M37920FCCGP, M37920FCCHP
M37920FGCGP, M37920FGCHP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER FLASH MEMORY VERSION

DESCRIPTION

These are single-chip microcomputers designed with high-performance CMOS silicon gate technology, including the internal flash memory. These are housed in 100-pin plastic molded QFP. 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. These microcomputers include the 4-channel DMA controller and the DRAM controller. Therefore, these microcomputers are suitable for office, business, and industrial equipment controller that require fast 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
  [M37920FCCGP, M37920FCCHP]
  Flash memory (User ROM area) ..... 120 Kbytes
  RAM ..... 4096 bytes
  [M37920FGCGP, M37920FGCHP]
  Flash memory (User ROM area) ..... 248 Kbytes
  RAM ..... 6144 bytes
  [All of the above computers]
  Flash memory (Boot ROM area) ..... 16 Kbytes
- Instruction execution time
  The fastest instruction at 20 MHz frequency ..... 50 ns
- Single power supply ..... 5 V ± 0.5 V
- Interrupts ..... 6 external sources, 20 internal sources, 7 levels
- Multi-functional 16-bit timer ..... 5 + 3
- Serial I/O (UART or Clock synchronous) 2
- 10-bit A-D converter ..... 4-channel inputs
- DMA controller ..... 4 channels
- DRAM controller
- 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–P12) 85

<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

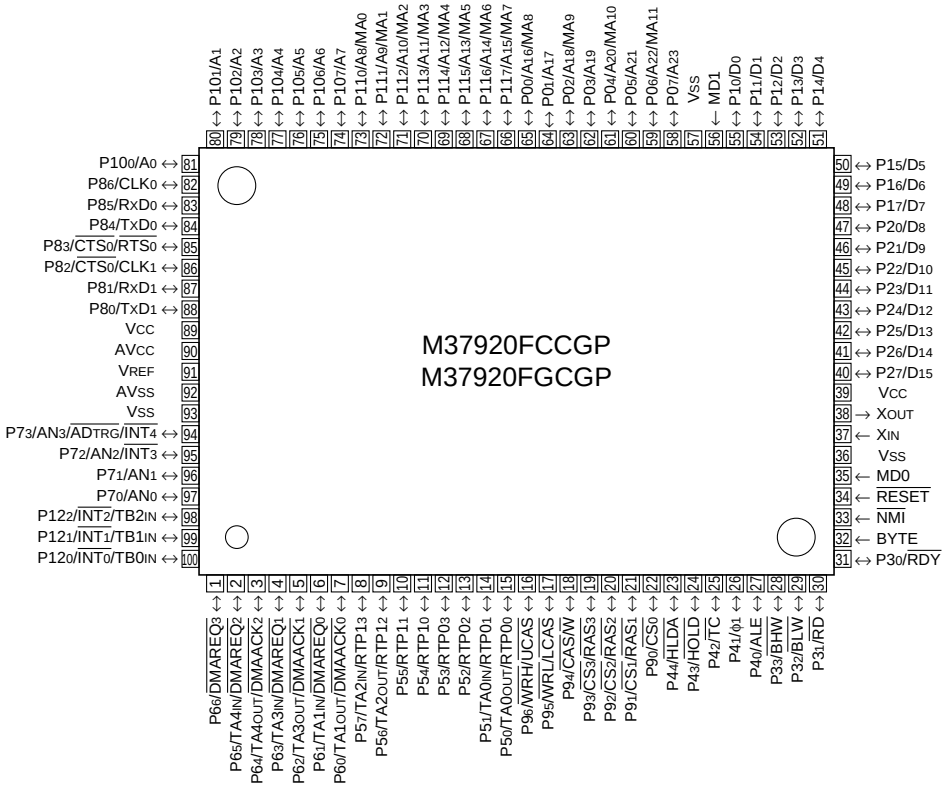
APPLICATION

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

Control devices for office equipment such as copiers and facsimiles

Control devices for industrial equipment such as communication and measuring instruments

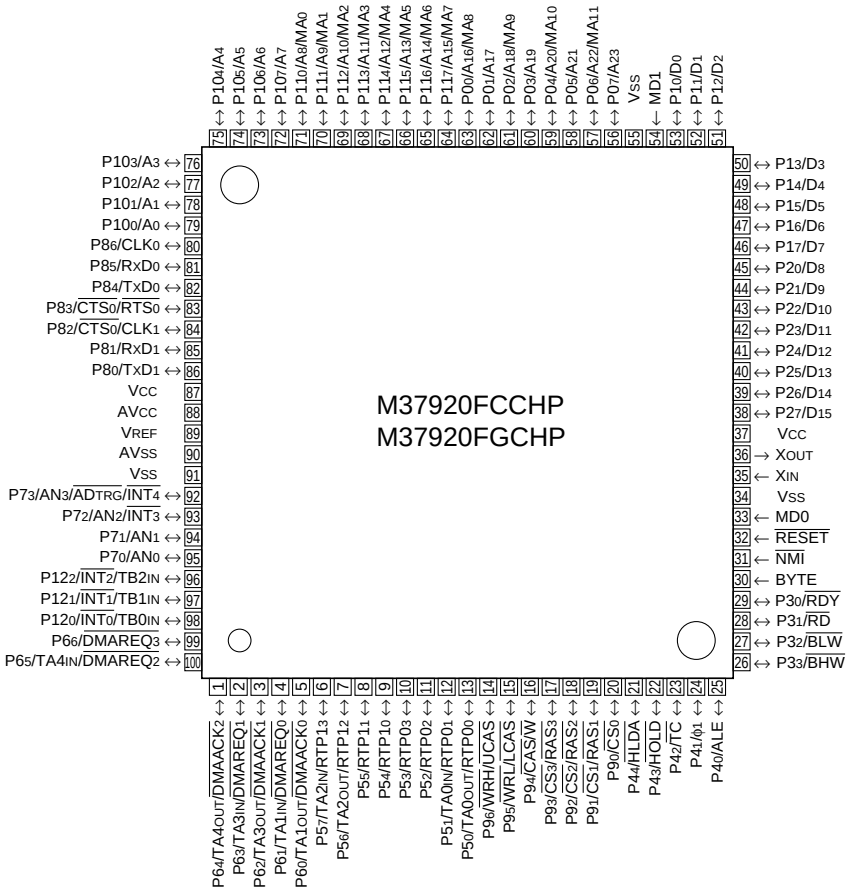
M37920FxCGP PIN CONFIGURATION (TOP VIEW)



Outline 100P6S-A

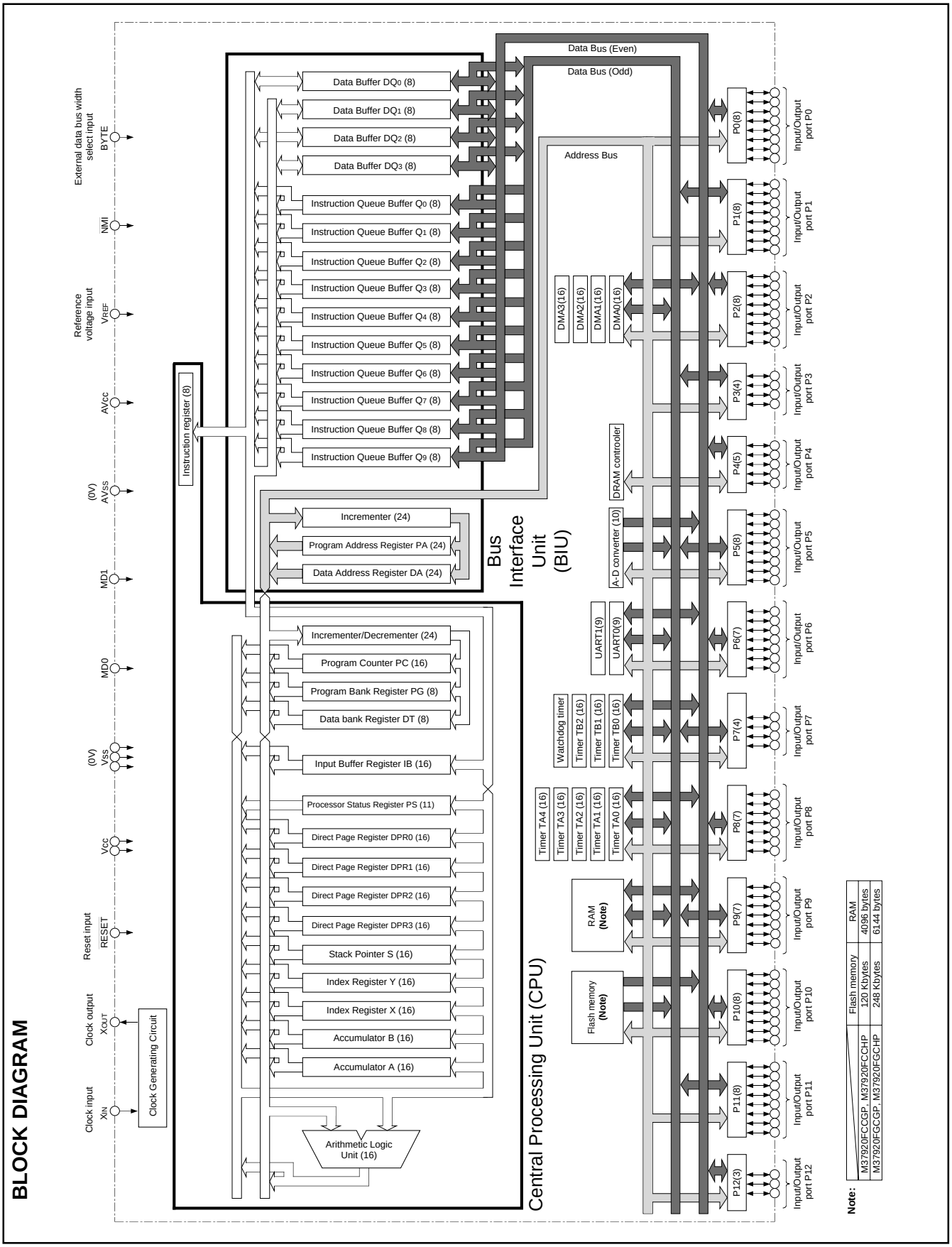
MITSUBISHI ELECTRIC

M37920FxCHP PIN CONFIGURATION (TOP VIEW)



Outline 100P6Q-A

BLOCK DIAGRAM



FUNCTIONS (Microcomputer mode)

Parameter		Functions
Number of basic machine instructions		203
Instruction execution time		50 ns (the fastest instruction at f(XIN) = 20 MHz)
External clock input frequency f(XIN)		20 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, P5, P10, P11	8-bit × 6
	P3, P7	4-bit × 2
	P4	5-bit × 1
	P6, P8, P9	7-bit × 3
	P12	3-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 (4 channels)
Watchdog timer		12-bit × 1
DMA controller		4 channels Maximum transfer rate 20 Mbytes/sec. (at f(XIN) = 20 MHz, 0 wait, 1-bus cycle transfer) 10 Mbytes/sec. (at f(XIN) = 20 MHz, 0 wait, 2-bus cycles transfer)
DRAM controller		1 channel Incorporates 8-bit refresh timer. Supports $\overline{CAS}$ before $\overline{RAS}$ refresh method or self refresh method.
Chip-select wait control		Chip select area × 4 ($\overline{CS0}$–$\overline{CS3}$). A wait number 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		6 external types, 20 internal types. Each interrupt except $\overline{NMI}$ can be set to a priority level within the range of 0–7 by software.
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz crystal resonator).
Power supply voltage		5 V±0.5 V
Power dissipation		125 mW (at f(XIN) = 20 MHz)
Ports' input/output characteristics	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Up to 16 Mbytes. Note that bank FF16 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)	M37920FCCGP, M37920FCCHP	120 Kbytes
	M37920FGCGP, M37920FGCHP	248 Kbytes
RAM	M37920FCCGP, M37920FCCHP	4096 bytes
	M37920FGCGP, M37920FGCHP	6144 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	M37920FCCGP, M37920FCCHP	5 blocks (8 Kbytes × 3, 32 Kbytes × 1, 64 Kbytes × 1), total 120 Kbytes
	M37920FGCGP, M37920FGCHP	7 blocks (8 Kbytes × 3, 32 Kbytes × 1, 64 Kbytes × 3), total 248 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 VSS for the single-chip mode or memory expansion mode, and VCC 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 applies to this pin.
XIN	Clock input	Input	These are input and output pins of the internal clock generating circuit. Connect a ceramic or quartz- crystal resonator between the XIN and XOUT pins. When an external clock is used, the clock source should be connected to the XIN pin, and the XOUT pin should be left open.
XOUT	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.
AVcc, AVss	Analog power supply input	—	Power supply input pin for the A-D converter. Connect AVcc to Vcc, and AVss to Vss externally.
VREF	Reference voltage input	Input	This is the reference voltage input pin for the A-D 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. In DRAM space is accessed, Multiplexed address (MA_8–MA_{11}) is output.
$P1_0$–$P1_7$	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.
$P2_0$–$P2_7$	I/O port P2	I/O	■ In single-chip mode or when 8-bit external data bus is used with “H” level applied to pin BYTE in memory expansion or microprocessor mode These pins have the same functions as port P0. ■ When the 16-bit external data bus is used with “L” level applied to pin BYTE 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. According to the register setting, this pin funtions as an output pin of $\overline{\text{RDY}}$. $P3_1$, $P3_2$, $P3_3$ funtion as output pins of $\overline{\text{RD}}$, $\overline{\text{BLW}}$, $\overline{\text{BHW}}$, respectively. ■ In microprocessor mode $P3_0$ functions as an input pin of $\overline{\text{RDY}}$; and $P3_1$, $P3_2$, $P3_3$ function as output pins of $\overline{\text{RD}}$, $\overline{\text{BLW}}$, $\overline{\text{BHW}}$, respectively.
$P4_0$–$P4_4$	I/O port P4	I/O	■ In single-chip mode These pins have the same functions as port P0. $P4_2$ also funtions as pin $\overline{\text{TC}}$. ■ In memory expansion mode $P4_0$–$P4_4$ function as I/O port pins. According to the register setting, these pins function as output pins or input pins of ALE, ϕ_1, $\overline{\text{TC}}$, $\overline{\text{HOLD}}$, $\overline{\text{HLDA}}$, respectively. ■ In microprocessor mode $P4_0$ and $P4_1$ function as outpout pins of ALE, ϕ_1. According to the register setting, these pins also funtion as I/O port pins. $P4_2$ funtions as an I/O port pin. According to the register setting, this pin also funtions as pin $\overline{\text{TC}}$. $P4_3$ functions as an input pin of $\overline{\text{HOLD}}$, and $P4_4$ functions as an output pin of $\overline{\text{HLDA}}$.

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, A2, and output pins for the real-time output.
$P6_0$–$P6_6$	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 timers A1, A3, A4, input pins for DMA requests, and output pins for DMA acknowledge signals.
$P7_0$–$P7_3$	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. $P7_2$ and $P7_3$ also function as input pins for $\overline{INT_3}$ and $\overline{INT_4}$.
$P8_0$–$P8_6$	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.
$P9_0$–$P9_6$	I/O port P9	I/O	■ In single-chip mode These pins have the same function as port P0. ■ In memory expansion or microprocessor mode According to the software setting, $P9_0$–$P9_3$ also funtion as chip select output pins. While DRAM space is selected, $P9_4$–$P9_6$ function as output pins for DRAM control signals. Some pins of $P9_1$–$P9_3$, coressponding to the selected DRAM space, function as pins $\overline{RAS}$.
$P10_0$–$P10_7$	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 or microprocessor mode Address (A_8–A_{15}) is output. While DRAM space is accessed, Multiplexed address (MA_0–MA_7) is output.
$P12_0$–$P12_2$	I/O port P12	I/O	In addition to having the same functions as port P0 in the single-ship mode, these pins also function as input pins for timers B0–B2.
$\overline{NMI}$	Non-maskable interrupt	Input	This pin is for a non-maskable interrupt.

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, I/O ports, clock generating circuit, etc.

MEMORY

Figures 1 and 2 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 and 2.

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, UART, timers, interrupt control registers, DMA controoler, DRAM controller, etc.

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

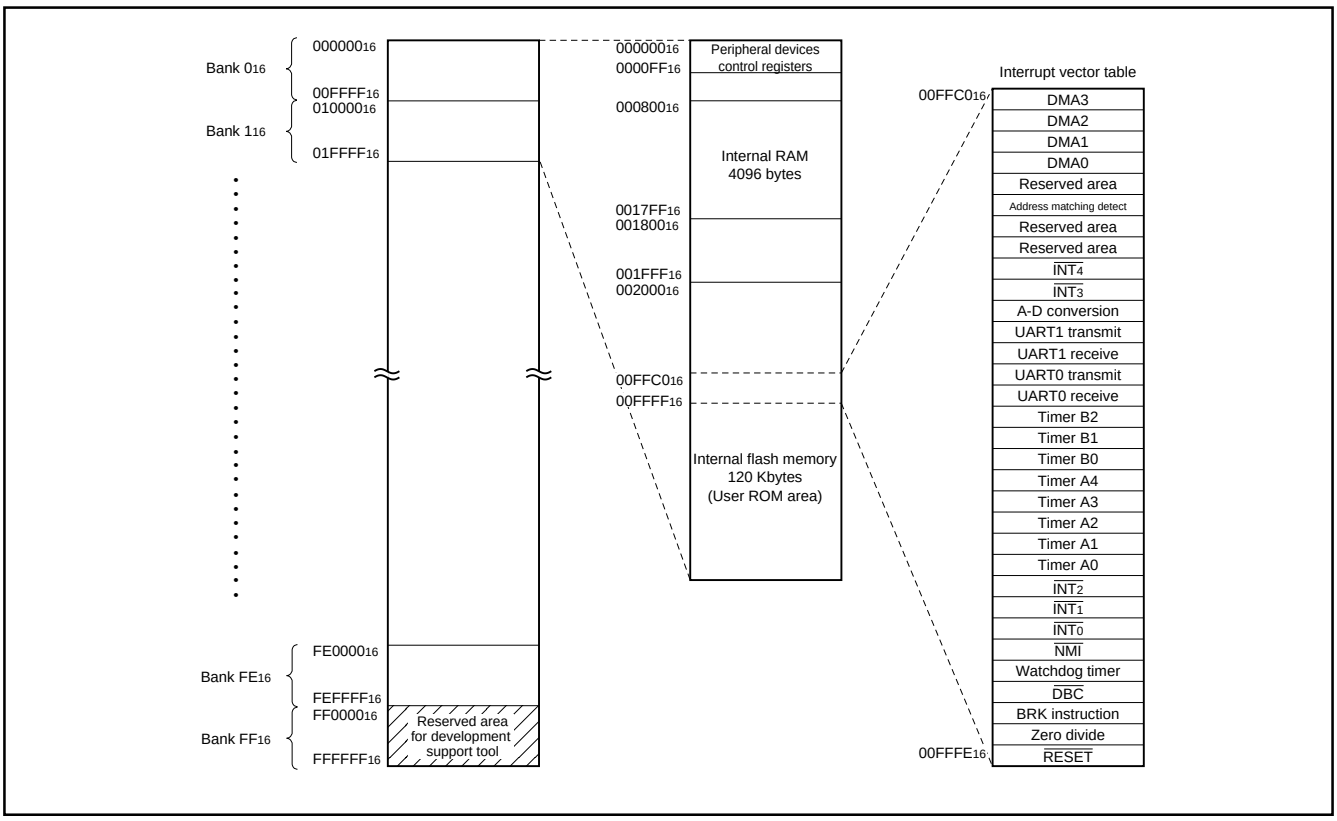


Fig. 1 Memory map of M37920FCCGP and M37920FCCHP (Single-chip mode)

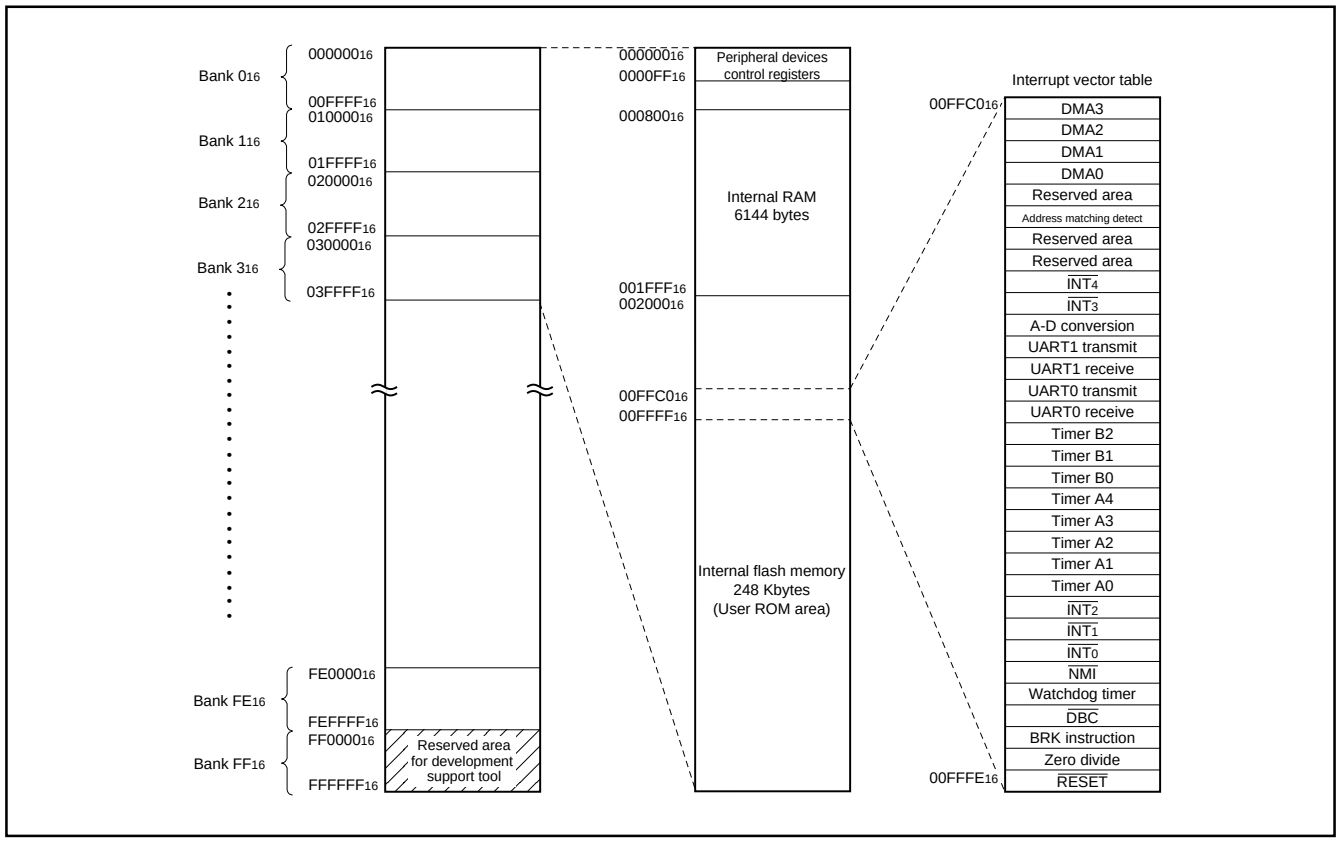


Fig. 2 Memory map of M37920FGCGP and M37920FGCHP (Single-chip mode)

Address (Hexadecimal notation)	
000000₁₆	Reserved area **(Note)**
000001₁₆	Reserved area **(Note)**
000002₁₆	Port P0 register
000003₁₆	Port P1 register
000004₁₆	Port P0 direction register
000005₁₆	Port P1 direction register
000006₁₆	Port P2 register
000007₁₆	Port P3 register
000008₁₆	Port P2 direction register
000009₁₆	Port P3 direction register
00000A₁₆	Port P4 register
00000B₁₆	Port P5 register
00000C₁₆	Port P4 direction register
00000D₁₆	Port P5 direction register
00000E₁₆	Port P6 register
00000F₁₆	Port P7 register
000010₁₆	Port P6 direction register
000011₁₆	Port P7 direction register
000012₁₆	Port P8 register
000013₁₆	Port P9 register
000014₁₆	Port P8 direction register
000015₁₆	Port P9 direction register
000016₁₆	Port P10 register
000017₁₆	Port P11 register
000018₁₆	Port P10 direction register
000019₁₆	Port P11 direction register
00001A₁₆	Port P12 register
00001B₁₆	
00001C₁₆	Port P12 direction register
00001D₁₆	
00001E₁₆	A-D control register 0
00001F₁₆	A-D control register 1
000020₁₆ 000021₁₆	A-D register 0
000022₁₆ 000023₁₆	A-D register 1
000024₁₆ 000025₁₆	A-D register 2
000026₁₆ 000027₁₆	A-D register 3
000028₁₆	
000029₁₆	
00002A₁₆	
00002B₁₆	
00002C₁₆	
00002D₁₆	
00002E₁₆	
00002F₁₆	
000030₁₆	UART0 transmit/receive mode register
000031₁₆	UART0 baud rate register (BRG0)
000032₁₆ 000033₁₆	UART0 transmit buffer register
000034₁₆	UART0 transmit/receive control register 0
000035₁₆	UART0 transmit/receive control register 1
000036₁₆ 000037₁₆	UART0 receive buffer register
000038₁₆	UART1 transmit/receive mode register
000039₁₆	UART1 baud rate register (BRG1)
00003A₁₆ 00003B₁₆	UART1 transmit buffer register
00003C₁₆	UART1 transmit/receive control register 0
00003D₁₆	UART1 transmit/receive control register 1
00003E₁₆ 00003F₁₆	UART1 receive buffer register

Address (Hexadecimal notation)	
000040₁₆	Count start register
000041₁₆	
000042₁₆	One-shot start register
000043₁₆	
000044₁₆	Up-down register
000045₁₆	Timer A clock division select register
000046₁₆ 000047₁₆	Timer A0 register
000048₁₆ 000049₁₆	Timer A1 register
00004A₁₆ 00004B₁₆	Timer A2 register
00004C₁₆ 00004D₁₆	Timer A3 register
00004E₁₆ 00004F₁₆	Timer A4 register
000050₁₆ 000051₁₆	Timer B0 register
000052₁₆ 000053₁₆	Timer B1 register
000054₁₆ 000055₁₆	Timer B2 register
000056₁₆	Timer A0 mode register
000057₁₆	Timer A1 mode register
000058₁₆	Timer A2 mode register
000059₁₆	Timer A3 mode register
00005A₁₆	Timer A4 mode register
00005B₁₆	Timer B0 mode register
00005C₁₆	Timer B1 mode register
00005D₁₆	Timer B2 mode register
00005E₁₆	Processor mode register 0
00005F₁₆	Processor mode register 1
000060₁₆	Watchdog timer register
000061₁₆	Watchdog timer frequency select register
000062₁₆	Particular function select register 0
000063₁₆	Particular function select register 1
000064₁₆	Particular function select register 2
000065₁₆	Reserved area **(Note)**
000066₁₆	Debug control register 0
000067₁₆	Debug control register 1
000068₁₆ 000069₁₆ 00006A₁₆	Address comparison register 0
00006B₁₆ 00006C₁₆ 00006D₁₆	Address comparison register 1
00006E₁₆	$\overline{INT_3}$ interrupt control register
00006F₁₆	$\overline{INT_4}$ interrupt control register
000070₁₆	A-D conversion interrupt control register
000071₁₆	UART0 transmit interrupt control register
000072₁₆	UART0 receive interrupt control register
000073₁₆	UART1 transmit interrupt control register
000074₁₆	UART1 receive interrupt control register
000075₁₆	Timer A0 interrupt control register
000076₁₆	Timer A1 interrupt control register
000077₁₆	Timer A2 interrupt control register
000078₁₆	Timer A3 interrupt control register
000079₁₆	Timer A4 interrupt control register
00007A₁₆	Timer B0 interrupt control register
00007B₁₆	Timer B1 interrupt control register
00007C₁₆	Timer B2 interrupt control register
00007D₁₆	$\overline{INT_0}$ interrupt control register
00007E₁₆	$\overline{INT_1}$ interrupt control register
00007F₁₆	$\overline{INT_2}$ interrupt control register

Note: Do not write to this address.

