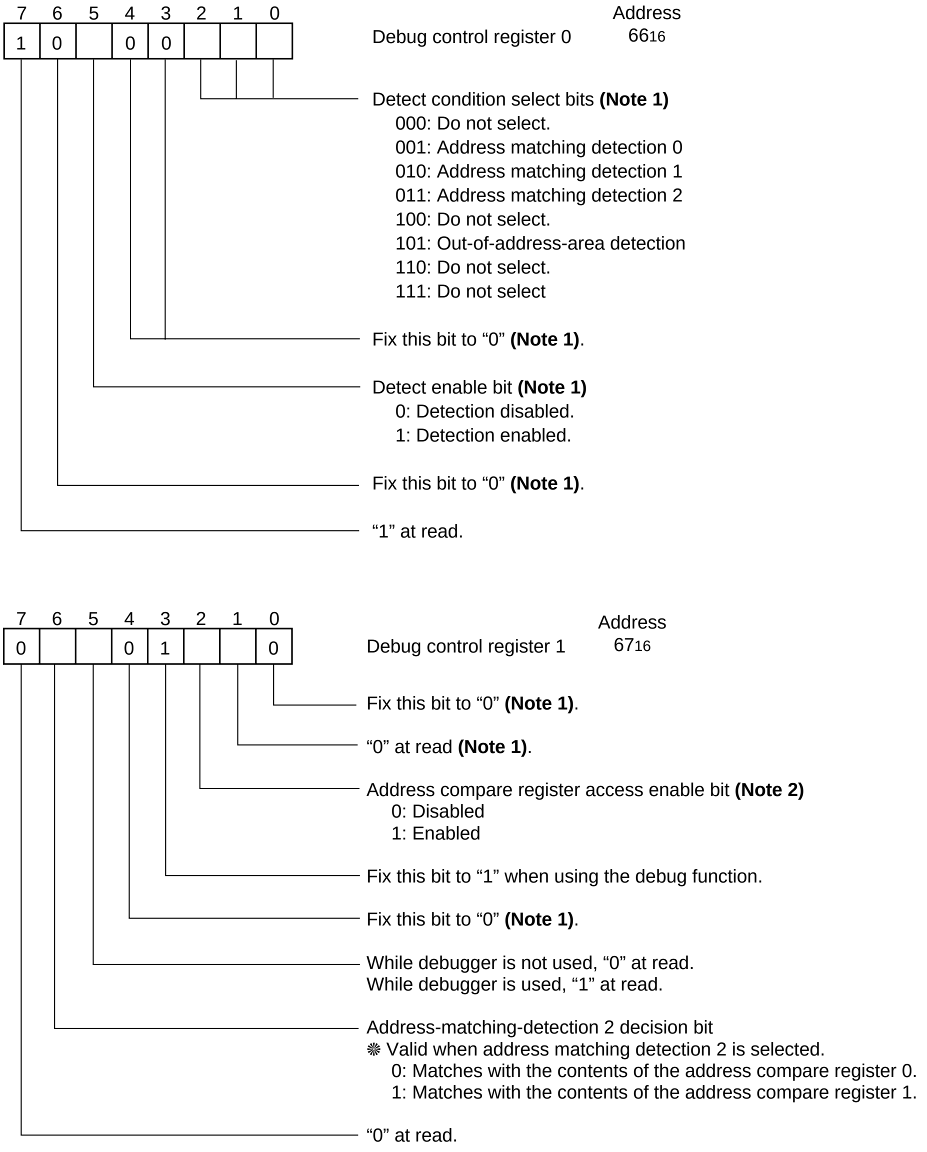

Internal data bus (DB0 to DB15)
Address compare register 0
Address compare register 1
Debug control register 0
Debug control register 1
Matching • Compare register
Matching • Compare register
Address matching detect circuit
Address matching detection interrupt
CPU bus (Address)

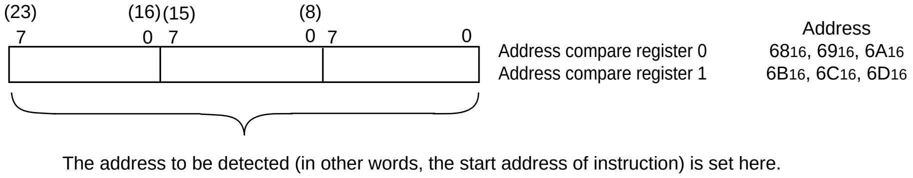
Fig 103. Block diagram of debug function


7 6 5 4 3 2 1 0
1 0 0 0
Debug control register 0
Address
66₁₆
Detect condition select bits (Note 1)
000: Do not select.
001: Address matching detection 0
010: Address matching detection 1
011: Address matching detection 2
100: Do not select.
101: Out-of-address-area detection
110: Do not select.
111: Do not select
Fix this bit to “0” (Note 1).
Detect enable bit (Note 1)
0: Detection disabled.
1: Detection enabled.
Fix this bit to “0” (Note 1).
“1” at read.
7 6 5 4 3 2 1 0
0 0 1 0
Debug control register 1
Address
67₁₆
Fix this bit to “0” (Note 1).
“0” at read (Note 1).
Address compare register access enable bit (Note 2)
0: Disabled
1: Enabled
Fix this bit to “1” when using the debug function.
Fix this bit to “0” (Note 1).
While debugger is not used, “0” at read.
While debugger is used, “1” at read.
Address-matching-detection 2 decision bit
❋ Valid when address matching detection 2 is selected.
0: Matches with the contents of the address compare register 0.
1: Matches with the contents of the address compare register 1.
“0” at read.

Notes 1: At power-on reset, these bits = “0”; at hardware reset or software reset, these bits retain the value just before reset.

2: Set this bit to “1” with the instruction just before the address compare register 0, 1 (addresses 68_{16} to $6D_{16}$) is accessed. And then, clear this bit to “0” with the instruction just after the access.

Fig. 104 Bit configuration of debug control register 0, 1



The address to be detected (in other words, the start address of instruction) is set here.

Fig. 105 Bit configuration of address compare register 0, 1

FLASH MEMORY MODE

These microcomputers contain the DINOR (DIvided bit line NOR)-type flash memory; and single-power-supply reprogramming is available to this. These microcomputers have the following three modes, enabling reading/programming/erasure for the flash memory:

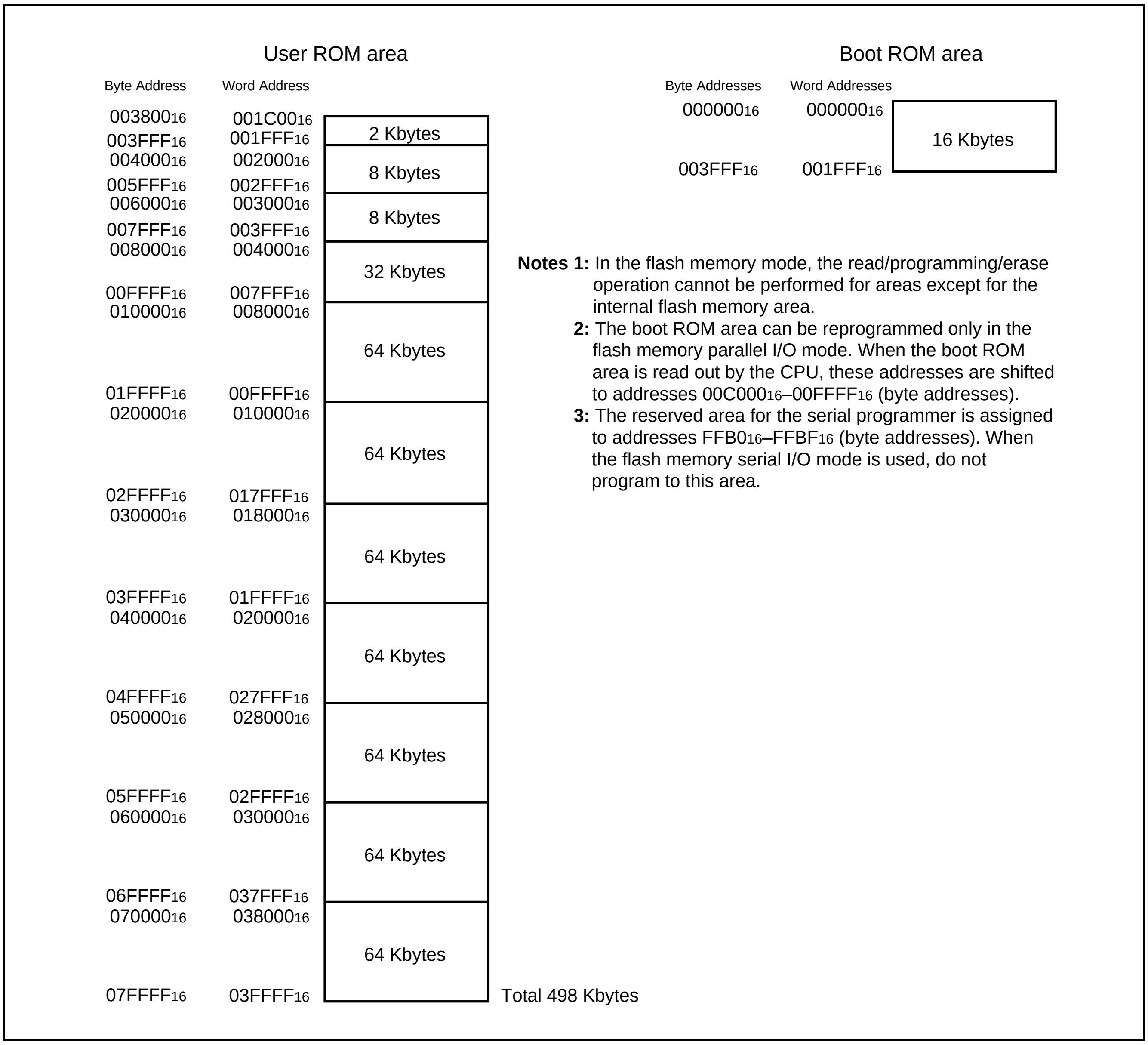
- Flash memory parallel I/O mode and Flash memory serial I/O mode, where the flash memory is handled by using an external programmer.
- CPU reprogramming mode, where the flash memory is handled by the central processing unit (CPU).

As shown in Figures 106 to 108, the flash memory is divided into several blocks, and erasure per block is possible.

Each of these blocks is provided with a lock bit, which determines the validity of erasure/program execution. Therefore, data protection per block is possible.

This internal flash memory has the boot ROM area storing the reprogramming control software for reprogramming in the CPU reprogramming mode and flash memory serial I/O mode, as well as the user ROM area storing a certain control software for the normal operation in the microcomputer mode.

Although our reprogramming control firmware for the flash memory serial I/O mode has been stored into this boot ROM area on shipment, the user-original reprogramming control software which is more appropriate for the user’s system is reprogrammable into this area, instead. Note that the reprogramming for the boot ROM area is enabled only in the flash memory parallel I/O mode.

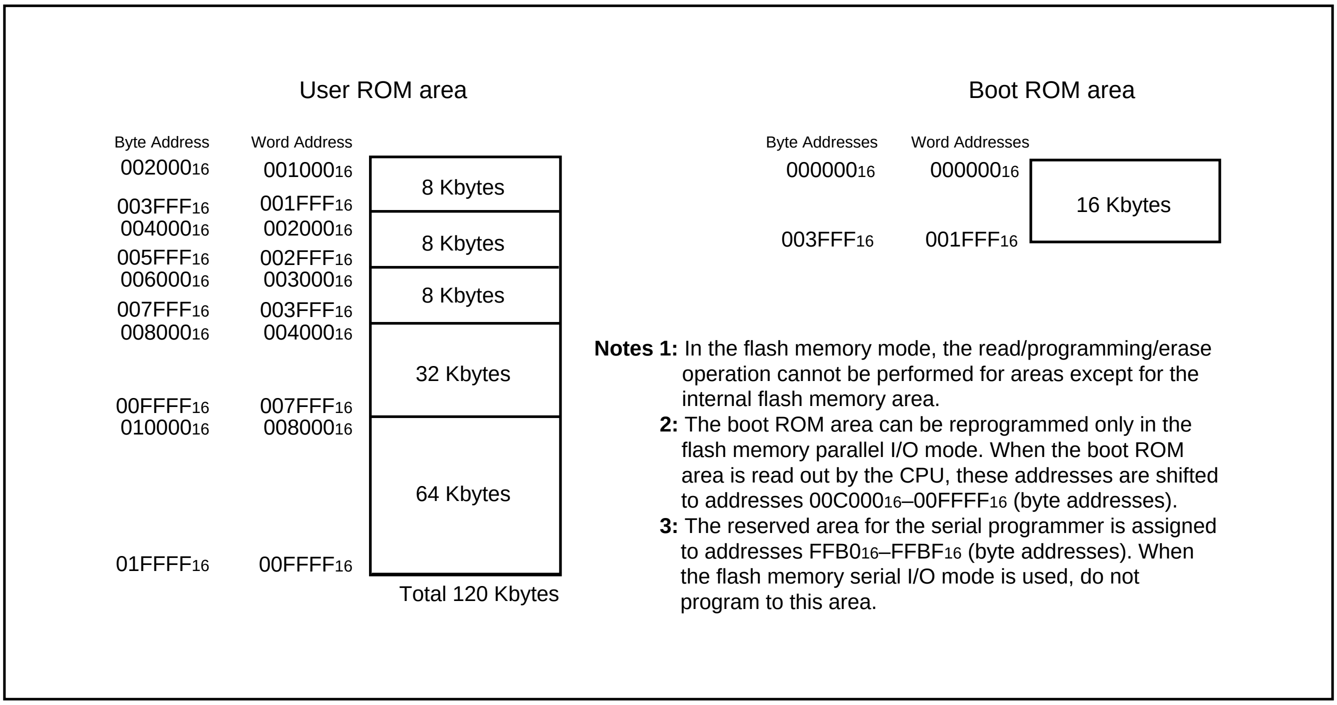


Notes 1: In the flash memory mode, the read/programming/erase operation cannot be performed for areas except for the internal flash memory area.

2: The boot ROM area can be reprogrammed only in the flash memory parallel I/O mode. When the boot ROM area is read out by the CPU, these addresses are shifted to addresses 00C000$_{16}$–00FFFF$_{16}$ (byte addresses).

3: The reserved area for the serial programmer is assigned to addresses FFB0$_{16}$–FFBF$_{16}$ (byte addresses). When the flash memory serial I/O mode is used, do not program to this area.

Fig 106. M37902FJCHP: block configuration of internal flash memory

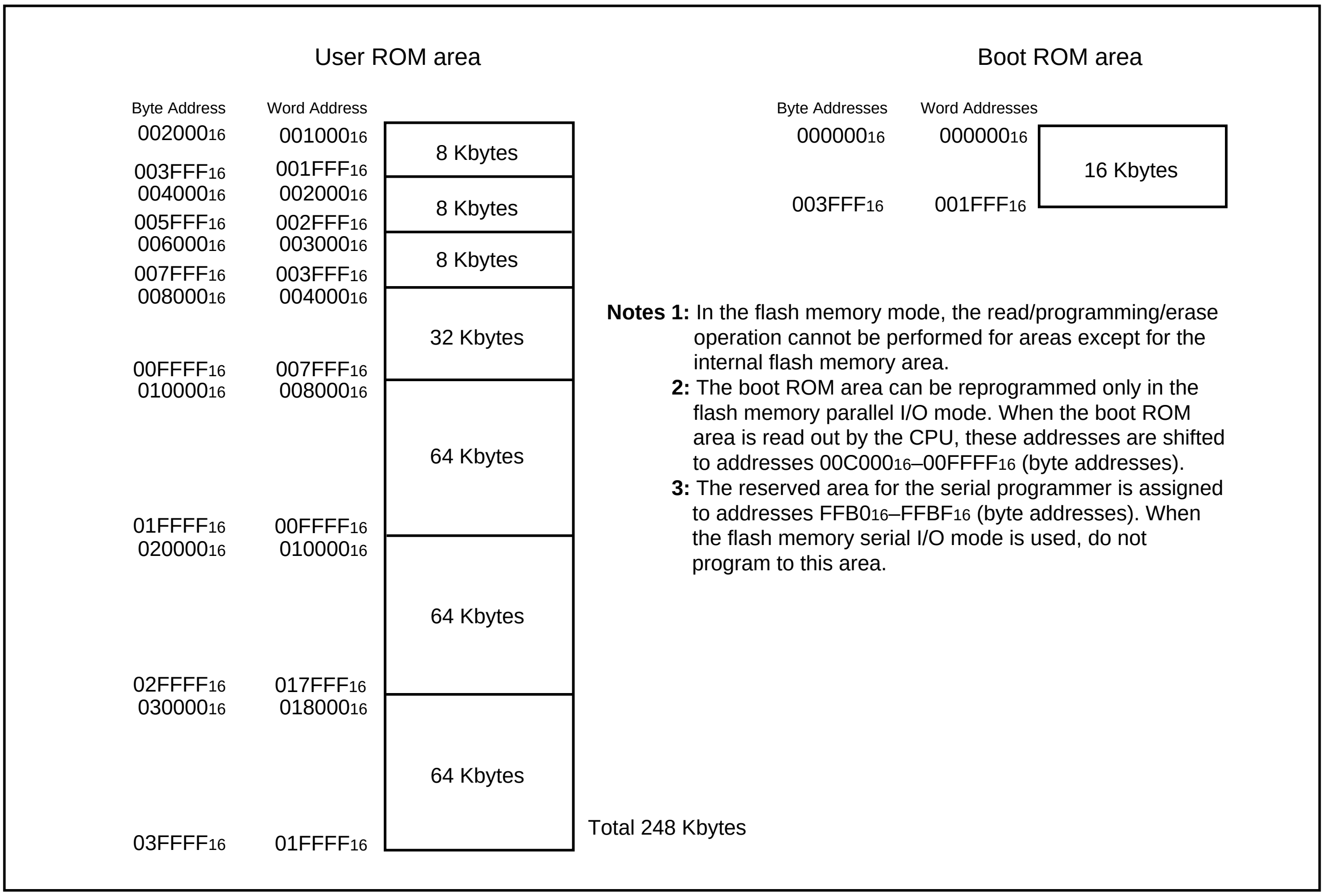


Notes 1: In the flash memory mode, the read/programming/erase operation cannot be performed for areas except for the internal flash memory area.

2: The boot ROM area can be reprogrammed only in the flash memory parallel I/O mode. When the boot ROM area is read out by the CPU, these addresses are shifted to addresses 00C000$_{16}$–00FFFF$_{16}$ (byte addresses).

3: The reserved area for the serial programmer is assigned to addresses FFB0$_{16}$–FFBF$_{16}$ (byte addresses). When the flash memory serial I/O mode is used, do not program to this area.

Fig 107. M37902FCCHP: block configuration of internal flash memory



Notes 1: In the flash memory mode, the read/programming/erase operation cannot be performed for areas except for the internal flash memory area.

2: The boot ROM area can be reprogrammed only in the flash memory parallel I/O mode. When the boot ROM area is read out by the CPU, these addresses are shifted to addresses 00C000$_{16}$–00FFFF$_{16}$ (byte addresses).

3: The reserved area for the serial programmer is assigned to addresses FFB0$_{16}$–FFBF$_{16}$ (byte addresses). When the flash memory serial I/O mode is used, do not program to this area.

Fig 108. M37902FGCHP: block configuration of internal flash memory

Flash Memory Parallel I/O Mode

The flash memory parallel I/O mode is used to manipulate the internal flash memory with a parallel programmer. This parallel programmer uses the software commands listed in Table 19 to do the flash memory manipulations, such as read/programming/erase operations.

In the flash memory parallel I/O mode, each block can be protected from erasing/programming (in other words, block lock).

Table 19. Software commands (flash memory parallel I/O mode)

Software Command
Read Array
Read Status Register
Clear Status Register
Page Programming **(Note)**
Block Erase
Erase All Unclocked Block
Lock Bit Programming
Read Lock Bit Status

Note: Programming is performed in a unit of 256 bytes, with the low-order address assigned in the range of 00_{16}—FF_{16} (byte addresses).

User ROM Area and Boot ROM Area

The user ROM area and boot ROM area can be reprogrammed in the flash memory parallel I/O mode.

The programming and block erase operations can be performed only to these areas.

The boot ROM area, 16 Kbytes in size, is assigned to addresses 0000_{16}–$3FFF_{16}$ (byte addresses), so that programming and block erase operations can be performed only to this area. (Access to any address out of this area is prohibited).

The erasable block in the boot ROM area is only one block, consisting of 16 Kbytes. The reprogramming control firmware to be used in the flash memory serial I/O mode has been stored to this boot ROM area on our shipment. Therefore, do not reprogram the boot ROM area if the user uses the flash memory serial I/O mode.

Addresses $FFB0_{16}$ to $FFBF_{16}$ are the reserved area for the serial programmer. Therefore, when the user uses the flash memory serial I/O mode, do not program to this area.

Note that, when the boot ROM area is read out from the CPU in the CPU reprogramming mode, described later, its addresses will be shifted to $C000_{16}$—$FFFF_{16}$ (byte addresses).

Flash Memory Serial I/O Mode

In the flash memory serial I/O mode, addresses, data, and software commands, which are required to read/program/erase the internal flash memory, are serially input and output with a fewer pins and the dedicated serial programmer.

In this mode, being different from the flash memory parallel I/O mode, the CPU controls reprogramming of the flash memory (using the CPU reprogramming mode), serial input of the reprogramming data, etc.

The reprogramming control firmware for the flash memory serial I/O mode has been stored in the boot ROM area on shipment of the product from us. Note that, then, the flash memory serial I/O mode will become unavailable if the boot ROM area has been reprogrammed in the flash memory parallel I/O mode.

Note that, also, this reprogramming control firmware for the flash memory serial I/O mode is subject to change.

Figure 112 shows the pin connection in the flash memory serial I/O mode.