Fig. 4 Location of SFRs (1)

Address (Hexadecimal notation)	
000080₁₆	$\overline{CS_0}$ control register L
000081₁₆	$\overline{CS_0}$ control register H
000082₁₆	$\overline{CS_1}$ control register L
000083₁₆	$\overline{CS_1}$ control register H
000084₁₆	$\overline{CS_2}$ control register L
000085₁₆	$\overline{CS_2}$ control register H
000086₁₆	$\overline{CS_3}$ control register L
000087₁₆	$\overline{CS_3}$ control register H
000088₁₆	
000089₁₆	
00008A₁₆	Area $\overline{CS_0}$ start address register
00008B₁₆	
00008C₁₆	Area $\overline{CS_1}$ start address register
00008D₁₆	
00008E₁₆	Area $\overline{CS_2}$ start address register
00008F₁₆	
000090₁₆	Area $\overline{CS_3}$ start address register
000091₁₆	
000092₁₆	
000093₁₆	
000094₁₆	
000095₁₆	
000096₁₆	
000097₁₆	
000098₁₆	
000099₁₆	
00009A₁₆	
00009B₁₆	
00009C₁₆	Reserved area **(Note)**
00009D₁₆	Reserved area **(Note)**
00009E₁₆	Flash memory control register
00009F₁₆	
0000A0₁₆	Real-time output control register
0000A1₁₆	
0000A2₁₆	Pulse output data register 0
0000A3₁₆	
0000A4₁₆	Pulse output data register 1
0000A5₁₆	
0000A6₁₆	Reserved area **(Note)**
0000A7₁₆	
0000A8₁₆	DRAM control register
0000A9₁₆	Refresh timer
0000AA₁₆	
0000AB₁₆	
0000AC₁₆	$\overline{CTS}$/$\overline{RTS}$ separate select register
0000AD₁₆	
0000AE₁₆	
0000AF₁₆	
0000B0₁₆	DMAC control register L
0000B1₁₆	DMAC control register H
0000B2₁₆	DMA0 interruput control register
0000B3₁₆	DMA1 interruput control register
0000B4₁₆	DMA2 interruput control register
0000B5₁₆	DMA3 interruput control register
0000B6₁₆	
0000B7₁₆	
0000B8₁₆	
0000B9₁₆	
0000BA₁₆	
0000BB₁₆	
0000BC₁₆	Reserved area **(Note)**
0000BD₁₆	Reserved area **(Note)**
0000BE₁₆	Reserved area **(Note)**
0000BF₁₆	Reserved area **(Note)**

Address (Hexadecimal notation)	
0000C0₁₆	Source address register 0 L
0000C1₁₆	Source address register 0 M
0000C2₁₆	Source address register 0 H
0000C3₁₆	
0000C4₁₆	Destination address register 0 L
0000C5₁₆	Destination address register 0 M
0000C6₁₆	Destination address register 0 H
0000C7₁₆	
0000C8₁₆	Transfer counter register 0 L
0000C9₁₆	Transfer counter register 0 M
0000CA₁₆	Transfer counter register 0 H
0000CB₁₆	
0000CC₁₆	DMA0 mode register L
0000CD₁₆	DMA0 mode register H
0000CE₁₆	DMA0 control register
0000CF₁₆	
0000D0₁₆	Source address register 1 L
0000D1₁₆	Source address register 1 M
0000D2₁₆	Source address register 1 H
0000D3₁₆	
0000D4₁₆	Destination address register 1 L
0000D5₁₆	Destination address register 1 M
0000D6₁₆	Destination address register 1 H
0000D7₁₆	
0000D8₁₆	Transfer counter register 1 L
0000D9₁₆	Transfer counter register 1 M
0000DA₁₆	Transfer counter register 1 H
0000DB₁₆	
0000DC₁₆	DMA1 mode register L
0000DD₁₆	DMA1 mode register H
0000DE₁₆	DMA1 control register
0000DF₁₆	
0000E0₁₆	Source address register 2 L
0000E1₁₆	Source address register 2 M
0000E2₁₆	Source address register 2 H
0000E3₁₆	
0000E4₁₆	Destination address register 2 L
0000E5₁₆	Destination address register 2 M
0000E6₁₆	Destination address register 2 H
0000E7₁₆	
0000E8₁₆	Transfer counter register 2 L
0000E9₁₆	Transfer counter register 2 M
0000EA₁₆	Transfer counter register 2 H
0000EB₁₆	
0000EC₁₆	DMA2 mode register L
0000ED₁₆	DMA2 mode register H
0000EE₁₆	DMA2 control register
0000EF₁₆	
0000F0₁₆	Source address register 3 L
0000F1₁₆	Source address register 3 M
0000F2₁₆	Source address register 3 H
0000F3₁₆	
0000F4₁₆	Destination address register 3 L
0000F5₁₆	Destination address register 3 M
0000F6₁₆	Destination address register 3 H
0000F7₁₆	
0000F8₁₆	Transfer counter register 3 L
0000F9₁₆	Transfer counter register 3 M
0000FA₁₆	Transfer counter register 3 H
0000FB₁₆	
0000FC₁₆	DMA3 mode register L
0000FD₁₆	DMA3 mode register H
0000FE₁₆	DMA3 control register
0000FF₁₆	

Note: Do not write to this address.

Fig. 5 Location of SFRs (2)

CENTRAL PROCESSING UNIT (CPU)

The CPU has 13 registers, and they are shown in Figure 6. 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).

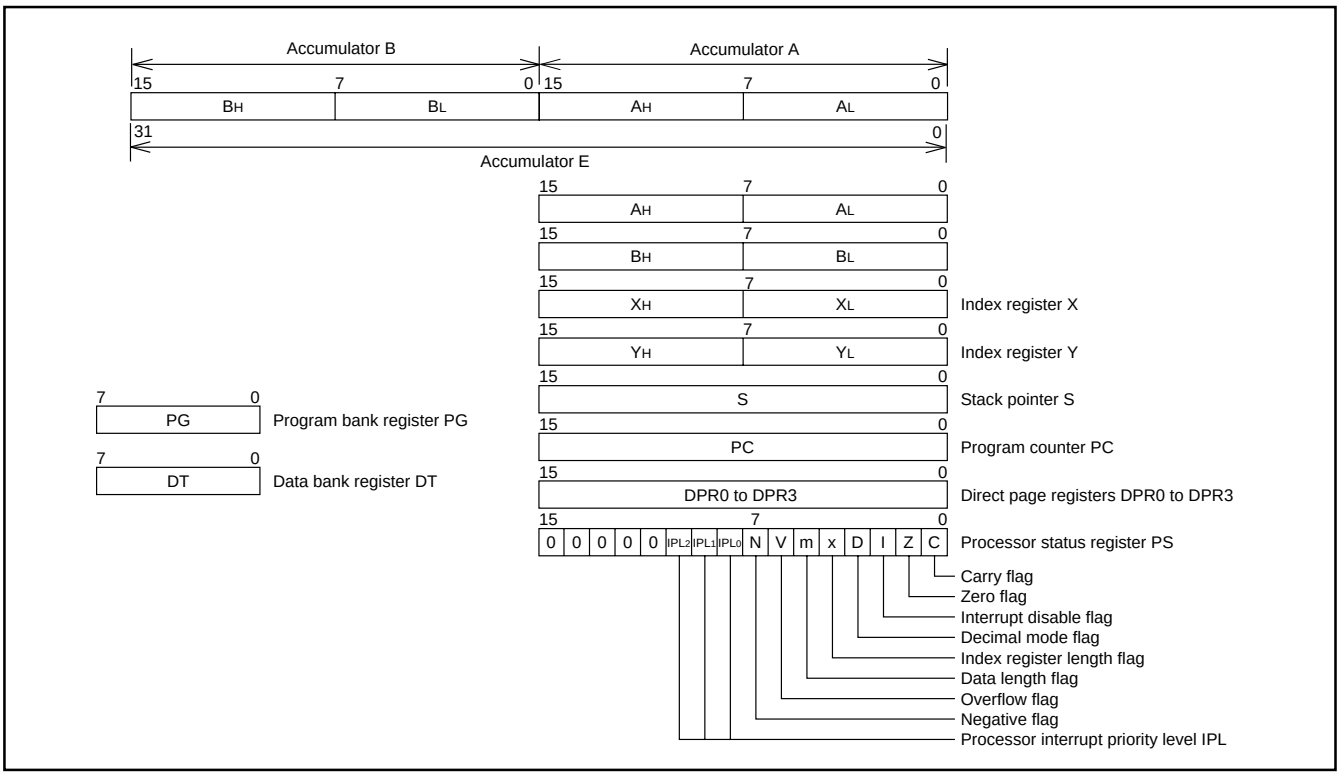


Fig. 6 Register structure

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 shown below is always performed via the bus interface unit (BIU), which is located between the CPU and the internal buses:

- Between the CPU and the internal memory, internal peripheral devices, external areas
- Between the DMA controller (DMAC) and the internal memory, internal peripheral devices, external areas

Figure 7 shows the BIU and the bus structure. The CPU and BIU, or DMAC and BIU are connected by a dedicated bus respectivery, and any transfer between the CPU and BIU, or DMAC 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.

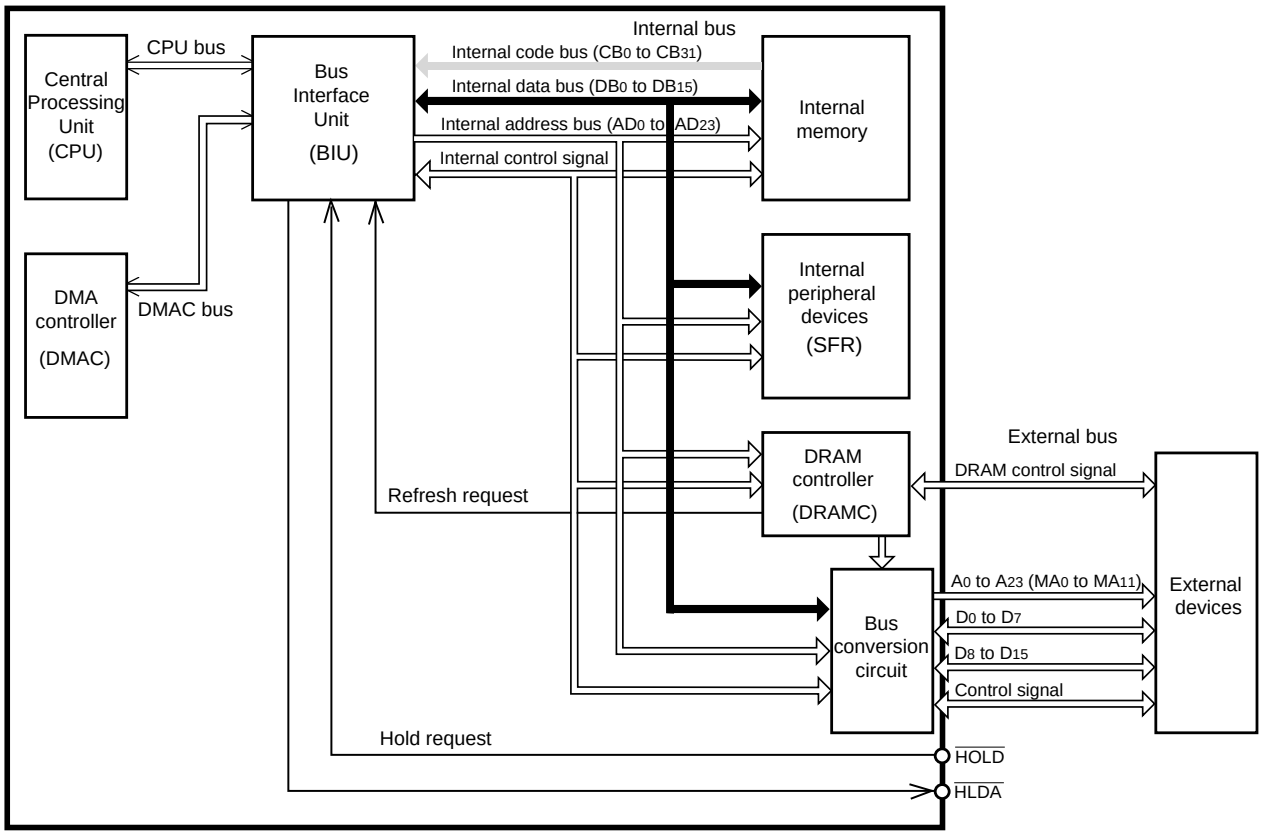
Each width of external buses are as follows: a 24-bit address bus, 16-bit data bus.

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.

When the DRAM is selected in external devices, the internal DMAC controller (DRAMC) is operated, and access for DRAM and DRAM refresh operation become enabled. For details, refer to the section on the chip select wait controller and DRAMC described later.

When accessing the external devices, it is possible to insert the recovery cycles. Refer to the section on the processor modes and chip select wait controller described later.

When the burst ROM is used as an external device, refer to the section on the chip select wait controller described later.



SFR : Special Function Register

✵ The CPU bus, DMAC bus, internal bus, and external bus separate out independently.

Fig. 7 BIU and bus structure

BIU structure

The BIU consists of four registers shown in Figure 8. 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 or DMAC. Consists of 32 bits.

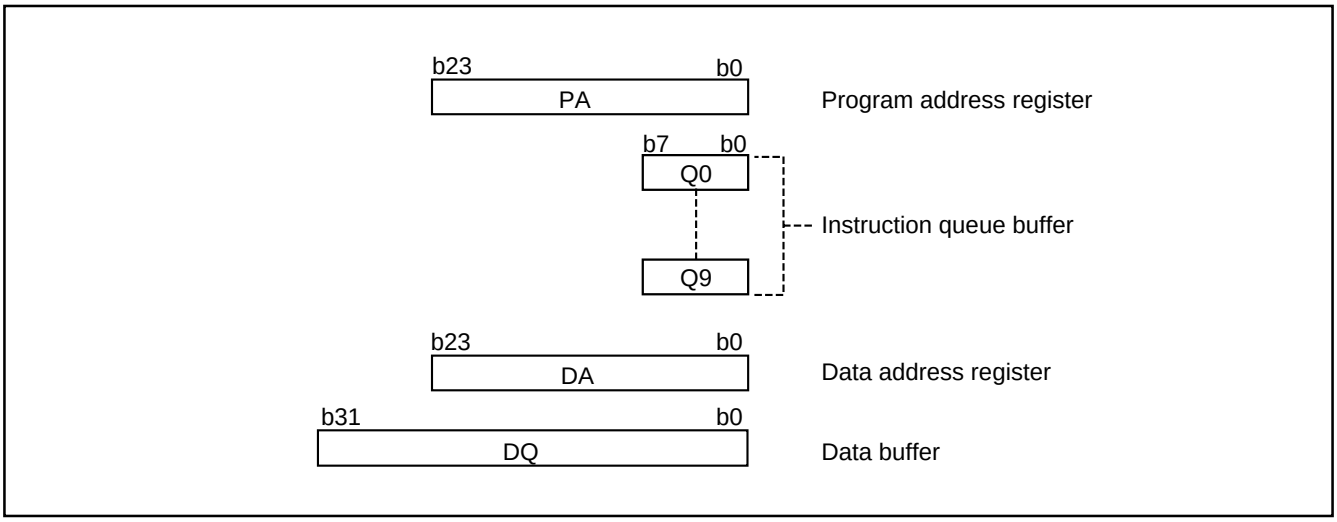


Fig. 8 Register structure of BIU

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 bus width of external devices = 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-numbered 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 Others: Whether any the burst ROM access and the DRAM space is specified or not. (For details, refer to the section on the chip select wait controller and DRAM controller described later.)

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-numbered 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 bus width of external devices = 16 bits or 8 bits:
  - ② Number of waits
  - ③ Others: Whether the DRAM space is specified or not. (For details, refer to the section on the chip select wait controller and DRAM controller described later.)

The BIU controls the bus cycle depending on the above conditions. Instruction prefetch and data access are performed as shown in Tables 3 to 10.

Table 3. Instruction prefetch

Access to internal area	Access to external area	
	When external data bus width = 16 bits	When external data bus width = 8 bits
When branched or at instruction prefetch	When address locates at 4-byte boundary or when branched: double consecutive access	When address is even-numbered address or when branched: double consecutive access
	When address of instruction to be prefetched locates at 8-byte boundary: quadruple consecutive access	When address of instruction to be prefetched locates at 4-byte boundary or 8-byte boundary: quadruple consecutive access

Table 4. Data access (1)

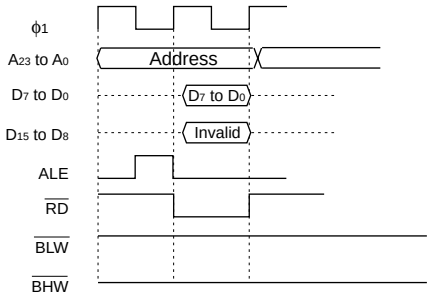
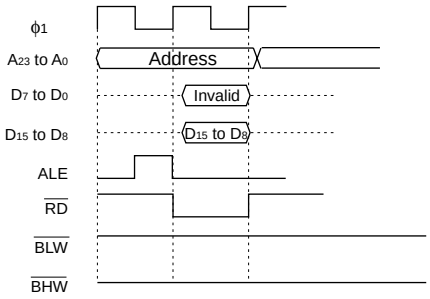
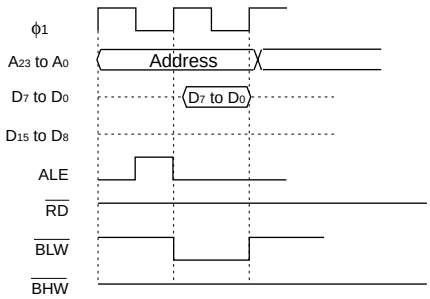
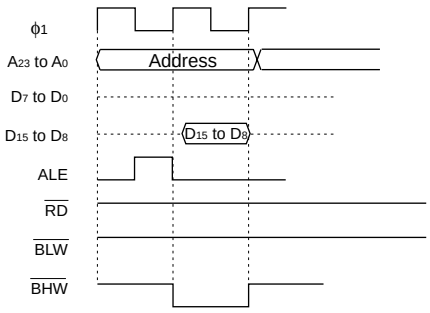
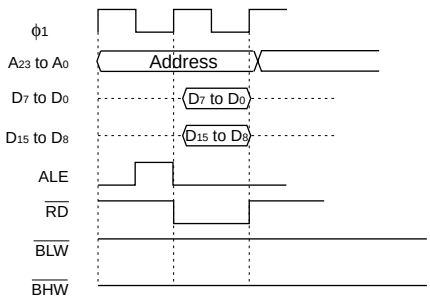
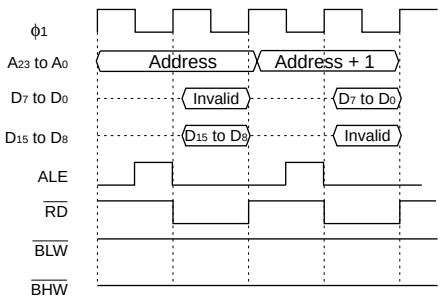
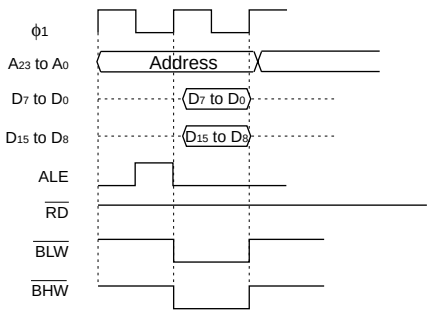
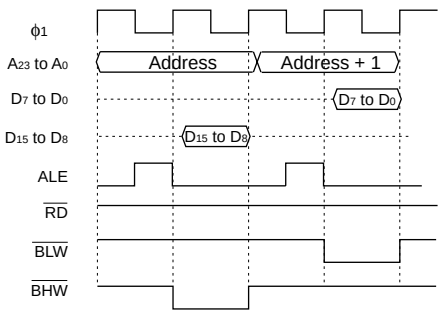
		Access starting from even-numbered address	Access starting from odd-numbered address
External data bus width = 16 bits	Byte data read		
	Byte data written		
	Word data read		
	Word data written		

Table 5. Data access (2)

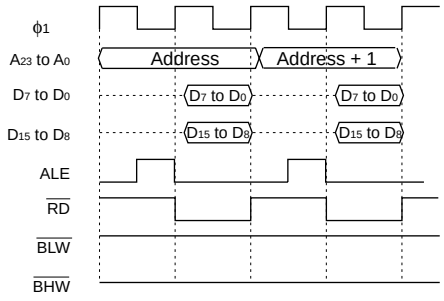
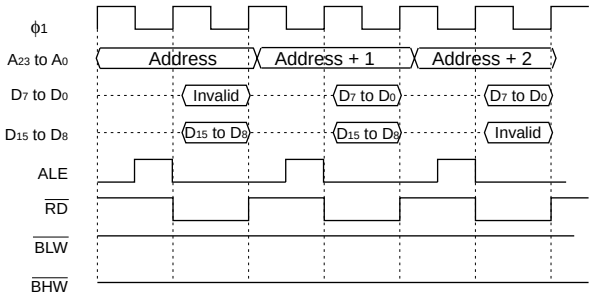
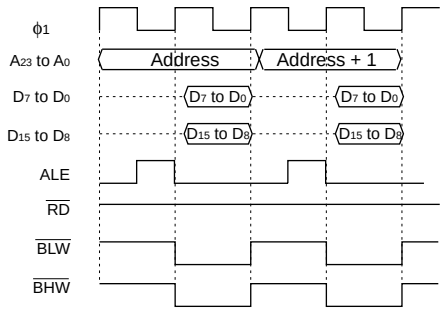
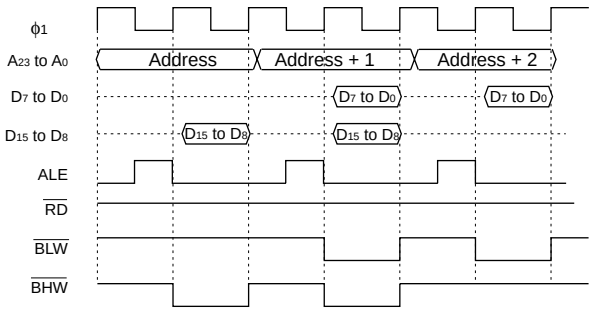
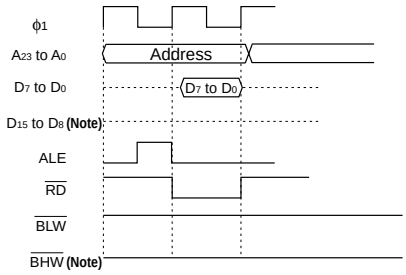
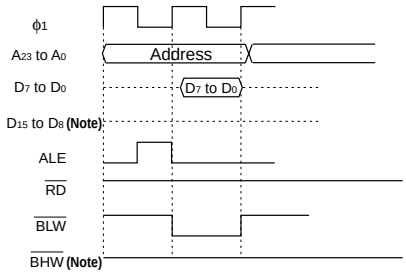
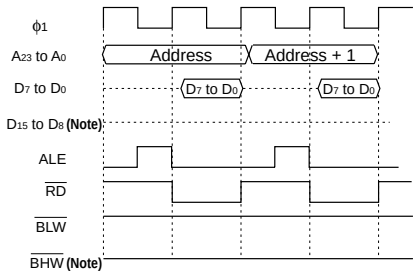
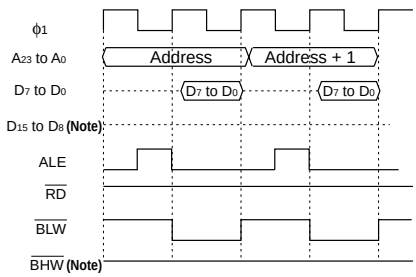
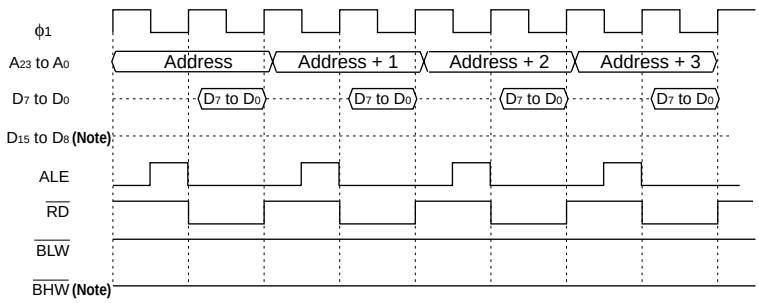
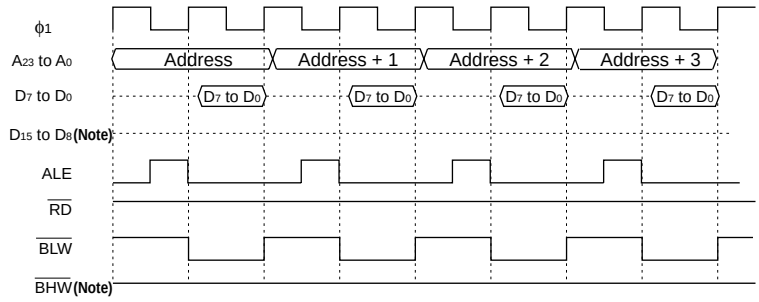
		Access starting from even-numbered address	Access starting from odd-numbered address
External data bus width = 16 bits	Double word data read		
	Double word data written		

Table 6. Data access (3)

		Access starting from even- or odd-numbered address
External data bus width = 8 bits	Byte data read	
	Byte data written	
	Word data read	
	Word data written	

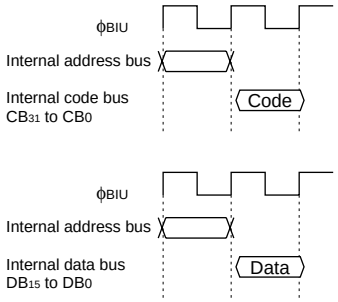
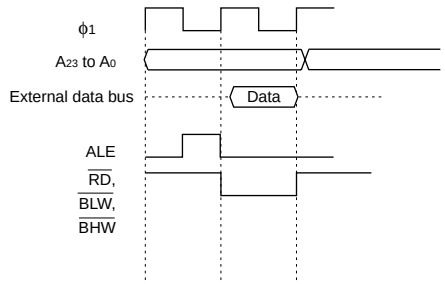
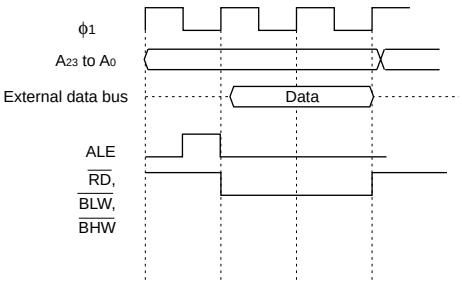
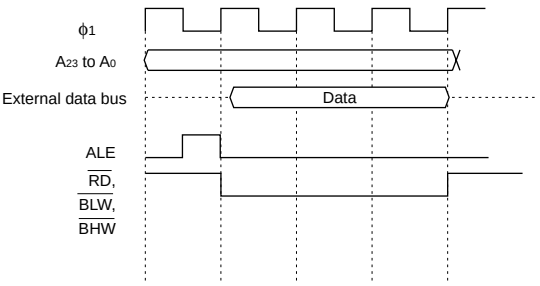
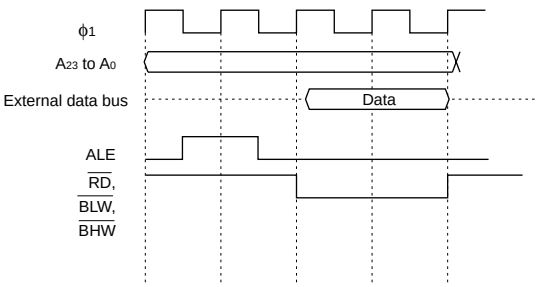
Note: When the voltage level at pin BYTE = “L”, functions as pins D_{15} to D_8 are valid. However, when 8-bit width is selected as the external bus width by the chip select wait controller, the functions as pins D_{15} to D_8 and $\overline{BHW}$ become invalid. When the voltage level at pin BYTE =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

Table 7. Data access (4)

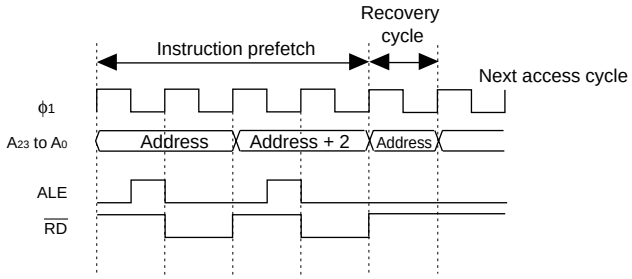
	Access starting from even- or odd-numbered address
External data bus width = 8 bits — Double word data read	
External data bus width = 8 bits — Double word data written	

Note: When the voltage level at pin BYTE = “L”, functions as pins D_{15} to D_8 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_{15} to D_8 and $\overline{BHW}$ become invalid. When the voltage level at pin BYTE = “H”, these pins function as programmable I/O port (P2) pins.

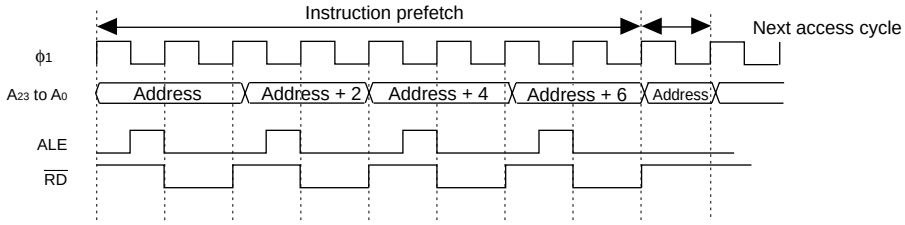
Table 8. Wait number (Instruction prefetch or data access)

Access to internal area	Access to external area
	0-wait access 
	1-wait access 
	2-wait access 
	ALE expansion wait access 

At double consecutive access (when address locates at 4-byte boundary or when branched)

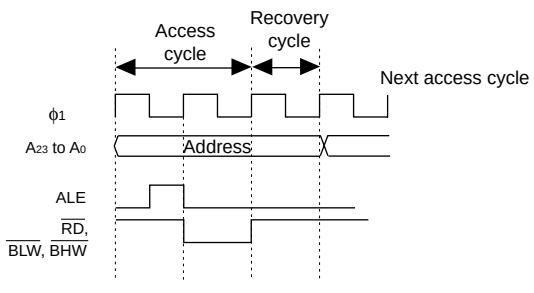


At quadruple consecutive access (when address locates at 8-byte boundary)



Note: External data bus width = 16 bits and at 0 wait.

Fig. 9 Recovery cycle (at instruction prefetch)



Note: At 0 wait.

Fig. 10 Recovery cycle (at data access)

Selection of processor mode

Figures 11, 12 show the bit configurations of the processor mode registers 0, 1.

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 5E16; Figure 11)

Table 9 lists the selection method of a processor mode.

The memory map which the CPU can access depends on the selected processor mode. Figure 13 shows the memory maps in three processor modes.

Also, the functions of ports P0 to P4, P10, P11, and part of port P9 depend on the selected processor mode. For details, see Table 10.

In the single-chip mode, ports P0 to P4, P10, P11, and P9 function as I/O ports. 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, and part of port P9 function as I/O pins for the address bus, data bus, bus control signals. (Some of port functions are selectable.)

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.

Table 11 lists each bus control signal's function.