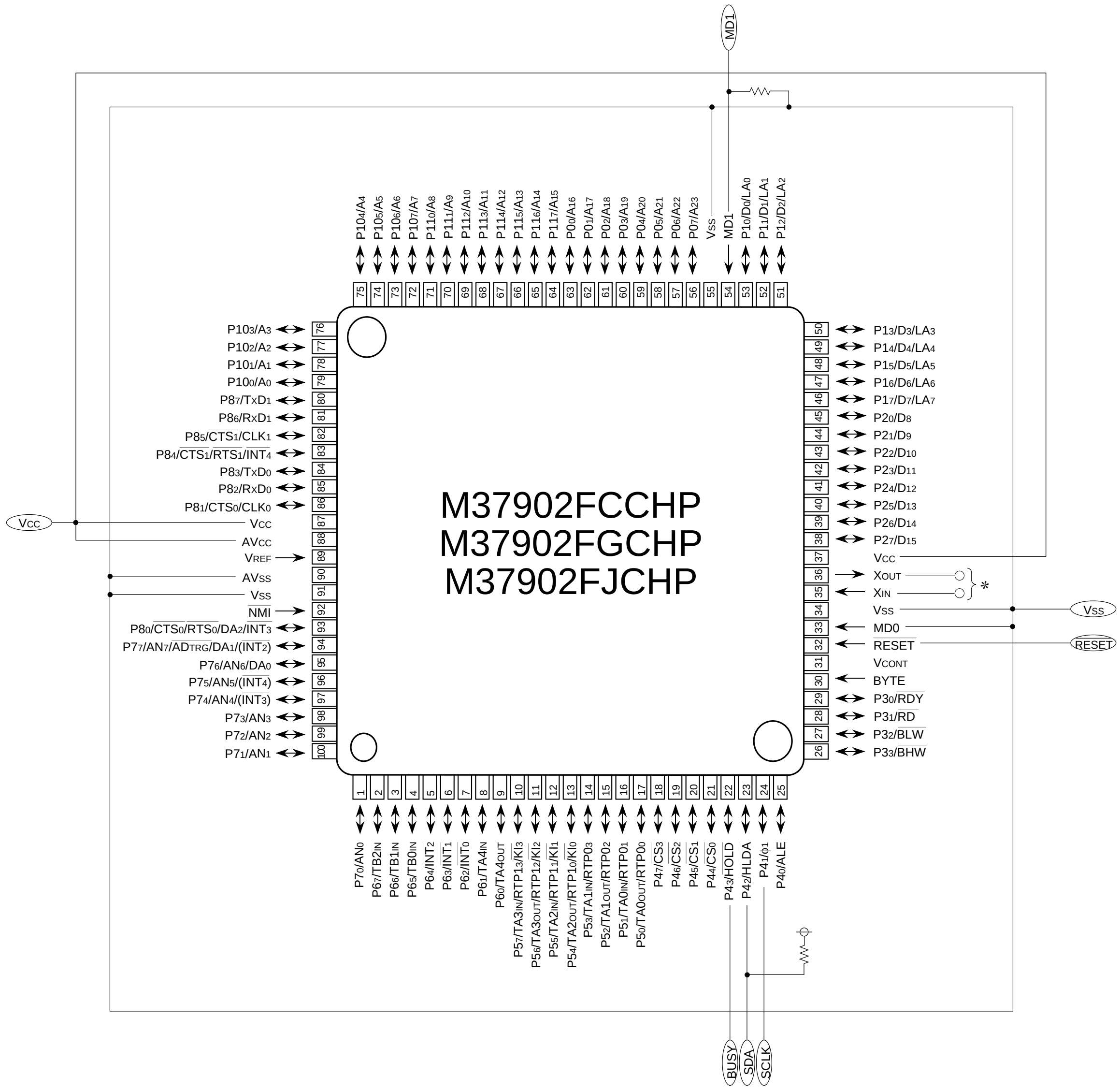
The three pins, SCLK, SDA, and BUSY, are used to input and output serial data.

The SCLK pin is the input pin of external transfer clocks. The SDA pin is the I/O pin of transmit and receive data, and its output acts as the N-channel open-drain output. To the SDA pin, connect an external pullup resistor (about 1 kΩ). The BUSY pin is the output pin of the BUSY flag (CMOS output) and goes “H” during BUSY periods owing to a certain operation, such as transmit, receive, erase, programming, etc.

Transmit and receive data are serially transferred 8 bits at a time.

In the flash memory serial I/O mode, only the user ROM area can be reprogrammed; the boot ROM area is not accessible.

Addresses FFB0$_{16}$ to FFBF$_{16}$ are the reserved area for the serial programmer. Therefore, when the user uses the flash memory serial I/O mode, do not program to this area.



✻: Connect to the ceramic oscillation circuit.

Output 100P6Q-A

Fig.112 Pin connection of M37902FxCHP in flash memory serial I/O mode

CPU Reprogramming Mode

The CPU reprogramming mode is used to perform the operations for the internal flash memory (reading, programming, erasing) under control of the CPU.

In this mode, only the user ROM area can be reprogrammed; the boot ROM area cannot be reprogrammed.

The user-original reprogramming control software for the CPU reprogramming mode can be stored in either the user ROM area or the boot ROM area. Because the CPU cannot read out the flash memory in the CPU reprogramming mode, the above software must be transferred to the internal RAM in advance to be executed.

Boot Mode

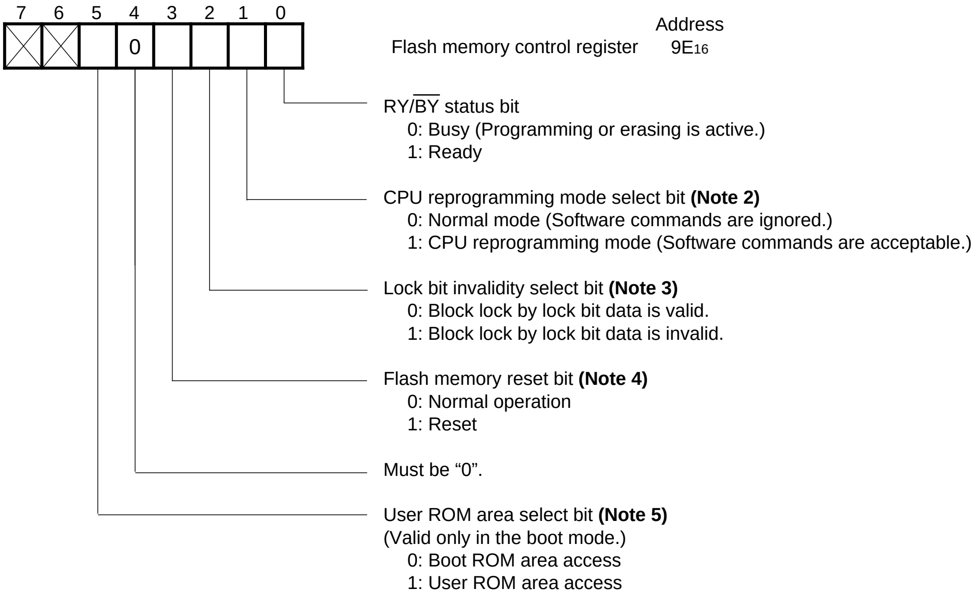
The user-original reprogramming control software for the CPU reprogramming mode must be stored into the user ROM area or the boot ROM area in the flash memory parallel I/O mode in advance. (If this program has been stored into the boot ROM area, the flash memory serial I/O mode will become unavailable).

Note that addresses of the boot ROM area depend on the accessing ways to the boot ROM area, When accessing in the flash memory parallel I/O mode, these addresses will be shifted to 0000$_{16}$ to 3FFF$_{16}$ (byte address). On the other hand, when accessing with the CPU, these addresses will be shifted to C000$_{16}$ to FFFF$_{16}$ (byte address).

Reset removal with both of the MD0 and MD1 pins held "L" invokes the normal microcomputer mode, and the CPU operates using the control software stored in the user ROM area. In this case, the boot ROM area is not accessible.

Removing reset with the MD0 pin held "L" and the MD1 pin "H", the CPU starts its operation using the reprogramming control software stored in the boot ROM area. This mode is called the boot mode. The reprogramming control software in the boot ROM area can also reprogram the user ROM area.

After reset removal, be sure not to change the status at pins MD0 and MD1.



Flash memory control register   Address 9E$_{16}$

RY/$\overline{\text{BY}}$ status bit
0: Busy (Programming or erasing is active.)
1: Ready

CPU reprogramming mode select bit **(Note 2)**
0: Normal mode (Software commands are ignored.)
1: CPU reprogramming mode (Software commands are acceptable.)

Lock bit invalidity select bit **(Note 3)**
0: Block lock by lock bit data is valid.
1: Block lock by lock bit data is invalid.

Flash memory reset bit **(Note 4)**
0: Normal operation
1: Reset

Must be "0".

User ROM area select bit **(Note 5)**
(Valid only in the boot mode.)
0: Boot ROM area access
1: User ROM area access

Notes 1: The contents of the flash memory control register after reset is removed are "XX000001$_2$".
2: To set "1", writing of "0" to bit 1 and subsequent writing of "1" to bit 1 are necessary. Writing to bit 1 must be performed by the user-original reprogramming control software in the internal RAM.
3: To set "1", writing of "0" to bit 2 and subsequent writing of "1" to bit 2 are necessary while bit 1 = "1".
4: Valid only when bit 1 = "1". Set bit 3 to "1" (reset), and then clear to "0". This bit 3 must be controlled with bit 1 = "1".
5: Writing to bit 5 must be performed by the user-original reprogramming control software in the internal RAM.

Fig. 114 Bit configuration of flash memory control register

Function overview (CPU reprogramming mode)

The CPU reprogramming mode is available in the single-chip mode, memory expansion mode, and boot mode to reprogram the user ROM area only.

In the CPU reprogramming mode, the CPU erases, programs, and reads the internal flash memory by writing software commands. Note that the user-original reprogramming control software must be transferred to the internal RAM in advance to be executed.

The CPU reprogramming mode becomes active when “1” is written into the flash memory control register’s bit 1 (the CPU reprogramming mode select bit) shown in Figure 114, and software commands become acceptable.

In the CPU reprogramming mode, software commands and data are all written to and read from even addresses (Note that address A_0 in byte addresses = “0”.) 16 bits at a time. Therefore, a software command consisting of 8 bits must be written to an even address; therefore, any command written to an odd address will be invalid. Since the write data at the 2nd cycle of a programming command consists of 16 bits, this data must be written to even and odd addresses.

The write state machine (WSM) in the flash memory controls the erase and programming operations. What the status of the WSM operation is and whether the programming or erase operation has been completed normally or terminated by an error can be examined by reading the status register.

Figure 114 shows the bit configuration of the flash memory control register.

Bit 0 (the RY/$\overline{\text{BY}}$ status bit) is a read-only bit for indicating the WSM operation. This bit goes to “0” (BUSY) while the automatic programming/erase operation is active and goes to “1” (READY) during the other operations.

Bit 1 serves as the CPU reprogramming mode select bit. Writing of “1” to this bit selects the CPU reprogramming mode, and software commands will be acceptable. Because the CPU cannot directly access the internal flash memory in the CPU reprogramming mode, writing to this bit 1 must be performed by the user-original reprogramming control software which has been transferred to the internal RAM in advance. To set bit 1 to “1”, it is necessary to write “0” and “1” to this bit 1 successively. On the other hand, to clear this bit to “0”, it is sufficient only to write “0”.

Bit 2 serves as the lock bit invalidity select bit, and setting this bit to “1” invalidates the protection by a lock bit against erasing and programming (block lock). The lock bit invalidity select bit can invalidates the lock bit function but set no lock bit itself. However, if erasing is performed with this bit = “1”, a lock bit with value “0” (the locked state) will be set to “1” (the unlocked state) after the erasing has been completed. To set the lock bit invalidity select bit to “1”, write “0” and “1” to this bit 2 successively with the CPU reprogramming mode select bit = “1”. The manipulation of bit 2 is allowed only when the CPU reprogramming mode select bit = “1”.

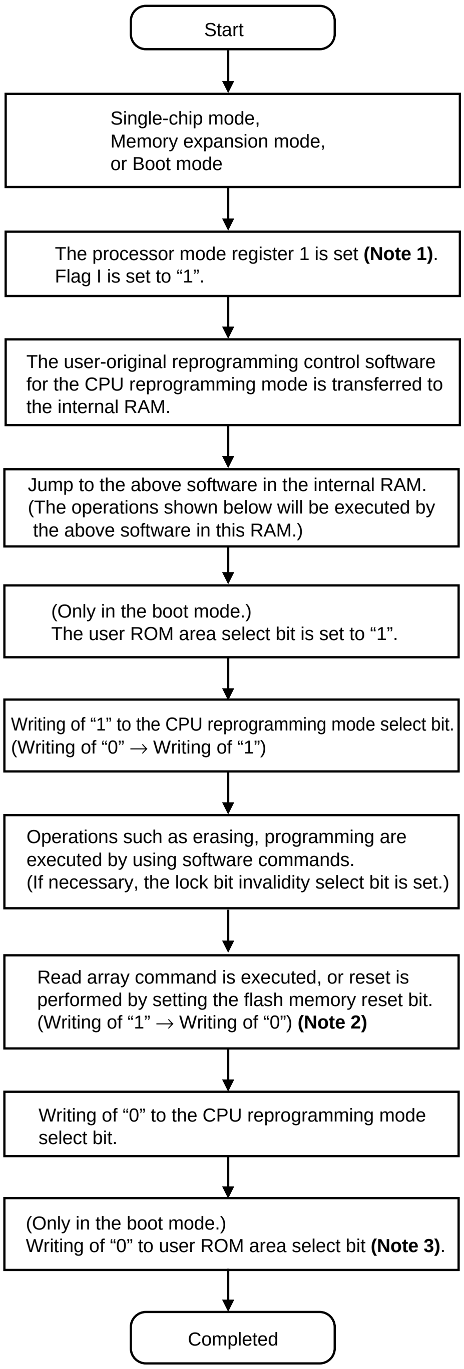
Bit 3 (the flash memory reset bit) resets the control circuit of the internal flash memory and is used when the CPU reprogramming mode is terminated or when an abnormal access to the flash memory happens. Writing of “1” to bit 3 with the CPU reprogramming mode select bit = “1” preforms the reset operation. To remove the reset, write “0” to bit 3 subsequently.

Bit 5 serves as the user ROM area select bit and is valid only in the boot mode. Setting this bit to “1” in the boot mode switches an accessible area from the boot ROM area to the user ROM area. To use the CPU reprogramming mode in the boot mode, set this bit to “1”. Note that when the microcomputer is booted up in the user ROM area, only the user ROM area is accessible and bit 5 is invalid; on the other hand, when the microcomputer is in the boot mode, bit 5 is valid independent of the CPU reprogramming mode. To rewrite bit 5, execute the user-original reprogramming control software transferred to the internal RAM in advance.

Figure 115 shows the CPU reprogramming mode set/termination flowchart, and be sure to follow this flowchart. As shown in Note 1 of Figure 115, before selecting the CPU reprogramming mode, set the processor mode register 1’s bit 7 (the internal ROM bus cycle select bit) to “0” and set flag I to “1” to avoid an interrupt request input.

When an $\overline{\text{NMI}}$ interrupt or a watchdog timer interrupt request is generated in the CPU reprogramming mode, when an input to the $\overline{\text{RESET}}$ pin is “L”, or when the software reset is performed, the flash memory control circuit and flash memory control register will be reset.

When the flash memory is reset during the erase or programming operation, this operation is cancelled and the target block’s data will be invalid. Just before writing a software command related to the erase/programming operation, be sure to write to the watchdog timer. Also, be sure to set the $\overline{\text{NMI}}$ pin to “H” to avoid an $\overline{\text{NMI}}$ interrupt request occurrence. In the CPU reprogramming mode, be sure not to use the **STP** and **WIT** instructions.


Start
Single-chip mode, Memory expansion mode, or Boot mode
The processor mode register 1 is set **(Note 1)**. Flag I is set to “1”.
The user-original reprogramming control software for the CPU reprogramming mode is transferred to the internal RAM.
Jump to the above software in the internal RAM. (The operations shown below will be executed by the above software in this RAM.)
(Only in the boot mode.) The user ROM area select bit is set to “1”.
Writing of “1” to the CPU reprogramming mode select bit. (Writing of “0” → Writing of “1”)
Operations such as erasing, programming are executed by using software commands. (If necessary, the lock bit invalidity select bit is set.)
Read array command is executed, or reset is performed by setting the flash memory reset bit. (Writing of “1” → Writing of “0”) **(Note 2)**
Writing of “0” to the CPU reprogramming mode select bit.
(Only in the boot mode.) Writing of “0” to user ROM area select bit **(Note 3)**.
Completed

Notes 1: The processor mode register 1’s bit 7 (address 5F$_{16}$, the internal ROM bus cycle select bit) must be “0” (bus cycle = 3ϕ).

2: To terminate the CPU reprogramming mode after the erase and programming operations have been completed, be sure to execute the read array command or perform the flash memory reset operation.

3: This bit may remain “1”. However, if this bit is “1”, the user ROM area access is specified.

Fig. 115 CPU reprogramming mode set/termination flowchart

Software Commands

Table 20 lists the software commands.

By writing a software command after the CPU reprogramming select bit has been set to “1”, erasing, programming, etc. can be specified. Note that, at software commands’ input, the high-order byte (D_8–D_{15}) is ignored. (Except for the write data at the 2nd cycle of a page programming command.)

Software commands are explained as below.

Read Array Command (FF$_{16}$)

By writing command code “FF$_{16}$” at the 1st bus cycle, the microcomputer enters the read array mode. If an address to be read is input in the next or the following bus cycles, the contents at the specified address are output to the data bus (D_0 to D_{15}) in a unit of 16 bits.

The read array mode is maintained until writing of another software command.

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

Writing command code “70$_{16}$” at the 1st bus cycle outputs the contents of the status register to the data bus (D_0-D_7) by a read at the 2nd bus cycle.

The status register is explained later.

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

This command clears three status bits (SR.3–5) each of which is set to “1” to indicate that the operation has been terminated by an error. To clear these bits, write command code “50$_{16}$” at the 1st bus cycle.

Page Programming Command (41$_{16}$)

Page programming facilitates quick programming of 128 words (a page = 256 bytes) at a time. To initiate page programming, write command code “41$_{16}$” at the 1st bus cycle; then, program a series of data, in a unit of 16 bits, sequentially from the 2nd to the 129th bus cycle. It is necessary, at this time, to increment address A_0–A_7 from "00$_{16}$" to “FE$_{16}$” by +2. (Programmed to even addresses.)

Upon completion of data loading, automatic programming (data programming and verification) operation is started.

The completion of the automatic programming operation is recognized by a read of the status register or a read of the flash memory control register. As the automatic programming operation starts, the microcomputer enters the read status register mode automatically to allow reading out the contents of the status register. Bit 7 of the status register (SR.7) is cleared to “0” simultaneously with the start of the automatic programming operation; and also, bit 7 returns to “1” by the end of it. Until writing of the read array command (FF$_{16}$), writing of the read lock bit status command (71$_{16}$), or performing the reset operation with the flash memory reset bit, this read status register mode is maintained. In continuous programming, if there is no programming error, page programming commands can be executed with the read status register mode kept.

Table 20. Software commands (CPU reprogramming mode)

Command	1st cycle			2nd cycle			3rd cycle		
	Mode	Address	Data (D$_0$ to D$_7$)	Mode	Address	Data	Mode	Address	Data
Read Array	Write	X **(Note 2)**	FF$_{16}$	—	—	—	—	—	—
Read Status Register	Write	X	70$_{16}$	Read	X	SRD **(Note 3)**	—	—	—
Clear Status Register	Write	X	50$_{16}$	—	—	—	—	—	—
Page Programming **(Note 3)**	Write	X	41$_{16}$	Write	WA0 **(Note 4)**	WD0 **(Note 4)**	Write	WA1	WD1
Block Erase	Write	X	20$_{16}$	Write	BA **(Note 5)**	D0$_{16}$	—	—	—
Erase All Unclocked Block	Write	X	A7$_{16}$	Write	X	D0$_{16}$	—	—	—
Lock Bit Programming	Write	X	77$_{16}$	Write	BA	D0$_{16}$	—	—	—
Read Lock Bit Status	Write	X	71$_{16}$	Read	BA	D$_6$ **(Note 6)**	—	—	—

Notes 1: At software commands' input, the high-order byte of data (D$_8$–D$_{15}$) is ignored.
2: X = An arbitrary address in the user ROM area. (Note that A$_0$ = "0".)
3: SRD = Status register data.
4: WA = Write address, WD = Write data (16 bits).
WA and WD must be set from "00$_{16}$" to "FE$_{16}$". (Byte addresses. Incremented by +2. Address A$_0$ = "0".) Page size = 128 words (128 × 16 bits).
5: Block address: the maximum address of each block must be input. Note that address A$_0$ = "0".
6: D$_6$ indicates the block lock status.
"1" = unlocked. "0" = locked.

The RY/$\overline{\text{BY}}$ status bit of the flash memory control register goes "0" during the automatic programming operation; and also, it goes "1" after the end of it, the same way as bit 7 of the status register.

Before execution of the next command, be sure to verify that bit 7 of the status register (SR.7) or the RY/$\overline{\text{BY}}$ status bit is set to "1" (READY). During the automatic programming operation, writing of commands and access to the flash memory must not be performed.

Reading out the status register after the automatic programming operation is completed reports the result of it. For details, refer to the section on the status register.

Figure 116 shows an example of the page programming flowchart.

Note that each block can be protected from programming by using a lock bit. For details, refer to the section on the data protect function.

Additional programming to any page that has already been programmed is prohibited.

Block Erase Command (20$_{16}$/D0$_{16}$)

Writing command code "20$_{16}$" at the 1st bus cycle and writing verify command code "D0$_{16}$" and the maximum address of the block (Note that address A$_0$ = "0".) at the subsequent 2nd bus cycle initiate the automatic erase (erasing and erase verification) operation for the specified block.

The completion of the automatic erase operation is verified by a read of the status register or a read of the flash memory control register. As the automatic erase operation starts, the microcomputer enters the read status register mode automatically to allow reading out the contents of the status register. Bit 7 of the status register (SR.7) is cleared to "0" simultaneously with the start of the automatic erase operation; and also, it returns to "1" by the end of it. The read status register mode is maintained until writing of the read array command (FF$_{16}$), writing of the read lock bit status command (71$_{16}$), or performing the reset operation with the flash memory reset bit.