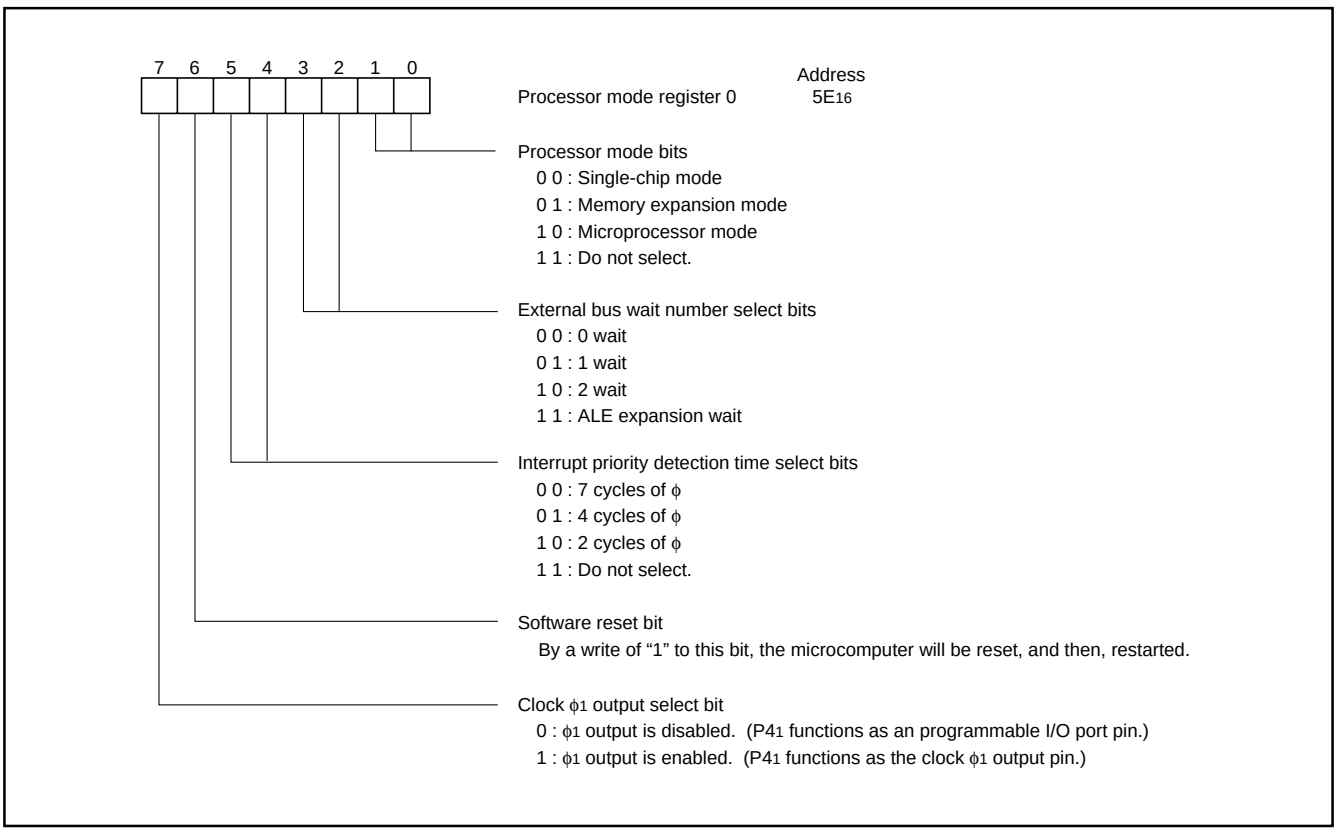
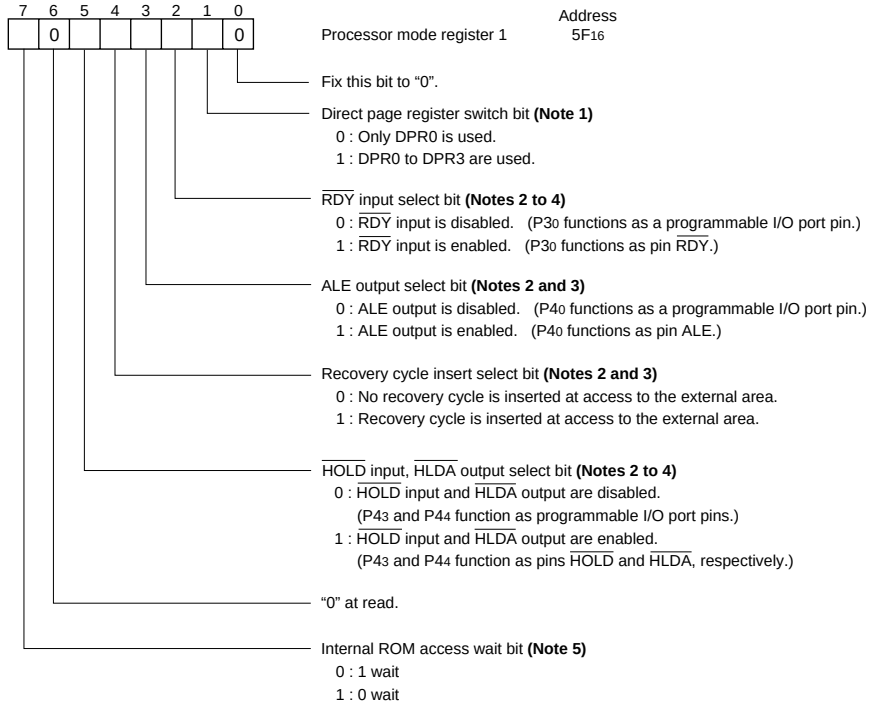


Fig. 11 Bit configuration of processor mode register 0



Notes 1: After reset, this bit's contents can be switched only once. During the software execution, be sure not to switch this bit's contents.
2: In the single-chip mode, these bits' functions are disabled regardless of these bits' contents.
3: While VSS level voltage is applied to pin MD0, each of these bits is "0" at reset. While VCC level voltage is applied to pin MD0, on the other hand, each of these bits is "1" at reset.
4: 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.
5: 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. 12 Bit configuration of processor mode register 1

Table 9. Selection method of processor mode

MD0	Mode	Description
VSS	• Single-chip mode • Memory expansion mode • Microprocessor mode	After reset is removed, the single-chip mode is selected. By changing the processor mode bits' contents by software, the single-chip mode, memory expansion mode or microprocessor mode can be selected.
VCC	• Microprocessor mode	After reset is removed, the microprocessor mode is selected.

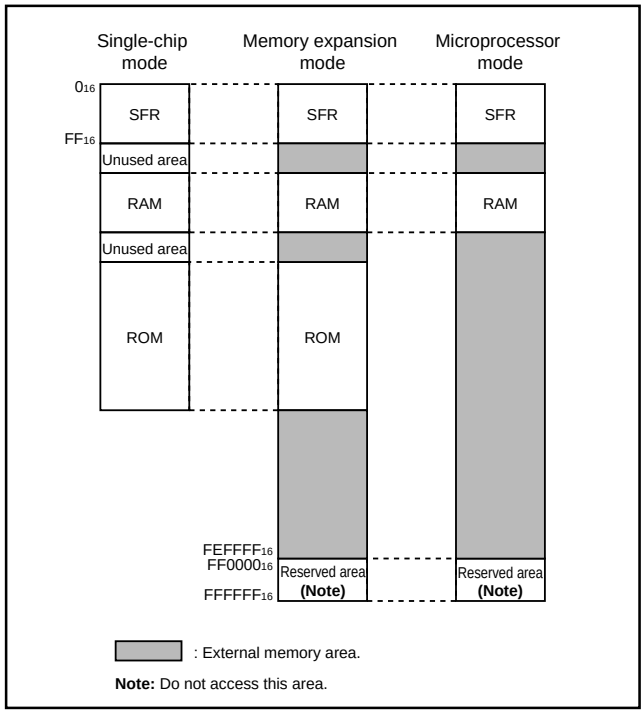


Fig. 13 Memory maps in three processor modes

Table 10. Relationship between processor modes, memory area, and port function

		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. Multiplexed address (MA_0 to MA_7) is output (Note 3)	Middle-order address (A_8 to A_{15}) is output. Multiplexed address (MA_0 to MA_7) is output (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. Multiplexed address (MA_8 to MA_{11}) is output (Note 3)	High-order address (A_{16} to A_{23}) is output. Multiplexed address (MA_8 to MA_{11}) is output (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-numbered address) is input/output.	Low-order data (D_0 to D_7, data at even-numbered address) is input/output.
	External data bus width = 8 bits		Low-order data (D_0 to D_7, data at even-/odd-numbered address) is input/output.	Low-order data (D_0 to D_7, data at even-/odd-numbered address) is input/output.
Port pins $P2_0$ to $P2_7$	External data bus width = 16 bits	I/O port pins $P2_0$ to $P2_7$	Low-order data (D_0 to D_7, data at odd-numbered address) is input/output.	Low-order data (D_0 to D_7, data at odd-numbered address) is input/output.
	External data bus width = 8 bits		I/O port pins $P2_0$ to $P2_7$	I/O port pins $P2_0$ to $P2_7$
Port pin $P3_0$		I/O port pin $P3_0$	I/O port pin $P3_0$ Ready signal $\overline{RDY}$ is input (Note 5).	Ready signal $\overline{RDY}$ is input. I/O port pin $P3_0$ (Note 5)
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-numbered address) is output.	Write signal $\overline{BLW}$ (write to even-numbered address) is output.
	External data bus width = 8 bits		Write signal $\overline{BLW}$ (write to even-/odd-numbered address) is output.	Write signal $\overline{BLW}$ (write to even-/odd-numbered 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-numbered address) is output.	Write signal $\overline{BHW}$ (write to odd-numbered address) is output.
	External data bus width = 8 bits		I/O port pin $P3_3$	I/O port pin $P3_3$
Port pin $P4_0$		I/O port pin $P4_0$	I/O port pin $P4_0$ Address latch enable signal ALE is output (Note 4).	Address latch enable signal ALE is output. I/O port pin $P4_0$ (Note 4)
Port pin $P4_1$		I/O port pin $P4_1$ Clock ϕ_1 is output (Note 4).	I/O port pin $P4_1$ Clock ϕ_1 is output (Note 4).	Clock ϕ_1 is output. I/O port pin $P4_1$ (Note 4)
Port pin $P4_2$		I/O port pin $P4_2$	I/O port pin $P4_2$ Hold acknowledge signa $\overline{HLDA}$ is output (Note 4).	Hold acknowledge signal $\overline{HLDA}$ is output. I/O port pin $P4_2$ (Note 4)
Port pin $P4_3$		I/O port pin $P4_3$	I/O port pin $P4_3$ Hold request signal $\overline{HOLD}$ is input (Note 4).	Hold request signal $\overline{HOLD}$ is input. I/O port pin $P4_3$ (Note 4)
Port pin $P9_0$		I/O port pin $P9_0$	I/O port pin $P9_0$ Chip select signal $\overline{CS_0}$ is output (Note 5).	Chip select signal $\overline{CS_0}$ is output.
Port pins $P9_1$ to $P9_3$		I/O port pins $P9_1$ to $P9_3$	I/O port pins $P9_1$ to $P9_3$ Chip select signals $\overline{CS_1}$ to $\overline{CS_3}$ are output (Note 6).	I/O port pin $P9_1$ to $P9_3$ Chip select signals $\overline{CS_1}$ to $\overline{CS_3}$ are output (Note 6).

Notes 1: For details of the processor mode setting, see Table 9.

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

3: While DRAM space is accessed, the multiplexed address is output.

4: 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.

5: In the memory expansion mode, port pin $P9_0$ 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}).

6: In the memory expansion and microprocessor modes, port pins $P9_1$ to $P9_3$ 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}).

Table 11. 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 the external.	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.

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 P90 to P93 can operate as chip select output pins ($\overline{CS0}$ to $\overline{CS3}$).

Figure 14 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, DRAM specification, burst ROM specification, and recovery cycle insertion validity.

For the external area except for areas $\overline{CS0}$ to $\overline{CS3}$, 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 $\overline{CS0}$ is automatically selected.

Table 12 lists the function of areas $\overline{CS0}$ to $\overline{CS3}$.

Figure 15 shows the bit configuration of the $\overline{CS0}$/$\overline{CS1}$/$\overline{CS2}$/$\overline{CS3}$ control register Ls. These registers determine the following items of a device to be connected: wait number, external data bus width (**Note 1:** The external data bus width of area $\overline{CS0}$ is determined by pin BYTE's level.), $\overline{RDY}$ control validity, DRAM space specification (**Note 2:** For area $\overline{CS0}$, this function is invalid.), burst ROM access specification, recovery cycle insertion validity. For DRAM access, refer to the section on the DRAM controller.

Figure 16 shows the bit configuration of the $\overline{CS0}$/$\overline{CS1}$/$\overline{CS2}$/$\overline{CS3}$ control register Hs. These registers determine block size of an external area to be connected. For areas $\overline{CS1}$ and $\overline{CS2}$, by selecting mode 1 with the area $\overline{CSk}$ setting mode select bit, an chip select area can be set to the external area in bank 0.

Figure 17 shows the bit configuration of the area $\overline{CS0}$/$\overline{CS1}$/$\overline{CS2}$/$\overline{CS3}$ start address registers. For details of these addresses' setting, see Figures 18 to 20.

● Burst ROM access

For ROM supporting the burst ROM access, the burst ROM access can be specified. 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.

Figure 21 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.

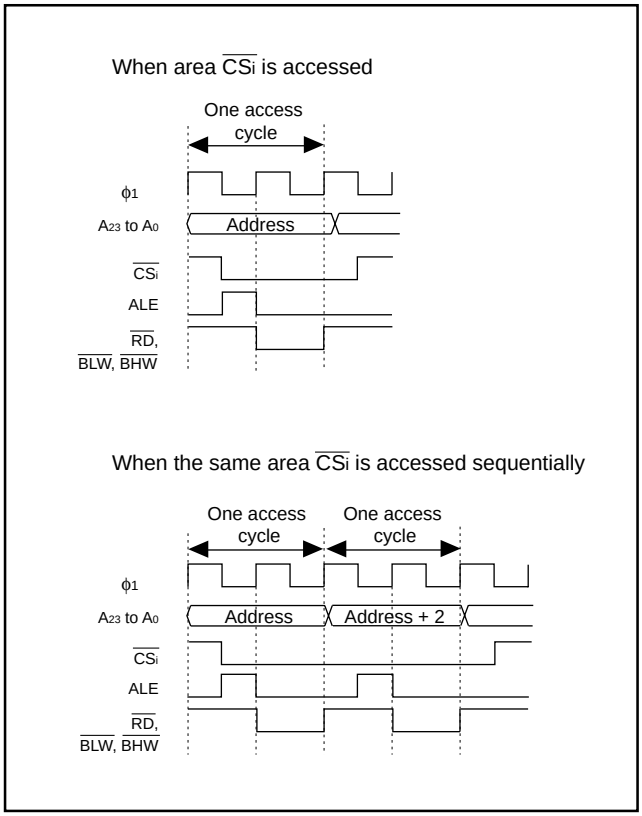
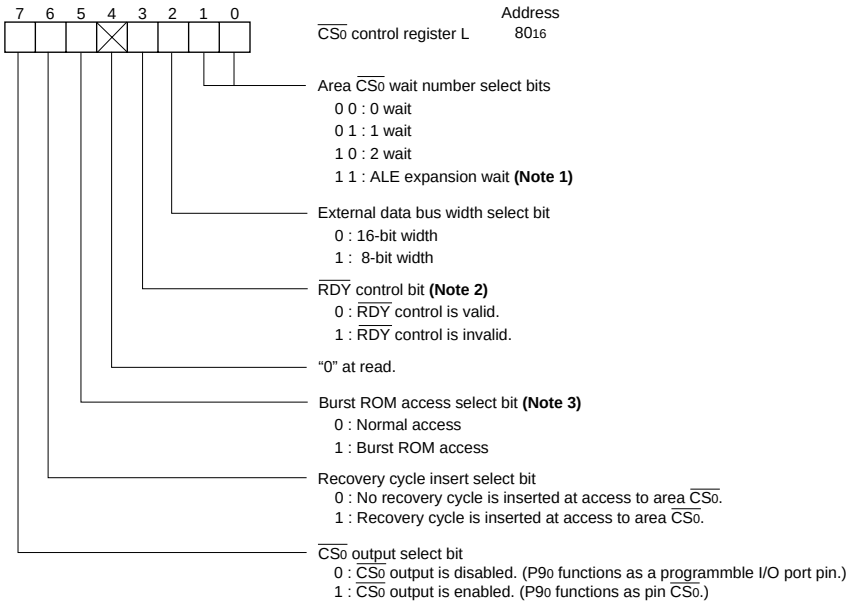


Fig. 14 Chip select output waveform example

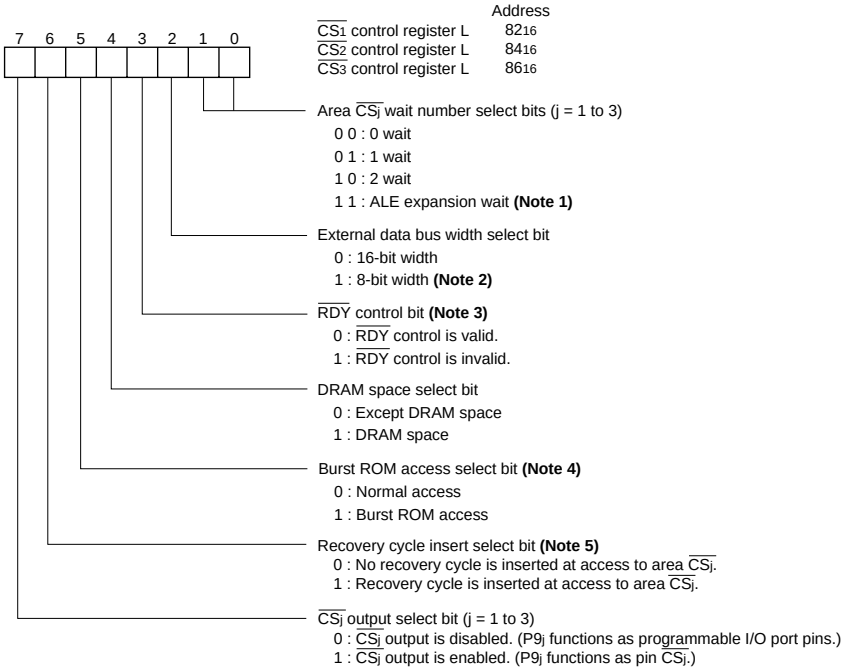
Table 12. Function of areas $\overline{CS_0}$ to $\overline{CS_3}$

	$\overline{CS_0}$	$\overline{CS_1}$, $\overline{CS_2}$		$\overline{CS_3}$	External area except for $\overline{CS_0}$ to $\overline{CS_3}$
		Mode 0	Mode 1		
Space where start address can be set	Bank 0_{16}	Banks 0_{16} to FE_{16}	Bank 0_{16}	Banks 2_{16} to FE_{16}	
Block size	128 Kbytes, 512 Kbytes, or 1 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	
Wait	0 wait, 1 wait, 2 wait, or ALE expansion wait (Selected by bits 0, 1 at address 80_{16}.)	0 wait, 1 wait, 2 wait, or ALE expansion wait (Selected by bits 0, 1 at addresses 82_{16}, 84_{16}.)		0 wait, 1 wait, 2 wait, or ALE expansion wait (Selected by bits 0, 1 at address 86_{16}.)	0 wait, 1 wait, 2 wait, or ALE expansion wait (Selected by bits 0, 1 at address $5E_{16}$.)
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 2)**.			Determined by pin BYTE's level
$\overline{RDY}$ control	Valid (Selected by bit 2 at address $5F_{16}$ and bit 3 at address 80_{16}.)	Valid (When DRAM space is specified, however, this control is invalid.) (Selected by bit 2 at address $5F_{16}$ and bit 3 at addresses 82_{16}, 84_{16}.)		Valid (When DRAM space is specified, however, this control is invalid.) (Selected by bit 2 at address $5F_{16}$ and bit 3 at address 86_{16}.)	Valid (Selected by bit 2 at address $5F_{16}$.)
DRAM space specification	Not available.	Available.		Available.	Not available.
Burst ROM access **(Note 1)**	Available.	Available.		Available.	Not available.
Recovery cycle insertion	Available.	Available.		Available.	Available.

Notes 1: Burst ROM access is valid only when the external data bus width is 16 bits at instruction prefetch.
2: When BYTE = Vcc level, the external data bus width is fixed to 8 bits.

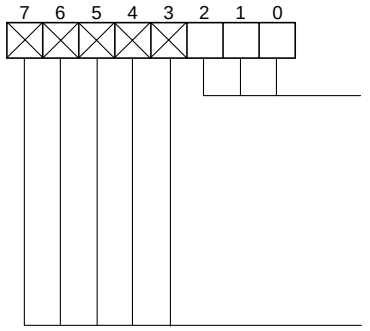


Notes 1: When the burst ROM access is specified (bit 5= 1), be sure not to select "112" (ALE expansion wait).
2: This bit is valid when the RDY input select bit (bit 2 at address 5F16) = "1".
3: While VCC level voltage is applied to pin BYTE, the normal access is selected regardless of this bit's contents.



Notes 1: When the DRAM space is specified (bit 4 = 1), fix these bits to "012" (1 wait). Also, when the burst ROM access is specified (bit 5 = 1), be sure not to select "112" (ALE expansion wait).
2: While VCC level voltage is applied to pin BYTE, this bit is fixed to "1" (8-bit width).
3: This bit is valid when the RDY input select bit (bit 2 at address 5F16) = "1". Also, when DRAM space is specified (bit 4 = 1), the RDY control is invalid regardless of this bit's contets.
4: When only the external data bus width select bit (bit 2) = "1" or while VCC level voltage is applied to pin BYTE, the normal access is selected regardless of this bit's contents.
5: When the DRAM space is specified (bit 4 = 1), fix this bit to "0" (no recovery cycle).

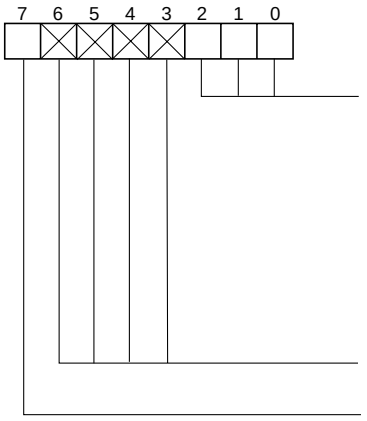
Fig. 15 Bit configuration of $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ control register Ls



$\overline{CS_0}$ control register H — Address 81₁₆

Area $\overline{CS_0}$ block size select bit
- 0 0 0 : 0 byte (Area $\overline{CS_0}$ is invalid.)
- 0 0 1 : 128 Kbytes
- 0 1 0 : Do not select.
- 0 1 1 : 512 Kbytes
- 1 0 0 : 1 Mbytes
- 1 0 1 : Do not select.
- 1 1 0 : Do not select.
- 1 1 1 : Do not select.

"0" at read.



$\overline{CS_1}$ control register H — Address 83₁₆
$\overline{CS_2}$ control register H — Address 85₁₆

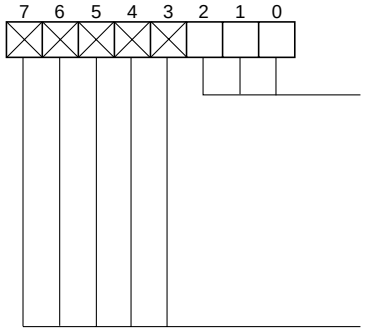
Area $\overline{CS_k}$ block size select bit (k = 1, 2)

When mode 0 is selected
- 0 0 0 : 0 byte (Area $\overline{CS_k}$ 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

When mode 1 is selected
- 0 0 0 : 0 byte (Area $\overline{CS_k}$ is invalid.)
- 0 0 1 : Do not select.
- 0 1 0 : Do not select.
- 0 1 1 : Do not select.
- 1 0 0 : 4 Kbytes
- 1 0 1 : 8 Kbytes
- 1 1 0 : Do not select.
- 1 1 1 : Do not select.

"0" at read.

Area $\overline{CS_k}$ setting mode select bit (k = 1, 2)
- 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.)

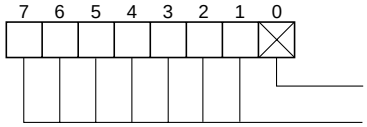


$\overline{CS_3}$ control register H — Address 87₁₆

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

"0" at read.

Fig. 16 Bit configuration of $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ control register Hs



Address

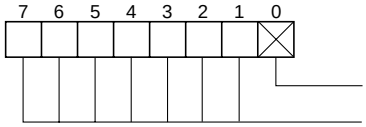
Area $\overline{CS_0}$ start address register 8A16

"0" at read.

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 18.



Address

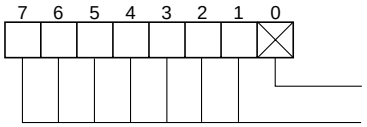
Area $\overline{CS_1}$ start address register 8C16
Area $\overline{CS_2}$ start address register 8E16

"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 19 and 20.



Address

Area $\overline{CS_3}$ start address register 9016

"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 20.

Fig. 17 Bit configuration of area $\overline{CS_0}/\overline{CS_1}/\overline{CS_2}/\overline{CS_3}$ start address registers

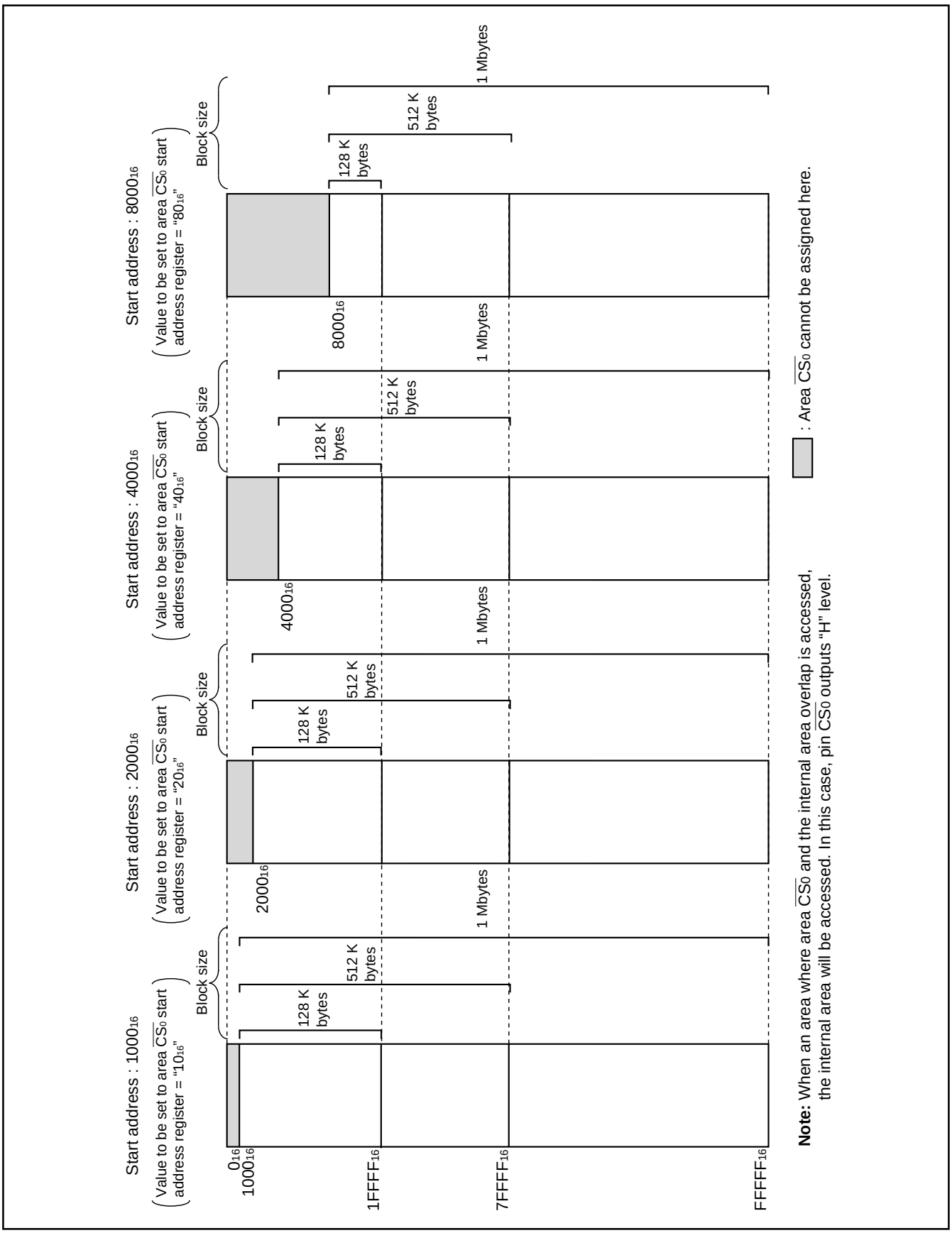
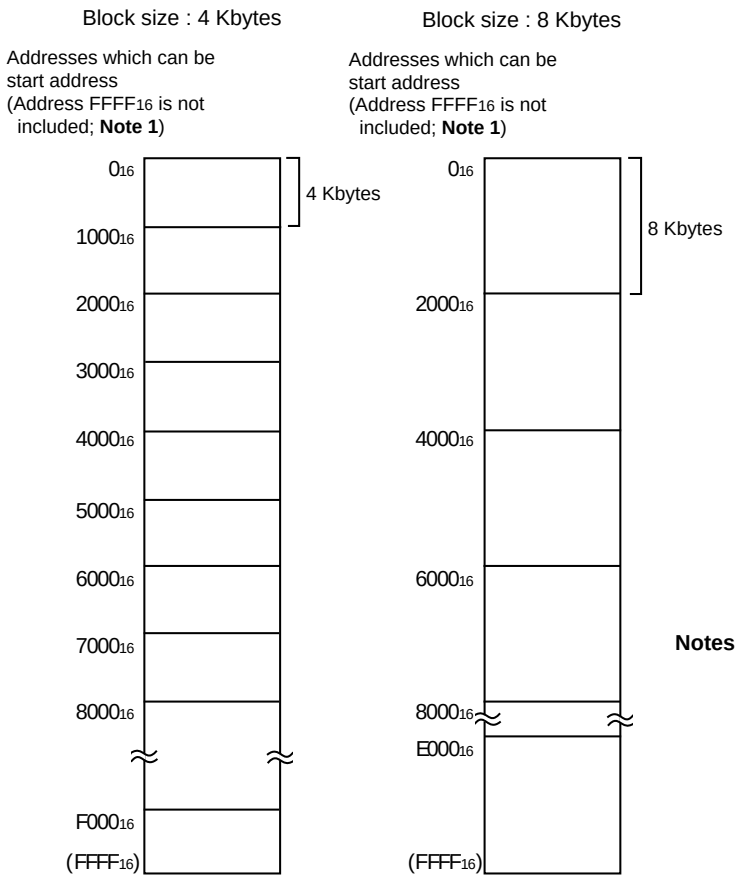


Fig. 18 Area $\overline{CS_0}$



Notes 1: Only A8 to A15 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. 19 Area $\overline{CS_1}$/$\overline{CS_2}$ (mode 1)

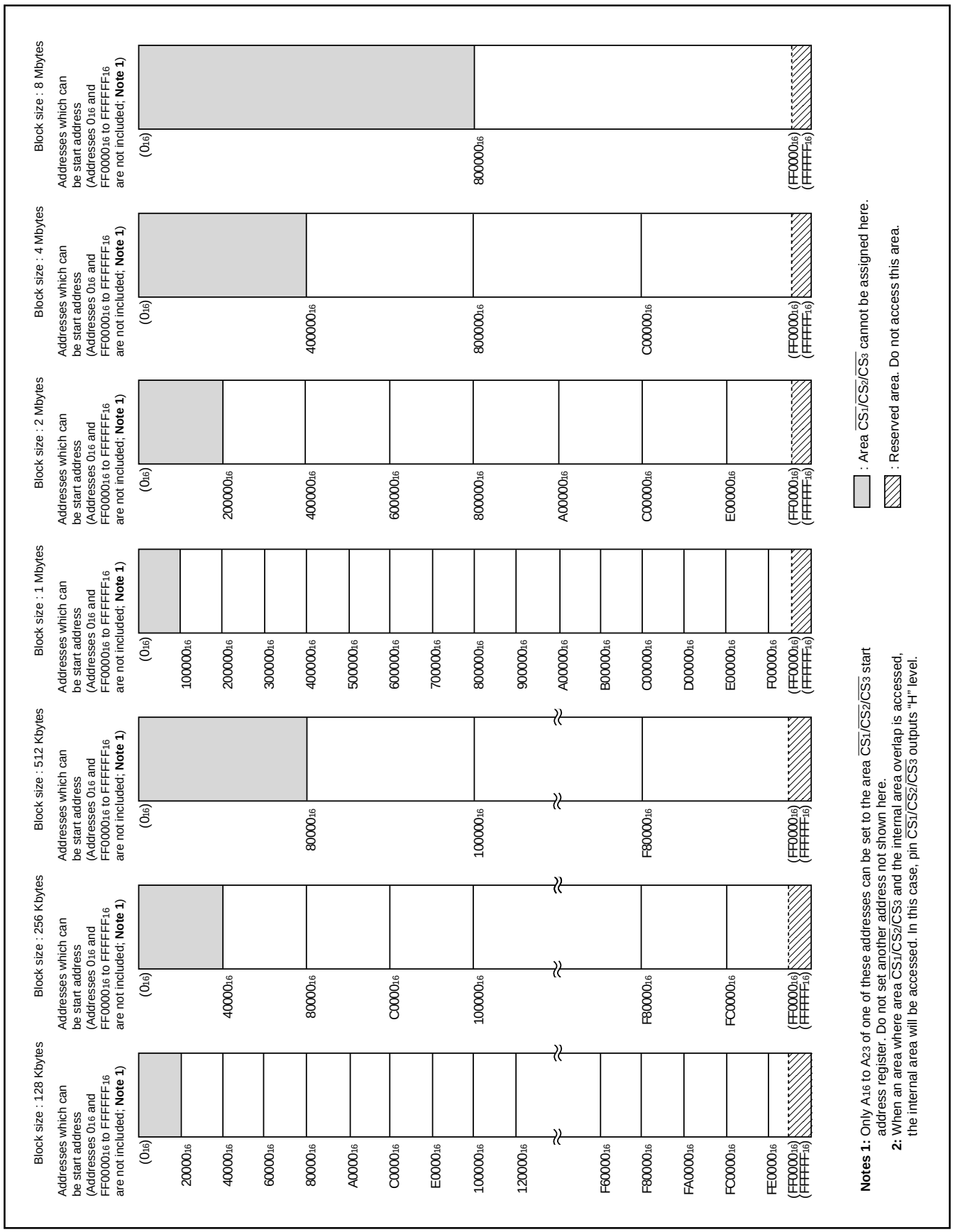
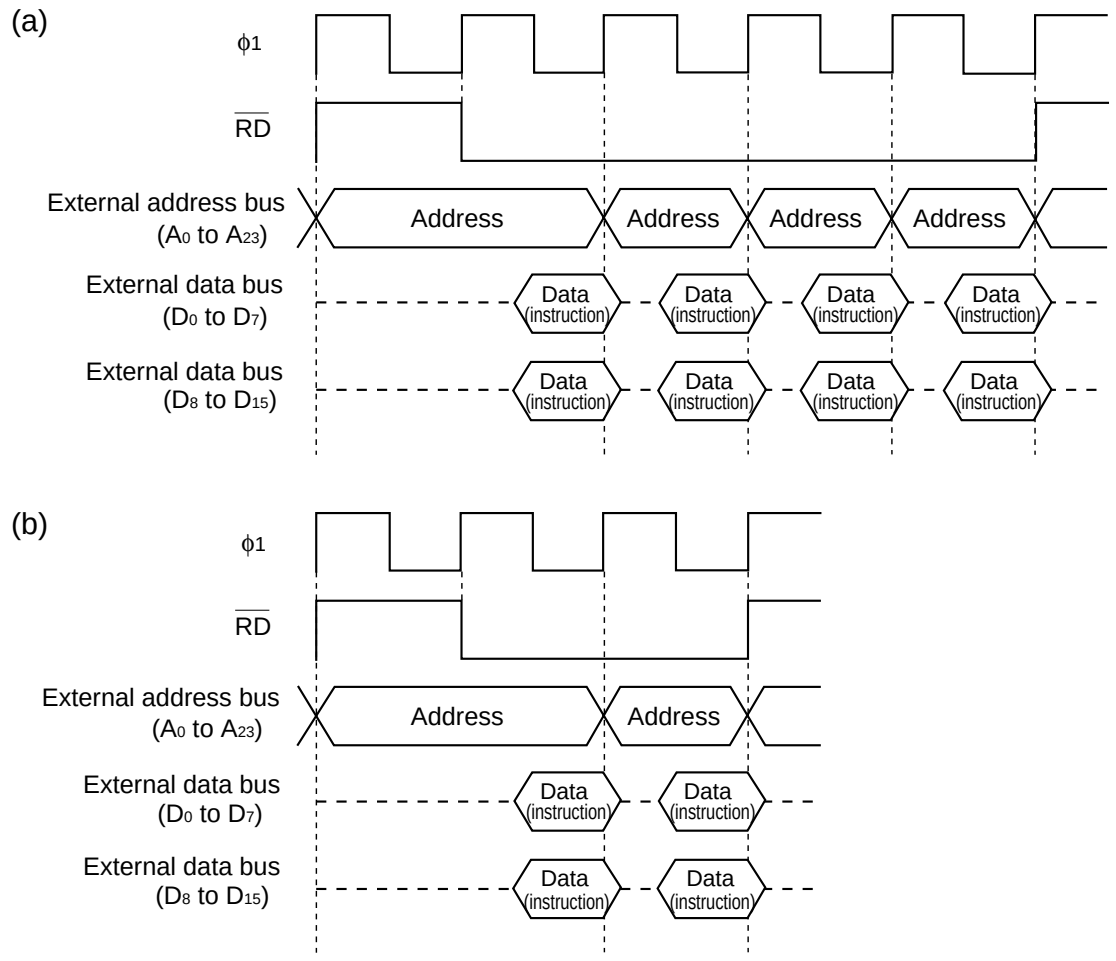


Fig. 20 Area $\overline{CS_1}/\overline{CS_2}$ (mode 0) and area $\overline{CS_3}$



Note: The above is applied when 0 wait is selected.

Fig. 21 Operating waveform example at burst ROM access

INTERRUPTS

Table 13 shows the interrupt sources and the corresponding interrupt vector addresses. Reset is also described as a type of interrupt in this section, too.

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

Interrupts other than reset, watchdog timer, zero divide, $\overline{NMI}$, and address matching detection all have interrupt control registers. Table 14 shows the addresses of the interrupt control registers, and Figure 22 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 $\overline{NMI}$ can be cleared by software.

Any of $\overline{INT_2}$ through $\overline{INT_0}$ interrupt requests is generated by an external input.

$\overline{INT_2}$ to $\overline{INT_0}$ 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.

Timer and UART interrupts are described in the respective section.

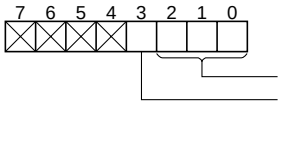
The priorities of interrupts when multiple interrupt requests are caused simultaneously are partially fixed by hardware, but, the other can be adjusted by software as shown in Figure 23.

The hardware priority is fixed as the following:

reset > $\overline{NMI}$ > watchdog timer > other interrupts

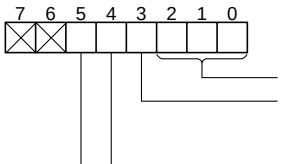
Table 13 Interrupt sources and interrupt vector addresses

Interrupts	Vector addresses
DMA3	$00FFC0_{16}$ $00FFC1_{16}$
DMA2	$00FFC2_{16}$ $00FFC3_{16}$
DMA1	$00FFC4_{16}$ $00FFC5_{16}$
DMA0	$00FFC6_{16}$ $00FFC7_{16}$
Address matching detection interrupt	$00FFCA_{16}$ $00FFCB_{16}$
$\overline{INT_4}$ external interrupt	$00FFD0_{16}$ $00FFD1_{16}$
$\overline{INT_3}$ 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	$00FFEC_{16}$ $00FFED_{16}$
$\overline{INT_2}$ external interrupt	$00FFEE_{16}$ $00FFEF_{16}$
$\overline{INT_1}$ external interrupt	$00FFF0_{16}$ $00FFF1_{16}$
$\overline{INT_0}$ external interrupt	$00FFF2_{16}$ $00FFF3_{16}$
$\overline{NMI}$ external interrupt	$00FFF4_{16}$ $00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$ $00FFF7_{16}$
$\overline{DBC}$ (Do not select.)	$00FFF8_{16}$ $00FFF9_{16}$
Break instruction (Do not select.)	$00FFFA_{16}$ $00FFFB_{16}$
Zero divide	$00FFFC_{16}$ $00FFFD_{16}$
Reset	$00FFFE_{16}$ $00FFFF_{16}$



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

Bit configuration of interrupt control registers for DMA0 to DMA3, A-D converter, UART0, UART1, timers A0 to A4, and timers B0 to B2, and $\overline{INT_3}$, $\overline{INT_4}$.



Interrupt priority level
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

Bit configuration of interrupt control registers for $\overline{INT_0}$– $\overline{INT_2}$.

Fig. 22 Bit configuration of interrupt control register

Table 14. Addresses of interrupt control registers

Interrupt control registers	Addresses
$\overline{INT_3}$ interrupt control register	00006E16
$\overline{INT_4}$ interrupt control register	00006F16
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	00007A16
Timer B1 interrupt control register	00007B16
Timer B2 interrupt control register	00007C16
$\overline{INT_0}$ interrupt control register	00007D16
$\overline{INT_1}$ interrupt control register	00007E16
$\overline{INT_2}$ interrupt control register	00007F16
DMA0 interrupt control register	0000B2 16
DMA1 interrupt control register	0000B3 16
DMA2 interrupt control register	0000B4 16
DMA3 interrupt control register	0000B5 16

Interrupts caused by the address matching detection and when dividing by zero are software interrupts and are not included in Figure 23.

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 24 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, $\overline{NMI}$, 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, $\overline{NMI}$, and address match detection interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 15.

The interrupt request bit and the interrupt priority level of each interrupt source are sampled and latched at each operation code fetch cycle while ϕ 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.

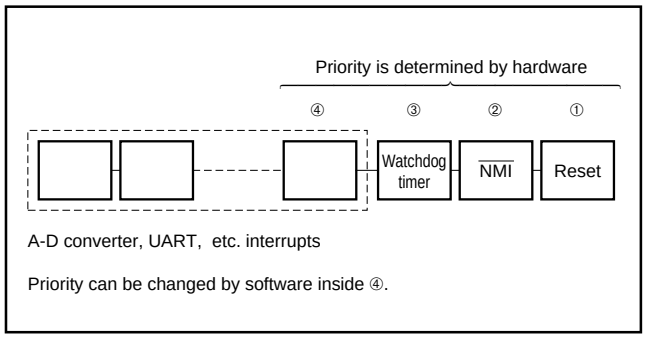


Fig. 23 Interrupt priority

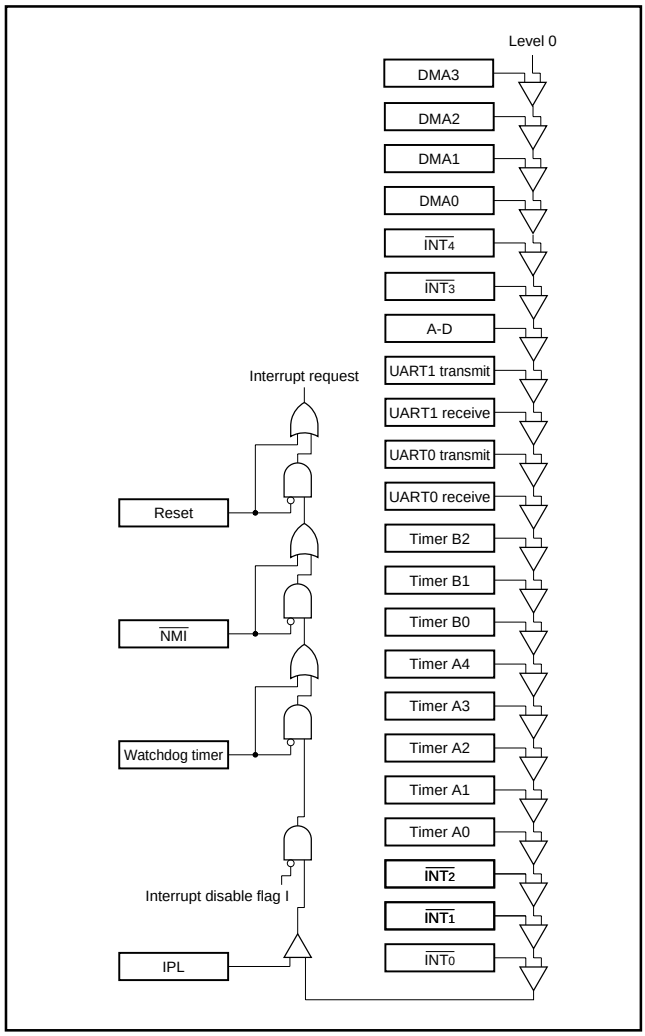


Fig. 24 Interrupt priority detection

As shown in Figure 25, 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 26. Table 16 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 15. Value loaded in processor interrupt level (IPL) during an interrupt

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

Table 16. 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 ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

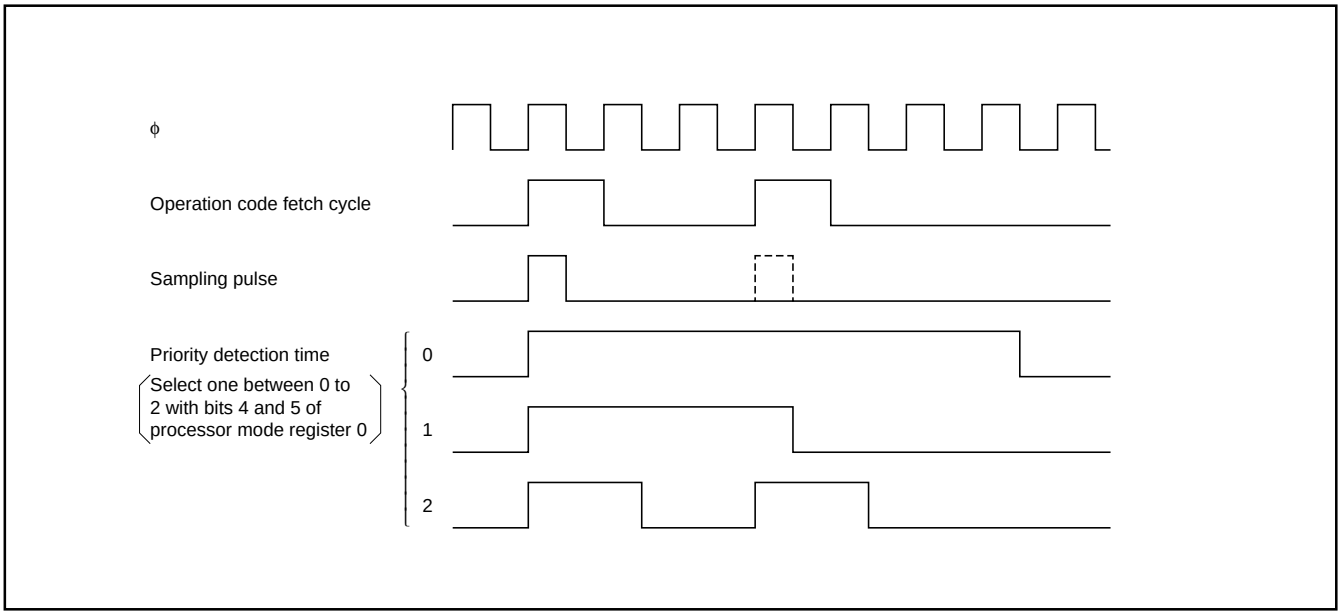


Fig. 25 Interrupt priority detection time

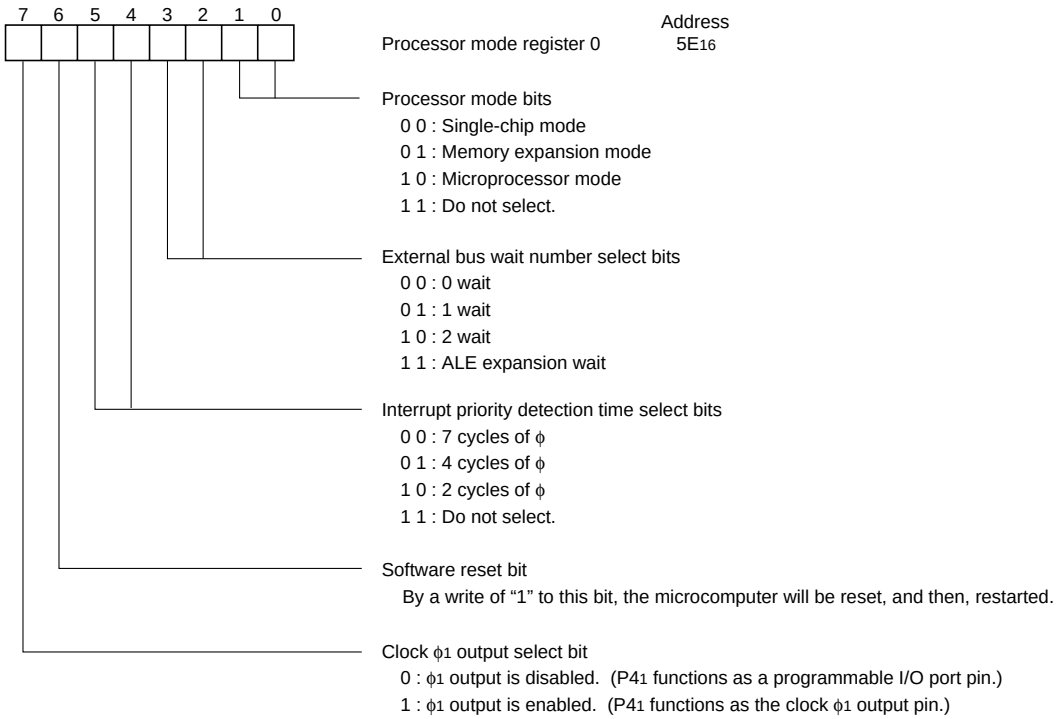


Fig. 26 Bit configuration of processor mode register 0

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 27 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 28 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 29 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 17.)

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

Figure 30 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.

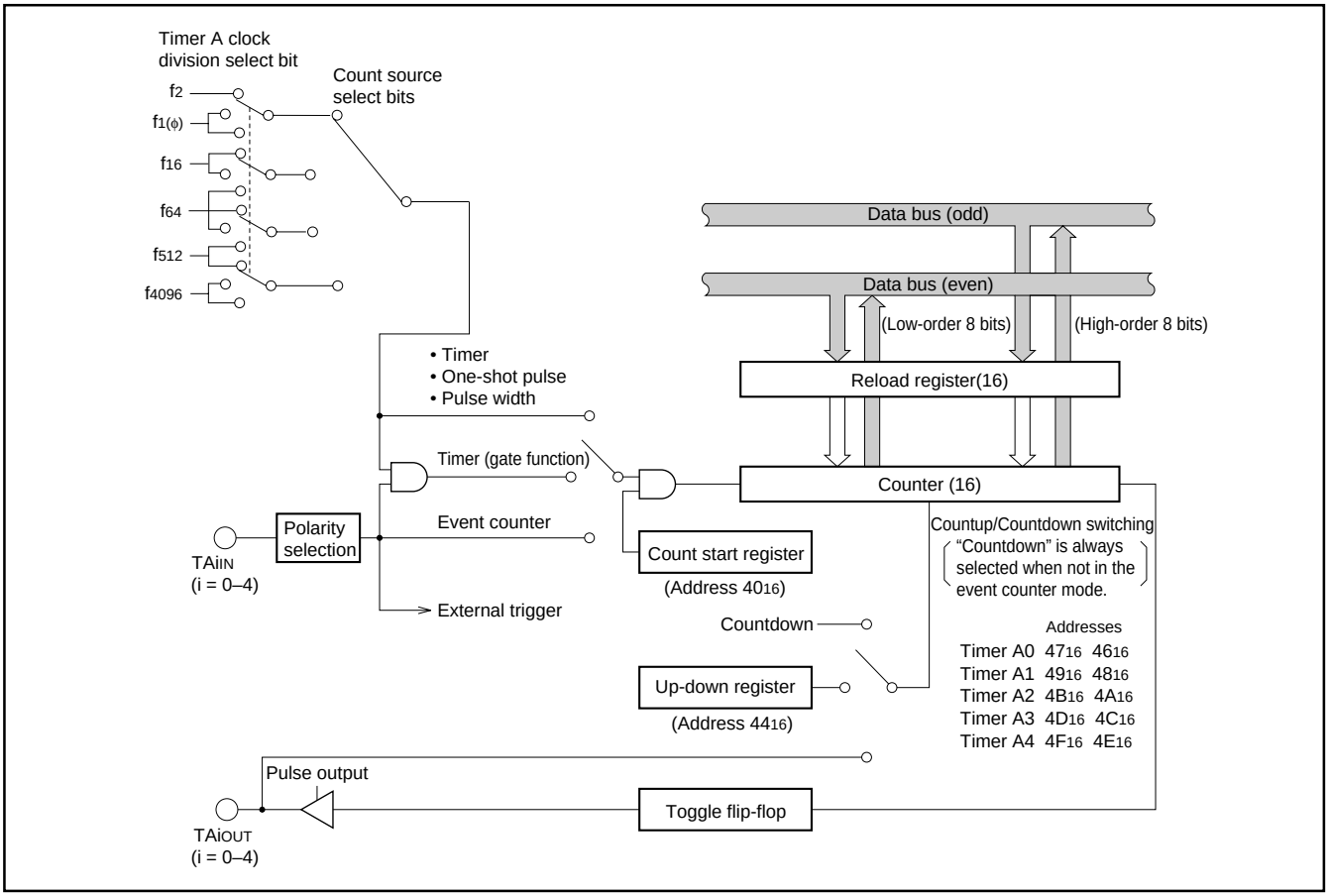


Fig. 27 Block diagram of timer A

When bit 2 of the timer Ai mode register is “1”, the output is generated from TAiOUT 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 TAiOUT pin.

When bit 2 is “0”, TAiOUT can be used as a normal port pin. When bit 4 is “0”, TAiIN can be used as a normal port pin.

When bit 4 is “1”, counting is performed only while the input signal from the TAiIN pin is “H” or “L” as shown in Figure 31. Therefore, this can be used to measure the pulse width of the TAiIN 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 TAiIN 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 TAiIN 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).

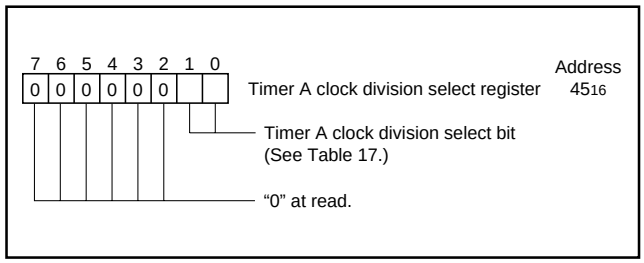


Fig. 28 Bit configuration of timer A clock division select register

Table 17. 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	$f_{1(\phi)}$	$f_{1(\phi)}$	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.

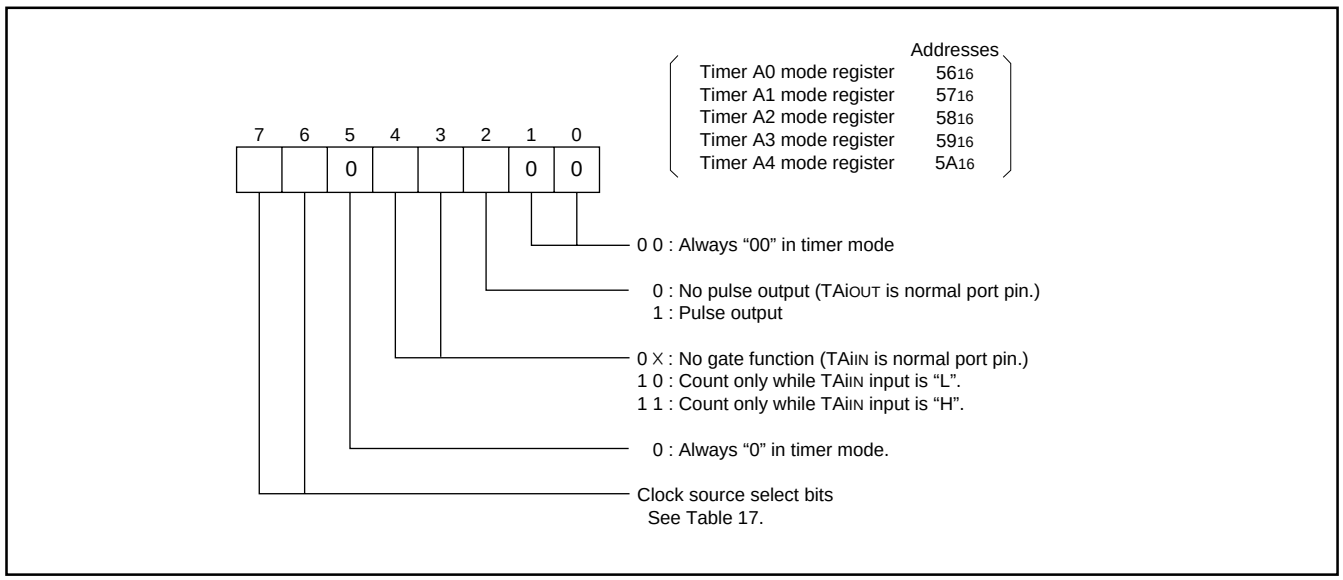


Fig. 29 Bit configuration of timer Ai mode register during timer mode

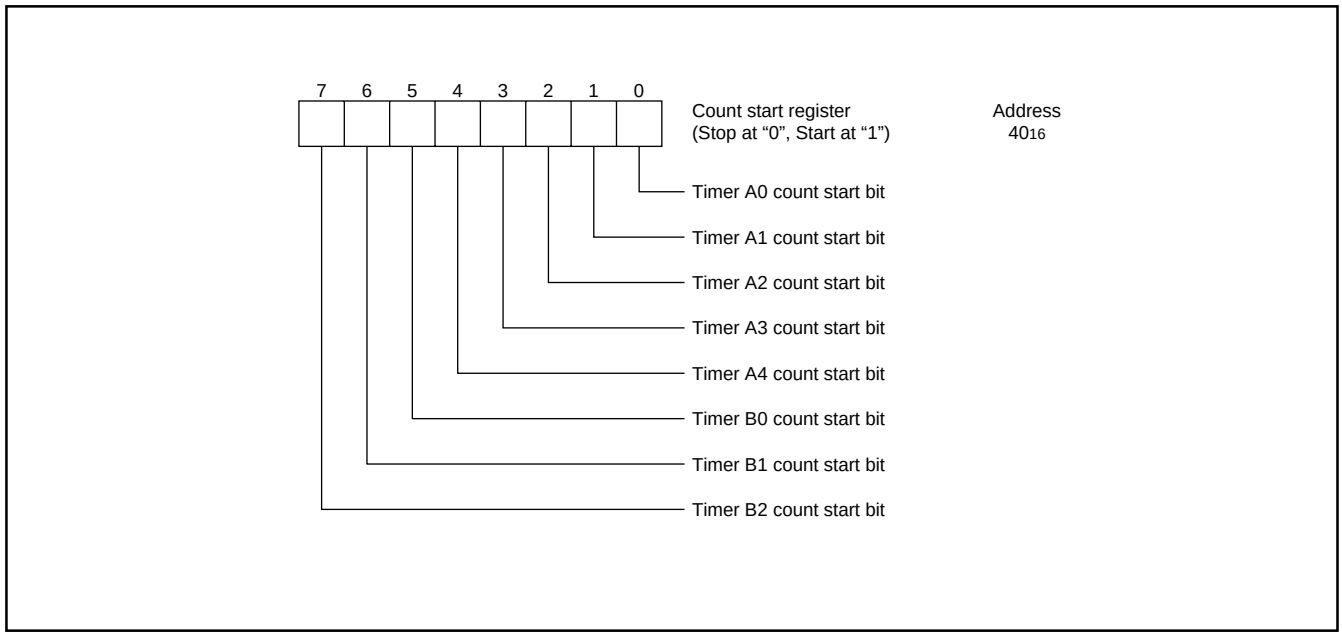


Fig. 30 Bit configuration of count start register

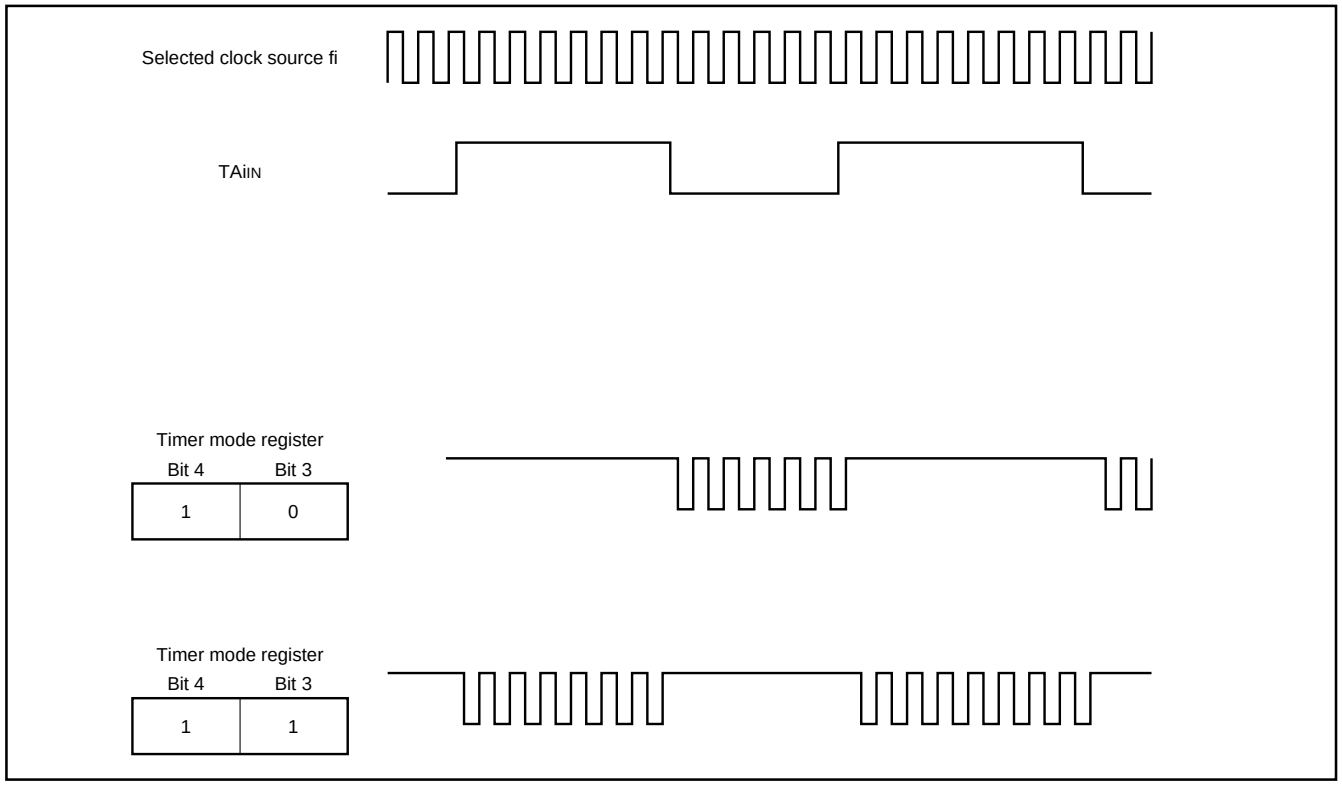


Fig. 31 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 32 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 30 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 33 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.