The RY/$\overline{\text{BY}}$ status bit of the flash memory control register goes "0" during the automatic erase operation; and also, it goes "1" after the end of it, the same way as bit 7 of the status register.

Before execution of the next command, be sure to verify that bit 7 of the status register (SR.7) or the RY/$\overline{\text{BY}}$ status bit is set to "1" (READY). During the automatic erase operation, writing of commands and access to the flash memory must not be performed.

Reading out the status register after the automatic erase operation is completed reports the result of it. For details, refer to the section on the status register.

Figure 117 shows an example of the block erase flowchart.

Note that each block can be protected from erasing by using a lock bit. For details, refer to the section on the data protect function.

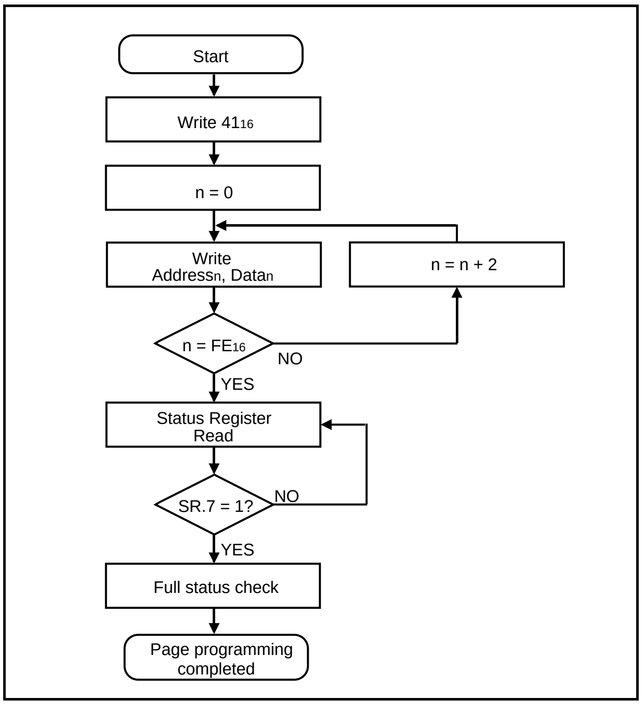


Fig. 116 Page programming flowchart

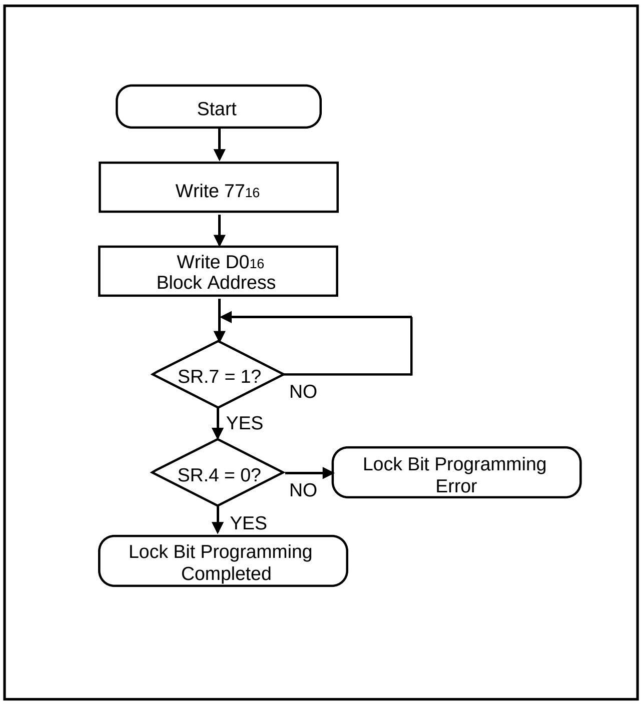


Fig. 118 Lock bit programming flowchart

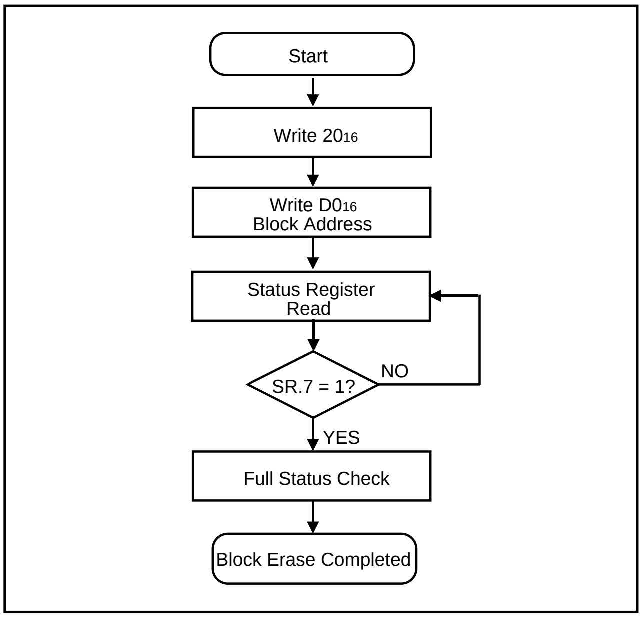


Fig. 117 Block erase flowchart

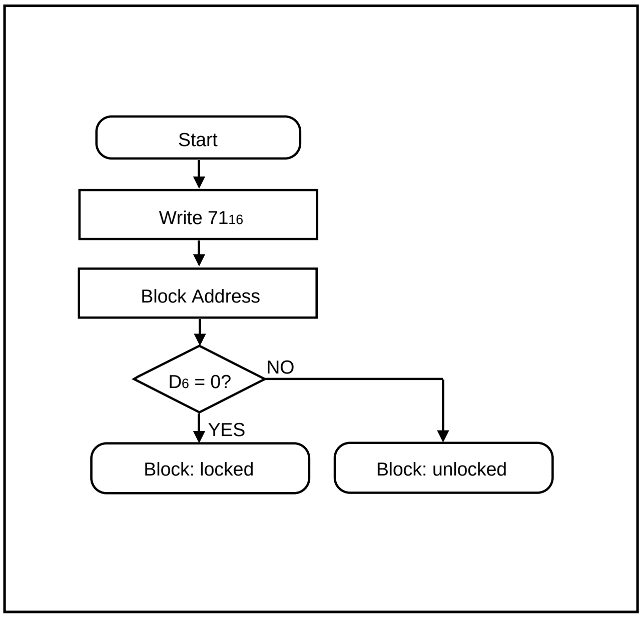


Fig. 119 Read lock bit status flowchart

Erase All Unlocked Block Command (A7$_{16}$/D0$_{16}$)

Writing command code "A7$_{16}$" at the 1st bus cycle and writing verify command code "D0$_{16}$" at the subsequent 2nd bus cycle initiate the continuous block erase (chip erase) operations for all the blocks.
The completion of the chip erase operation, as well as of the block erase operation, is verified by a read of the status register or a read of the flash memory control register. The result of the automatic erase operation is also reported by a read of the status register.
During the automatic erase operation (when the RY/$\overline{\text{BY}}$ status bit = "0"), writing of commands and access to the flash memory must not be performed.
When the lock bit invalidity select bit = "1", all the blocks are erased regardless of the status of their lock bits. When the lock bit invalidity select bit = "0", on the contrary, the status of each lock bit becomes valid, so only the blocks in the unlocked state (lock bit = "1") are erased.

Lock Bit Programming Command (77$_{16}$/D0$_{16}$)

By writing of command code "77$_{16}$" at the 1st bus cycle and writing of verify command code "D0$_{16}$" and the block's maximum address (Note that address A_0 = "0".) at the subsequent 2nd bus cycle, "0" (the locked state) is written into the lock bit of the specified block.
Figure 118 shows an example of the lock bit programming flowchart.
The status of the lock bit can be read out by the read lock bit status command.
The completion of the lock bit programming operation, as well as of the page programming operation, is verified by a read of the status register or a read of the flash memory control register.
For details of the lock bit's function and the method of reset, refer to the section on the data protect function.

Read Lock Bit Status Command (71$_{16}$)

By writing of command code "71$_{16}$" at the 1st bus cycle and writing of the block's maximum address (Note that address A_0 = "0".) at the subsequent 2nd bus cycle, the status of the lock bit of the specified block is output to the data bus (D_6).
Figure 119 shows an example of the read lock bit programming flowchart.

Data Protect Function (Block Lock)

Each block is implemented with a nonvolatile lock bit to protect the block from erasing/programming (block lock). A "0" (the locked state) can be written to a lock bit using the lock bit programming command, and the lock bit of each block can be read out by using the read lock bit status command.
Whether a block lock is valid or invalid is determined by the status of the lock bit and the lock bit invalidity select bit of the flash memory control register.

(1) When the lock bit invalidity select bit = "0", a lock bit determines whether to lock or unlock the corresponding block. A block with its lock bit = "0" is locked and inhibited from erasing and programming. On the other hand, a block with its lock bit = "1" remains unlocked and allows to be erased/programmed.
(2) When the lock bit invalidity select bit = "1", all the blocks are unlocked and allows to be erased/programmed regardless of the values of their lock bits. In this case, a lock bit with a value "0" (the locked state) is set to "1" (the unlocked state) after completion of the erase operation, and the locked state by the lock bit is terminated.

To perform erase or programming, be sure to do one of the following.
- By executing the read lock bit status command, verify that the lock of the target block is invalid.
- Set the lock bit invalidity select bit to "1" to invalidate the lock.

When the block erase or programming is performed with the lock valid, the erase status bit (SR.5) and programming status bit (SR.4) are set to "1" (terminated by error).

Status Register

The status register is used to indicate what the status of the write state machine (WSM) operation is and whether the programming/erase operation has been completed normally or terminated by an error. By writing the read status register command (70$_{16}$), the contents of the status register can be read out; by writing the clear status register command (50$_{16}$), the contents of the status register can be cleared.
Table 21 lists the definition of each bit of the status register.
The status register outputs "80$_{16}$" after reset is removed.
The status of each bit is described below.

Write State Machine (WSM) Status Bit (SR.7)

This bit reports the operation status of the WSM. This bit is set to “1” (READY) after the system power is turned on or after reset is removed.

During the automatic programming or erase operation, this bit is cleared to “0” (BUSY), however, set to “1” upon completion of them.

Erase Status Bit (SR.5)

This bit reports the status of the automatic erase operation. This bit is set to “1” if an erase error occurs and returns to “0” if one of the following conditions is satisfied:

- the system power is turned on.
- reset is removed.
- the clear status register command (50_{16}) is executed.

Programming Status Bit (SR.4)

This bit reports the status of the automatic programming operation. This bit is set to “1” if a programming error occurs and returns to “0” if one of the following conditions is satisfied:

- the system power is turned on.
- reset is removed.
- the clear status register command (50_{16}) is executed.

Block Status After Programming Bit (SR.3)

This bit is set to “1”, upon completion of the page programming operation, if the excessive programming **(Note)** occurs. That is, the status register becomes “80_{16}” when the programming operation is terminated normally, “90_{16}” when the programming operation is failed, and “88_{16}” when the excessive programming occurs.

Under the condition that any of SR.5, SR.4 and SR.3 = “1”, none of the page programming, block erase, erase all unlocked block, and lock bit programming commands can be accepted. To execute these commands, in advance, execute the clear status register command (50_{16}) to clear the status register.

Both of SR.4 and SR.5 are set to “1” under the following conditions (Command Sequence Error):

(1) when data other than “$D0_{16}$” and “FF_{16}” is written to the data in the 2nd bus cycle of the lock bit programming command (77_{16}/$D0_{16}$)

(2) when data other than “$D0_{16}$” and “FF_{16}” is written to the data in the 2nd bus cycle of the block erase command (20_{16}/$D0_{16}$)

(3) when data other than “$D0_{16}$” and “FF_{16}” is written to the data in the 2nd bus cycle of the erase all unlocked block command ($A7_{16}$/$D0_{16}$)

Note that, writing of “FF_{16}” forces the microcomputer into the read array mode. Simultaneously with this, the command written in the 1st bus cycle will be canceled.

Note: The excessive programming means the status that memory cells are too depleted, so data cannot be read out correctly.

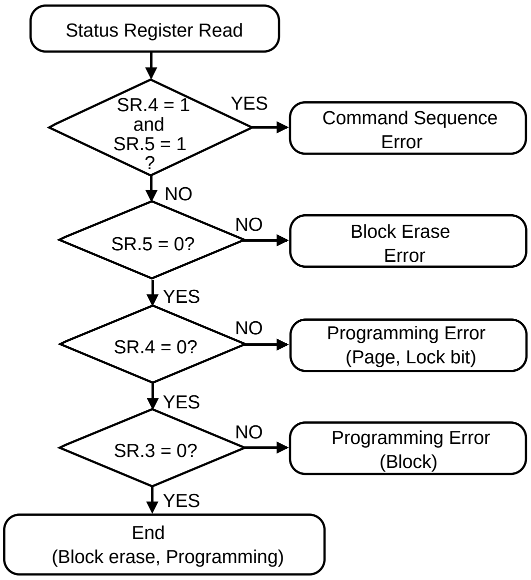
Full Status Check

The full status check reports the results of the erase or programming operation.

Figure 120 shows the full status check flowchart and actions to be taken if an error has occurred.

Table 21. Bit definition of status register

Symbol	Status	Definition	
		“1”	“0”
SR.7 (D_7)	Write State Machine (WSM) Status	Ready	Busy
SR.6 (D_6)	Reserved	—	—
SR.5 (D_5)	Erase Status	Terminated by error.	Terminated normally.
SR.4 (D_4)	Programming Status	Terminated by error.	Terminated normally.
SR.3 (D_3)	Block Status After Programming	Terminated by error.	Terminated normally.
SR.2 (D_2)	Reserved	—	—
SR.1 (D_1)	Reserved	—	—
SR.0 (D_0)	Reserved	—	—



Command Sequence Error · · · Execute the clear status register command (50_{16}) to clear the status register. After verifying the command to be correctly input, start the operation again.

Block Erase Error · · · Examine whether a lock is active or not by executing the read lock bit status command (71_{16}). After removing the lock, perform block erase again. If the same error still occurs, this page cannot be used.

Programming Error (Page, Lock bit) · · · Examine whether a lock is active or not by executing the read lock bit status command (71_{16}). After removing the lock, perform programming again. If the same error still occurs, this page cannot be used.

Programming Error (Block) · · · After erasing the block where an error has occured, perform programming again. If the same error still occurs, this block cannot be used.

Note: Under the condition that any of SR.5, SR.4 and SR.3 = “1”, none of the page programming, block erase, erase all unlocked block, and lock bit programming commands can be accepted. To execute these commands, in advance, execute the clear status register command (50_{16}).

Fig. 120 Full status check flowchart and actions to be taken if an error has ocurred

DC Electrical Characteristics (V_{CC} = 5 V ± 0.5 V, Ta = 0 to 60 °C, f(f_{sys}) = 26 MHz (Note))

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
I_{CC1}	V_{CC} power source current (at read)		30	48	mA
I_{CC2}	V_{CC} power source current (at write)			48	mA
I_{CC3}	V_{CC} power source current (at programming)			54	mA
I_{CC4}	V_{CC} power source current (at erasing)			54	mA

Limits of V_{IH}, V_{IL}, V_{OH}, V_{OL}, I_{IH}, and I_{IL} for each pin are the same as those in the microcomputer mode.
Note: f(f_{sys}) indicates the system clcok (fsys) frequency.

AC Electrical Characteristics (V_{CC} = 5 V ± 0.5 V, Ta = 0 to 60 °C, f(f_{sys}) = 26 MHz (Note))

Parameter	Limits			Unit
	Min.	Typ.	Max.	
Page programming time		8	120	ms
Block erase time		50	600	ms
Erase all unlocked block time		50 × n	600 × n	ms
Lock bit programming time		8	120	ms

n = Number of blocks to be erased

The limits of parameters other than the above are same as those in the microcomputer mode.

Note: f(f_{sys}) indicates the system clock (fsys) frequency.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Ratings	Unit
V_{CC}	Power source voltage	–0.3 to 6.5	V
AV_{CC}	Analog power source voltage	–0.3 to 6.5	V
V_I	Input voltage P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$, V_{REF}, X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1, $\overline{NMI}$, V_{CONT}	–0.3 to V_{CC}+0.3	V
V_O	Output voltage P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$, X_{OUT}	–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	400	mW
T_{opr}	Operating ambient temperature	–20 to 85	°C
T_{stg}	Storage temerature	–40 to 150	°C

RECOMMENDED OPERATING CONDITIONS (Vcc = 5 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Power source voltage	4.5	5.0	5.5	V
AV_{CC}	Analog power source voltage		V_{CC}		V
V_{SS}	Power source voltage		0		V
AV_{SS}	Analog power source voltage		0		V
V_{IH}	High-level input voltage X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1	0.8 Vcc		Vcc	V
V_{IH}	High-level input voltage P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$	0.7 V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage P0$_0$–P0$_7$ (When the port P0 input level select bit = “0”)	0.7 Vcc		Vcc	V
V_{IH}	High-level input voltage P0$_0$–P0$_7$ (When the port P0 input level select bit = “1”)	0.43 Vcc		Vcc	V
V_{IH}	High-level input voltage D$_0$–D$_7$, D$_8$–D$_{15}$	0.43 Vcc		Vcc	V
V_{IH}	High-level input voltage $\overline{RDY}$, $\overline{HOLD}$, TA0$_{IN}$–TA4$_{IN}$, TA0$_{OUT}$–TA4$_{OUT}$, TB0$_{IN}$–TB2$_{IN}$, $\overline{KI_0}$–$\overline{KI_3}$, $\overline{INT_0}$–$\overline{INT_4}$, $\overline{NMI}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK$_0$, CLK$_1$, RxD$_0$, RxD$_1$	0.43 Vcc		Vcc	V
V_{IH}	High-level input voltage SCLK, SDA **(Note 1)**	0.43 Vcc		Vcc	V
V_{IH}	Low-level input voltage X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage P0$_0$–P0$_7$ (When the port P0 input level select bit = “0”)	0		0.2 V_{CC}	V
V_{IL}	Low-level input voltage P0$_0$–P0$_7$ (When the port P0 input level select bit = “1”)	0		0.16 V_{CC}	V
V_{IL}	Low-level input voltage D$_0$–D$_7$, D$_8$–D$_{15}$	0		0.16 V_{CC}	V
V_{IL}	Low-level input voltage $\overline{RDY}$, $\overline{HOLD}$, TA0$_{IN}$–TA4$_{IN}$, TA0$_{OUT}$–TA4$_{OUT}$, TB0$_{IN}$–TB2$_{IN}$, $\overline{KI_0}$–$\overline{KI_3}$, $\overline{INT_0}$–$\overline{INT_4}$, $\overline{NMI}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK$_0$, CLK$_1$, RxD$_0$, RxD$_1$	0		0.16 V_{CC}	V
V_{IL}	Low-level input voltage SCLK, SDA **(Note 1)**	0		0.16 V_{CC}	V
$I_{OH(peak)}$	High-level peak output current P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$			–10	mA
$I_{OH(avg)}$	High-level average output current P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$			–5	mA
$I_{OL(peak)}$	Low-level peak output current P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$			10	mA
$I_{OL(avg)}$	Low-level average output current P0$_0$–P0$_7$, P1$_0$–P1$_7$, P2$_0$–P2$_7$, P3$_0$–P3$_3$, P4$_0$–P4$_7$, P5$_0$–P5$_7$, P6$_0$–P6$_7$, P7$_0$–P7$_7$, P8$_0$–P8$_7$, P10$_0$–P10$_7$, P11$_0$–P11$_7$			5	mA
f(X_{IN})	External clock input frequency **(Note 2)**			26	MHz
f(f_{sys})	System clock frequency			26	MHz

Notes 1: Pins SCLK and SDA are used only in the flash memory serial I/O mode.

2: When using the PLL frequency multiplier, be sure that f(f_{sys}) = 26 MHz or less.

3: Average output current is the average value of an interval of 100 ms.

4: The sum of $I_{OL(peak)}$ for ports P0–P2, P8, P10, and P11 must be 80 mA or less, the sum of $I_{OH(peak)}$ for ports P0–P2, P8, P10, and P11 must be 80 mA or less, the sum of $I_{OL(peak)}$ for ports P3–P7 must be 80 mA or less, the sum of $I_{OH(peak)}$ for ports P3–P7 must be 80 mA or less.

DC ELECTRICAL CHARACTERISTICS (Vcc = 5 V, Vss = 0 V, Ta = –20 to 85 °C, f(f_{sys}) = 26 MHz **(Note)**)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P10_0$–$P10_7$, $P11_0$–$P11_7$	I_{OH} = –10 mA		3			V
V_{OH}	High-level output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$, $P4_2$, $P4_4$–$P4_7$, $P10_0$–$P10_7$, $P11_0$–$P11_7$	I_{OH} = –400 μA		4.7			V
V_{OH}	High-level output voltage $P3_1$–$P3_3$	I_{OH} = –10 mA		3.4			V
		I_{OH} = –400 μA		4.8			
V_{OL}	Low-level output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P10_0$–$P10_7$, $P11_0$–$P11_7$	I_{OL} = 10 mA				2	V
V_{OL}	Low-level output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P4_0$, $P4_2$, $P4_4$–$P4_7$, $P10_0$–$P10_7$, $P11_0$–$P11_7$	I_{OL} = 2 mA				0.45	V
V_{OL}	Low-level output voltage $P3_1$–$P3_3$	I_{OL} = 10 mA				1.6	V
		I_{OL} = 2 mA				0.4	
V_{T+} —V_{T-}	Hysteresis $\overline{\text{RDY}}$, $\overline{\text{HOLD}}$, $TA0_{IN}$–$TA4_{IN}$, $TA0_{OUT}$–$TA4_{OUT}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{KI_0–KI_3}$, $\overline{INT_0}$–INT_4, $\overline{\text{NMI}}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1, RxD_0, RxD_1			0.2		0.7	V
V_{T+} —V_{T-}	Hysteresis $\overline{\text{RESET}}$			0.5		1.5	V
V_{T+} —V_{T-}	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, X_{IN}, $\overline{\text{RESET}}$, BYTE, MD0, MD1, $\overline{\text{NMI}}$	V_I = 5.0 V				5	μA
I_{IL}	Low-level input current $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_3$, $P5_0$–$P5_3$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, X_{IN}, $\overline{\text{RESET}}$, BYTE, MD0, MD1	V_I = 0 V				–5	μA
I_{IL}	Low-level input current $P4_4$–$P4_7$, $P5_4$–$P5_7$, $\overline{\text{NMI}}$	V_I = 0 V, No pullup transistor				–5	μA
		V_I = 0 V, Pullup transistor used		–0.4	–0.7	–1.1	mA
V_{RAM}	RAM hold voltage	When clock is stoped.		2			V
I_{CC}	Power source current	Output-only pins are open, and the other pins are connected to Vss or Vcc. An external square-waveform clock is input. (Pin X_{OUT} is open.) The PLL frequency multiplier stops its operation.	f(f_{sys}) = 26 MHz. CPU operates.		30	54	mA
			Ta = 25 °C when clock is stopped.			1	μA
			Ta = 85 °C when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS

(V_{CC} = AV_{CC} = 5 V ± 0.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits		Unit
				Min.	Max.	
—	Resolution	V_{REF} = V_{CC}			10	Bits
—	Absolute accuracy	V_{REF} = V_{CC}	10-bit resolution mode		± 3	LSB
			8-bit resolution mode		± 2	LSB
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}		5		kΩ
t_{CONV}	Conversion time	f(f_{sys}) ≤ 26 MHz	10-bit resolution mode	4.54		µs
			8-bit resolution mode	1.89 **(Note)**		
V_{REF}	Reference voltage			2.7	V_{CC}	V
V_{IA}	Analog input voltage			0	V_{REF}	V

Note: This is applied when A-D conversion freguency (ϕ_{AD}) = f_1.

D-A CONVERTER CHARACTERISTICS

(V_{CC} = 5 V, V_{SS} = AV_{SS} = 0 V, V_{REF} = 5 V, T_a = –20 to 85 °C, unless otherwise noted)

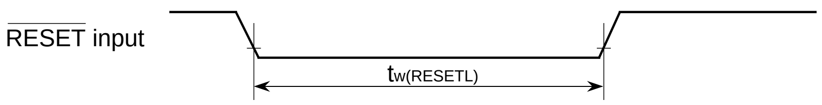
Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute accuracy				± 1.0	%
t_{su}	Set time				3	µs
R_O	Output resistance		1	2.5	4	kΩ
I_{VREF}	Reference power source input current	**(Note)**			3.2	mA

Note: The test conditions are as follows:
- One D-A converter is used.
- The D-A register value of the unused D-A converter is “00_{16}.”
- The reference power source input current for the ladder resistance of the A-D converter is excluded.

RESET INPUT

Reset input timing requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(RESETL)}$	$\overline{\text{RESET}}$ input low-level pulse width	2			µs



PERIPHERAL DEVICE INPUT/OUTPUT TIMING

(V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(f_{sys}) = 26 MHz unless otherwise noted)

✻ For limits depending on f(f_{sys}), their calculation formulas are shown below. Also, the values at f(f_{sys}) = 26 MHz are shown in ( ).

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TA$_{i\text{IN}}$ input cycle time	80		ns
$t_{w(TAH)}$	TA$_{i\text{IN}}$ input high-level pulse width	40		ns
$t_{w(TAL)}$	TA$_{i\text{IN}}$ input low-level pulse width	40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TA$_{i\text{IN}}$ input cycle time	f(f_{sys}) ≤ 26 MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (615)		ns
$t_{w(TAH)}$	TA$_{i\text{IN}}$ input high-level pulse width	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns
$t_{w(TAL)}$	TA$_{i\text{IN}}$ input low-level pulse width	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns

Note : The TA$_{i\text{IN}}$ input cycle time requires 4 or more cycles of a count source. The TA$_{i\text{IN}}$ input high-level pulse width and the TA$_{i\text{IN}}$ input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at f(f_{sys}) ≤ 26 MHz.

Timer A input (External trigger input in one-shot pulse mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TA$_{i\text{IN}}$ input cycle time	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns
$t_{w(TAH)}$	TA$_{i\text{IN}}$ input high-level pulse width		80		ns
$t_{w(TAL)}$	TA$_{i\text{IN}}$ input low-level pulse width		80		ns

Timer A input (External trigger input in pulse width modulation mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA$_{i\text{IN}}$ input high-level pulse width	80		ns
$t_{w(TAL)}$	TA$_{i\text{IN}}$ input low-level pulse width	80		ns

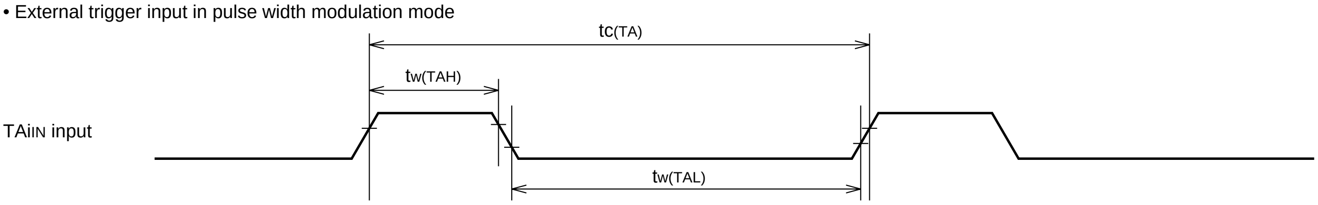
Timer A input (Up-down input and Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TA$_{i\text{OUT}}$ input cycle time	2000		ns
$t_{w(UPH)}$	TA$_{i\text{OUT}}$ input high-level pulse width	1000		ns
$t_{w(UPL)}$	TA$_{i\text{OUT}}$ input low-level pulse width	1000		ns
$t_{su(UP-T_{IN})}$	TA$_{i\text{OUT}}$ input setup time	400		ns
$t_{h(T_{IN}-UP)}$	TA$_{i\text{OUT}}$ input hold time	400		ns

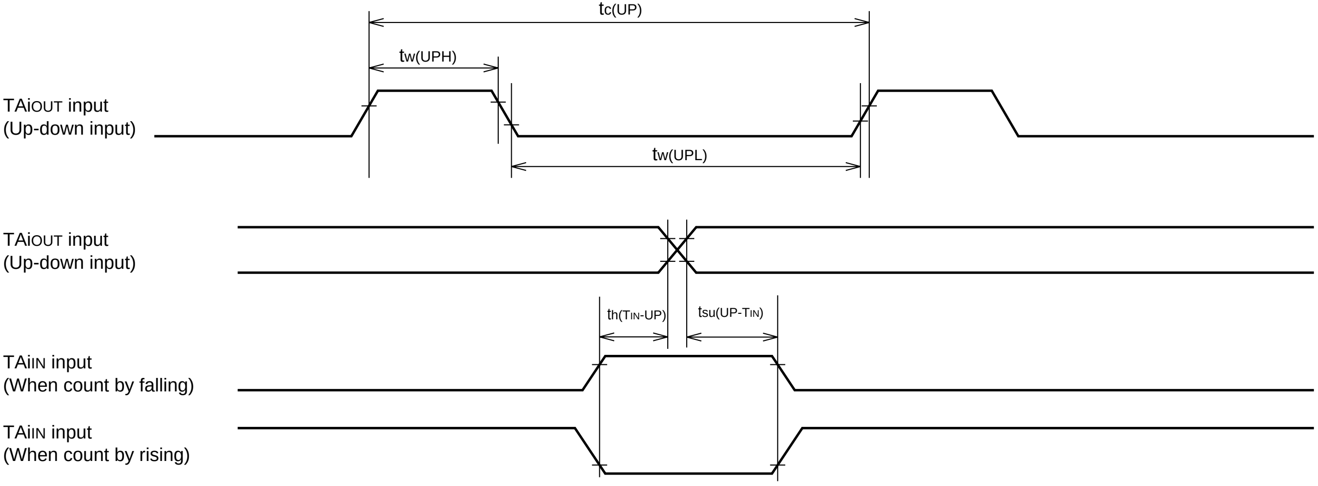
Timer A input (Two-phase pulse input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	800		ns
$t_{su(TAjIN-TAjOUT)}$	TAjIN input setup time	200		ns
$t_{su(TAjOUT-TAjIN)}$	TAjOUT input setup time	200		ns

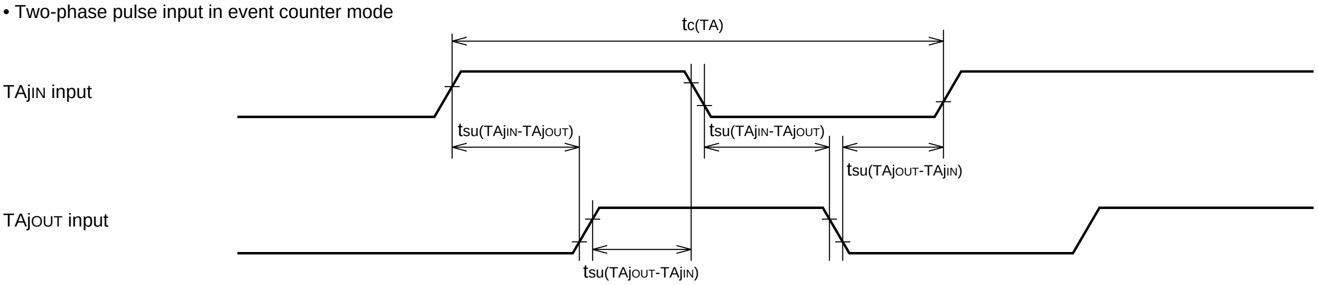
- Gating input in timer mode
- Count input in event counter mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode



- Up-down and Count input in event counter mode



- Two-phase pulse input in event counter mode



Test conditions
- V_{CC} = 5 V±0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.15 V

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time (one edge count)	80		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (one edge count)	40		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (one edge count)	40		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (both edge count)	160		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width (both edge count)	80		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width (both edge count)	80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	f(f_{sys}) ≤ 26 MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (615)		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns

Note: The TBi_{IN} input cycle time requires 4 or more cycles of a count source. The TBi_{IN} input high-level pulse width and the TBi_{IN} input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at f(f_{sys}) ≤ 26 MHz.

Timer B input (Pulse width measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	f(f_{sys}) ≤ 26 MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (615)		ns
$t_{w(TBH)}$	TBi_{IN} input high-level pulse width	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns
$t_{w(TBL)}$	TBi_{IN} input low-level pulse width	f(f_{sys}) ≤ 26 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (307)		ns

Note: The TBi_{IN} input cycle time requires 4 or more cycles of a count source. The TBi_{IN} input high-level pulse width and the TBi_{IN} input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at f(f_{sys}) ≤ 26 MHz.

A-D trigger input

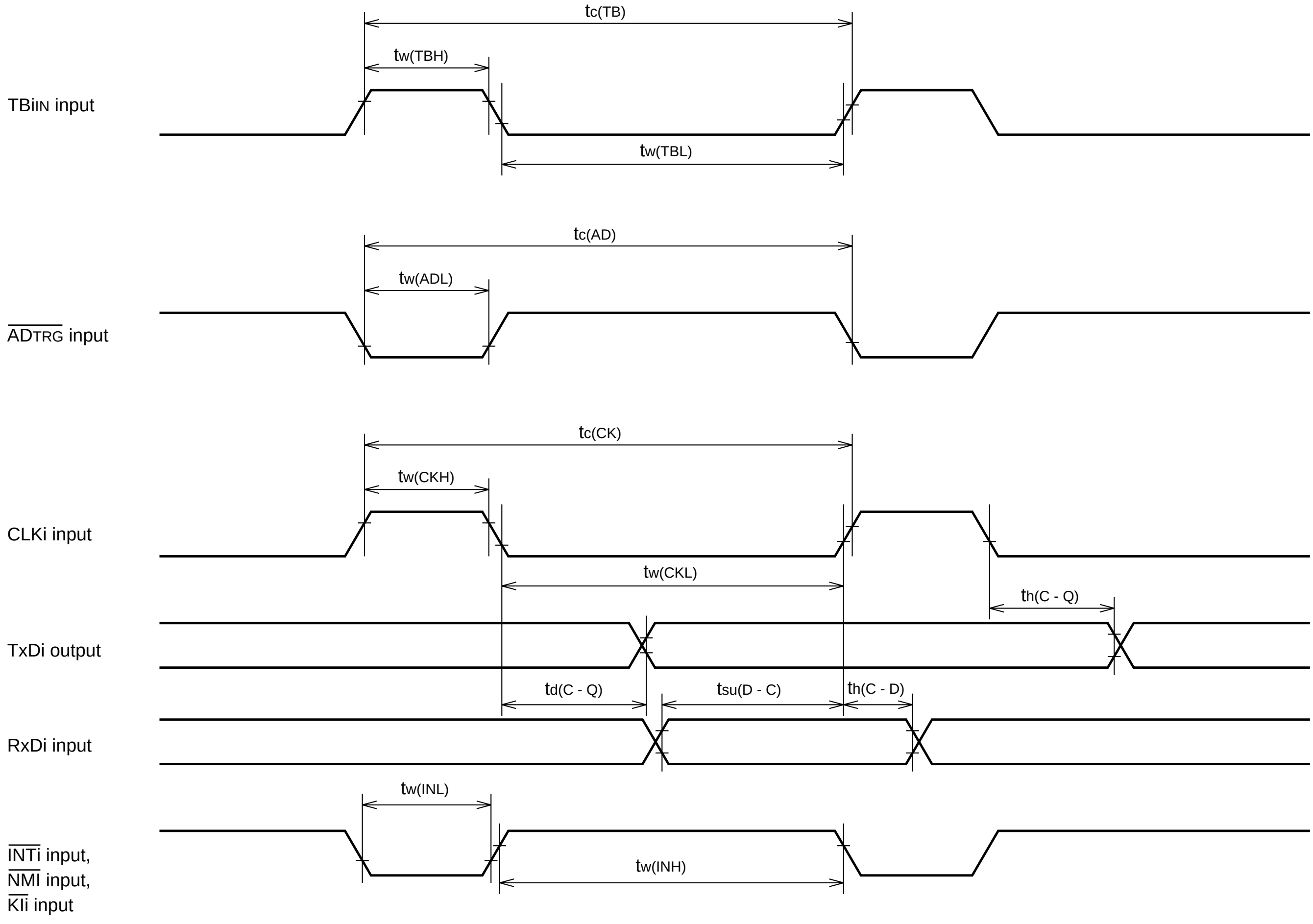
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (minimum allowable trigger)	1000		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input low-level pulse width	125		ns

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK$_i$ input cycle time	200		ns
$t_{w(CKH)}$	CLK$_i$ input high-level pulse width	100		ns
$t_{w(CKL)}$	CLK$_i$ input low-level pulse width	100		ns
$t_{d(C-Q)}$	TxD$_i$ output delay time		80	ns
$t_{h(C-Q)}$	TxD$_i$ hold time	0		ns
$t_{su(D-C)}$	RxD$_i$ input setup time	20		ns
$t_{h(C-D)}$	RxD$_i$ input hold time	90		ns

External interrupt ($\overline{\text{INT}_i}$) input, $\overline{\text{NMI}}$ input, Key input interrupt ($\overline{\text{KIi}}$) input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INT}_i}$ input/$\overline{\text{NMI}}$ input/$\overline{\text{KIi}}$ input high-level pulse width	250		ns
$t_{w(INL)}$	$\overline{\text{INT}_i}$ input/$\overline{\text{NMI}}$ input/$\overline{\text{KIi}}$ input low-level pulse width	250		ns



Test conditions
- Vcc = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 0.8 V, V_{IH} = 2.15 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 50 pF

READY, HOLD TIMING

Timing requirements (V

Timing requirements (V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(f_{sys}) = 26 MHz, unless otherwise noted)

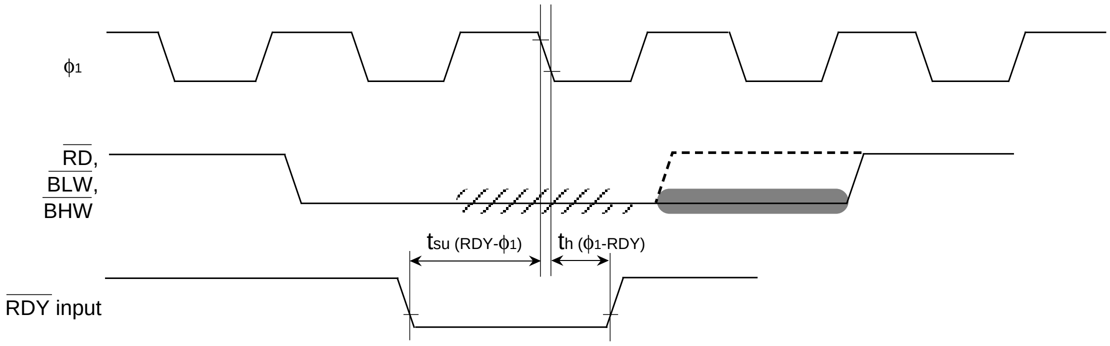
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(RDY-\phi 1)}$	$\overline{\text{RDY}}$ input setup time	40		ns
$t_{su(HOLD-\phi 1)}$	$\overline{\text{HOLD}}$ input setup time	40		ns
$t_{h(\phi 1-RDY)}$	$\overline{\text{RDY}}$ input hold time	0		ns
$t_{h(\phi 1-HOLD)}$	$\overline{\text{HOLD}}$ input hold time	0		ns

Switching characteristics (V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(f_{sys}) = 26 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(\phi 1-HLDAL)}$	$\overline{\text{HLDA}}$ output delay time		20	ns
$t_{d(RDH-HLDAL)}$	$\overline{\text{HLDA}}$ low-level output delay time after read	t_c –15 **(Note)**		ns
$t_{d(BXWH-HLDAL)}$	$\overline{\text{HLDA}}$ low-level output delay time after write	t_c –15 **(Note)**		ns
$t_{pxz(HLDAL-RDZ)}$	Floating start delay time	–15	10	ns
$t_{pxz(HLDAL-BXWZ)}$	Floating start delay time	–15	10	ns
$t_{pxz(HLDAL-CSiZ)}$	Floating start delay time	–15	10	ns
$t_{pxz(HLDAL-ALEZ)}$	Floating start delay time	–15	10	ns
$t_{pxz(HLDAL-AZ)}$	Floating start delay time	–15	10	ns
$t_{pzx(HLDAL-RDZ)}$	Floating release delay time	0		ns
$t_{pzx(HLDAL-BXWZ)}$	Floating release delay time	0		ns
$t_{pzx(HLDAL-CSiZ)}$	Floating release delay time	0		ns
$t_{pzx(HLDAL-ALEZ)}$	Floating release delay time	0		ns
$t_{pzx(HLDAL-AZ)}$	Floating release delay time	0		ns

Note: t_c = 1/f(f_{sys}).

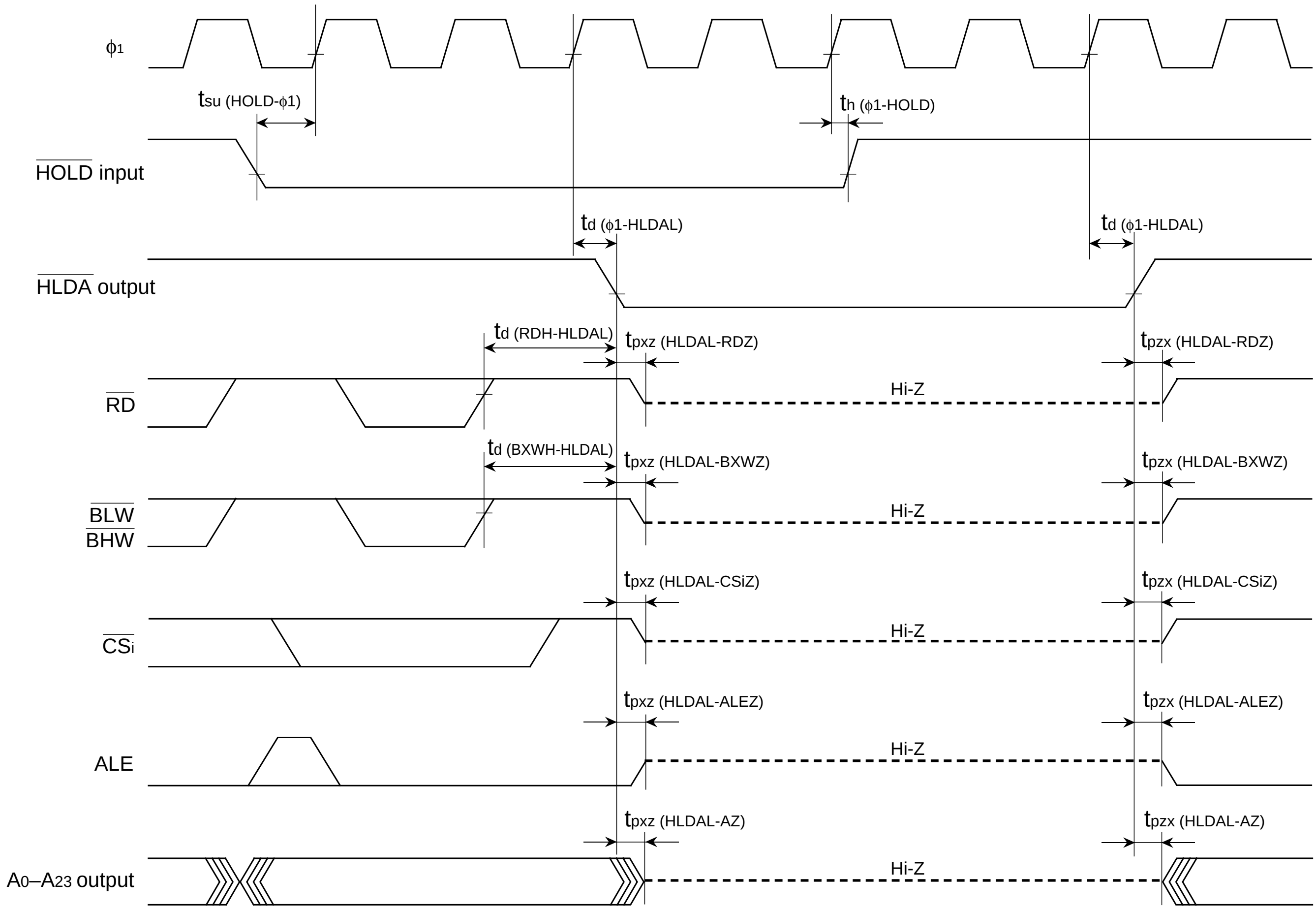
$\overline{\text{RDY}}$ input



▨ : Wait inserted by software (The above is applied when bus cycle = 1φ + 2φ)

▬ : Wait inserted by ready function

$\overline{\text{HOLD}}$ input



Test conditions

- V_{CC} = 5 V ± 0.5 V, Ta= –20 to 85 °C
- $\overline{\text{RDY}}$ input, $\overline{\text{HOLD}}$ input : V_{IL} = 0.8V, V_{IH} = 2.15 V
- $\overline{\text{HLDA}}$ output : V_{OL} = 0.8V, V_{OH} = 2.0 V, C_L = 50 pF

External bus timing

For limits depending on f(f_{sys}), their calculation formulas are shown below.