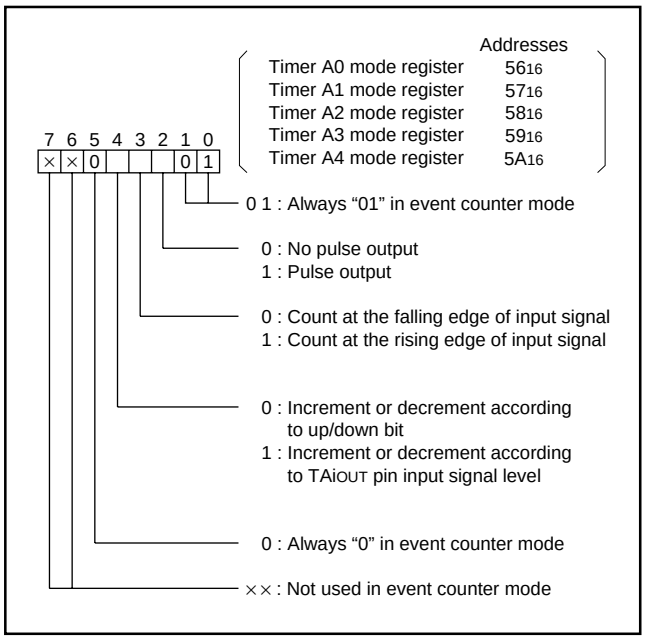


Fig. 32 Bit configuration of timer Ai mode register during event counter mode

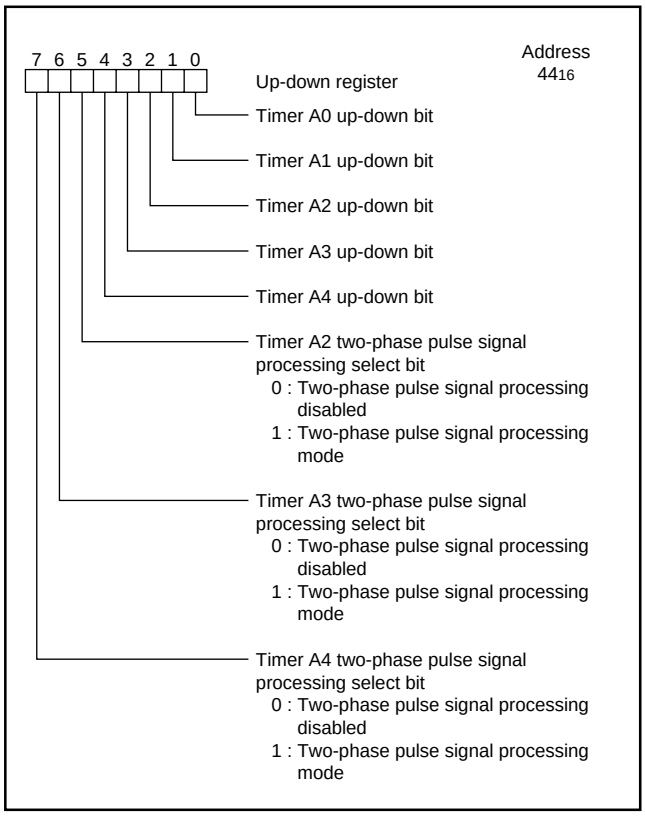


Fig. 33 Bit configuration of up-down register

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 TAjOUT (j = 2 to 4) pin and TAjIN pin respectively.

When timers A2 and A3 are used, as shown in Figure 34, the count is incremented when a rising edge is input to the TAkIN pin after the level of TAkOUT(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 35, when a phase-related pulse with a rising edge input to the TA4IN pin is input after the level of TA4OUT pin changes from “L” to “H”, the count is incremented at the respective rising edge and falling edge of the TA4OUT pin and TA4IN pin.

When a phase-related pulse with a falling edge input to the TA4OUT pin is input after the level of TA4IN pin changes from “H” to “L”, the count is decremented at the respective rising edge and falling edge of the TA4IN pin and TA4OUT 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 36.) 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.

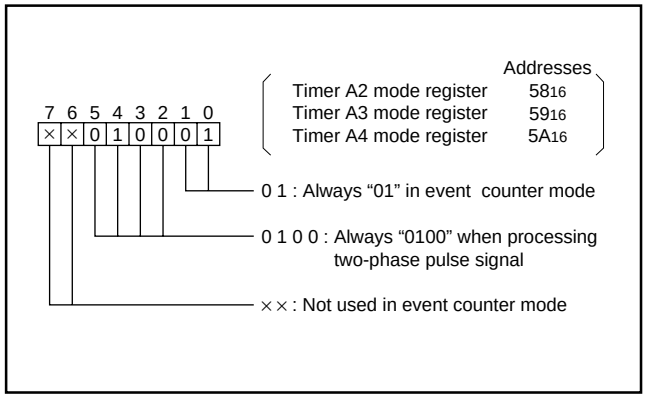


Fig. 36 Bit configuration of timer Aj mode register when performing two-phase pulse signal processing in event counter mode

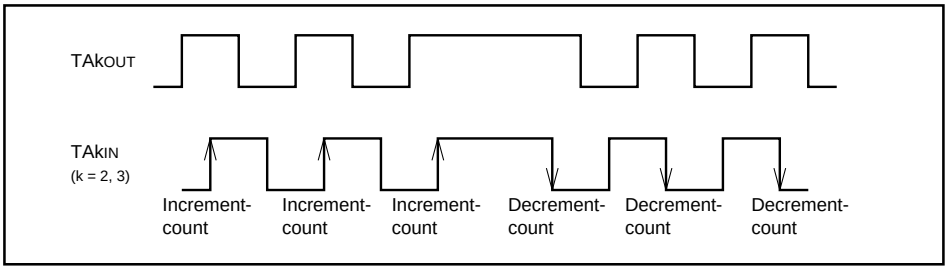


Fig. 34 Two-phase pulse processing operation of timers A2 and A3

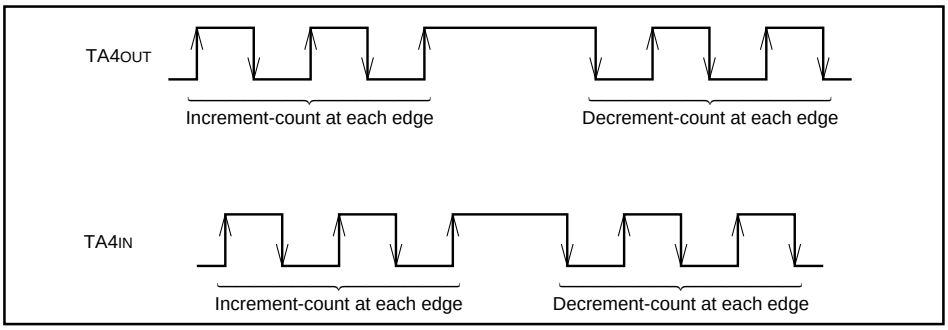


Fig. 35 Two-phase pulse processing operation of timer A4

(3) One-shot pulse mode [10]

Figure 37 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 38 shows the bit configuration of the one-shot start register.

As shown in Figure 39, 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 17.)

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 40, 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.

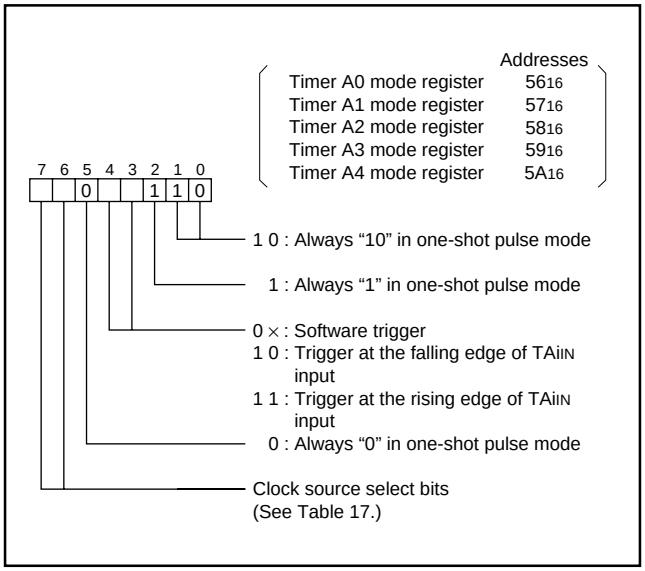


Fig. 37 Bit configuration of timer Ai mode register during one-shot pulse mode

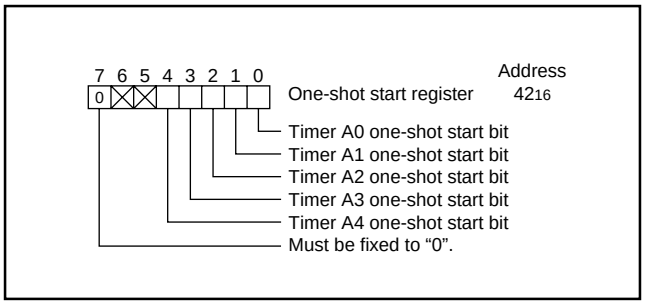


Fig. 38 Bit configuration of one-shot start register

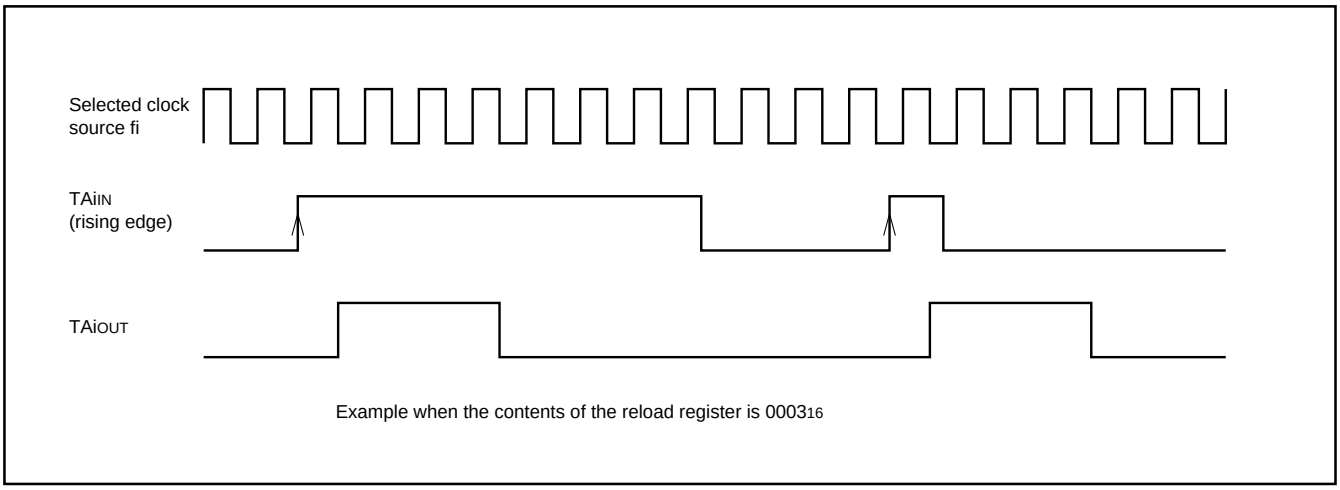


Fig. 39 Pulse output example when external rising edge is selected

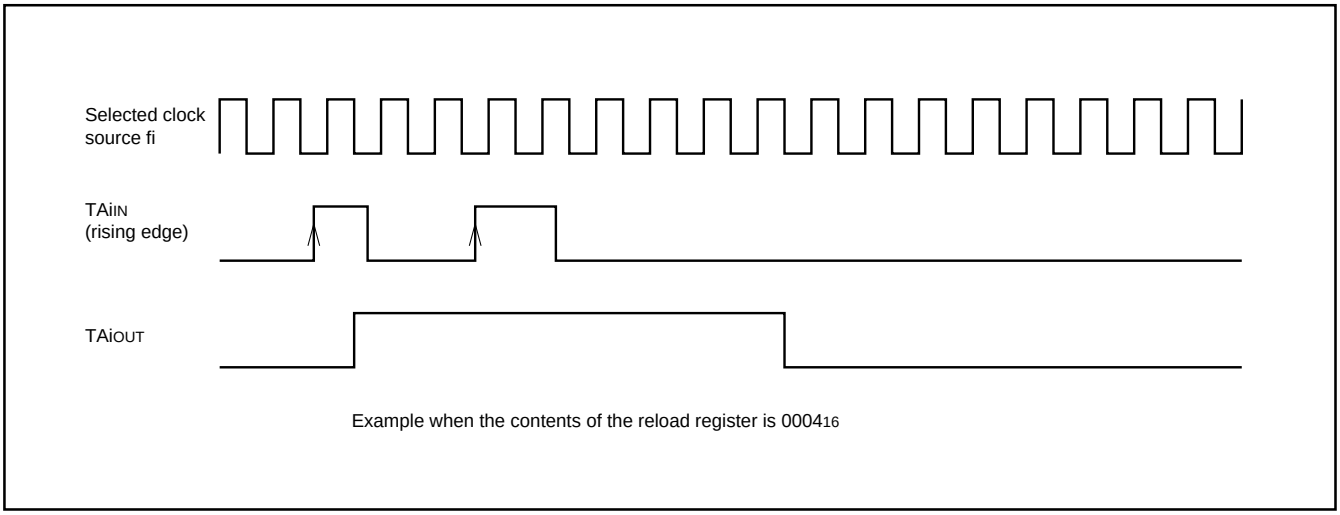


Fig. 40 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode [11]

Figure 41 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 TAiIN 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 TAiOUT 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 TAiIN 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 42 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 17.) A pulse is generated when the counter reaches 0000_{16} as shown in Figure 43. At the same time, the contents of the reload register is transferred to the counter and count is continued.

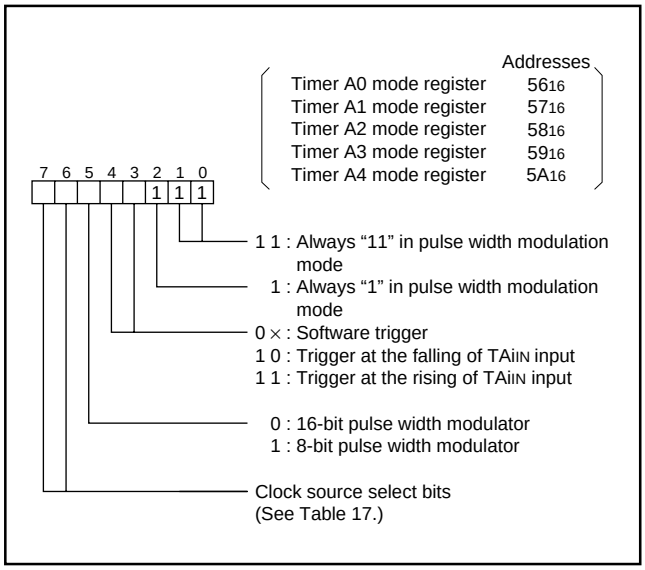


Fig. 41 Bit configuration of timer Ai mode register during pulse width modulation mode

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

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

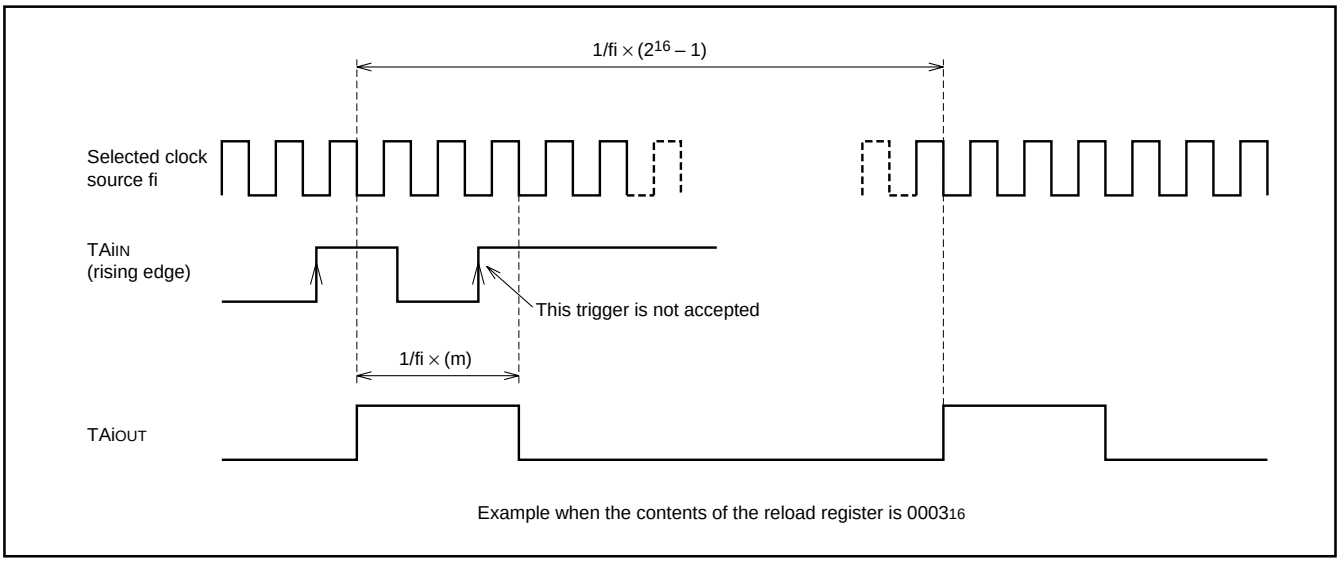


Fig. 42 16-bit length pulse width modulator output pulse example

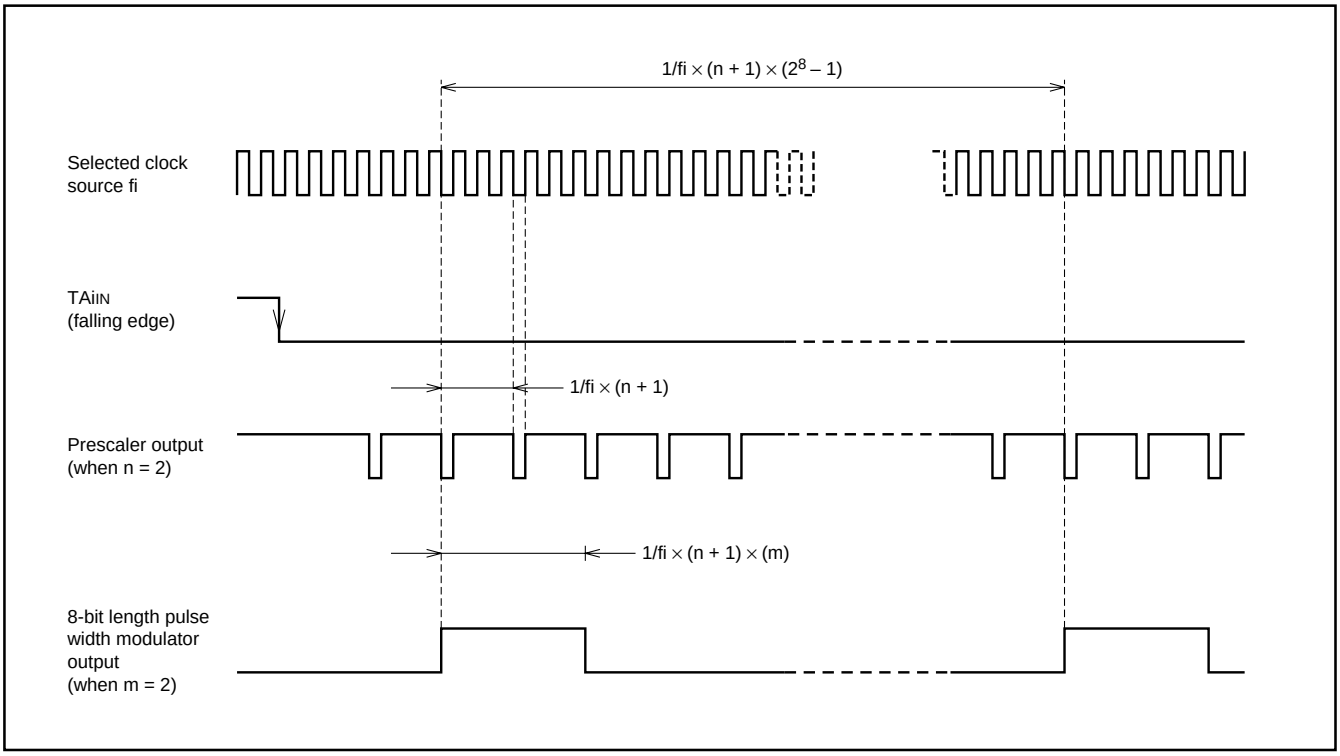


Fig. 43 8-bit length pulse width modulator output pulse example

TIMER B

Figure 44 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 45 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.

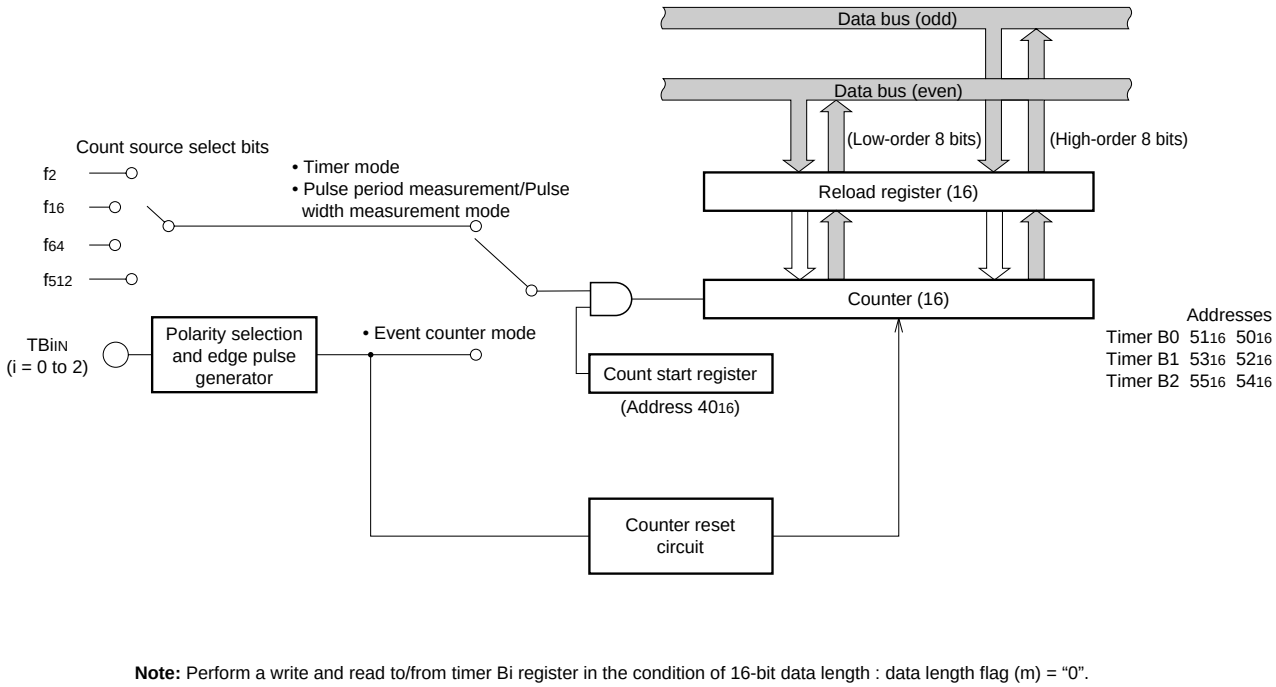
Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start bit is “1” and stops when “0”.

As shown in Figure 30, 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.



Note: Perform a write and read to/from timer Bi register in the condition of 16-bit data length : data length flag (m) = “0”.

Fig. 44 Block diagram of timer B

(2) Event counter mode [01]

Figure 46 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 TBiIN 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.

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

Figure 47 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 TBiIN 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 48, when the fall of the input signal from TBiIN 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 TBiIN pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 49.

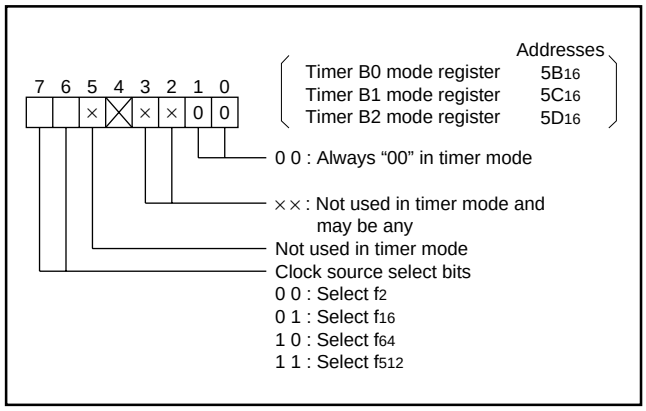


Fig. 45 Bit configuration of timer Bi mode register during timer mode

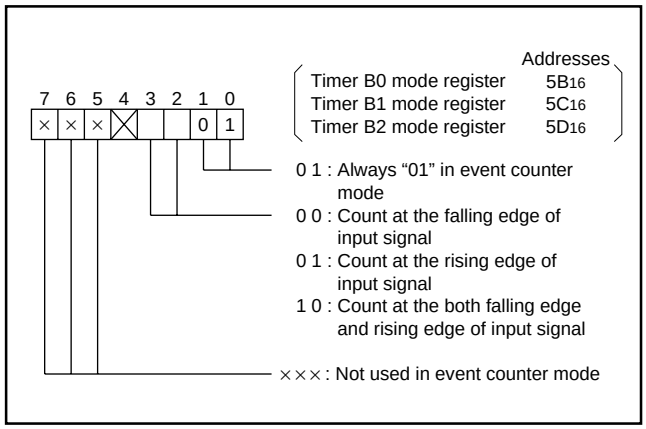


Fig. 46 Bit configuration of timer Bi mode register during event counter mode

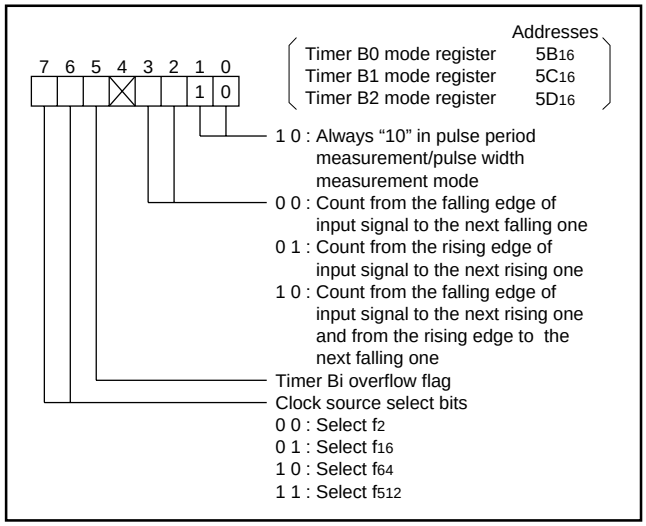


Fig. 47 Bit configuration of timer Bi mode register during pulse period measurement/pulse width measurement mode

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 TBiIN 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.