Bus cycle	W_H	W_L	Bus cycle	W_H	W_L
$1\phi+1\phi$	1	1	$2\phi+3\phi$	2	3
$1\phi+2\phi$	1	2	$2\phi+4\phi$	2	4
$1\phi+3\phi$	1	3	$3\phi+3\phi$	3	3
$2\phi+2\phi$	2	2	$3\phi+4\phi$	3	4

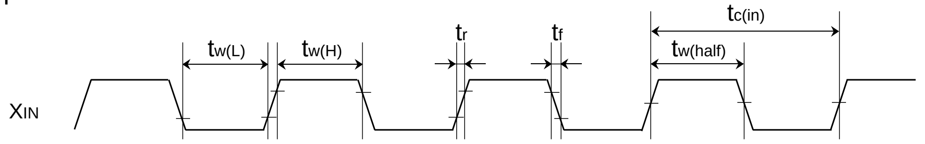
t_c = 1/f(f_{sys}).

Timing Requirements (V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 26 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(in)}$	External clock input cycle time	38		ns
$t_{w(half)}$	External clock input pulse width with half input-volage	0.45 tc	0.55 tc	ns
$t_{w(H)}$	External clock input high-level pulse width	0.5 t_c – 6		ns
$t_{w(L)}$	External clock input low-level pulse width	0.5 t_c – 6		ns
t_r	External clock input rise time	6		ns
t_f	External clock input fall time	6		ns
$t_{a(A-D)}$	Address access time (the address output select bit = 0)		(W_H + W_L) tc-45	ns
$t_{a(A-D)}$	Address access time (the address output select bit = 1)		(W_H + W_L-0.5) tc-35	ns
$t_{a(CSiL-D)}$	Chip select access time		(W_H + W_L-0.5) tc-35	ns
$t_{a(RDL-D)}$	Read access time		W_L × tc-30	ns
$t_{su(D-RDL)}$	Read data setup time	15		ns
$t_{h(RDH-D)}$	Data input hold time after read	0		ns
$t_{a(BA-D)}$	Address access time at burst ROM access		W_L × tc-35	ns
$t_{h(BA-D)}$	Data hold time after address at burst ROM access	8		ns
$t_{a(LA-D)}$	Address access time (the multiplexed bus select bit = 1)	(W_H + W_L-0.5)tc-35 **(Note)**		ns

Note: This is independent of the address output select bit's contents.

External clock input



Test conditions
- Vcc = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V ($t_{w(H)}$, $t_{w(L)}$, t_r, t_f)
- Output timing voltage : 2.5 V ($t_{c(in)}$, $t_{w(half)}$)

Switching characteristics (V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(f_{sys}) = 26 MHz, unless otherwise noted)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{d(\phi1\text{-}RDL)}$	Read low-level output delay time		–18	0	ns
$t_{d(\phi1\text{-}RDH)}$	Read high-level output delay time		–18	0	ns
$t_{d(\phi1\text{-}BXWL)}$	Write low-level output delay time		–18	0	ns
$t_{d(\phi1\text{-}BXWH)}$	Write high-level output delay time		–18	0	ns
$t_{d(\phi1L\text{-}CSiL)}$	Chip select low-level output delay time		–20	0	ns
$t_{d(\phi1L\text{-}CSiH)}$	Chip select high-level output delay time		–22	10	ns
$t_{d(\phi1H\text{-}A)}$	Address output delay time (the address output select bit = 0)		–5	25	ns
$t_{d(\phi1L\text{-}A)}$	Address output delay time (the address output select bit = 1)		–20	16	ns
$t_{w(ALEH)}$	ALE pulse width	Bus cycle = 1φ + 1φ, 1φ + 2φ, 1φ + 3φ	0.5tc-19		ns
		Bus cycle = 2φ + 2φ	tc-20		ns
		Bus cycle = 2φ + 3φ, 2φ + 4φ, 3φ + 3φ, 3φ + 4φ	1.5tc-20		ns
$t_{d(A\text{-}ALEL)}$	ALE completion delay time after address stabilization (when the address output select bit = 0)	Bus cycle = 1φ + 1φ, 1φ + 2φ, 1φ + 3φ	tc-30		ns
		Bus cycle = 2φ + 2φ	1.5tc-30		ns
		Bus cycle = 2φ + 3φ, 2φ + 4φ, 3φ + 3φ, 3φ + 4φ	2tc-30		ns
	ALE completion delay time after address stabilization (when the address output select bit = 1)	Bus cycle = 1φ + 1φ, 1φ + 2φ, 1φ + 3φ	0.5tc-19		ns
		Bus cycle = 2φ + 2φ	tc-20		ns
		Bus cycle = 2φ + 3φ, 2φ + 4φ, 3φ + 3φ, 3φ + 4φ	1.5tc-20		ns
$t_{w(RDL)}$	Read output pulse width		W_L × tc-15		ns
$t_{w(RDH)}$	Read output high-level width **(Note 1)**		W_H × tc-15		ns
$t_{d(RDH\text{-}BXWH)}$	Write disable valid time after read **(Note 2)**		tc-15		ns
$t_{d(A\text{-}RDH)}$	Address valid time before read (when the address output select bit = 0)		W_H × tc-30		ns
$t_{d(A\text{-}RDH)}$	Address valid time before read (when the address output select bit = 1)		(W_H-0.5)tc-19		ns
$t_{h(RDH\text{-}A)}$	Address hold time after read (when the address output select bit = 0) **(Note 2)**		8		ns
$t_{h(RDH\text{-}A)}$	Address hold time after read (when the address output select bit = 1) **(Note 2)**		0.5tc-10		ns
$t_{d(RDH\text{-}ALEL)}$	ALE completion delay time after read start			20	ns
$t_{d(ALEL\text{-}RDH)}$	Read disable valid time after ALE completion	Bus cycle = 2φ + 2φ	0.5tc-19		ns
		Bus cycle = 3φ + 3φ, 3φ + 4φ	tc-15		ns
$t_{d(CSiL\text{-}RDH)}$	Chip select valid time before read		(W_H-0.5)tc-19		ns
$t_{d(CSiL\text{-}RDL)}$	Chip select output valid time before read completion		(W_H + W_L-0.5)tc-20		ns
$t_{h(RDH\text{-}CSiL)}$	Chip select hold time after read		0.5tc-14		ns
$t_{d(RDH\text{-}D)}$	Next write cycle data output delay time after read **(Note 2)**		tc-15		ns
$t_{w(BXWL)}$	Write output pulse width		W_L × tc-15		ns
$t_{w(BXWH)}$	Write output high-level width **(Note 1)**		W_H × tc-15		ns
$t_{d(BXWH\text{-}RDH)}$	Read disable valid time after write **(Note 2)**		tc-15		ns
$t_{d(A\text{-}BXWH)}$	Address valid time before write (when the address output select bit = 0)		W_H × tc-30		ns
$t_{d(A\text{-}BXWH)}$	Address valid time before write (when the address output select bit = 1)		(W_H-0.5)tc-19		ns
$t_{h(BXWH\text{-}A)}$	Address hold time after write (when the address output select bit = 0) **(Note 2)**		8		ns
$t_{h(BXWH\text{-}A)}$	Address hold time after write (when the address output select bit = 1) **(Note 2)**		0.5tc-10		ns
$t_{d(BXWH\text{-}ALEL)}$	ALE completion delay time after write start			20	ns
$t_{d(ALEL\text{-}BXWH)}$	Write disable valid time after ALE completion	Bus cycle = 2φ + 2φ	0.5tc-19		ns
		Bus cycle = 2φ + 3φ, 2φ + 4φ, 3φ + 3φ, 3φ + 4φ	tc-15		ns
$t_{d(CSiL\text{-}BXWH)}$	Chip select valid time before write		(W_H-0.5)tc-19		ns
$t_{d(CSiL\text{-}BXWL)}$	Chip select output valid time before write completion		(W_H + W_L-0.5)tc-20		ns
$t_{h(BXWH\text{-}CSiL)}$	Chip select hold time after write		0.5tc-14		ns
$t_{d(D\text{-}BXWL)}$	Data output valid time before write completion		W_L × tc-20		ns
$t_{h(BXWH\text{-}D)}$	Data hold time after write **(Note 3)**		0.5tc-10		ns
$t_{pxz(BXWH\text{-}DZ)}$	Floating start delay time after write **(Note 3)**			0.5tc + 10	ns

Notes 1: When the bus cycle just before this parameter is for the area where the recovery cycle insertion is selected, this parameter is extended by tc (ns: one recovery cycle is inserted.) or by 2tc (ns: two recovery cycles are inserted.).

2: When accessing the area where the recovery cycle insertion is selected, this parameter is extended by tc (ns: one recovery cycle is inserted.) or by 2tc (ns: two recovery cycles are inserted.).

3: This parameter is extended by tc (ns) when both of the following conditions are satisfied:

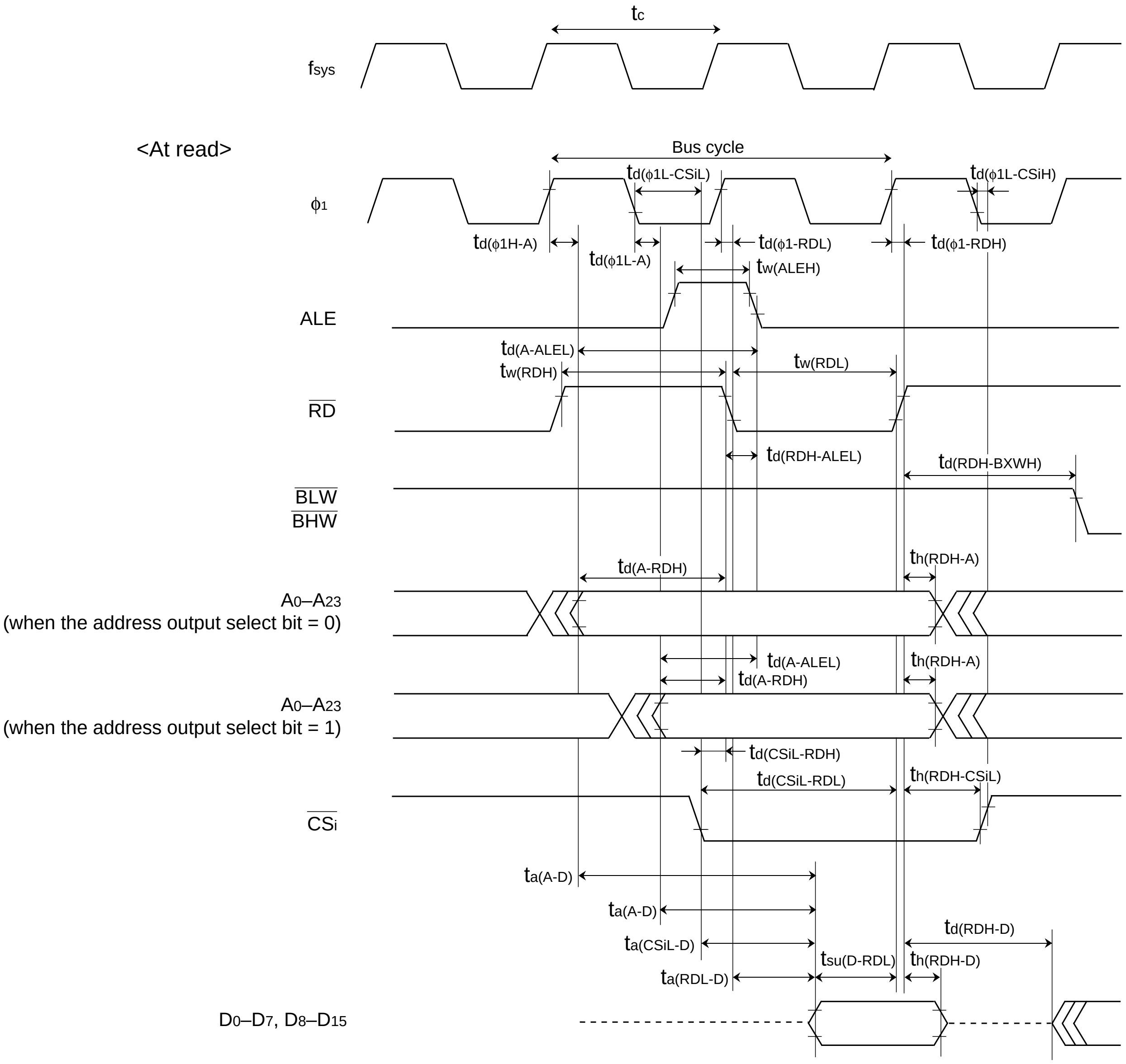
- When accessing the area where the recovery cycle insertion is selected.
- When two recovery cycles are inserted.

Switching characteristics (V_{CC} = 5 V±0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(f_{sys}) = 26 MHz, unless otherwise noted)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{d(LA-RDH)}$	Address valid time before read		(W_H-0.5)t_c-19 **(Note)**		ns
$t_{d(LA-ALEL)}$	ALE completion delay time after address stabilization	Bus cycle = 2ϕ + 2ϕ	t_c-20 **(Note)**		ns
		Bus cycle = 3ϕ + 3ϕ, 3ϕ + 4ϕ	1.5t_c-20 **(Note)**		ns
$t_{h(ALEL-LA)}$	Address hold time after ALE completion	Bus cycle = 2ϕ + 2ϕ	0.5t_c-19		ns
		Bus cycle = 3ϕ + 3ϕ, 3ϕ + 4ϕ	t_c-15		ns
$t_{pxz(RDH-LAZ)}$	Floating start delay time			5	ns
$t_{d(LA-BXWH)}$	Address valid time before write		(W_H-0.5)t_c-19 **(Note)**		ns
$t_{pzx(RDH-DZ)}$	Floating release delay time		0.5t_c-19 **(Note)**		ns

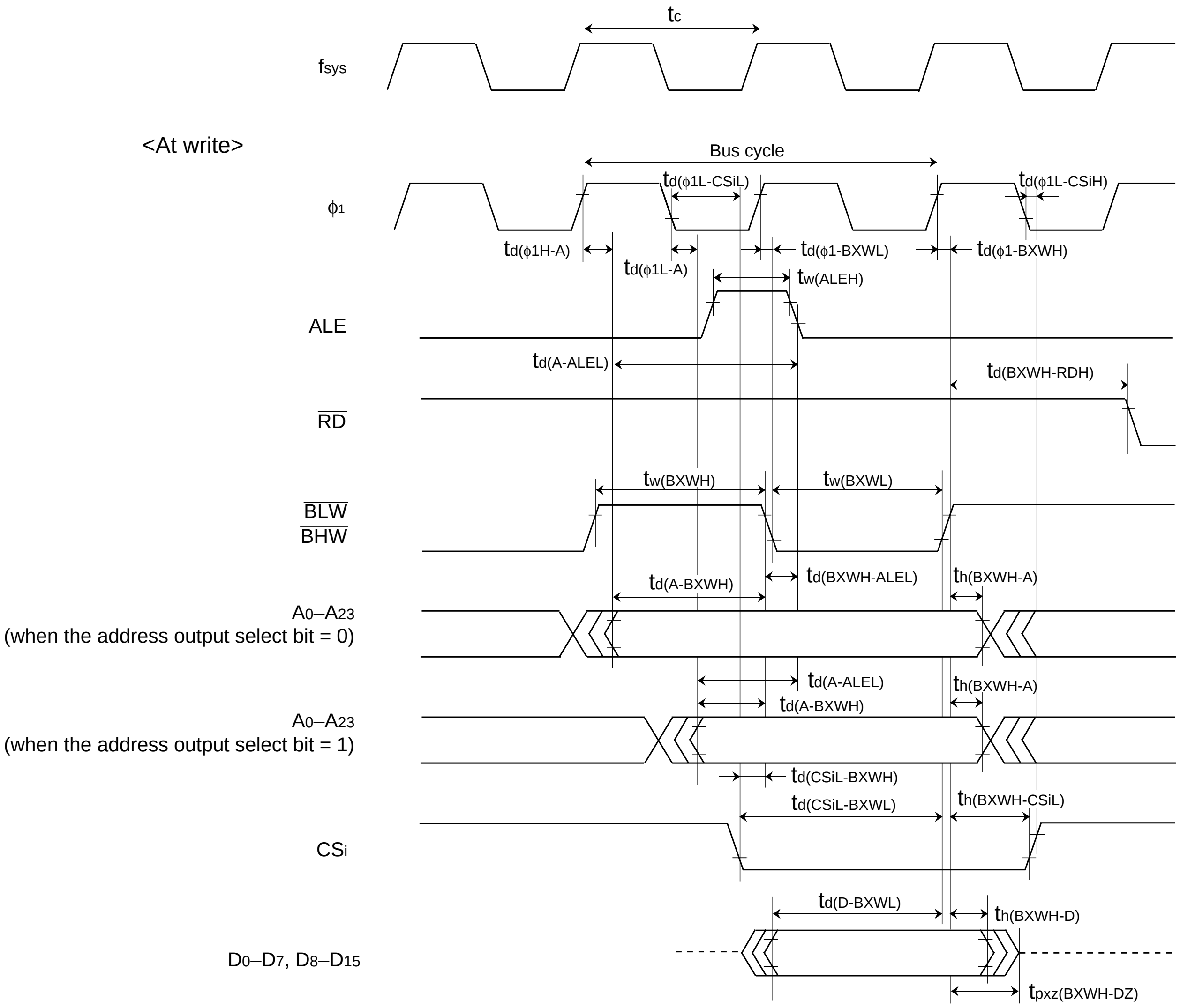
Note: This is independent of the address output select bit’s contents.

Normal access: bus cycle = 1ϕ + 1ϕ, 1ϕ + 2ϕ, 1ϕ+ 3ϕ,
2ϕ + 3ϕ, or 2ϕ + 4ϕ



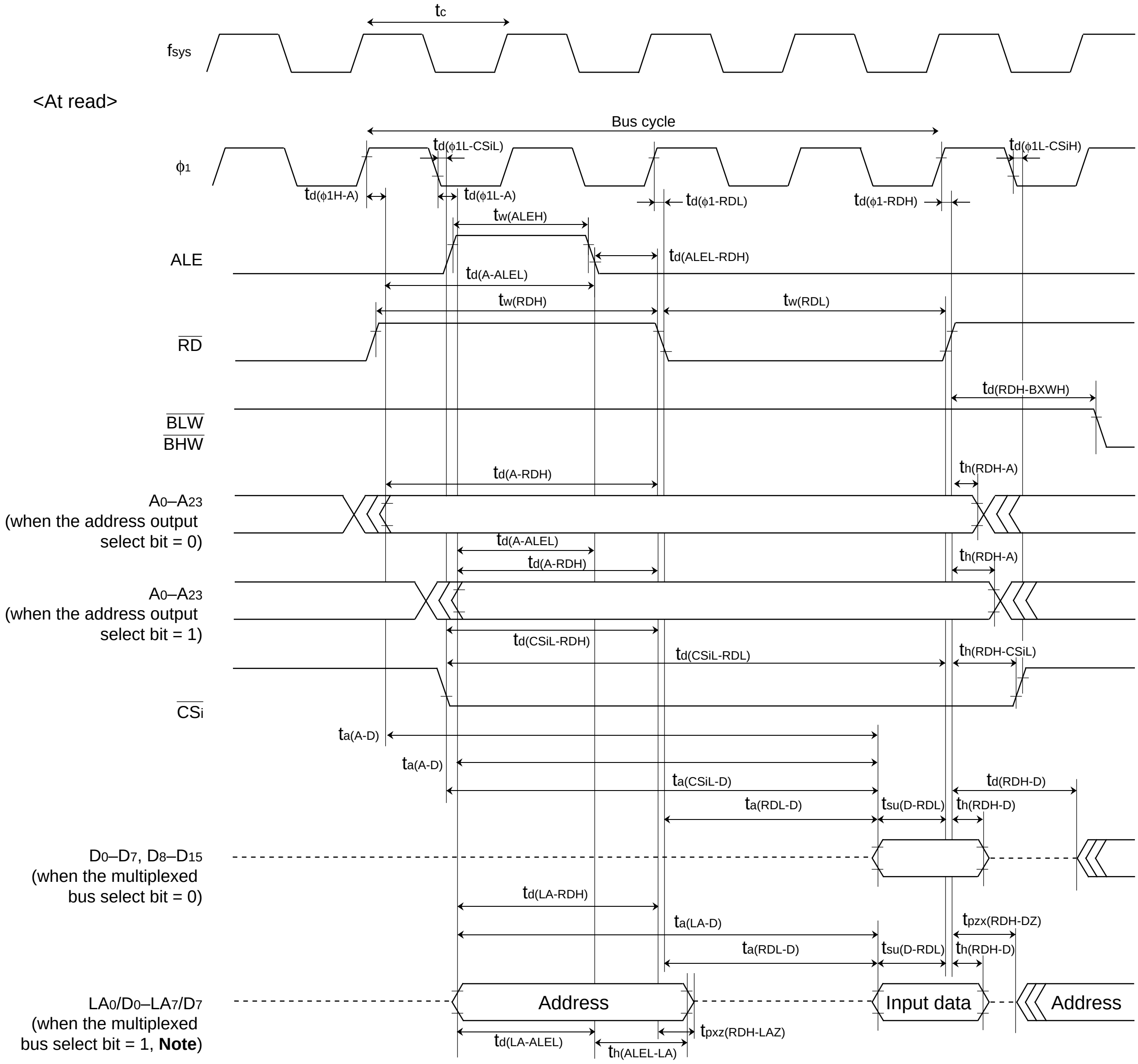
Test conditions
- V_{CC} = 5 V ±0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.15 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{CS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{CS_i}$)