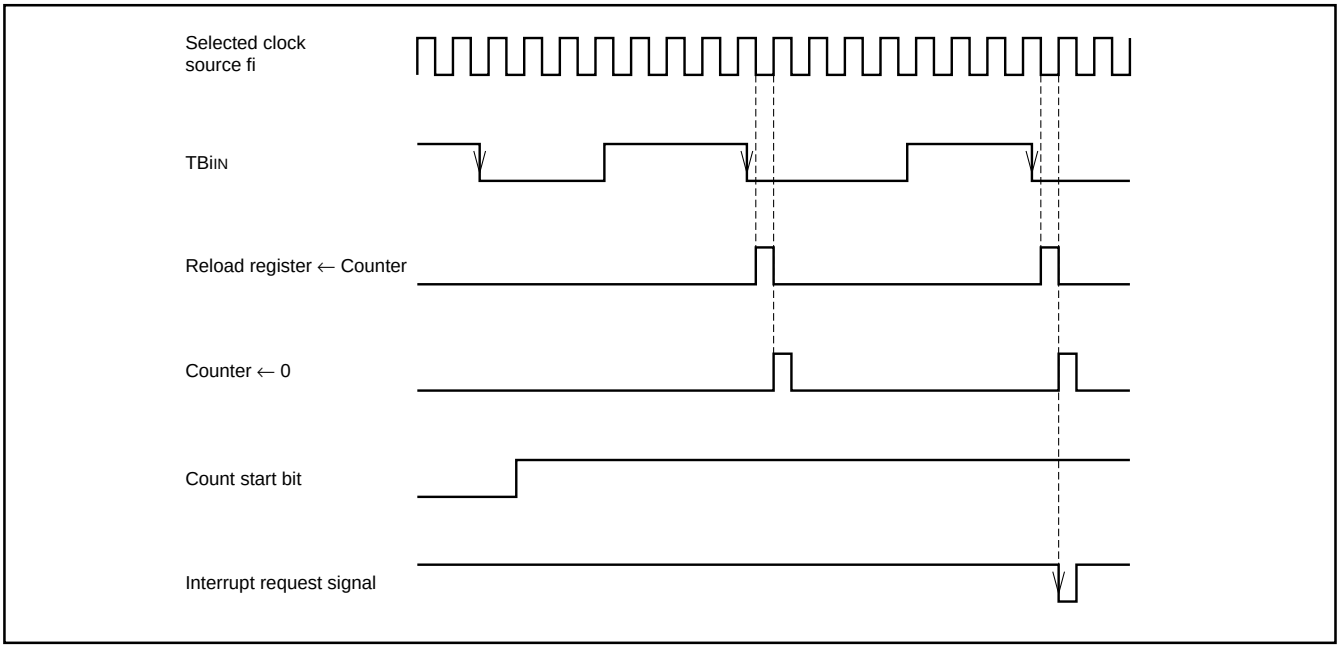


Fig. 48 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

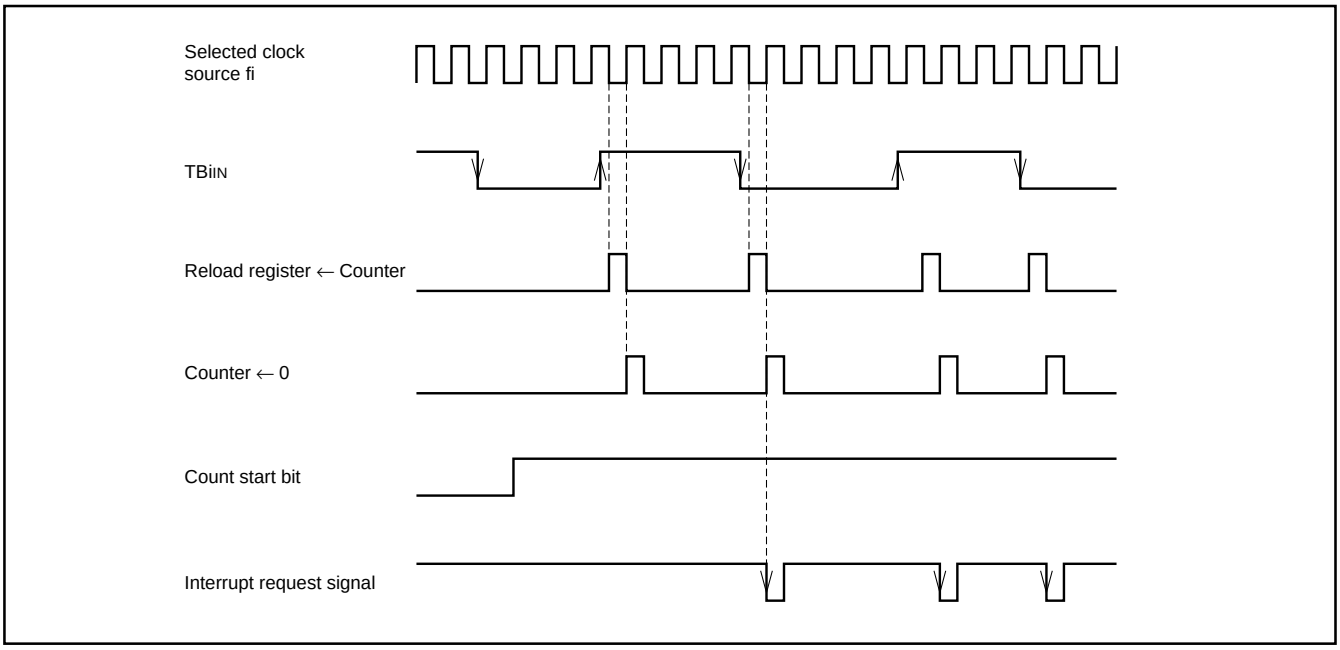


Fig. 49 Pulse width measurement mode operation

SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 50 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 51 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 52 and 53 show the block diagrams of the receiver/transmitter.

Figure 54 shows the bit configuration of the UARTi transmit/receive control register.

Each communication method is described below.

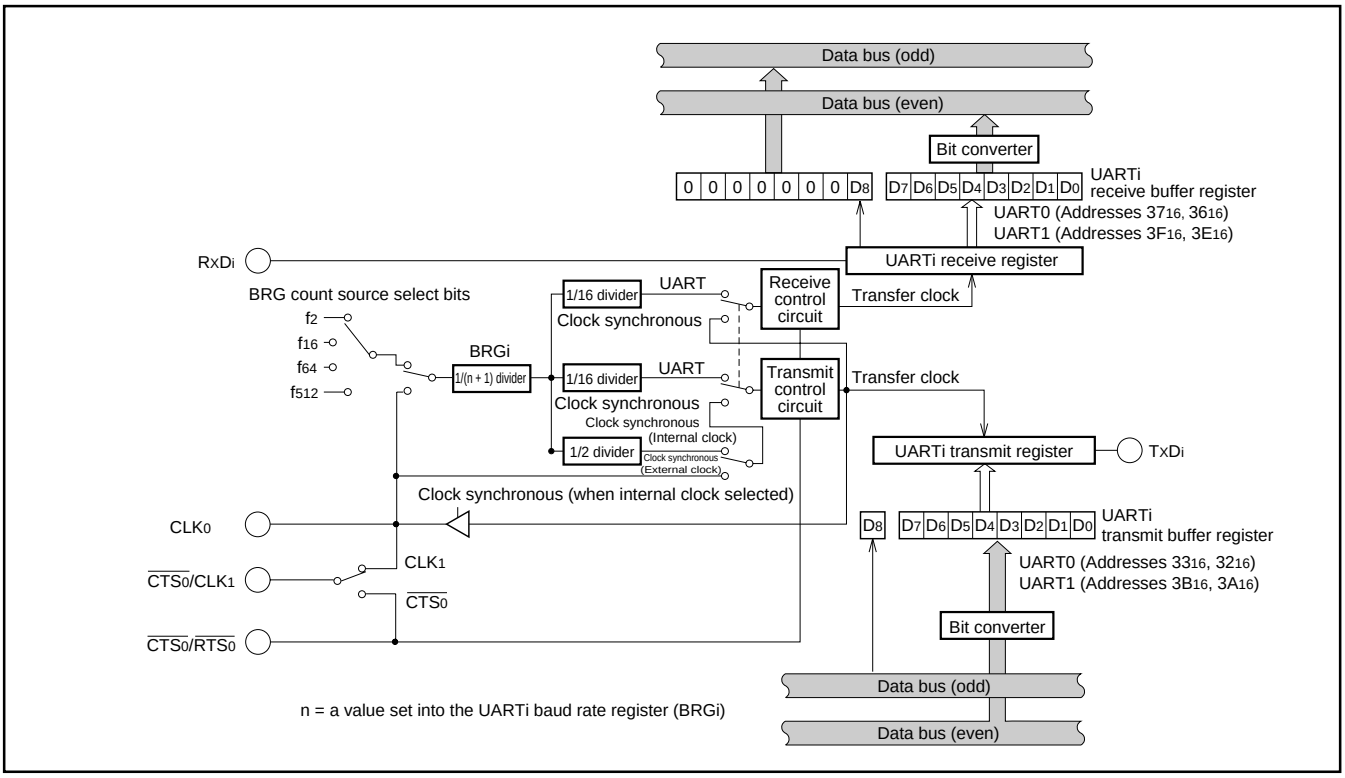


Fig. 50 Block diagram of serial I/O port

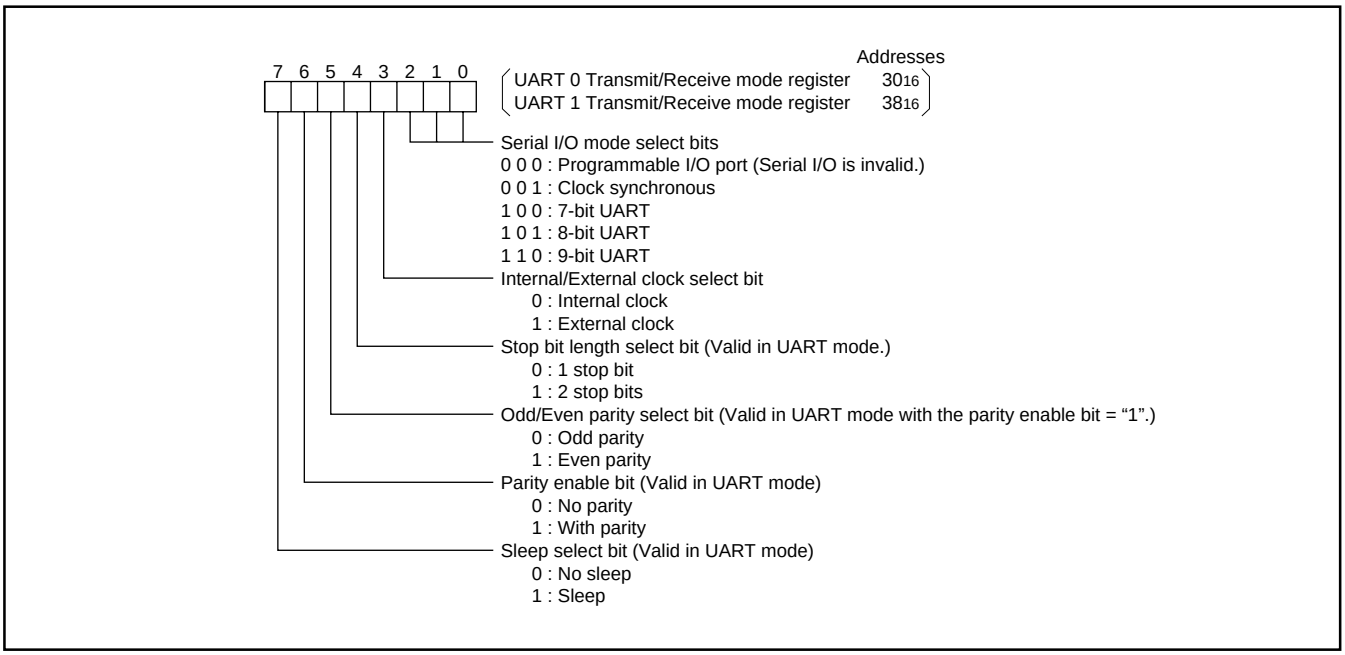


Fig. 51 Bit configuration of UARTi transmit/receive mode register

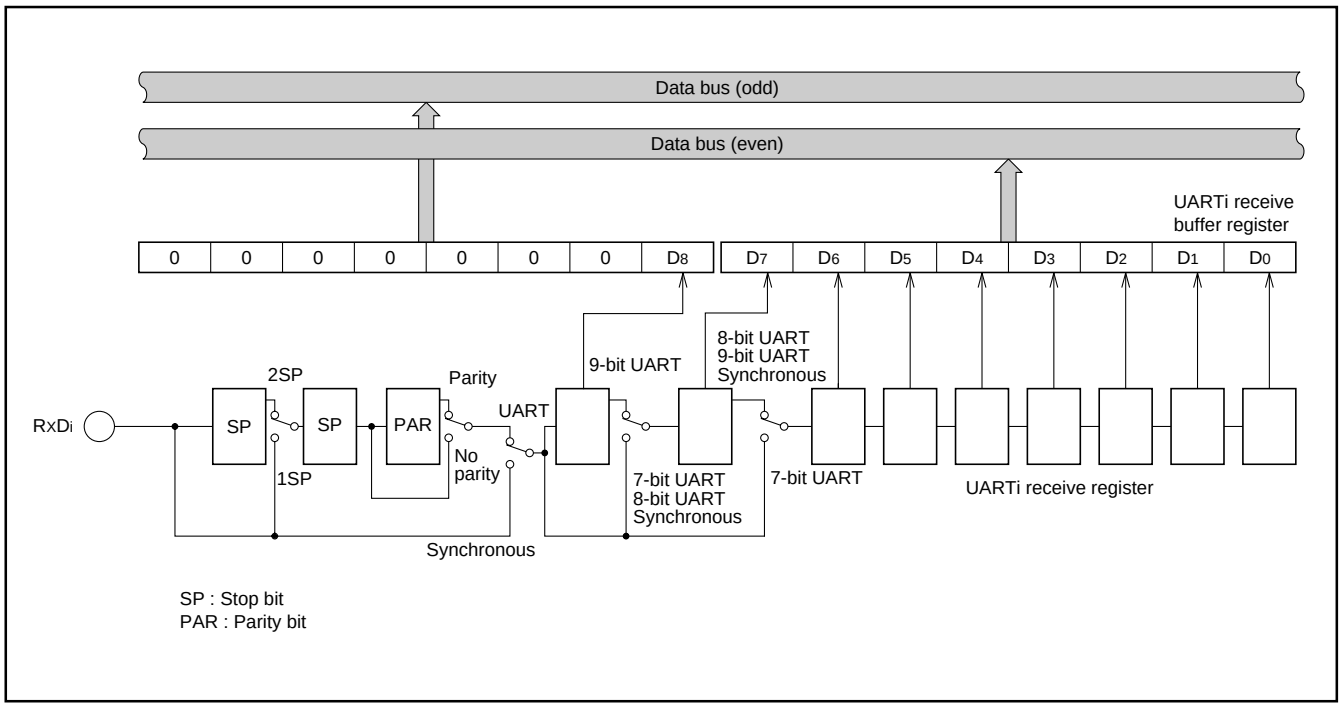


Fig. 52 Block diagram of receiver

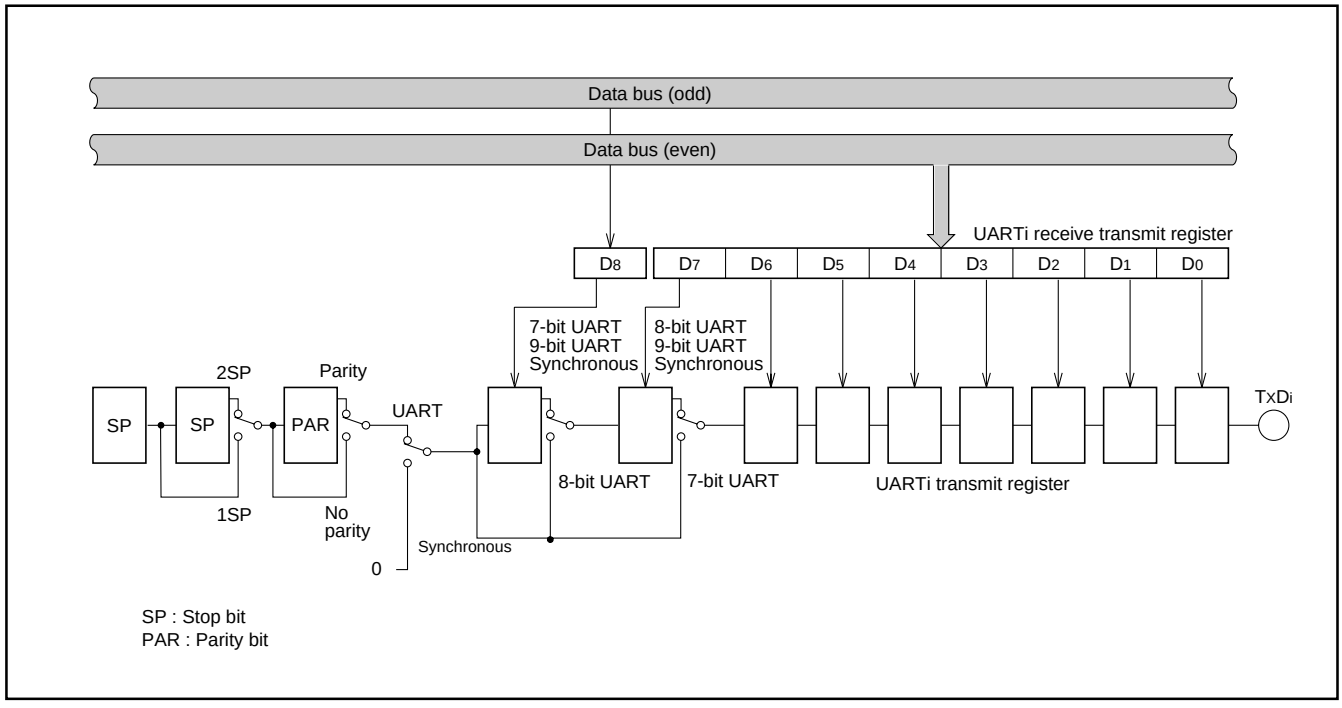
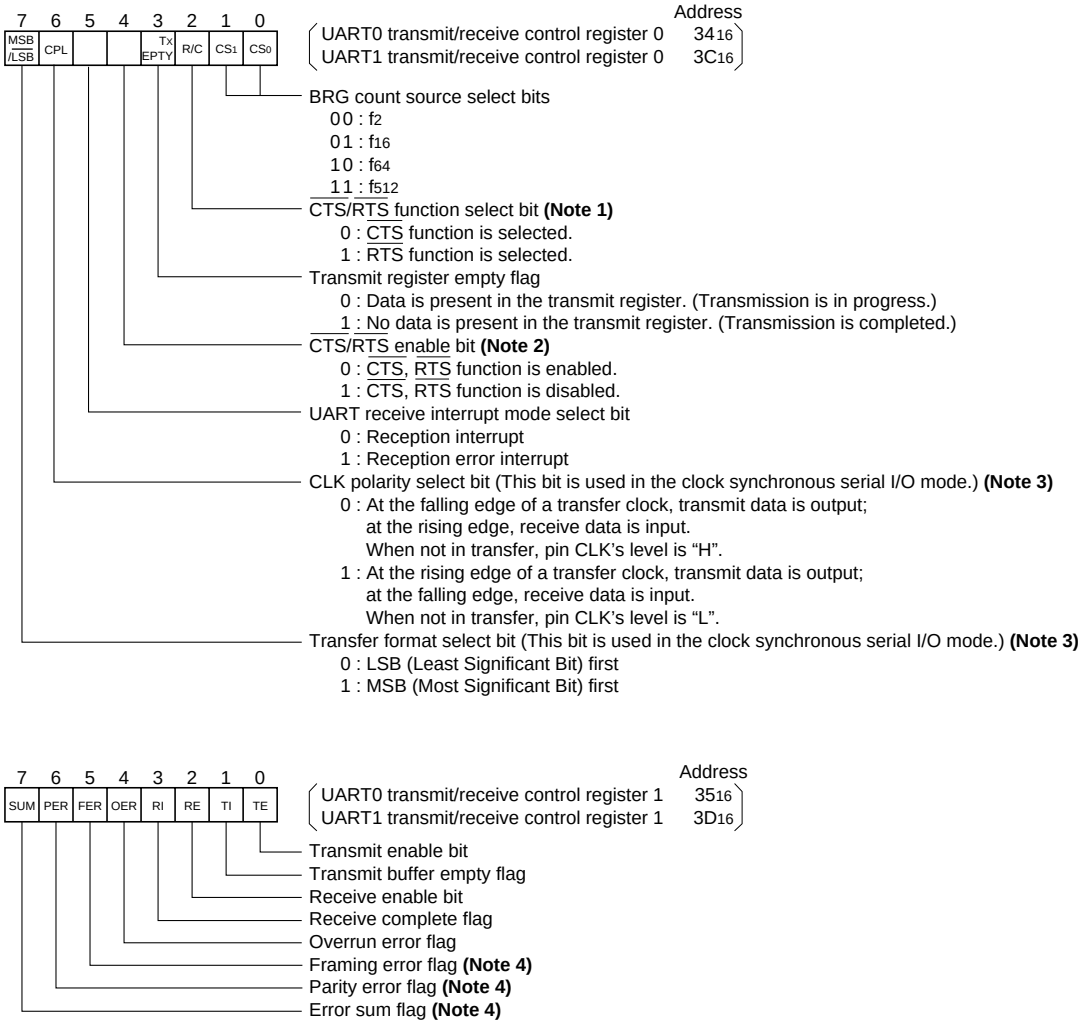


Fig. 53 Block diagram of transmitter



Notes 1: Valid when the CTS/RTS enable bit = "0".
2: Fix this bit to "1" in UART1 transmit/receive control register 0. (UART1 is not equipped with the CTS/RTS function.)
3: Fix this bit to "0" in UART mode or when serial I/O is invalid.
4: Valid in UART mode.

Fig. 54 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 55 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 50, 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.

UART0 is equipped with the $\overline{CTS}$ and $\overline{RTS}$ functions.

UART1 is not equipped with the $\overline{CTS}/\overline{RTS}$ function.

Bit 4 of the UART0 transmit/receive control register 0 is used to determine whether to use $\overline{CTS_0}$ or $\overline{RTS_0}$ signal. Bit 4 must be "0" when $\overline{CTS_0}$ or $\overline{RTS_0}$ signal is used. Bit 4 must be "1" when $\overline{CTS_0}$ and $\overline{RTS_0}$ signals are not used. When $\overline{CTS_0}$ and $\overline{RTS_0}$ signals are not used, $\overline{CTS_0}/\overline{RTS_0}$ pin can be used as a normal port pin.

When using this pin as pin $\overline{CTS_0}/\overline{RTS_0}$, :

- If bit 2 of the UART0 transmit/receive control register 0 is cleared to "0", $\overline{CTS_0}$ input is selected.
- If bit 2 is set to "1", $\overline{RTS_0}$ output is selected.

Figure 56 shows the bit configuration of the $\overline{CTS}/\overline{RTS}$ separate select register. By using bit 0 of the $\overline{CTS}/\overline{RTS}$ separate select register ($\overline{CTS}/\overline{RTS}$ separate select bit), the function of the $\overline{CTS_0}/\overline{RTS_0}$ pin can be separated into two functions. When bit 0 = "1", the above separation is performed. When bit 0 = "0", no separation is performed. When the $\overline{CTS_0}/\overline{RTS_0}$ pin is separated, $\overline{RTS_0}$ function is selected. When the $\overline{CTS_0}/CLK_1$ pin is separated, $\overline{CTS_0}$ function is selected.

The following describes the case where the $\overline{CTS}$ and $\overline{RTS}$ signals are used. When the $\overline{CTS}$ and $\overline{RTS}$ signals are not used, however, the $\overline{CTS}$ input is not necessary, and there is no $\overline{RTS}$ output.

Since UART1 is not equipped with the $\overline{CTS}/\overline{RTS}$ function, UART1 is regarded as the case where the $\overline{CTS}$ and $\overline{RTS}$ signals are not used.

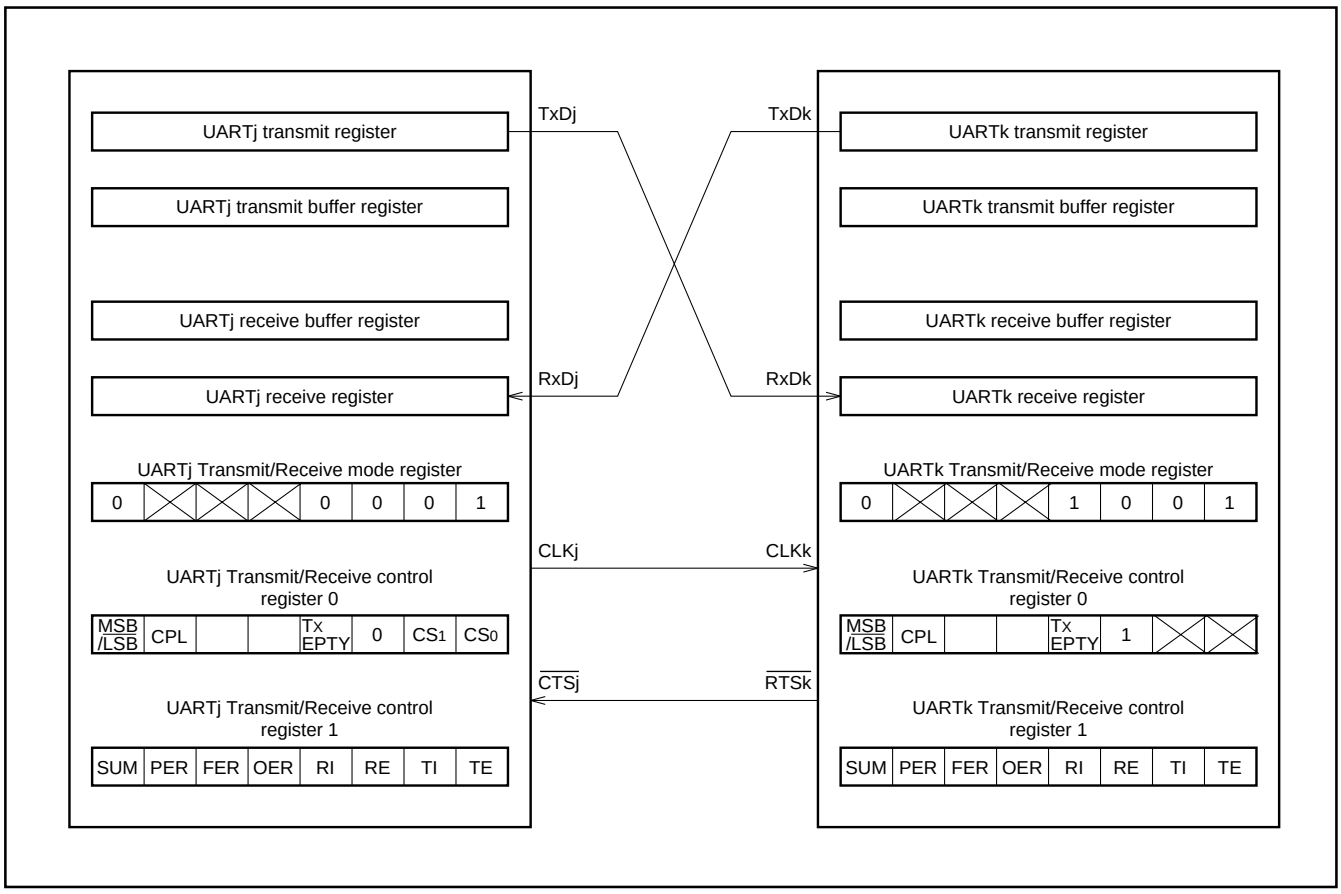


Fig. 55 Clock synchronous serial communication

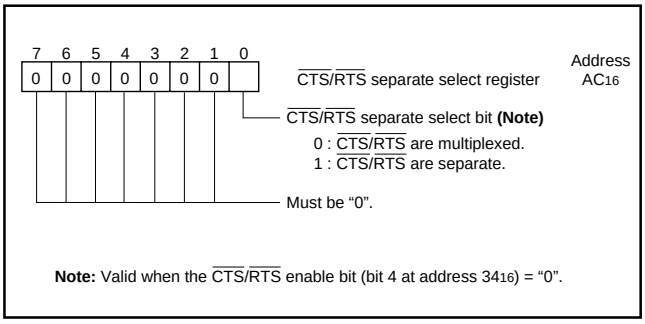


Fig. 56 Bit configuration of $\overline{\text{CTS}}/\overline{\text{RTS}}$ separate select register

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 $\overline{\text{CTS}}$j 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 57, 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 $\overline{\text{CTS}}$j input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, TIj flag, and $\overline{\text{CTS}}$j is checked while the TENDj signal (shown in Figure 57) 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 $\overline{\text{RTS}}$k output is "H" when the REK flag is "0". When the REK flag is set to "1", the $\overline{\text{RTS}}$k output becomes "L". This informs the transmitter side that reception becomes enabled. When the receive operation starts, the $\overline{\text{RTS}}$k 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 $\overline{\text{RTS}}$k 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 50, 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.

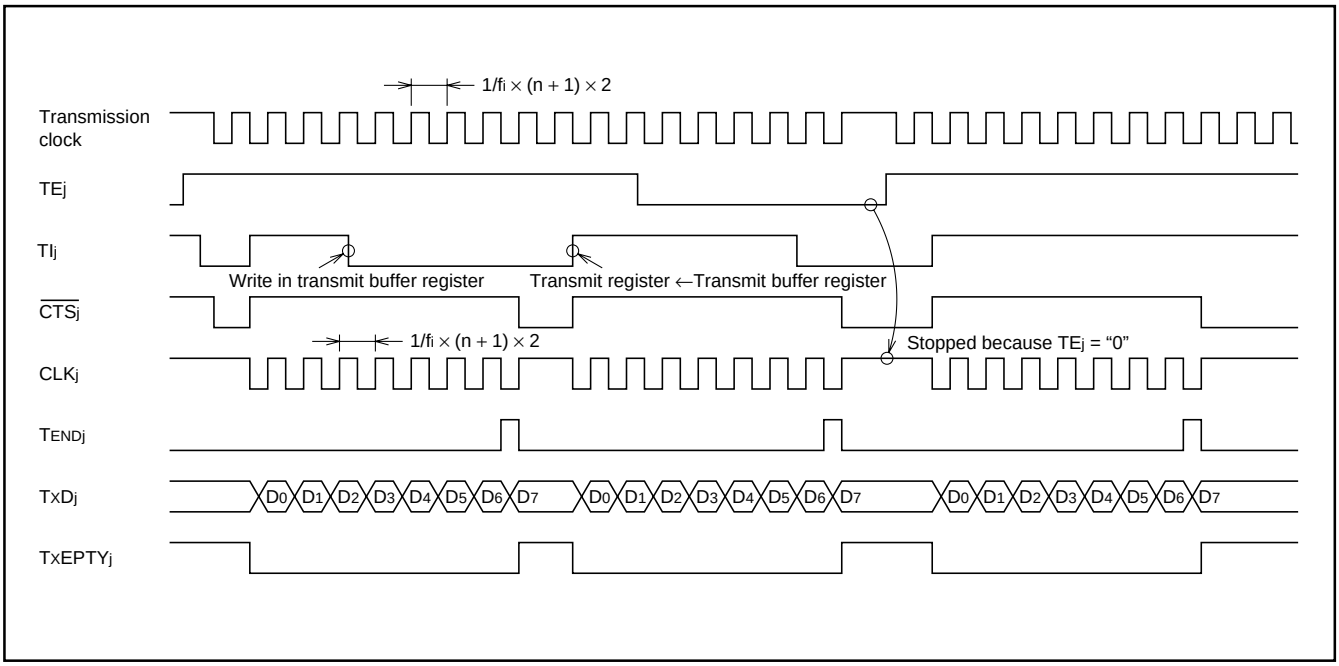


Fig. 57 Clock synchronous serial I/O timing

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 (UARTk 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".)

Note that a DMA request is affected by the UART receive interrupt mode select bit if the UARTi reception is selected as a DMA request source of the DMA controller.