Normal access: bus cycle = 1ϕ + 1ϕ, 1ϕ + 2ϕ, 1ϕ + 3ϕ,
2ϕ + 3ϕ, or 2ϕ + 4ϕ


t_c
f_{sys}
<At write>
Bus cycle
$t_{d(\phi 1L\text{-}CSiL)}$
$t_{d(\phi 1L\text{-}CSiH)}$
ϕ_1
$t_{d(\phi 1H\text{-}A)}$
$t_{d(\phi 1\text{-}BXWL)}$
$t_{d(\phi 1\text{-}BXWH)}$
$t_{d(\phi 1L\text{-}A)}$
$t_{w(ALEH)}$
ALE
$t_{d(A\text{-}ALEL)}$
$t_{d(BXWH\text{-}RDH)}$
$\overline{RD}$
$t_{w(BXWH)}$
$t_{w(BXWL)}$
$\overline{BLW}$
$\overline{BHW}$
$t_{d(A\text{-}BXWH)}$
$t_{d(BXWH\text{-}ALEL)}$
$t_{h(BXWH\text{-}A)}$
A_0–A_{23}
(when the address output select bit = 0)
$t_{d(A\text{-}ALEL)}$
$t_{h(BXWH\text{-}A)}$
$t_{d(A\text{-}BXWH)}$
A_0–A_{23}
(when the address output select bit = 1)
$t_{d(CSiL\text{-}BXWH)}$
$t_{d(CSiL\text{-}BXWL)}$
$t_{h(BXWH\text{-}CSiL)}$
$\overline{CS_i}$
$t_{d(D\text{-}BXWL)}$
$t_{h(BXWH\text{-}D)}$
D_0–D_7, D_8–D_{15}
$t_{pxz(BXWH\text{-}DZ)}$

Test conditions
- V_{CC} = 5 V ±0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.15 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{CS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{CS_i}$)

Normal access: bus cycle = 2ϕ + 2ϕ, 3ϕ + 3ϕ, 3ϕ + 4ϕ

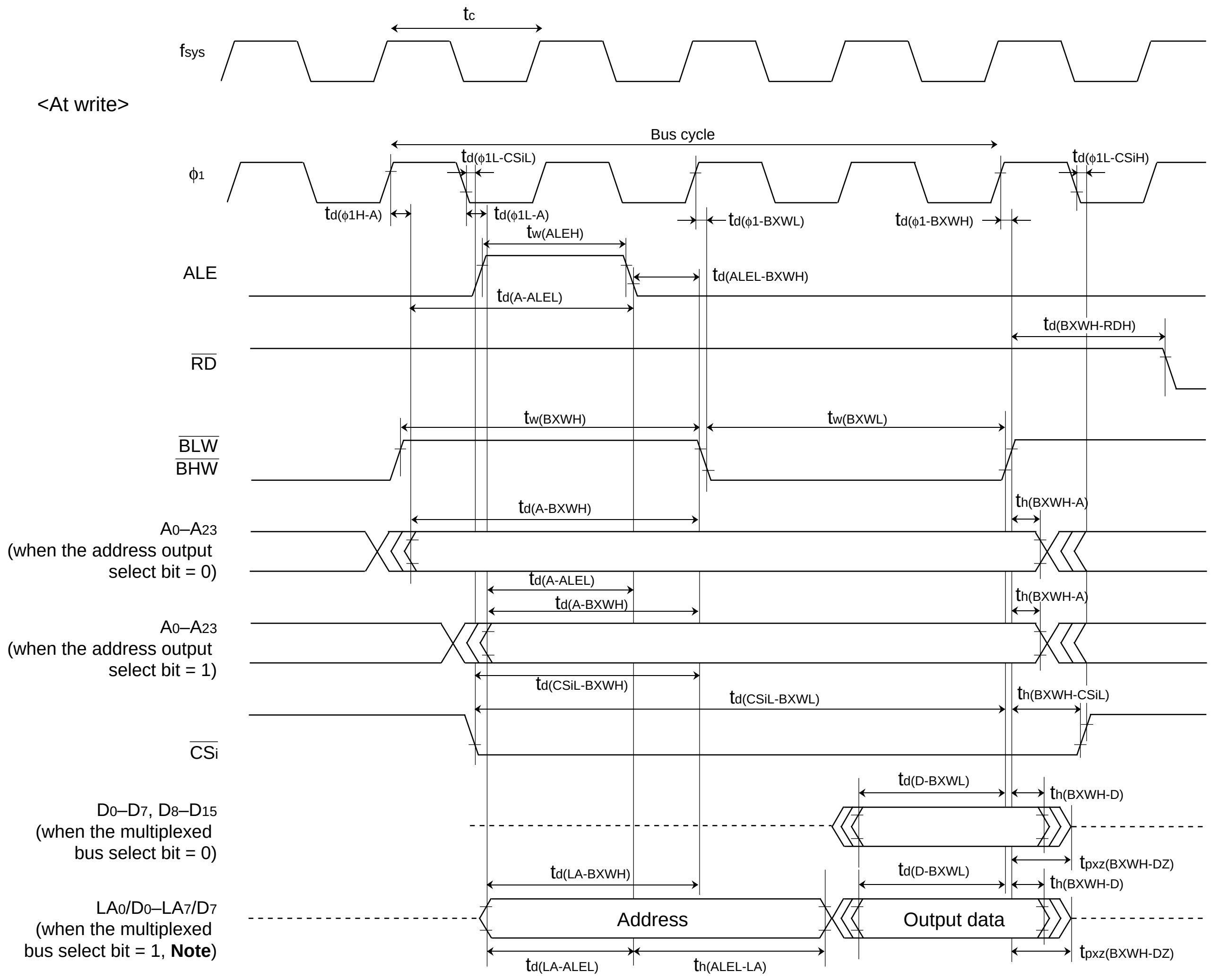


Note: Valid only when area $\overline{\text{CS2}}$ is accessed with the external data bus width = 8 bits.

Test conditions

- V_{CC} = 5 V ±0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.15 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{\text{CS}}_i$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{\text{CS}}_i$)

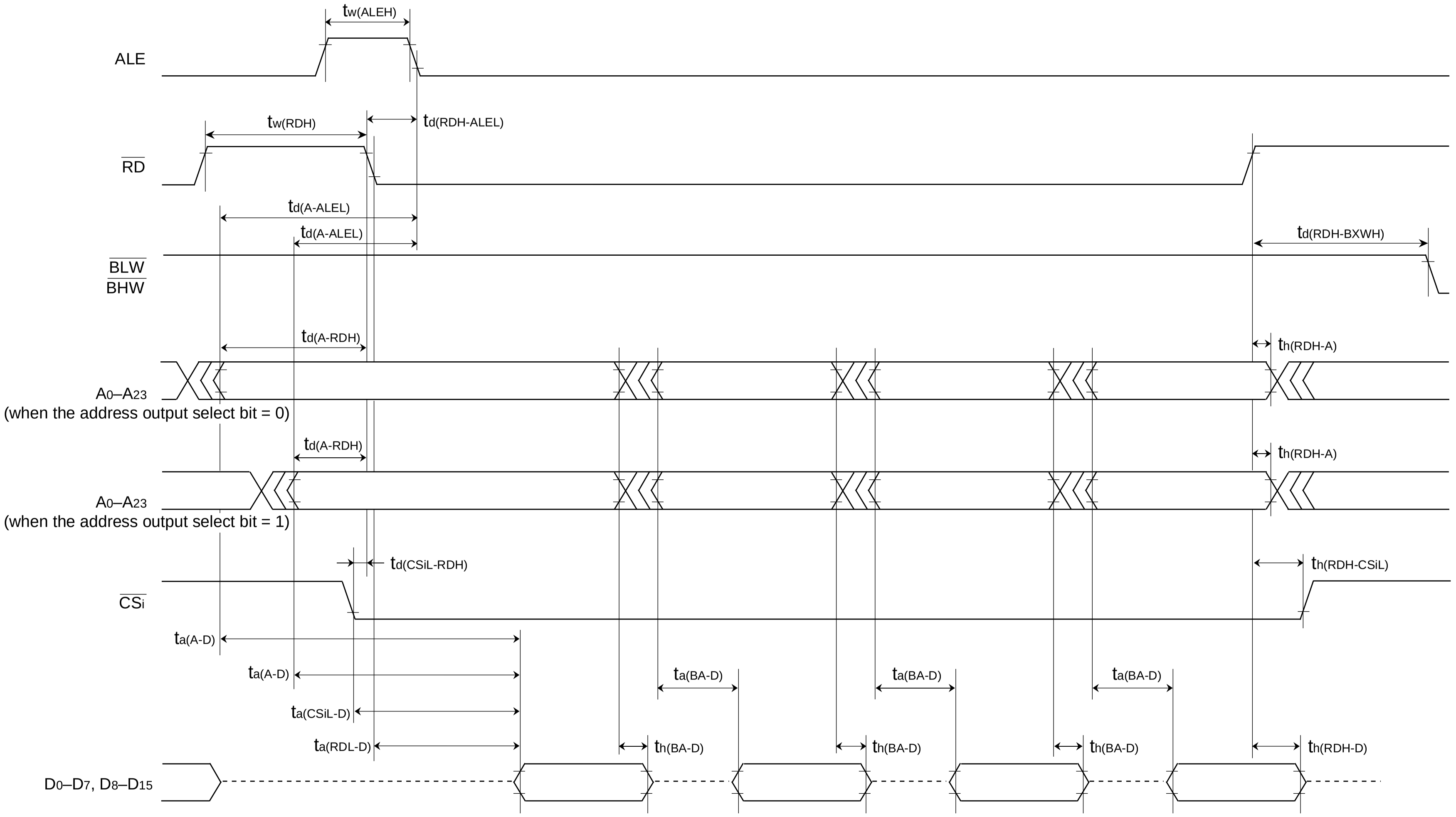
Normal access: bus cycle = 2ϕ + 2ϕ, 3ϕ + 3ϕ, 3ϕ + 4ϕ



Note: Valid only when area $\overline{CS2}$ is accessed with the external data bus width = 8 bits.

Test conditions
- V_{CC} = 5 V ±0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.15 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{CS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{CS_i}$)

Burst ROM access: bus cycle = 1ϕ + 1ϕ, 1ϕ + 2ϕ, 1ϕ + 3ϕ, 2ϕ + 3ϕ, 2ϕ + 4ϕ



Test conditions

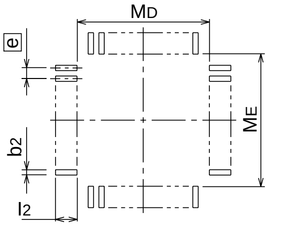
- V_{CC} = 5 V ±0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.15 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{CS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{CS_i}$)

PACKAGE OUTLINE

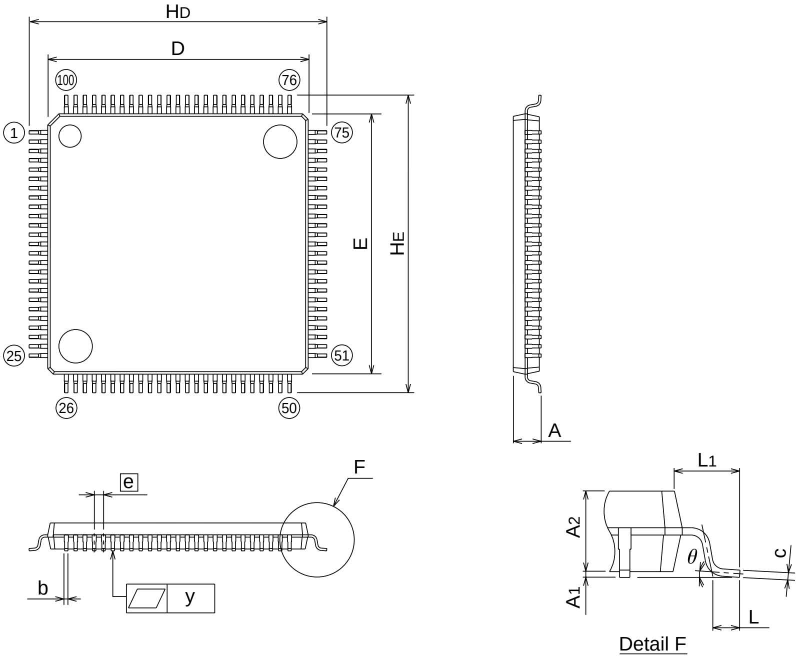
100P6Q-A

Plastic 100pin 14×14mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	–		Cu Alloy



Recommended Mount Pad



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A_1	0	0.1	0.2
A_2	–	1.4	–
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	13.9	14.0	14.1
$\boxed{e}$	–	0.5	–
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
L	0.3	0.5	0.7
L_1	–	1.0	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.225	–
l_2	1.0	–	–
M_D	–	14.4	–
M_E	–	14.4	–

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

New publication, effective Jun., 2000.
Specifications subject to change without notice.

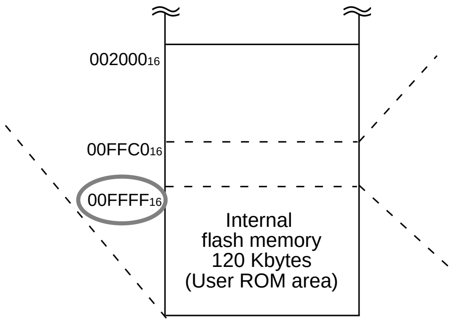
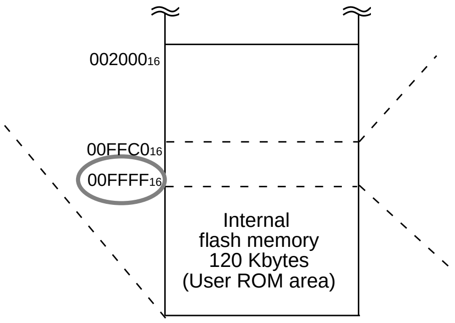
Revision History	M37902FxCHP Datasheet

<table>
<tr><th>Rev.
No.</th><th>Revision Description</th><th>Rev.
date</th></tr>
<tr><td>1.0</td><td>First Edition</td><td>990305</td></tr>
<tr><td>2.0</td><td>Refer to <b>Corrections and Supplementary Explanation for “M37902FxC Datasheet (REV.A)”.</b></td><td>990625</td></tr>
<tr><td>3.0</td><td>The following are revised/added points in this edition:
• Figure 26 in page 40; the bit’s name (bit 7) of the port function control register
<Error> Pin $\overline{\text{NMI}}$ pullup connection select bit <b>(Note 2)</b>
<Correction> Pin $\overline{\text{NMI}}$ pullup select bit <b>(Note 2)</b>
• Page 95; <b>CLOCK GENERATING CIRCUIT</b>, Right column, line 10
<Error> ••••• the PLL output clock (f_{PLL}). (In other words, set bit 5 to “1”.)
<Correction> ••••• the PLL output clock (f_{PLL}). (In other words, set bit 5 to “1”.) Note that, after reset, the PLL multiplication ratio select bits are allowed to be changed only once.
• Table 15 in page 95; Note is revised.
<Error> ••••• f(X_{IN}) means the frequency of the input clock from pin X_{IN} f(X_{IN}).
<Correction> ••••• f(X_{IN}) means the frequency of the input clock from pin X_{IN} f(X_{IN}). After reset, the PLL multiplication ratio select bits are allowed to be changed only once.
• Page 120; <b>RECOMMENDED OPERATING CONDITIONS</b>
&lt;Error&gt;<table><tr><td>f(f_{sys})</td><td>External clock input frequency <b>(Note 2)</b></td><td>••••••••••••</td></tr></table>&lt;Correction&gt;<table><tr><td>f(X_{IN})</td><td>External clock input frequency <b>(Note 2)</b></td><td>••••••••••••</td></tr></table></td><td>990917</td></tr>
<tr><td>4.0</td><td>The following are revised/added points in this edition:
• Page 83; <b>D-A CONVERTER</b>, Left column, line 15
<Error> The D-A output enable bit is cleared to “0” at reset. ••••
<Correction> The contents of the corresponding D-A output enable bit and D-A register are cleared to “0” at reset. ••••
• Page 83; <b>D-A CONVERTER</b>, Right column, line 1
<Error> with pin D-Ai.
<Correction> with pin D-Ai. Also, when not using the D-A converter, be sure to clear the contents of the corresponding D-A output enable bit and D-A register to “0”.</td><td>991008</td></tr>
<tr><td>5.0</td><td>Refer to <b>Corrections and Supplementary Explanation for “M37902FxC Datasheet (REV.B)”.</b>
<b>Notes 1:</b> ★ represents the new information added in Rev.5.0.
<b>2:</b> The revised/added points informed in Rev.3.0 and Rev.4.0 are included in <b>Corrections and Supplementary Explanation for “M37902FxC Datasheet (REV.B)”.</b></td><td>000629</td></tr>
</table>

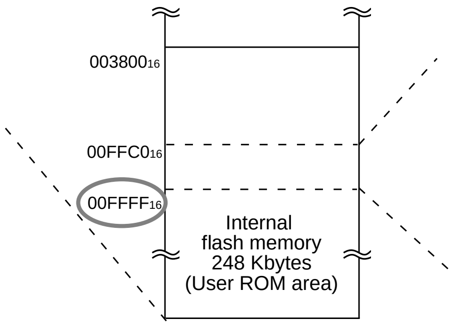
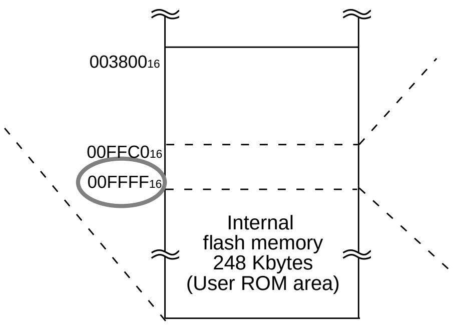
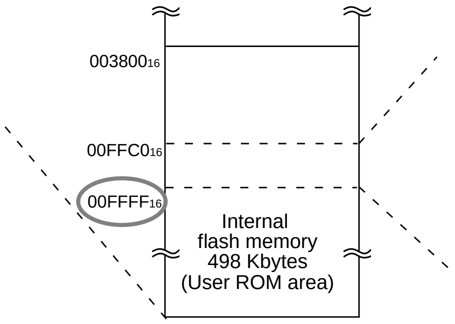
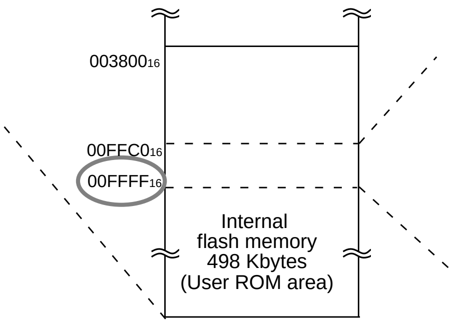
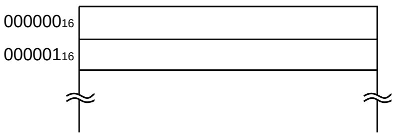
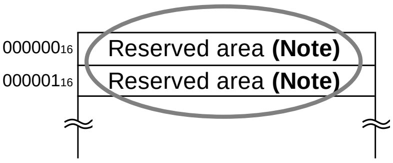
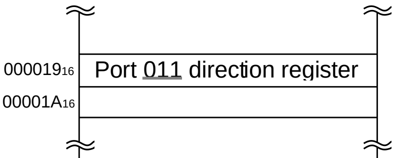
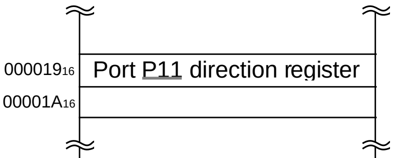
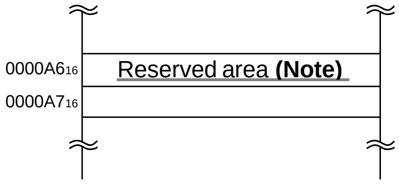
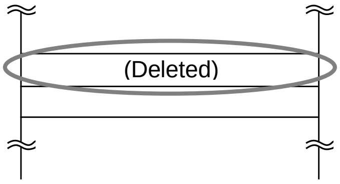
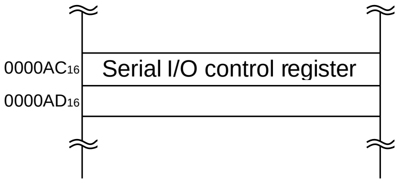
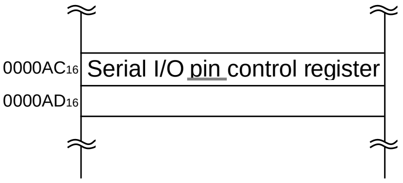
Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.1

<table>
<tr><th></th><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td rowspan="2">★</td><td rowspan="2">All pages,
Header</td><td>PRELIMINARY
Notice: This is not a final specification.
Some parametric limits are subject to change.</td><td>(Deleted)</td></tr>
<tr><td>M37902F8CGP, M37902F8CHP, M37902FCCGP, M37902FCCHP, M37902FECGP, M37902FECHP, M37902FGCGP, M37902FGCHP, M37902FHCGP, M37902FHCHP, M37902FJCGP, M37902FJCHP</td><td>M37902FCCHP, M37902FGCHP, M37902FJCHP</td></tr>
<tr><td rowspan="6">★</td><td rowspan="6">Page 1,
DISTINCTIVE FEATURES ;
Memory</td><td>[M37902F8CGP, M37902F8CHP]
Flash memory (User ROM area) ....60 Kbytes
RAM...2048 bytes</td><td>(Deleted)</td></tr>
<tr><td>[M37902FCCGP, M37902FCCHP]
Flash memory (User ROM area) ....120 Kbytes
RAM...4096 bytes</td><td>[M37902FCCHP]
Flash memory (User ROM area) ....120 Kbytes
RAM...4096 bytes</td></tr>
<tr><td>[M37902FECGP, M37902FECHP]
Flash memory (User ROM area) ....184 Kbytes
RAM...6144 bytes</td><td>(Deleted)</td></tr>
<tr><td>[M37902FGCGP, M37902FGCHP]
Flash memory (User ROM area) ....248 Kbytes
RAM...6144 bytes</td><td>[M37902FGCHP]
Flash memory (User ROM area) ....248 Kbytes
RAM...6144 bytes</td></tr>
<tr><td>[M37902FHCGP, M37902FHCHP]
Flash memory (User ROM area) ....370 Kbytes
RAM...12288 bytes</td><td>(Deleted)</td></tr>
<tr><td>[M37902FJCGP, M37902FJCHP]
Flash memory (User ROM area) ....498 Kbytes
RAM...12288 bytes</td><td>[M37902FJCHP]
Flash memory (User ROM area) ....498 Kbytes
RAM...12288 bytes</td></tr>
<tr><td>★</td><td>Page 1,
APPLICATION</td><td>Control devices for personal computer peripheral equipment such as CD-ROM drives, DVD-ROM drives, hard disk drives, high density FDD, printers
Control devices for office equipment such as copiers and facsimiles
Control devices for industrial equipment such as communication and measuring instruments</td><td>Control devices for personal computer peripheral equipment such as CD-ROM drives, DVD-ROM drives, hard disk drives, high density FDD, printers</td></tr>
<tr><td>★</td><td>———</td><td>M37902FxCGP PIN CONFIGURATION (TOP VIEW)</td><td>(Deleted)</td></tr>
<tr><td rowspan="2">★</td><td rowspan="2">Page 2,
PIN CONFIGURATION
Page 105,
Fig. 112,
Pin connection of M37902FxCHP in flash memory</td><td>P6$_3$/$\overline{\text{INT}_2}$</td><td>P6$_4$/$\overline{\text{INT}_2}$</td></tr>
<tr><td>(Type)
M37902F8CHP
M37902FCCHP
M37902FECHP
M37902FGCHP
M37902FHCHP
M37902FJCHP</td><td>(Type)
M37902FCCHP
M37902FGCHP
M37902FJCHP</td></tr>
<tr><td>★</td><td>Page 3,
BLOCK DIAGRAM,
<b>Note:</b></td><td><b>Note:</b>
<table>
<tr><th></th><th>Flash memory</th><th>RAM</th></tr>
<tr><td>M37902F8CGP,M37902F8CHP</td><td>60 Kbytes</td><td>2048 bytes</td></tr>
<tr><td>M37902FCCGP,M37902FCCHP</td><td>120 Kbytes</td><td>4096 bytes</td></tr>
<tr><td>M37902FECGP,M37902FECHP</td><td>184 Kbytes</td><td>6144 bytes</td></tr>
<tr><td>M37902FGCGP,M37902FGCHP</td><td>248 Kbytes</td><td>6144 bytes</td></tr>
<tr><td>M37902FHCGP,M37902FHCHP</td><td>370 Kbytes</td><td>12288 bytes</td></tr>
<tr><td>M37902FJCGP,M37902FJCHP</td><td>498 Kbytes</td><td>12288 bytes</td></tr>
</table></td><td><b>Note:</b>
<table>
<tr><th></th><th>Flash memory</th><th>RAM</th></tr>
<tr><td>M37902FCCHP</td><td>120 Kbytes</td><td>4096 bytes</td></tr>
<tr><td>M37902FGCHP</td><td>248 Kbytes</td><td>6144 bytes</td></tr>
<tr><td>M37902FJCHP</td><td>498 Kbytes</td><td>12288 bytes</td></tr>
</table></td></tr>
<tr><td></td><td>Page 4,
Chip-select wait control</td><td>Chip select area × 4 ($\overline{\text{CS}_0}$–$\overline{\text{CS}_3}$). A wait number and bus width can be set for each chip select area.</td><td>Chip select area × 4 ($\overline{\text{CS}_0}$–$\overline{\text{CS}_3}$). A bus cycle type and bus width can be set for each chip select area.</td></tr>
</table>

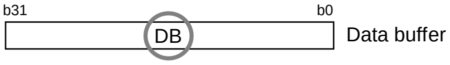
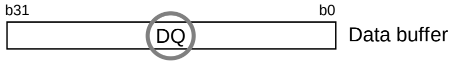
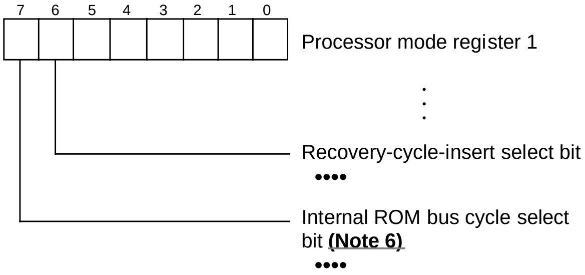
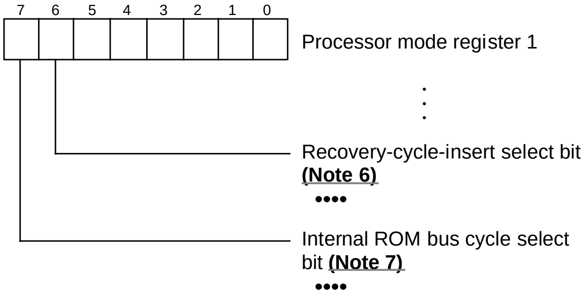
Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.2

<table>
<tr><th></th><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td>★</td><td>Page 4,
Parameter</td><td><table><tr><td>Operating temperature range</td><td>••••••</td></tr></table></td><td><table><tr><td>Operating ambient temperature range</td><td>•••••</td></tr></table></td></tr>
<tr><td>★</td><td>Page 4,
<b>Note:</b></td><td><b>Note:</b><table>
<tr><td>Flash memory (User ROM area)</td><td>M37902F8CGP,M37902F8CHP</td><td>60 Kbytes</td></tr>
<tr><td></td><td>M37902FCCGP,M37902FCCHP</td><td>120 Kbytes</td></tr>
<tr><td></td><td>M37902FECGP,M37902FECHP</td><td>184 Kbytes</td></tr>
<tr><td></td><td>M37902FGCGP,M37902FGCHP</td><td>248 Kbytes</td></tr>
<tr><td></td><td>M37902FHCGP,M37902FHCHP</td><td>370 Kbytes</td></tr>
<tr><td></td><td>M37902FJCGP,M37902FJCHP</td><td>498 Kbytes</td></tr>
<tr><td>RAM</td><td>M37902F8CGP,M37902F8CHP</td><td>2048 bytes</td></tr>
<tr><td></td><td>M37902FCCGP,M37902FCCHP</td><td>4096 bytes</td></tr>
<tr><td></td><td>M37902FECGP,M37902FECHP</td><td>6144 bytes</td></tr>
<tr><td></td><td>M37902FGCGP,M37902FGCHP</td><td>6144 bytes</td></tr>
<tr><td></td><td>M37902FHCGP,M37902FHCHP</td><td>12288 bytes</td></tr>
<tr><td></td><td>M37902FJCGP,M37902FJCHP</td><td>12288 bytes</td></tr>
</table></td><td><b>Note:</b><table>
<tr><td>Flash memory (User ROM area)</td><td>M37902FCCHP</td><td>120 Kbytes</td></tr>
<tr><td></td><td>M37902FGCHP</td><td>248 Kbytes</td></tr>
<tr><td></td><td>M37902FJCHP</td><td>498 Kbytes</td></tr>
<tr><td>RAM</td><td>M37902FCCHP</td><td>4096 bytes</td></tr>
<tr><td></td><td>M37902FGCHP</td><td>6144 bytes</td></tr>
<tr><td></td><td>M37902FJCHP</td><td>12288 bytes</td></tr>
</table></td></tr>
<tr><td>★</td><td>Page 5,
<b>Notes 1:</b></td><td><table>
<tr><td>User ROM area</td><td>M37902F8CGP, M37902F8CHP</td><td>4 blocks ••••••</td></tr>
<tr><td></td><td>M37902FCCGP, M37902FCCHP</td><td>5 blocks ••••••</td></tr>
<tr><td></td><td>M37902FECGP, M37902FECHP</td><td>6 blocks ••••••</td></tr>
<tr><td></td><td>M37902FGCGP, M37902FGCHP</td><td>7 blocks ••••••</td></tr>
<tr><td></td><td>M37902FHCGP, M37902FHCHP</td><td>9 blocks ••••••</td></tr>
<tr><td></td><td>M37902FJCGP, M37902FJCHP</td><td>11 blocks •••••</td></tr>
</table></td><td><table>
<tr><td>User ROM area</td><td>M37902FCCHP</td><td>5 blocks ••••••</td></tr>
<tr><td></td><td>M37902FGCHP</td><td>7 blocks••••••</td></tr>
<tr><td></td><td>M37902FJCHP</td><td>11 blocks •••••</td></tr>
</table></td></tr>
<tr><td>★</td><td>Page 6,
P4$_0$–P4$_7$</td><td>•••••••••••••••
■ In microprocessor mode
••••••••. According to the register setting, P4$_0$–P4$_4$ also ••••••••••</td><td>•••••••••••••••
■ In microprocessor mode
••••••••. According to the register setting, P4$_0$–P4$_3$ also ••••••••••</td></tr>
<tr><td></td><td>Page 9,
MEMORY,
Line 2</td><td>•••••••••. The address space is 16 Mbytes from address 0$_{16}$ to FFFFFF$_{16}$. •••••••••</td><td>•••••••••. The address space is 16 Mbytes from addresses 0$_{16}$ to FFFFFF$_{16}$. •••••••••</td></tr>
<tr><td>★</td><td>———</td><td>Memory map of M37902F8CGP and M37902F8CHP (Single-chip mode)</td><td>(Deleted)</td></tr>
<tr><td></td><td>★
Page 9,
Fig. 1</td><td>Fig. 2. Memory map of M37902FCCGP and M37902FCCHP (Single-chip mode)
 </td><td>Fig. 1. Memory map of M37902FCCHP (Single-chip mode)
 </td></tr>
<tr><td>★</td><td>———</td><td>Memory map of M37902FECGP and M37902FECHP (Single-chip mode)</td><td>(Deleted)</td></tr>
<tr><td></td><td></td><td>Memory map of M37902FHCGP and M37902FHCHP (Single-chip mode)</td><td>(Deleted)</td></tr>
</table>

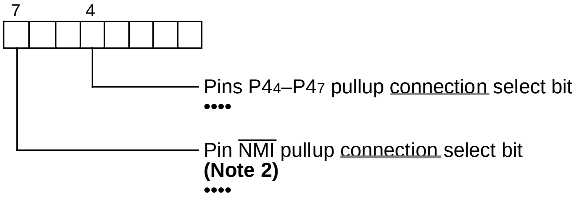
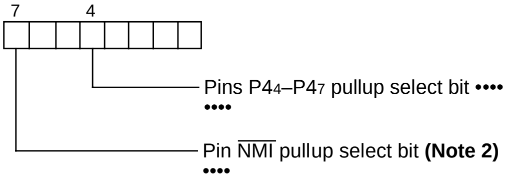
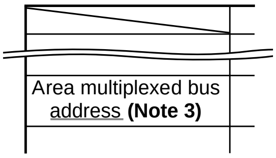
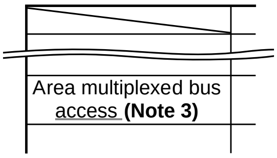
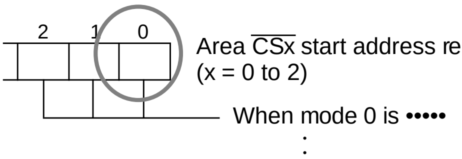
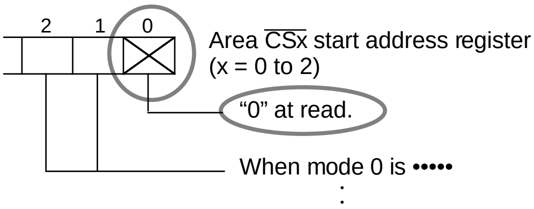
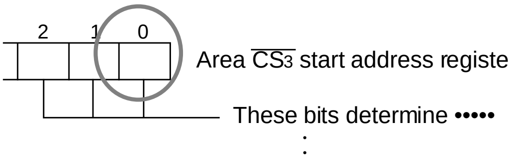
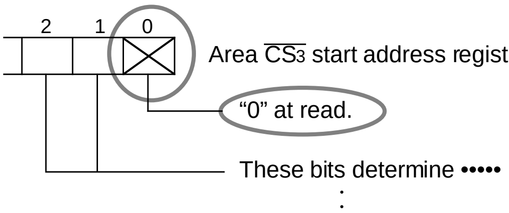
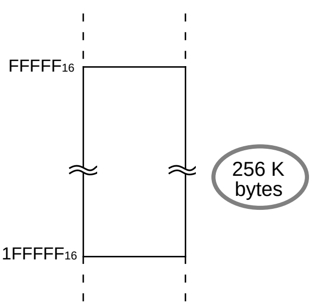
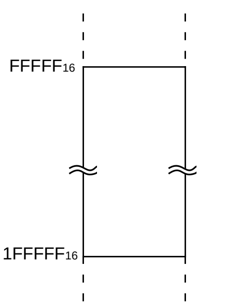
Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.3

Page	Error	Correction
★ Page 10, Fig. 2	Fig. 4. Memory map of M37902FGCGP and M37902FGCHP (Single-chip mode)  003800_{16} $00FFC0_{16}$ $00FFFF_{16}$ Internal flash memory 248 Kbytes (User ROM area)	Fig. 2. Memory map of M37902FGCHP (Single-chip mode)  003800_{16} $00FFC0_{16}$ $00FFFF_{16}$ Internal flash memory 248 Kbytes (User ROM area)
★ Page 10, Fig. 3	Fig. 6. Memory map of M37902FJCGP and M37902FJCHP (Single-chip mode)  003800_{16} $00FFC0_{16}$ $00FFFF_{16}$ Internal flash memory 498 Kbytes (User ROM area)	Fig. 3. Memory map of M37902FJCHP (Single-chip mode)  003800_{16} $00FFC0_{16}$ $00FFFF_{16}$ Internal flash memory 498 Kbytes (User ROM area)
Page 11, Fig. 7 ★ address 00_{16}, 01_{16}		 000000_{16} Reserved area **(Note)** 000001_{16} Reserved area **(Note)**
address 19_{16}		
Page 12, Fig. 8 ★ address $A6_{16}$		
address AC_{16}, AD_{16}		 $0000AC_{16}$ Serial I/O pin control register $0000AD_{16}$

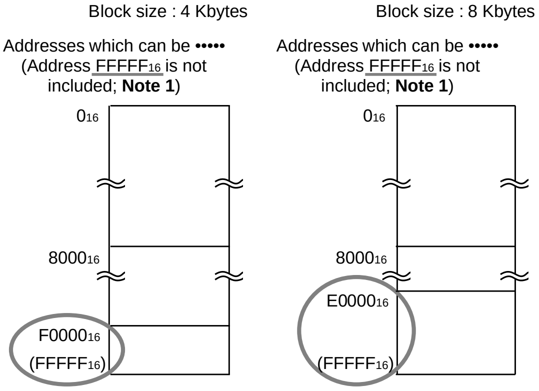
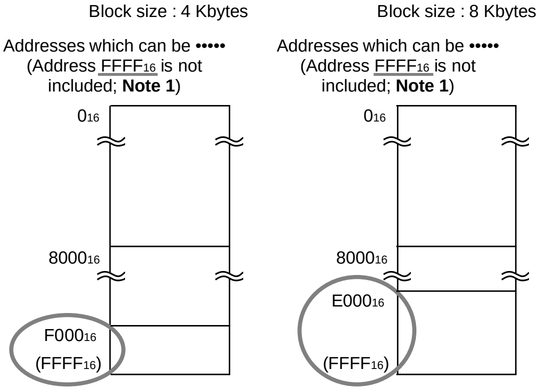
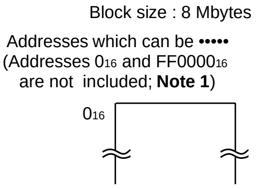
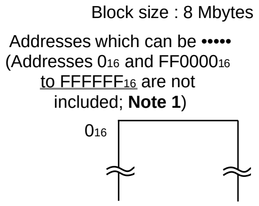
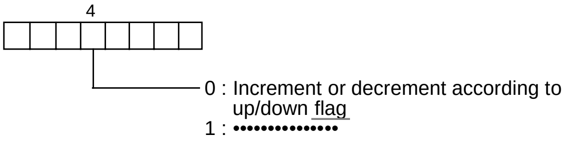
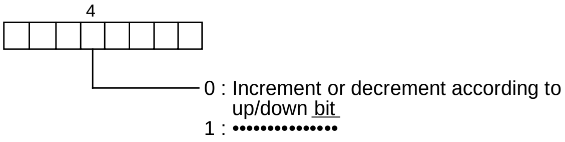
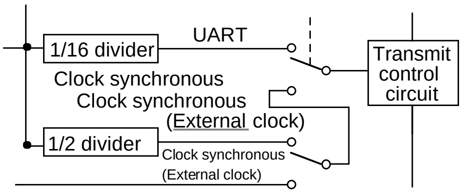
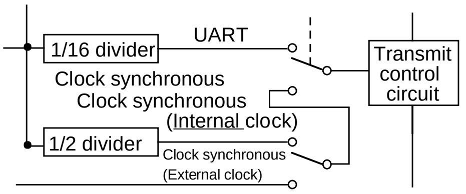
Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.4

	Page	Error	Correction
★	Page 18, Table 1 Instruction queue buffer	Temporarity stores an instruction which ••••••••.	Temporarily stores an instruction which ••••••••.
	Data buffer	Temporarity stores data which has been ••••••••, and external areas by the BIU or which is to be writeen to internal memory, •••••••• .	Temporarily stores data which has been ••••••••, and external areas by the BIU; or temporarily stores data which is to be written to internal memory, •••••••• .
★	Page 18, Fig. 11		
	Page 26, Fig. 18, **Notes 1**	**Notes 1:** The number of bus cycles is determined by the following bits: ••••••••••••	**Notes 1:** The bus cycle type is determined by the following bits: ••••••••••••
	Page 31, Right column Line 5	••••••••. Therefore, ports P0 or P4, P10, P11 function as I/O pins for the address bus, ••••••••	••••••••. Therefore, ports P0 to P4, P10, P11 function as I/O pins for the address bus, ••••••••
	Page 33, Table 5	Mode **(Note 1)** | Pin MD0 Processor mode **(Note 2)**	Mode **(Note 1)** | Pin MD0 Processor mode bits **(Note 2)**
★	Page 35, Fig. 24, **Note**	**Notes 1:** While VSS •••••, bit 1 is cleared to “0”. While VCC •••••, bit 1 is set to “1” at reset. (Fixed to “1”.)	**Notes 1:** While VSS •••••, this bit’s state is cleared to “0” at reset. While VCC •••••, this bit’s state is set to “1” at reset. (Fixed to “1”.)
		3: While VSS •••••, bit 7 is cleared to “0”. While VCC •••••, bit 7 is set to “1” at reset.	**3:** While VSS •••••, this bit’s state is cleared to “0” at reset. While VCC •••••, this bit’s state is set to “1” at reset.
★	Page 36, Fig. 25	 7 6 5 4 3 2 1 0 Processor mode register 1 Recovery-cycle-insert select bit •••• Internal ROM bus cycle select bit **(Note 6)** ••••	 7 6 5 4 3 2 1 0 Processor mode register 1 Recovery-cycle-insert select bit **(Note 6)** •••• Internal ROM bus cycle select bit **(Note 7)** ••••
		2: After reset, this bit can be set only once. During the software execution, be sure not to change this bit.	**2:** After reset, this bit’s contents can be switched only once. During the software execution, be sure not to switch this bit’s contents.
		4: While VSS •••••, these bits are cleared to “0”. While VCC •••••, on the other hand, these bits are set to “1”.	**4:** While VSS •••••, each of these bits is “0” at reset. While VCC •••••, on the other hand, each of these bits is “1” at reset.
		5: After reset, these bits can be set to “1” only once. Once these bits have been cleared to “0” from “1”, they cannot be set to “1” again. (Fixed to “0”.)	**5:** In the memory expansion or microprocessor mode, if this bit’s contents is switched from “1” to “0”, this bit will be cleared to “0”. After this clearance, this bit cannot return to “1”. If it is necessary to set this bit to “1”, be sure to reset the microcomputer.