When the UARTk receive interrupt mode select bit is cleared to "0", a DMA request is generated at each UART reception. When the UARTk receive interrupt mode select bit is set to "1", a DMA request is generated only at normal UART reception. (In other words, no DMA request is generated when an error has occurred.)

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.

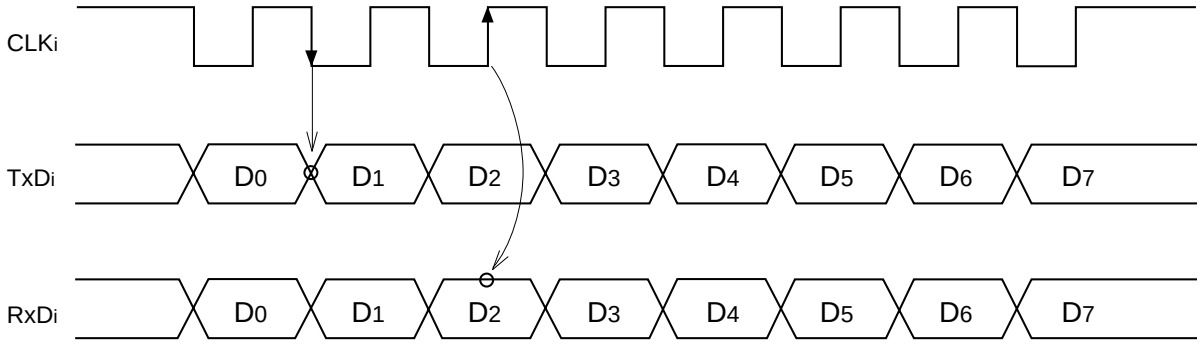
As shown in Figure 58, when bit 6 = "0", the polarity is as follows:

- 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:

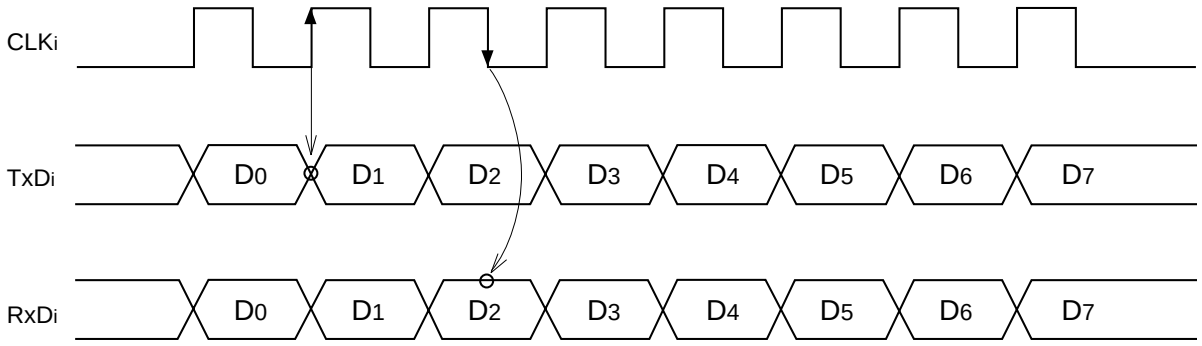
- In transmission, transmit data is output at the rising edge of CLKj.
- In reception, receive data is input at the rising edge of CLKk.
- When not in transfer, CLKi is at "L" level.

■ 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. 58 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 59 shows the connection relation.

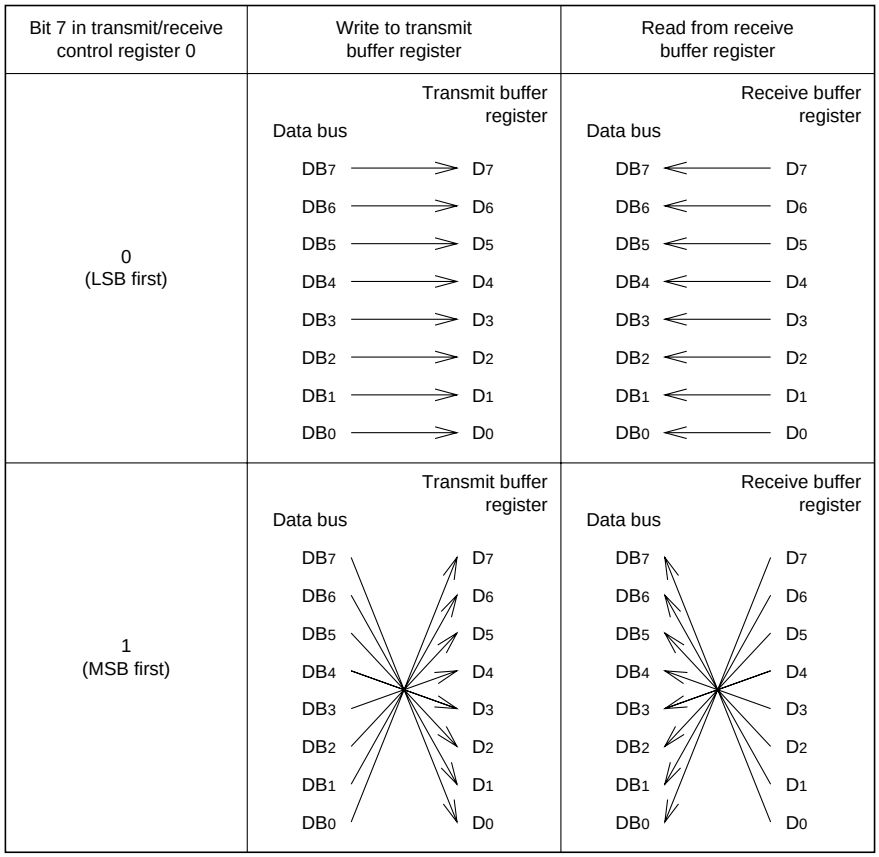


Fig. 59 Connection relation between transmit buffer register, receive buffer register, and data bus

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},

$$\text{Bit Rate} = (\text{fi or } f_{EXT}) / \{(n + 1) \times 16\}$$

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.

The function and select method of the $\overline{CTS}/\overline{RTS}$ pin are the same as those of the clock synchronous serial communication mode.

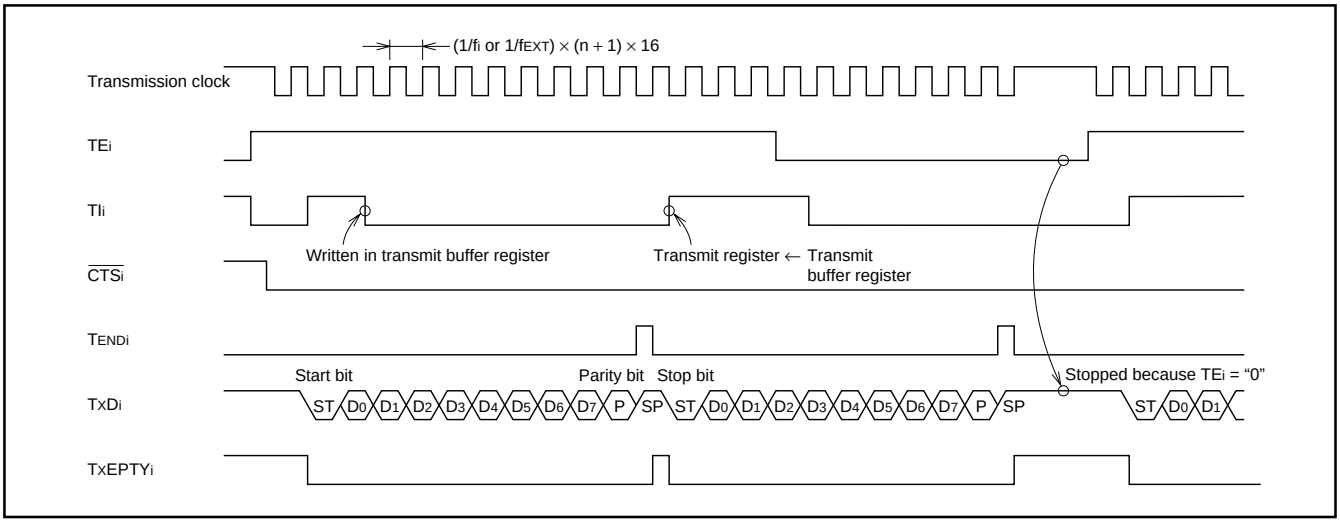


Fig. 60 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit selected

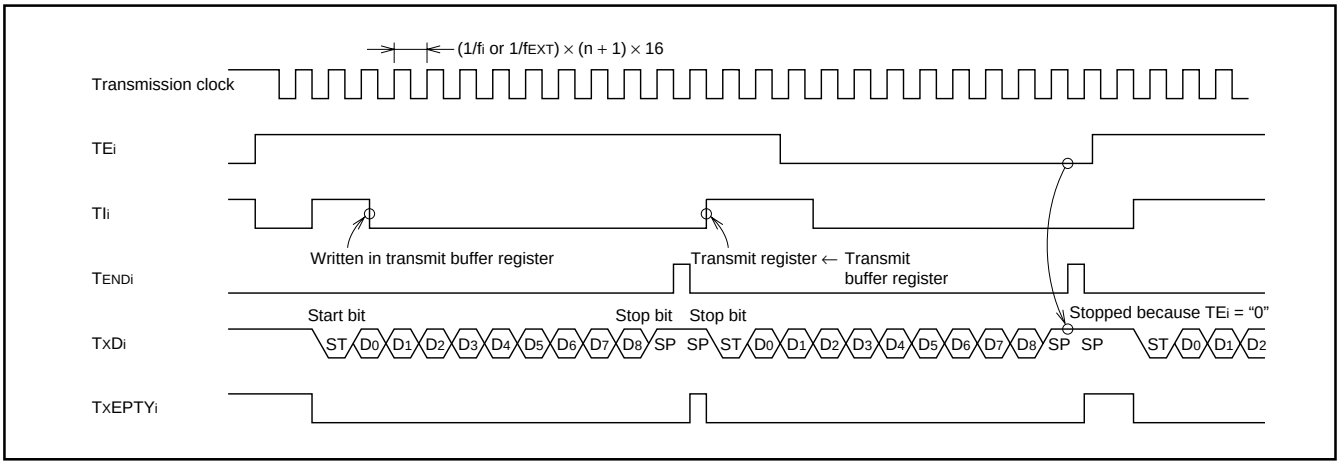


Fig. 61 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits selected

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 60 and 61, 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 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 60 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”.

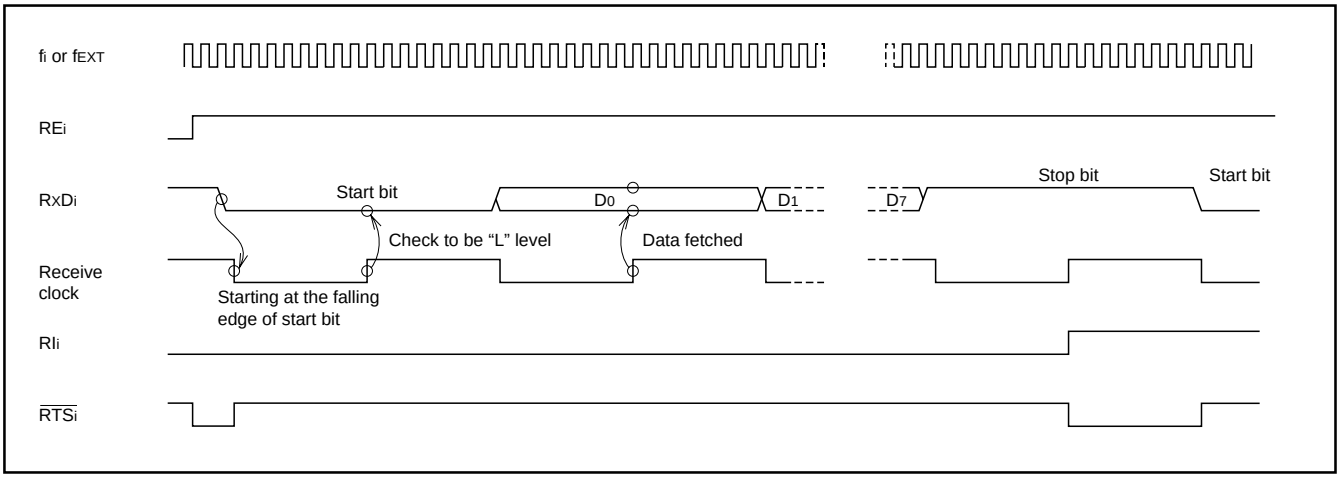


Fig. 62 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit selected

Receive

Receive is enabled when bit 2 (REi flag) of UARTi transmit/receive control register 1 is set to "1." As shown in Figure 62, the frequency divider circuit (1/16) at the receiving side begin to work when a start bit arrives and the data is received.

If $\overline{\text{RTS}}$i output is selected by setting bit 2 of UARTi transmit/receive control register 0 to "1", the $\overline{\text{RTS}}$i output is "H" when the REi flag is "0". When the REi flag changes to "1", the $\overline{\text{RTS}}$i output goes "L" to inform the receiver that reception has become enabled. When the receive operation starts, the $\overline{\text{RTS}}$i 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 52. 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, $\overline{\text{RTS}}$i 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 RIi, OERi, FERi, and PERi flags are set to "1" while transferring the contents of the receive register into the receive buffer register.

The FERi, PERi, and SUMi flags are cleared to "0" when the low-order byte of the receive buffer register has been read out or when "0" has been written to the REi flag.

The OERi flag is cleared to "0" when "0" has been written 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.

A-D CONVERTER

The A-D converter is a 10-bit successive approximation converter. Figure 63 shows the block diagram of the A-D converter, Figure 64 shows the bit configuration of the A-D control register 0 (address $1E_{16}$), and 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-numbered address of the corresponding A-D register and the high-order 2 bits are stored in bits 0 and 1 at the odd-numbered address of the corresponding A-D register. Bits 2 to 7 of the A-D register odd-numbered address are "000000_2" when read.

When the conversion result is used as 8-bit data, the conversion result are stored in even-numbered address of the corresponding A-D register. In this case, the value at the A-D register's odd-numbered 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.

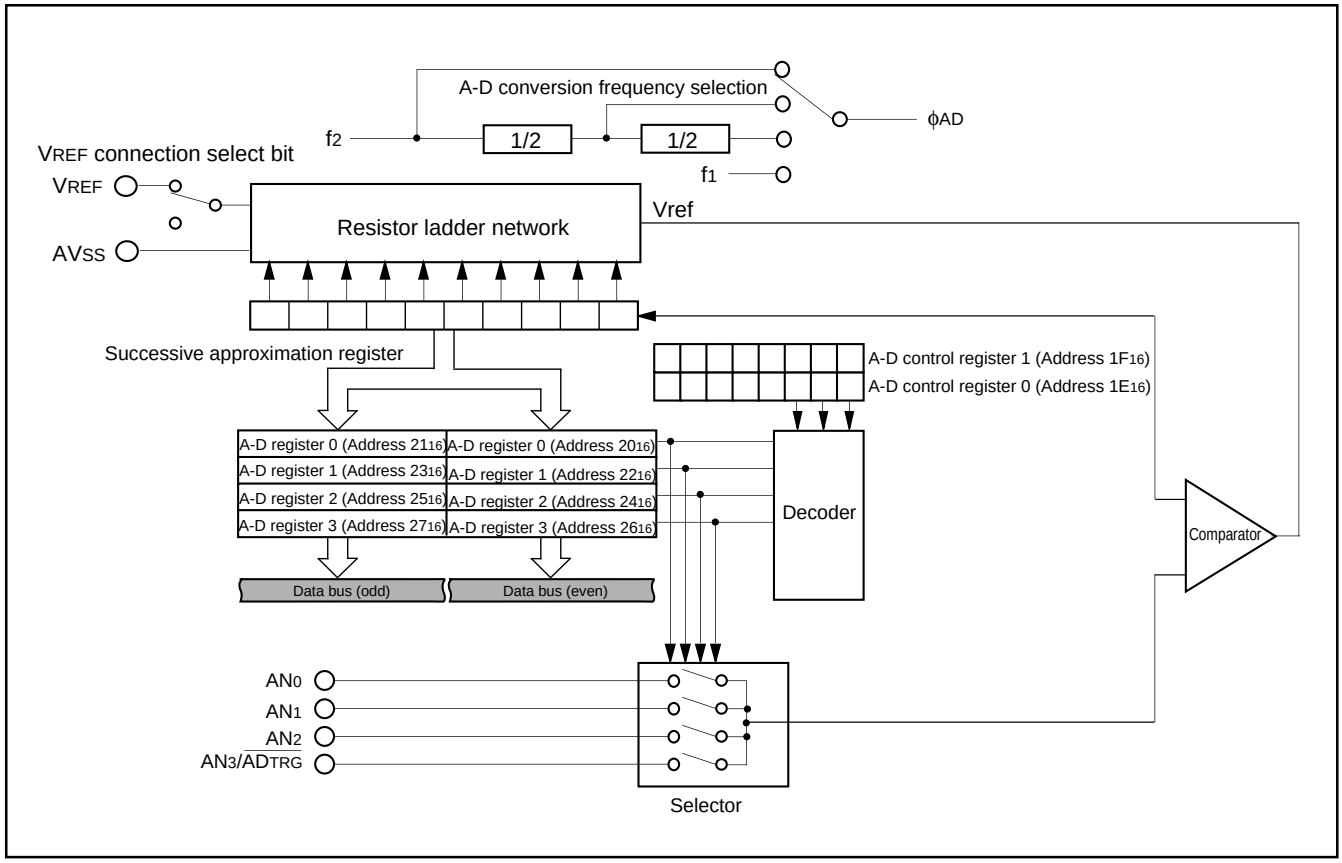


Fig. 63 Block diagram of A-D converter

Trigger

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

Software 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 a software 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.

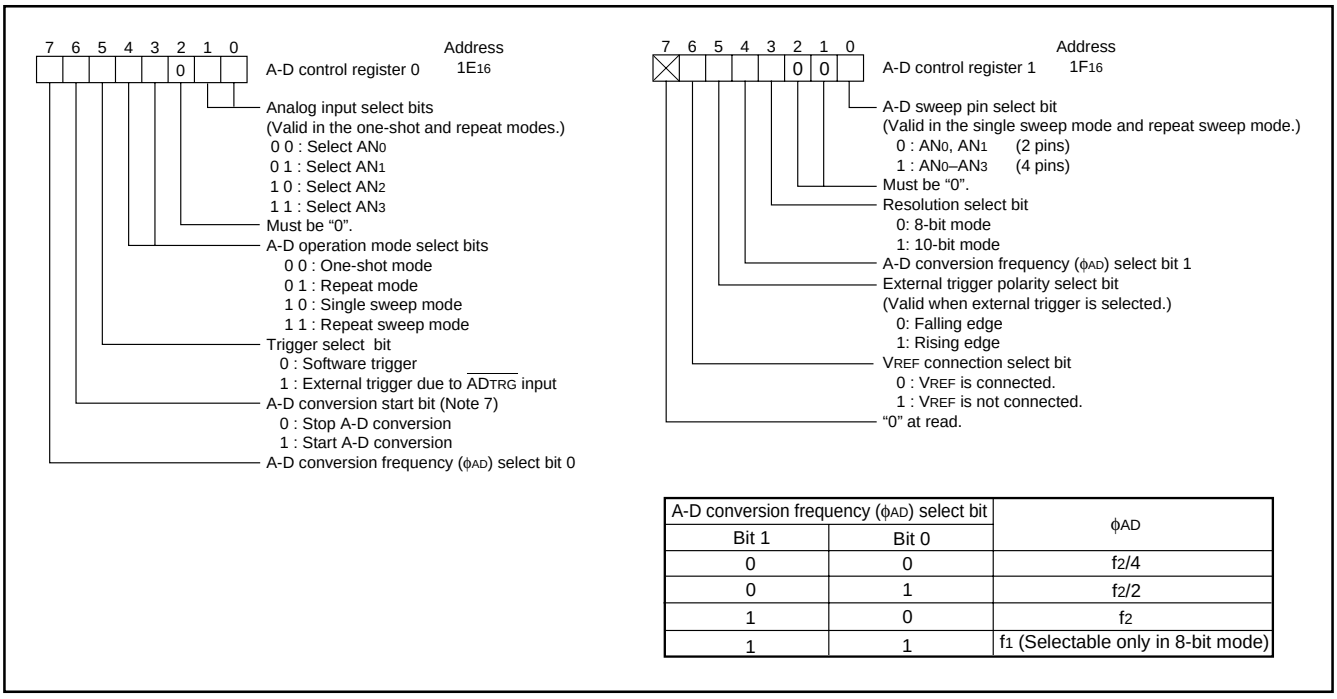
A-D conversion starts when the A-D conversion start bit is “1” and the $\overline{\text{ADTRG}}$ input changes from “H” to “L” (or “L” to “H.”) In this case, the pins that can be used for A-D conversion are AN_0 to AN_2 because the $\overline{\text{ADTRG}}$ pin is multiplexed with an analog voltage input pin, AN_3. 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.

VREF connection

Whether to connect the reference voltage input (VREF) with the resistor ladder network or not depends on bit 6 of the A-D control register 1. The VREF 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 VREF pin to the resistor ladder network can be cut off by disconnecting resistor ladder network from the VREF pin.

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



A-D conversion frequency (φAD) select bit		φAD
Bit 1	Bit 0	
0	0	f2/4
0	1	f2/2
1	0	f2
1	1	f1 (Selectable only in 8-bit mode)

Fig. 64 Bit configuration of A-D control register 0

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 and 1 of A-D control register 0.

When a software 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, bit 6 of A-D control register 0 (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 $\overline{\text{ADTRG}}$ pin, This operation is the same as that for software 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 a software 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 pin is selected by bit 0 of the A-D control register 1 (address $1F_{16}$). Two pins, or four 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 a software trigger or with an external trigger input. A software 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 a software trigger is selected, A-D conversion is started when bit 6 of A-D control register 0 (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 $\overline{\text{ADTRG}}$ pin. In this case, the A-D conversion result which is stored in the A-D register 3 becomes invalid.

The operation by external trigger is the same as that by a software trigger except that the A-D conversion start bit is not cleared to “0” after A-D conversion and that 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 a software 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 an A-D interrupt. It is because this interrupt request bit is undefined just after reset.

DMA CONTROLLER

The DMA (direct memory access) controller is a 4-channel controller which provides high-speed data transfers from memory to memory, memory to input/output ports of external devices (herein referred to as external I/O), and external I/O to memory without using the CPU. Figure 65 shows the block diagram of the DMA controller, Figure 66 shows the DMA control-related register memory map, and Figure 67 shows the bit configuration of the DMAC control registers L and H. DMA transfers are performed by the DMA control circuit via the bus interface unit (BIU).

Each of DMAC control registers L and H consists of 8 bits. For DMAC control register L, bit 0 is the priority select bit, and bit 1 is the $\overline{TC}$ pin validity bit. Bits 4 to 7 are DMAi request bits (i = 0 to 3). Reading these bits indicates whether a DMA request for each channel has occurred or not. For DMAC control register H, bits 0 to 3 are software DMA request bits, and each of them is used to generate a DMA request by software. Bits 4 to 7 are DMAi enable bits (i = 0 to 3). The DMA request is accepted only when the corresponding DMAi enable bit is set to “1”. All of these DMAi enable bits are cleared to “0” after reset removal.

Figure 68 shows the bit configuration of the DMAi control register (i = 0 to 3). Each channel of the DMAi control register consists of 8 bits. Bits 0 to 3 are DMA request source select bits.

Bit 4 determines whether the edge or level sense function is to be used for selecting a request source from pin $\overline{DMAREQi}$ (DMA request input). Bit 5 is the $\overline{DMAACKi}$ validity bit. When bit 5 is “0”, pin $\overline{DMAACKi}$ (the DMA acknowledge signal output pin) is invalid; when “1”, pin $\overline{DMAACKi}$ is valid.

Figure 69 shows the bit configuration of the DMAi mode registers L and H. Each channel of both registers consists of 8 bits. Refer to the corresponding section for more details.

Pin description

Pins $\overline{DMAREQi}$, $\overline{DMAACKi}$, $\overline{TC}$ are used for DMA transfers.

Pin $\overline{DMAREQi}$ is a DMA request input pin. Port pins P61, P63, P65, and P66 are multiplexed with pins $\overline{DMAREQ0}$, $\overline{DMAREQ1}$, $\overline{DMAREQ2}$ and $\overline{DMAREQ3}$, respectively. These pins are used in order to request a DMA transfer from the external.

When the DMA request source select bits (bits 0 to 3) of the DMAi control register are set to “0001”, the input signal from this pin becomes the DMA request signal. In order to use any of the above pin as pin $\overline{DMAREQi}$, be sure to set the corresponding bit of the port P6 direction register to the input mode.

Pin $\overline{DMAACKi}$ is the DMA acknowledge signal output pin. Port pins P60, P62, P64 are multiplexed with pins $\overline{DMAACK0}$, $\overline{DMAACK1}$, and $\overline{DMAACK2}$, respectively. When bit 5 ($\overline{DMAACKi}$ validity bit) of the DMAi control register for each channel is set to “1”, pin $\overline{DMAACKi}$ serves as the output-only pin for signal $\overline{DMAACKi}$. (DMA3 is not equipped with pin $\overline{DMAACKi}$.) During DMA transfer, the operating channel acknowledge signal is output regardless of the data transfer method (the 1-bus cycle transfer or 2-bus cycle transfer). When the acknowledge signal is not needed, clear the $\overline{DMAACKi}$ validity bit to “0”, so that pin $\overline{DMAACKi}$ can serve as an I/O pin.

Pin $\overline{TC}$ is a terminal count pin and is multiplexed with port pin P42. Pin $\overline{TC}$ is valid when “1” has been written to bit 1 of the DMAC control register L. At this time, pin $\overline{TC}$ serves as the N-channel open drain output pin. When the value of the transfer counter register or transfer block counter is “0”, pin $\overline{TC}$ outputs “L” level for 1 cycle of $\phi 1$. Furthermore, when the $\overline{TC}$ pin validity bit is “1”, any ongoing channel DMA transfer can be cancelled by changing the input level at pin $\overline{TC}$ from “H” to “L”.

Data transfer method

Two different data transfer methods are available: 2-bus cycle transfer, effective for memory-to-memory data transfer, and 1-bus cycle transfer, effective for memory-to-I/O or I/O-to-memory data transfer. Both methods are described in detail below.

(1) 2-bus cycle transfer

When bit 1 of the DMAi mode register L, as shown in Figure 69, is cleared to “0”, the 2-bus cycle transfer method is selected. This method makes data to be transferred by the “1 transfer unit”, by using 1 read bus cycle and 1 write bus cycle. The “1 transfer unit” refers to the number of bits which can be transferred in 1 DMA transfer operation, and it is determined by bit 0 of the DMAi mode register L. When bit 0 is cleared to “0”, “1 transfer unit” consists of 16 bits (2 bytes); when “1”, “1 transfer unit” consists of 8 bits (1 byte).

In the 2-bus cycle transfer, be sure to clear bit 0 of the DMAi mode register H to “0”.

Figure 70 shows an connection example with external memories in 2-bus cycle transfer. In the read cycle, the transfer source address is output to the address bus, and the data at this address is read out by the “1 transfer unit” and then stored into the BIU’s data buffer. When 16-bit data is read out from an odd-numbered address or when 16-bit data is read out with the external data bus width = 8 bits, the microcomputer will enter the write cycle after the above 16-bit data is stored into the BIU’s data buffer in 2 accesses.

In the write cycle, the transfer destination address is output to the address bus, and the data which has been stored in the BIU’s data buffer is written to the transfer destination address. When 16-bit data is read out from an odd-numbered address or when 16-bit data is read out with the external data bus width = 8 bits, the microcomputer will preforms the write operation in 2 accesses.

(2) 1-bus cycle transfer

When bit 1 of the DMAi mode register L is set to “1”, the 1-bus cycle transfer method is selected. When data transfer is to be made between an external I/O and the external memory, this method allows the memory to be read at the same time the data is written to the external I/O, and vice versa, resulting in fast data transfer. Bit 0 of the DMAi mode register H determines whether the 1-bus cycle transfer is to be made from the external memory to the external I/O or from the external I/O to the external memory. When the bit is “1”, the data transfer is made from the external I/O to the external memory.

Figure 71 shows an connection example with external memories and external I/Os in 1-bus cycle transfer (the external data bus width = 16 bits and “1 transfer unit” = 16 bits).

For the transfer from the external memory to external I/O, the external-memory-side address (transfer source address) is output to the address bus, pin $\overline{RD}$ goes to “L”, and the read operation will be performed.

This ensures that the data is read out from the external memory. At the same time, pin $\overline{DMAACK}$i corresponding to the operating DMAi channel (i = 0 to 2) goes to “L”, the external I/O is selected, and the data read from the external memory is directly fetched at the rising of signal $\overline{RD}$. In this manner, data is transferred from external memory to external I/O in 1 bus cycle.

For the transfer from the external I/O to the external memory, the data is read out from the external I/O, selected by the acknowledge signal from pin $\overline{DMAACK}$i, to the data bus. At the same time, the external-memory-side address (transfer destination address) is output to the address bus, pin $\overline{BLW}$ (write signal for even-numbered addresses) and pin $\overline{BHW}$ (write signal for odd-numbered addresses) go to “L”, and the write operation to the external memory is performed.

The 1-bus cycle transfer cannot perform operations for a read from or a write to the internal memory. In order to perform the transfer from the internal memory to the external I/O or from the external I/O to the internal memory, be sure to select the 2-bus cycle transfer method.