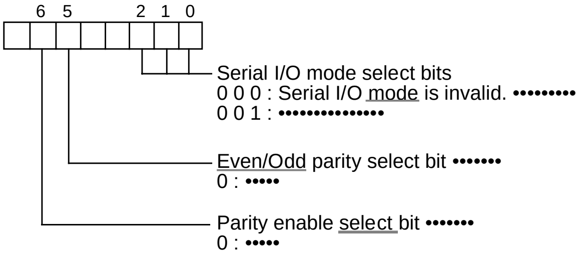
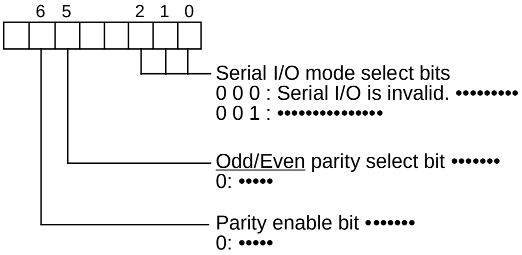
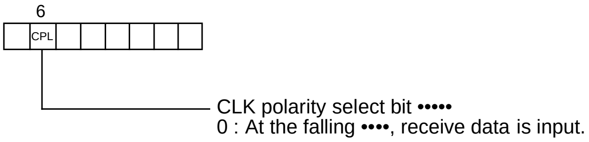
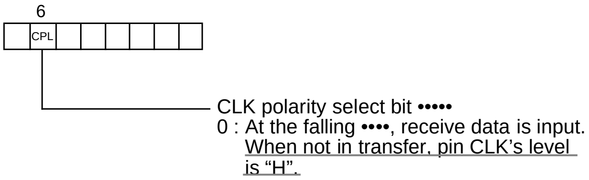
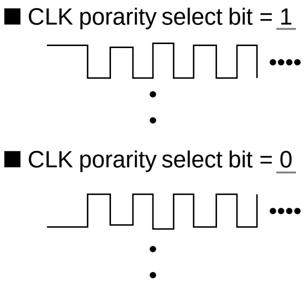
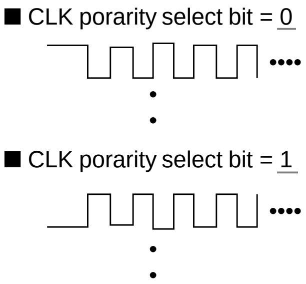
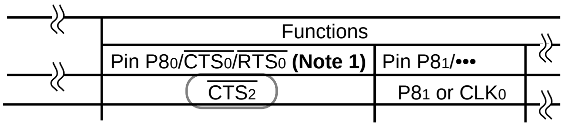
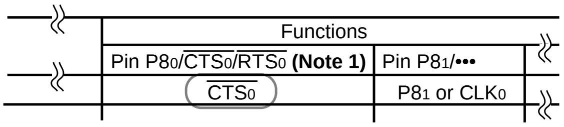
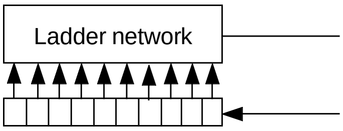
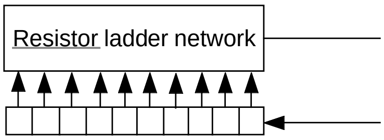
Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.5

Page	Error	Correction
★ Page 36, Fig. 25	———————	**6:** The program which switches this bit's contents must be assigned to the internal area.
	6: In the microprocessor mode, this bit is invalid.	**7:** In the microprocessor mode, this bit is invalid. When the internal flash memory is reprogrammed in the CPU reprogramming mode, be sure to clear this bit to "0".
Page 37, Fig. 26, bit 4	 7 4 Pins $P4_4$–$P4_7$ pullup connection select bit •••• Pin $\overline{NMI}$ pullup connection select bit **(Note 2)** ••••	 7 4 Pins $P4_4$–$P4_7$ pullup select bit •••• •••• Pin $\overline{NMI}$ pullup select bit **(Note 2)** ••••
Page 39, Table 7	 Area multiplexed bus address **(Note 3)**	 Area multiplexed bus access **(Note 3)**
★ Page 40, Fig. 28, $\overline{CS_0}$ control register L; **Note**	**Notes 1:** While V_{SS} •••••, this bit is cleared to "0". While V_{CC} •••••, this bit is set to "1" at reset.	**Notes 1:** While V_{SS} •••••, this bit's state is cleared to "0" at reset. While V_{CC} •••••, this bit's state is set to "1" at reset.
	5: While V_{SS} •••••, this bit is cleared to "0". While V_{CC} •••••, this bit is set to "1" at reset. (Fixed to "1".)	**5:** While V_{SS} •••••, this bit's state is cleared to "0" at reset. While V_{CC} •••••, this bit's state is set to "1" at reset. (Fixed to "1".)
★ Page 42, Fig. 30, Area $\overline{CS}$x start address register (x = 0 to 3)	 2 1 0 Area $\overline{CS}$x start address register (x = 0 to 2) When mode 0 is ••••• :	 2 1 0 Area $\overline{CS}$x start address register (x = 0 to 2) "0" at read. When mode 0 is ••••• :
	 2 1 0 Area $\overline{CS_3}$ start address register These bits determine ••••• :	 2 1 0 Area $\overline{CS_3}$ start address register "0" at read. These bits determine ••••• :
Page 43, Fig. 31	Start address : 4000_{16}  $FFFFF_{16}$ $1FFFFF_{16}$ 256 K bytes	Start address : 4000_{16}  $FFFFF_{16}$ $1FFFFF_{16}$

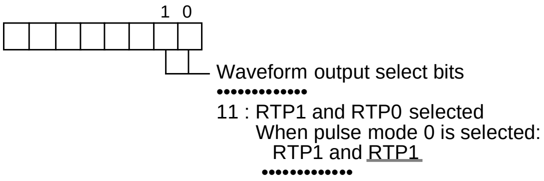
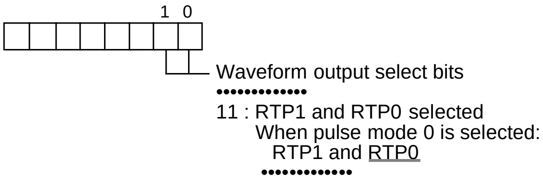
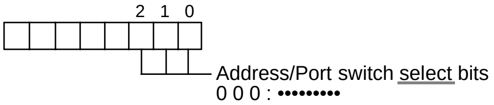
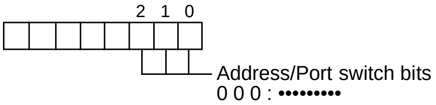
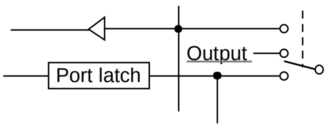
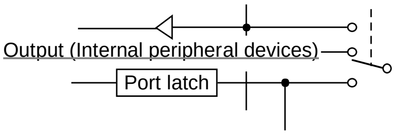
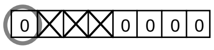
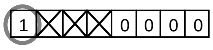
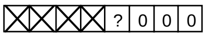
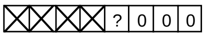
Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.6

Page	Error	Correction
Page 44, Fig. 32	 Block size : 4 Kbytes Addresses which can be ••••• (Address FFFFF$_{16}$ is not included; **Note 1**) 0$_{16}$ 8000$_{16}$ F0000$_{16}$ (FFFFF$_{16}$) Block size : 8 Kbytes Addresses which can be ••••• (Address FFFFF$_{16}$ is not included; **Note 1**) 0$_{16}$ 8000$_{16}$ E0000$_{16}$ (FFFFF$_{16}$)	 Block size : 4 Kbytes Addresses which can be ••••• (Address FFFF$_{16}$ is not included; **Note 1**) 0$_{16}$ 8000$_{16}$ F000$_{16}$ (FFFF$_{16}$) Block size : 8 Kbytes Addresses which can be ••••• (Address FFFF$_{16}$ is not included; **Note 1**) 0$_{16}$ 8000$_{16}$ E000$_{16}$ (FFFF$_{16}$)
Page 45, Fig. 33	 Block size : 8 Mbytes Addresses which can be ••••• (Addresses 0$_{16}$ and FF0000$_{16}$ are not included; **Note 1**) 0$_{16}$	 Block size : 8 Mbytes Addresses which can be ••••• (Addresses 0$_{16}$ and FF0000$_{16}$ to FFFFFF$_{16}$ are not included; **Note 1**) 0$_{16}$
Page 45, Fig. 33, title	Fig. 33 Area $\overline{CS_0}$/$\overline{CS_1}$/$\overline{CS_2}$ (mode 1) and area $\overline{CS_3}$	Fig. 33 Area $\overline{CS_0}$/$\overline{CS_1}$/$\overline{CS_2}$ (mode 0) and area $\overline{CS_3}$
Page 46, Left column, Line 2	Table 8 shows the interrupt types and the ••••••••	Table 8 shows the interrupt sources and the ••••••••
Page 46, Table 8, title	Table 8. Interrupt types and the interrupt vector addresses	Table 8. Interrupt sources and interrupt vector addresses
Page 52, Left column, Line 5	To use these pins as timer input pins, the data direction register •••••	To use these pins as timer input pins, the port direction register •••••
Page 55, Fig. 46, bit 4	 4 0 : Increment or decrement according to up/down flag 1 : •••••••••••••••	 4 0 : Increment or decrement according to up/down bit 1 : •••••••••••••••
★ Page 59, Left column, Lines 14, 17, 22	••••• timer Ai start bit •••••	••••• count start bit •••••
★ Page 62, (2) Event counter mode [01]	(Line 15) particular function select register 1 (bit 7 at address 63$_{16}$) •••••	(Line 15) particular function select register 1 (bit 6 at address 63$_{16}$) •••••
Page 64, Fig. 64	 1/16 divider UART Transmit control circuit Clock synchronous Clock synchronous (External clock) 1/2 divider Clock synchronous (External clock)	 1/16 divider UART Transmit control circuit Clock synchronous Clock synchronous (Internal clock) 1/2 divider Clock synchronous (External clock)

Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.7

Page	Error	Correction
Page 64, Fig. 65, bits 2 to 0, bit 5, bit 6	 Serial I/O mode select bits 0 0 0 : Serial I/O mode is invalid. ••••••••• 0 0 1 : ••••••••••••••• Even/Odd parity select bit ••••••• 0 : ••••• Parity enable select bit ••••••• 0 : •••••	 Serial I/O mode select bits 0 0 0 : Serial I/O is invalid. ••••••••• 0 0 1 : ••••••••••••••• Odd/Even parity select bit ••••••• 0: ••••• Parity enable bit ••••••• 0: •••••
Page 66, Fig. 68, UART0/1 transmit/receive control register 0, bit 6	 CLK polarity select bit ••••• 0 : At the falling ••••, receive data is input.	 CLK polarity select bit ••••• 0 : At the falling ••••, receive data is input. When not in transfer, pin CLK's level is "H".
Page 68, Left column, Last line	read out the RTS$_k$ output turns back to "L". •••••••	read out the $\overline{\text{RTS}}_k$ output turns back to "L". •••••••
Page 70, Fig. 71	 ■ CLK porarity select bit = 1 ■ CLK porarity select bit = 0	 ■ CLK porarity select bit = 0 ■ CLK porarity select bit = 1
★ Page 74, Table 13	 Functions Pin P8$_0$/$\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$ **(Note 1)** | Pin P8$_1$/••• $\overline{\text{CTS}_2}$ | P8$_1$ or CLK$_0$	 Functions Pin P8$_0$/$\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$ **(Note 1)** | Pin P8$_1$/••• $\overline{\text{CTS}_0}$ | P8$_1$ or CLK$_0$
★ Page 76, Fig. 77	 Ladder network	 Resistor ladder network
Page 77, Left column, Line 12	AD$_{TRG}$ input changes from "H" to "L" (or "L" to "H".) ••••••	$\overline{\text{AD}}_{TRG}$ input changes from "H" to "L" (or "L" to "H".) ••••••
Page 77, Left column, Line 14	AD$_{TRG}$ pin is multiplexed with an analog voltage ••••••	$\overline{\text{AD}}_{TRG}$ pin is multiplexed with an analog voltage ••••••
★ Page 77, Right column, Lines 5 to 6	••••••• the ladder network or not •••••••••••	••••••• the resistor ladder network or not •••••••••••
★ Page 77, Right column, Line 10	the ladder network can be cut off by disconnecting ladder network •••••••	the resistor ladder network can be cut off by disconnecting resistor ladder network •••••••
★ Page 80, Left column, Line 10	•••••••, the corresponding pin (DA$_0$ to DA$_2$) outputs •••••••	•••••••, the corresponding pin (D-A$_0$ to D-A$_2$) outputs •••••••

Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.8

Page	Error	Correction
Page 80, Left column, Line 15	The D-A output enable bit is cleared to“0” at reset.•••••••	The contents of the corresponding D-A output enable bit and D-A register are cleared to “0” at reset.•••••••
Page 80, Right column, Line 1	with pin D-Ai.	with pin D-Ai. Also, when not using the D-A converter, be sure to clear the contents of the corresponding D-A output enable bit and D-A register to “0”.
Page 81, Fig. 83, bits 1 and 0	 1 0 Waveform output select bits ••••••••••••• 11 : RTP1 and RTP0 selected When pulse mode 0 is selected: RTP1 and RTP1 •••••••••••••	 1 0 Waveform output select bits ••••••••••••• 11 : RTP1 and RTP0 selected When pulse mode 0 is selected: RTP1 and RTP0 •••••••••••••
★ Page 82, Right column, Lines 1 to 3	When the waveform output select bits are set to “11” (bit 1 = bit 0 = “1”), $RTP1_3$ to $RTP1_0$ and $RTP0_3$ to $RTP0_0$ become pulse output port pins. When the waveform output •••••••••	When the waveform output select bits are set to “11” (bit 1 = bit 0 = “1”), pulse output port pins are divided into two groups; one consists of $RTP1_3$ to $RTP1_0$, $RTP0_3$, $RTP0_2$ and the other consists of $RTP0_1$ and $RTP0_0$. When the waveform output •••••••••
Page 86, Fig. 91, bits 2 to 0	 2 1 0 Address/Port switch select bits 0 0 0 : •••••••••	
Page 87, Fig. 89, 2nd diagram	[Inside dotted-line not included] $P4_0$/ALE, $P4_1$/ϕ_1, $P4_2$/$\overline{HLDA}$, •••••••••••	[Inside dotted-line not included] $P4_0$/ALE, $P4_1$/ϕ_1, $P4_2$/$\overline{HLDA}$, •••••••••••
		
Page 88, Fig. 90, 3rd diagram	[Inside dotted-line not included] $P7_7$/AN_7/$\overline{AD}_{TRG}$/DA_1/($\overline{INT_2}$)	[Inside dotted-line included] $P7_7$/AN_7/$\overline{AD}_{TRG}$/DA_1/($\overline{INT_2}$)
★ Page 89, Fig. 93, address 81_{16}	$\overline{CS_0}$ control register H (81_{16}) •••• 	$\overline{CS_0}$ control register H (81_{16}) •••• 
Page 90, Fig. 92, address 70_{16}	A-D interrupt control register (70_{16}) •••• 	A-D conversion interrupt control register (70_{16}) ••• 
★ Page 91, Left column, Lines 11, 12	•••••••, the oscillation circuit stops it’s operation and resumes the current dissipation.	•••••••, the oscillation circuit stops it’s operation, and the current dissipation is reduced.
★ Page 91, Left column, Line 17	••••••• from pin X_{IN} and output a multiplied clock.	••••••• from pin X_{IN} and generates a multiplied clock.
Page 92, Right column, Lines 4 to 5	••••••• In this selection, be sure that multiplied f(X_{IN}) does not exceed 26 MHz. •••••••••	••••••• The PLL multiplication ratio must be set so that the frequency of the PLL output clock (f_{PLL}) must be in the range from 10 MHz to 26 MHz. ••••••••
Page 92, Right column, Lines 10 to 11	••••• the PLL output clock (f_{PLL}). (In other words, set bit 5 to “1”.)	••••• the PLL output clock (f_{PLL}). (In other words, set bit 5 to “1”.) Note that, after reset, the PLL multiplication ratio select bits are allowed to be changed only once.

Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.9

	Page	Error	Correction
	Page 92, Table 15, **Note**	**Note:** Be sure that system clock f_{sys} does not exceed 26 MHz. f(X_{IN}) means the frequency of the input clock from pin X_{IN} (fX_{IN}).	**Note:** The PLL multiplication ratio must be set so that the frequency of the PLL output clock (f_{PLL}) must be in the range from 10 MHz to 26 MHz. f(X_{IN}) means the frequency of the input clock from pin X_{IN} (fX_{IN}). After reset, the PLL multiplication ratio select bits are allowed to be changed only once.
★	Page 96, Left column, Line 7	••••• the oscillation circuit and PLL circuit have been restarted ••••••	••••• the oscillation circuit has been restarted ••••••
★	Page 98, Right column, Line 8	the ladder network of the A-D converter will •••••	the resistor ladder network of the A-D converter will •••••
★	Page 98, Right column, Line 10	pin V_{REF} to the ladder network, and the power dissipation •••••	pin V_{REF} to the resistor ladder network, and the power dissipation •••••
★	Page 101, Fig. 106	Fig. 106. M37902FJCGP, M37902FJCHP : block configuration of internal flash memory	Fig. 106. M37902FJCHP : block configuration of internal flash memory
★	———	M37902F8CGP, M37902F8CHP : block configuration of internal flash memory	(Deleted)
		M37902FECGP, M37902FECHP : block configuration of internal flash memory	(Deleted)
★	Page 102, Fig. 107	Fig. 108. M37902FCCGP, M37902FCCHP : block configuration of internal flash memory	Fig. 107. M37902FCCHP : block configuration of internal flash memory
★	Page 102, Fig. 108	Fig. 110. M37902FGCGP, M37902FGCHP : block configuration of internal flash memory	Fig. 108. M37902FGCHP : block configuration of internal flash memory
★	———	M37902FHCGP, M37902FHCHP : block configuration of internal flash memory	(Deleted)
★	Page 103, Right column, Lines 15 to 17	area if the user uses the flash memory serial I/O mode. Note that, when the boot ROM area is read •••••••••	area if the user uses the flash memory serial I/O mode. Addresses $FFB0_{16}$ to $FFBF_{16}$ are the reserved area for the serial programmer. Therefore, when the user uses the flash memory serial I/O mode, do not program to this area. Note that, when the boot ROM area is read •••••••••
★	———	Pin connection of M37902FxCGP in flash memory serial I/O mode	(Deleted)
★	Page 106, Right column, After line 13	program the user ROM area.	program the user ROM area. After reset removal, be sure not to change the status at pins MD0 and MD1.
★	Page 106, Fig. 114, Notes 4	**4:** Valid only ••••• clear to “0”.	**4:** Valid only ••••• clear to “0”. This bit 3 must be controlled with bit 1 = “1”.
★	Page 107, Left column, Lines 16 to 20	••••• command consists of 8-bit units must be written only to an even address; therefore, any data written to an odd address will be invalid. The write state ••••••	••••• command consisting of 8 bits must be written to an even address; therefore, any command written to an odd address will be invalid. Since the write data at the 2nd cycle of a programming command consists of 16 bits, this data must be written to even and odd addresses. The write state ••••••
★	Page 107, Right column, After line 24	request occurrence.	request occurrence. In the CPU reprogramming mode, be sure not to use the **STP** and **WIT** instructions.

Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.10

<table>
<tr><th></th><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td>★</td><td>Page 108,
Fig. 115</td><td>↓
The CPU reprogramming mode select bit is set to “1”.
(Writing of “0” → Writing of “1”)
↓</td><td>↓
Writing of “1” to the CPU reprogramming mode select bit.
(Writing of “0” → Writing of “1”)
↓</td></tr>
<tr><td>★</td><td>Page 108,
Software
Commands</td><td>(Lines 6, 7)
••••••• (D_8–D_{15}) is ignored.</td><td>(Lines 6, 7)
••••••• (D_8–D_{15}) is ignored. (Except for the write data at the 2nd cycle of a page programming command.)</td></tr>
<tr><td>★</td><td rowspan="2">Page 108,
Page
programming
Command</td><td>(Titlle)
Page Program Command (41_{16})</td><td>(Titlle)
Page Programming Command (41_{16})</td></tr>
<tr><td></td><td>(After line 20)
••••••• mode is maintained.</td><td>(After line 20)
••••••• mode is maintained. In continuous programming, if there is no programming error, page programming commands can be executed with the read status register mode kept.</td></tr>
<tr><td>★</td><td>Page 109,
Page
programming
Command</td><td>(Lines 4 to 7)
•••••••, the same way as bit 7 of the status register.
Reading out the •••••••</td><td>(Lines 4 to 7)
•••••••, the same wayas bit 7 of the status register.
Before execution of the next command, be sure to verify that bit 7 of the status register (SR.7) or the RY/$\overline{BY}$ status bit is set to “1” (READY). During the automatic programming operation, writing of commands and access to the flash memory must not be performed.
Reading out the •••••••</td></tr>
<tr><td>★</td><td>Page 109,
Block Erase
Command</td><td>(Lines 20 to 23)
•••••••, the same way as bit 7 of the status register.
Reading out the •••••••</td><td>(Lines 20 to 23)
•••••••, the same way as bit 7 of the status register.
Before execution of the next command, be sure to verify that bit 7 of the status register (SR.7) or the RY/$\overline{BY}$ status bit is set to “1” (READY). During the automatic erase operation, writing of commands and access to the flash memory must not be performed.
Reading out the •••••••</td></tr>
<tr><td>★</td><td>Page 111,
Erase All
Unlocked Block
Command</td><td>(Lines 9 to 11)
••••••• is also reported by a read of the status register.
When the lock bit •••••••</td><td>(Lines 9 to 11)
••••••• is also reported by a read of the status register.
During the automatic erase operation (when the RY/$\overline{BY}$ status bit = “0”), writing of commands and access to the flash memory must not be performed.
When the lock bit •••••••</td></tr>
<tr><td>★</td><td>Page 111,
Data Protect
Function
(Block Lock)</td><td>(After line 20)
lock bit is terminated.</td><td>(After line 20)
lock bit is terminated.
To perform erase or programming, be sure to do one of the following.
• By executing the read lock bit status command, verify that the lock of the target block is invalid.
• Set the lock bit invalidity select bit to “1” to invalidate the lock.
When the block erase or programming is performed with the lock valid, the erase status bit (SR.5) and programming status bit (SR.4) are set to “1” (terminated by error).</td></tr>
<tr><td>★</td><td>Page 114,
ABSOLUTE
MAXIMUM
RATINGS</td><td><table><tr><th>Symbol</th><th>Parameter</th><th>Ratings</th><th>Unit</th></tr><tr><td>P_d</td><td>Power dispation</td><td>300</td><td>mW</td></tr><tr><td>T_{opr}</td><td>Operating temperature</td><td></td><td></td></tr></table></td><td><table><tr><th>Symbol</th><th>Parameter</th><th>Ratings</th><th>Unit</th></tr><tr><td>P_d</td><td>Power dispation</td><td>400</td><td>mW</td></tr><tr><td>T_{opr}</td><td>Operating ambient temperature</td><td></td><td></td></tr></table></td></tr>
</table>

Corrections and Supplementary Explanation for M37902FxC Datasheet (REV.B) NO.11

Page	Error	Correction
Page 114, RECOMMENDED OPERATING CONDITIONS	$f(f_{SYS})$ External clcok input frequency **(Note 2)** ••••••••	$f(X_{IN})$ External clcok input frequency **(Note 2)** ••••••••
★	$f(_{SYS})$ System clcok frequency ••••••••	$f(f_{SYS})$ System clcok frequency ••••••••
★ Page 123, Timing Requirements	$t_{w(harf)}$ External clock •••••••• ••••••	$t_{w(half)}$ External clock •••••••• ••••••
★ Page 131, PACKEGE OUTLINE	100P6S-A packege outline	(Deleted)