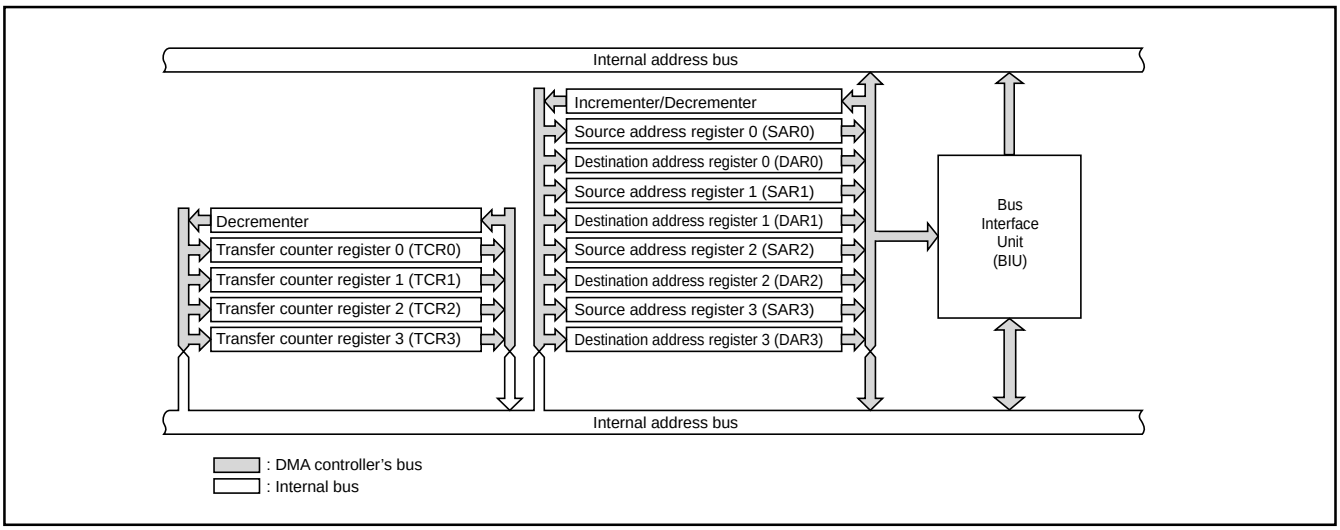
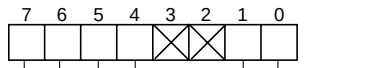


Fig. 65 Block diagram of DMA controller

Address (Hexadecimal notation)	
0000B0₁₆	DMAC control register L
0000B1₁₆	DMAC control register H
0000B2₁₆	DMA0 interruput control register
0000B3₁₆	DMA1 interruput control register
0000B4₁₆	DMA2 interruput control register
0000B5₁₆	DMA3 interruput control register

Address (Hexadecimal notation)		
0000C0₁₆		L
0000C1₁₆	Source address register 0	M
0000C2₁₆		H
0000C3₁₆		
0000C4₁₆		L
0000C5₁₆	Destination address register 0	M
0000C6₁₆		H
0000C7₁₆		
0000C8₁₆		L
0000C9₁₆	Transfer counter register 0	M
0000CA₁₆		H
0000CB₁₆		
0000CC₁₆	DMA0 mode register L	
0000CD₁₆	DMA0 mode register H	
0000CE₁₆	DMA0 control register	
0000CF₁₆		
0000D0₁₆		L
0000D1₁₆	Source address register 1	M
0000D2₁₆		H
0000D3₁₆		
0000D4₁₆		L
0000D5₁₆	Destination address register 1	M
0000D6₁₆		H
0000D7₁₆		
0000D8₁₆		L
0000D9₁₆	Transfer counter register 1	M
0000DA₁₆		H
0000DB₁₆		
0000DC₁₆	DMA1 mode register L	
0000DD₁₆	DMA1 mode register H	
0000DE₁₆	DMA1 control register	
0000DF₁₆		
0000E0₁₆		L
0000E1₁₆	Source address register 2	M
0000E2₁₆		H
0000E3₁₆		
0000E4₁₆		L
0000E5₁₆	Destination address register 2	M
0000E6₁₆		H
0000E7₁₆		
0000E8₁₆		L
0000E9₁₆	Transfer counter register 2	M
0000EA₁₆		H
0000EB₁₆		
0000EC₁₆	DMA2 mode register L	
0000ED₁₆	DMA2 mode register H	
0000EE₁₆	DMA2 control register	
0000EF₁₆		
0000F0₁₆		L
0000F1₁₆	Source address register 3	M
0000F2₁₆		H
0000F3₁₆		
0000F4₁₆		L
0000F5₁₆	Destination address register 3	M
0000F6₁₆		H
0000F7₁₆		
0000F8₁₆		L
0000F9₁₆	Transfer counter register 3	M
0000FA₁₆		H
0000FB₁₆		
0000FC₁₆	DMA3 mode register L	
0000FD₁₆	DMA3 mode register H	
0000FE₁₆	DMA3 control register	
0000FF₁₆		

Fig. 66 DMA controll-related register memory map



DMAC control register L — Address $B0_{16}$

Priority select bit
0: Fixed.
1: Rotating.

$\overline{TC}$ pin validity bit
0: Invalid. Pin $P4_2$ functions as a programmable I/O port (CMOS) pin.
1: Valid. Pin $P4_2$ functions as pin $\overline{TC}$ (N-channel open-drain).

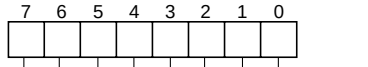
DMA0 request bit
0: No request.
1: Requested **(Note)**.

DMA1 request bit
0: No request.
1: Requested **(Note)**.

DMA2 request bit
0: No request.
1: Requested **(Note)**.

DMA3 request bit
0: No request.
1: Requested **(Note)**.

Note: Even when "1"s are written to bits 4 to 7 by software, these bits' status do not change.



DMAC control register H — Address $B1_{16}$

Software DMA0 request bit
1: DMA request. Valid when the software DMA source is selected.
The value is "0" at reading.

Software DMA1 request bit
1: DMA request. Valid when the software DMA source is selected.
The value is "0" at reading.

Software DMA2 request bit
1: DMA request. Valid when the software DMA source is selected.
The value is "0" at reading.

Software DMA3 request bit
1: DMA request. Valid when the software DMA source is selected.
The value is "0" at reading.

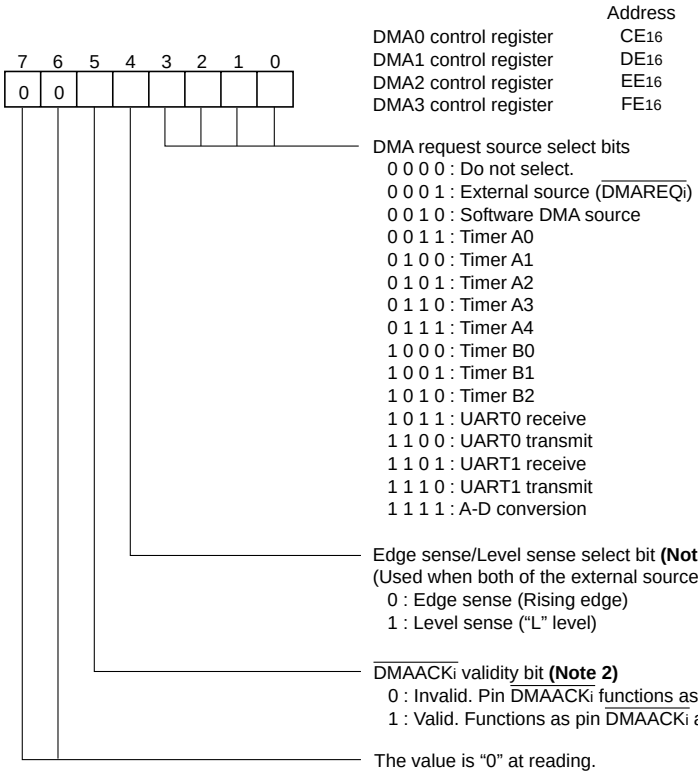
DMA0 enable bit
0: Disabled.
1: Enabled.

DMA1 enable bit
0: Disabled.
1: Enabled.

DMA2 enable bit
0: Disabled.
1: Enabled.

DMA3 enable bit
0: Disabled.
1: Enabled.

Fig. 67 Bit configuration of DMAC control registers L and H

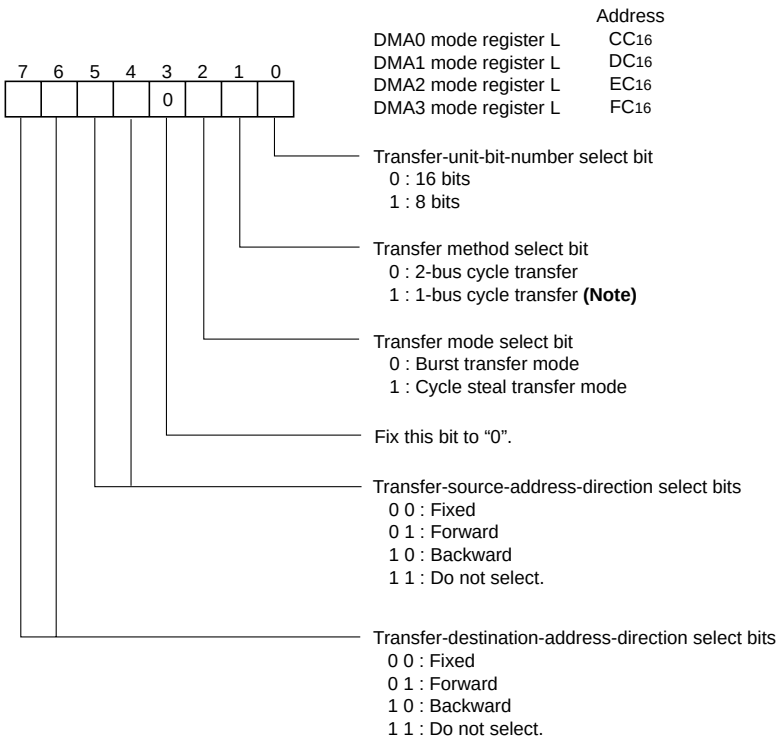


Notes 1: Be sure to fix this bit to “0” in any of the following cases:
- When the external source is selected by using bits 0 to 3
- In the cycle steal transfer mode

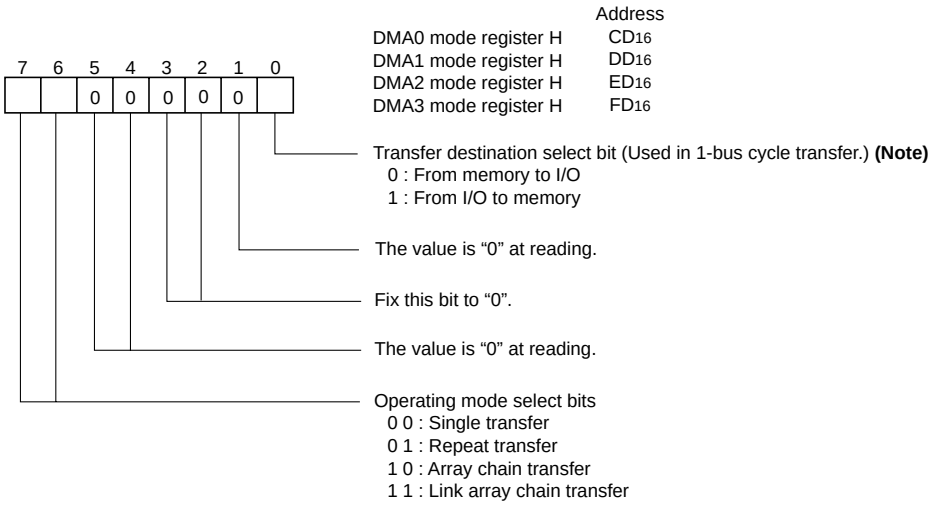
Level sense can be selected only when both of the external source and burst transfer mode are selected.

2: DMA3 is not equipped with the DMAACK output function. For the DMA3 control register, be sure to clear this bit to “0”.

Fig. 68 Bit configuration of DMAi control register

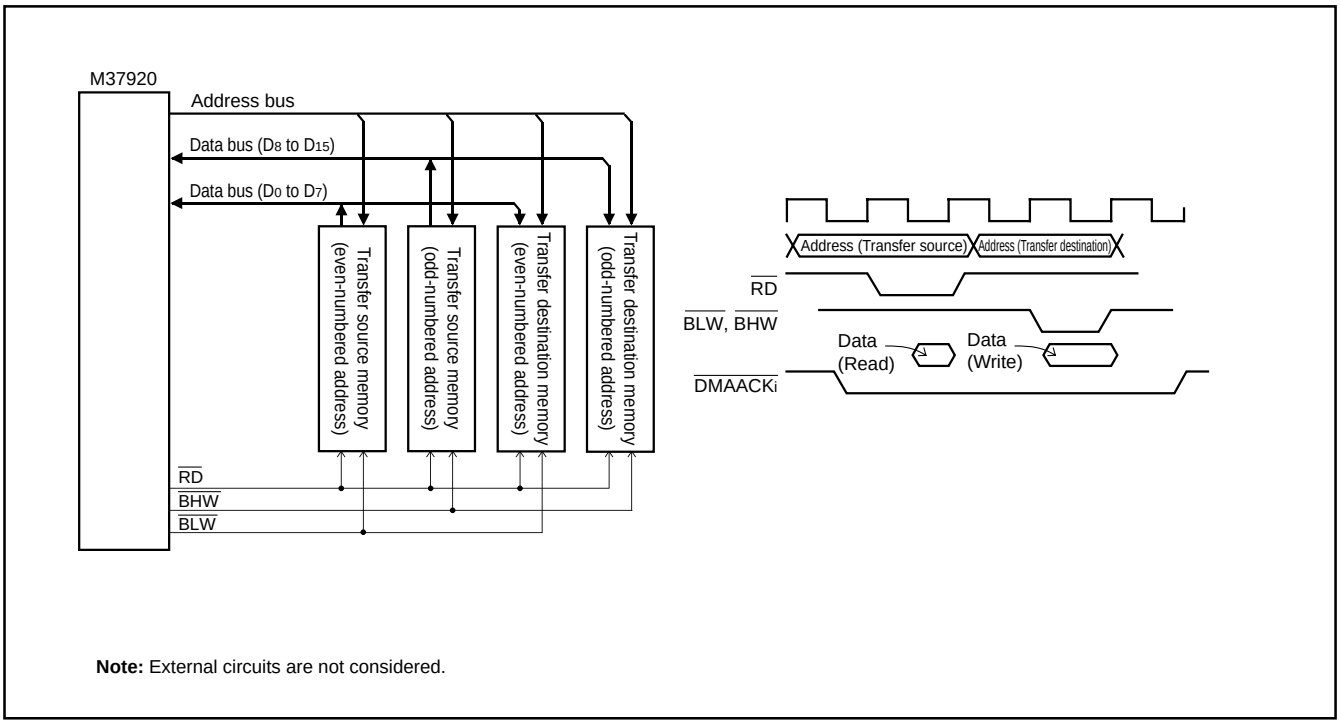


Note: For the DMA3 mode register L, be sure to fix this bit to "0" (2-bus cycle transfer). In this case, 1-bus cycle transfer cannot be used. Additionally, be sure to fix this bit to "0" when either transfer source or transfer destination is in an internal area. In this case, also, 1-bus cycle transfer cannot be used.



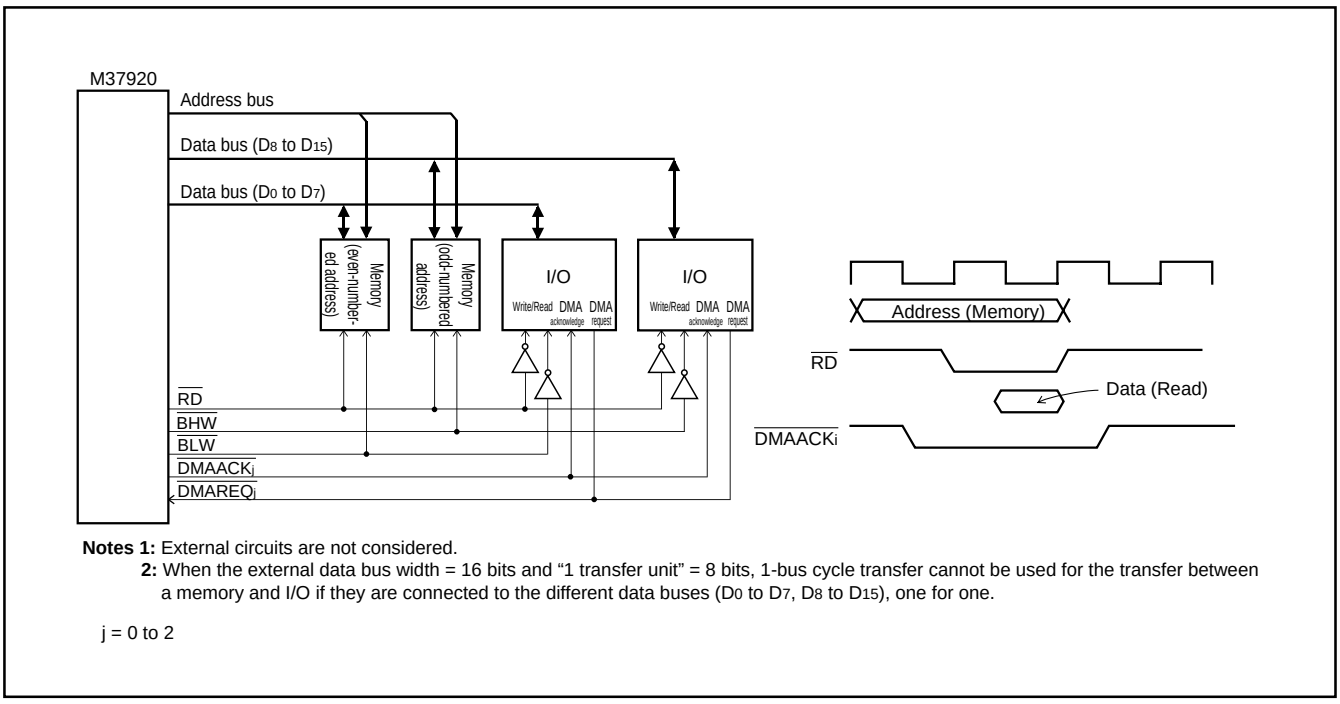
Note: Be sure to fix this bit to "0" in 2-bus cycle transfer.

Fig. 69 Bit configuration of DMA$_i$ control registers L and H



Note: External circuits are not considered.

Fig. 70 Connection example with external memories in 2-bus cycle transfer



Notes 1: External circuits are not considered.
2: When the external data bus width = 16 bits and "1 transfer unit" = 8 bits, 1-bus cycle transfer cannot be used for the transfer between a memory and I/O if they are connected to the different data buses (D0 to D7, D8 to D15), one for one.

j = 0 to 2

Fig. 71 Connection example with external memories and external I/Os in 1-bus cycle transfer

DMA request sources

One out of fifteen DMA request sources can be selected for each channel. There are a total of fifteen DMA request sources. Thirteen internal request sources (A-D conversion, UART0 transmit/receive, UART1 transmit/receive, timers A0 to A4, timers B0 to B2), one software DMA source issued by programs, and one external source by input to pin $\overline{\text{DMAREQ}}$i. For DMA request source selection, use the DMAi control register's DMAi request source select bits (bits 0 to 3) as shown in Figure 68. Table 18 lists the relationship between DMA request source select bits (bits 0 to 3) and DMA request sources. The request timing is the same as that for interrupts.

When the software DMA request source is selected with the DMA request source select bits, by writing "1" to any of the DMAC control register H's software DMA request bits (bits 0 through 3), the correspomding DMA request bit is set to "1". When a DMA request bit has been set to "1", the software DMA request bits are automatically cleared to "0". When the external source is selected with the DMA request source select bits, the input from pin $\overline{\text{DMAREQ}}$i sets the correspomding DMA request bit to "1". The DMA transfer request will not be accepted until both of the DMA request bit and DMA enable bit of the DMAC control registers L and H are "1". Therefore, if the DMA enable bit is "0", no DMA request will be accepted even when the DMA request bit is "1". Note that the DMA enable bit is "0" at reset. Therefore, after the DMA transfer parameter and other data have been setup, be sure to set the DMA enable bit of the DMA channel to be rendered valid to "1". This assures that the transfer request of that channel becomes valid, making the DMA transfer enabled.

Transfer mode

Two DMA transfer modes are available: burst transfer mode and cycle steal transfer mode. Mode selection is made variously for each channele, using bit 2 of the DMAi mode register L. When this bit is cleared to "0", the burst transfer mode is selected. This mode is automatically selected after reset removal.

(1) Burst transfer mode

In the burst transfer mode, either the edge sense or level sense mode can be selected only when the input from pin $\overline{\text{DMAREQ}}$i (external source) is selected as a request source.

When the DMAi control register's bit 4 is cleared to "0", the edge sense mode is selected. The edge sense mode is automatically selected after reset removal. In the edge sense mode, the DMA request bit is set to "1" at the falling edge of the input from pin $\overline{\text{DMAREQ}}$i. In the burst transfer's edge sense mode, the DMA request bit is cleared to "0" when any of the following conditions is satisfied.

1. Channel i's DMA enable bit is cleared to "0" (forced termination of transfer).
2. Channel i's DMA request bit is cleared to "0".
3. All of channel i's DMA transfers are completed (normal termination of transfer).
4. "L" level is input to pin $\overline{\text{TC}}$ during channel i's transfer (forced termination of transfer).

Figure 72 shows a burst transfer example in edge sense mode. When a DMA request is received from a certain channel in the edge sense mode's burst transfer, no DMA request from the other channels will be accepted until the DMA transfer on the former channel is completed. In this example, pin $\overline{\text{DMAREQ}}$i's input (external source) is selected as the DMA request source. When pin $\overline{\text{DMAREQ}}$i's input changes from the "H" to "L" level during CPU operation, the DMA request bit will be set to "1" and the DMA controller will acquire the right to use bus and initiate transfer. From high to low, the bus use priority is for DRAM refresh, $\overline{\text{HOLD}}$, DMA controller, and CPU. Therefore, if a request is made by the DRAM refresh, which has a higher priority than the DMA controller, the DMA controller halts any ongoing transfer operation at the end of the current transfer bus cycle and passes the right to use bus to the DRAM controller as shown in Figure 72. Upon getting the right, the DRAM controller generates the refresh cycle. When refreshing is terminated, the DMA controller resumes the execution of the interrupted DMA transfer at the point of interruption. Once a DMA request is accepted in the burst transfer mode, no request from the other channels is accepted until the DMA transfer is entirely completed or the transfer operation is brought to a forced stop. Therefore, even when the request bit of channel 0, which has a high priority, is set to "1" in the middle of transfer as shown in Figure 72, such a request will not be accepted. (The priority is explained in the next section.)

Table 18. Relationship between DMA request source select bits (bits 3 to 0) and DMA request sources

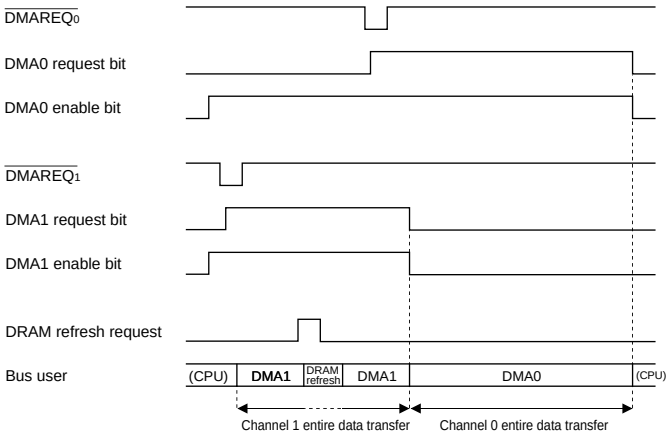
b3	b2	b1	b0	DMA request source
0	0	0	0	Do not select.
0	0	0	1	External source ($\overline{\text{DMAREQ}}$i)
0	0	1	0	Software DMA source
0	0	1	1	Timer A0
0	1	0	0	Timer A1
0	1	0	1	Timer A2
0	1	1	0	Timer A3
0	1	1	1	Timer A4
1	0	0	0	Timer B0
1	0	0	1	Timer B1
1	0	1	0	Timer B2
1	0	1	1	UART0 receive
1	1	0	0	UART0 transmit
1	1	0	1	UART1 receive
1	1	1	0	UART1 transmit
1	1	1	1	A-D conversion

When channel 1's DMA transfer is entirely completed, the right to use bus is once passed to the CPU, and the DMA transfer request from channel 0 is later accepted at the end of the current bus cycle. When bit 4 of the DMAi control register is set to "1", the level sense mode is selected. The level sense mode can be used only for the DMA request from pin $\overline{\text{DMAREQ}}_i$. When selecting another source, be sure to select the edge sense mode.

In the level sense mode, the DMAi request bit is set to "1" to initiate the DMA transfer only while pin $\overline{\text{DMAREQ}}_i$'s input level is "L". If pin $\overline{\text{DMAREQ}}_i$'s input level returns to "H" in the middie of transfer, the DMA operation is interrupted at the end of the current transfer bus cycle or next transfer bus cycle so that the right to use bus is returned to the CPU. At this time, the DMA enable bit is not cleared.

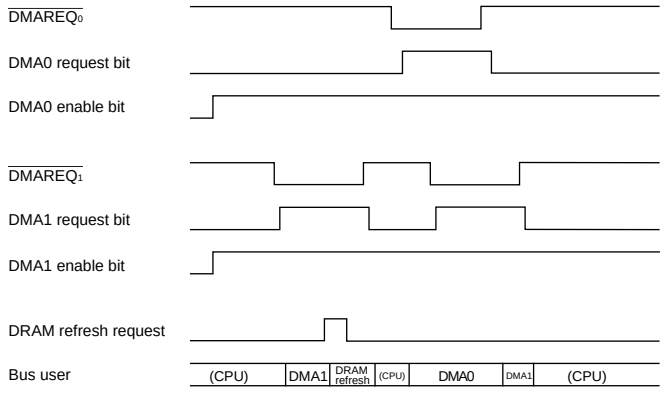
When pin $\overline{\text{DMAREQ}}_i$'s input level returns to "L", the transfer operation is resumed at the address which is next to the point of interruption. In the level sense mode, the DMA request bit varies only with the input level at pin $\overline{\text{DMAREQ}}_i$. Therefore, while pin $\overline{\text{DMAREQ}}_i$'s input level is "L", the DMA request bit remains to be "1" even if the transfer is completed.

Figure 73 shows a burst transfer example in level sense mode. When pin $\overline{\text{DMAREQ}}_i$'s input level for channel 1 changes from "H" to "L" during CPU operation, the DMA1 request bit will be set to "1" so that the DMA controller will acquire the right to use bus and initiate transfer. When pin $\overline{\text{DMAREQ}}_i$'s input level returns to "H", the DMA1 request bit is cleared to "0". This causes the DMA transfer operation to be interrupted and returns the right to use bus to the CPU.



The above example applies on the following conditions :
• DMA request sources of DMA0 and DMA1: external source (edge sense)
• Channel priority : fixed (channel 0 > channel 1)

Fig. 72 Burst transfer example (in edge sense mode)



The above example applies on the following conditions :
• DMA request sources of DMA0 and DMA1: external source (level sense)
• Channel priority : fixed (channel 0 > channel 1)

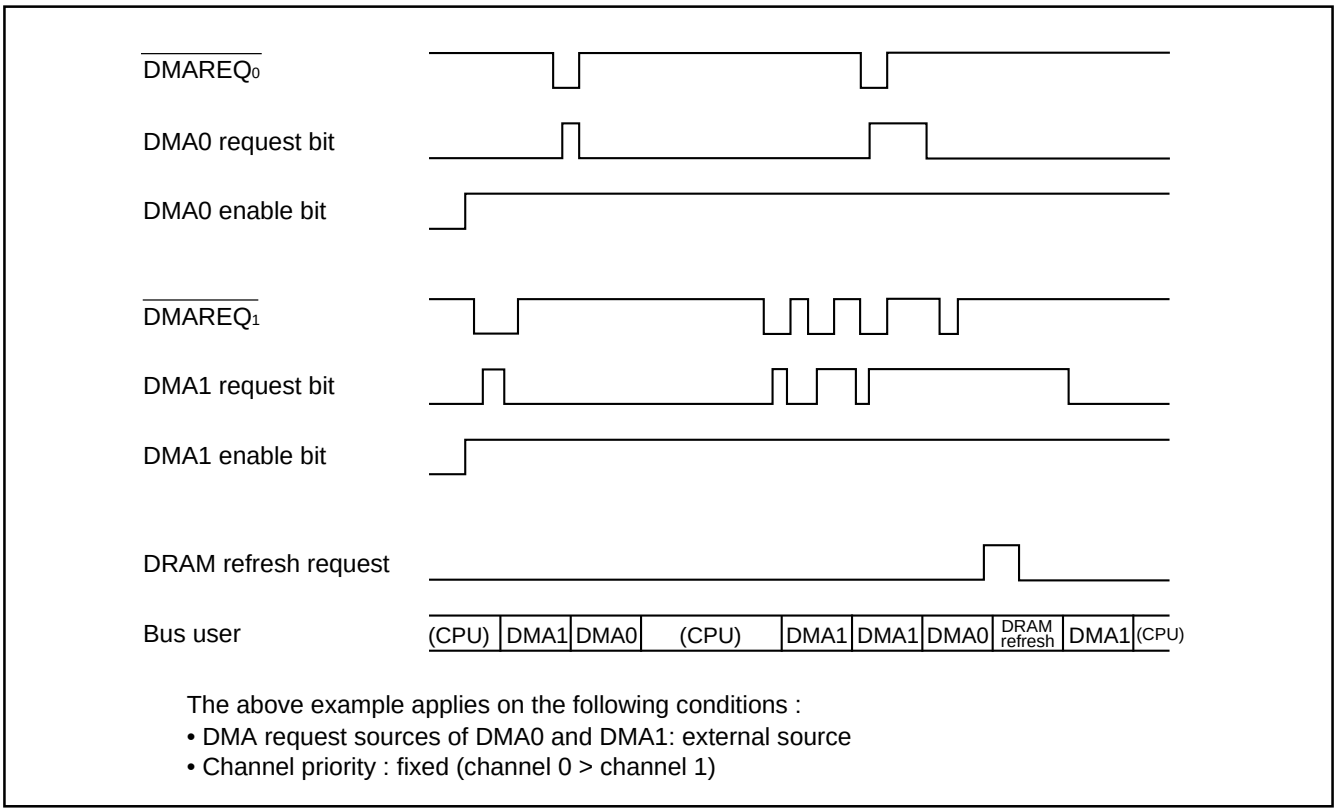
Fig. 73 Burst transfer example (in level sense mode)

(2) Cycle steal transfer mode

When bit 2 of the DMAi mode register L is set to “1”, the cycle steal transfer mode is selected. In the cycle steal transfer mode, be sure to select the edge sense mode.

When a DMA request occurs in the cycle steal transfer mode, the corresponding DMA request bit is set to “1” as in the burst transfer mode. When the DMA request from the channel is accepted, DMA transfer starts. However, the DMA request bit is automatically cleared to “0” at the start of the first DMA transfer cycle. Therefore, if there is no DMA request from any channel when 1-transfer-unit data has been transferred, the DMA controller returns the right to use bus to the CPU. If there is a DMA request from a channel, the DMA controller continues to use the bus and initiates DMA transfer for the channel. In the cycle steal transfer mode, the priorities of the channels are detected at all times to assure that the DMA request from a channel having the highest priority is accepted to initiate the DMA transfer execution. The DMA request bit is cleared to “0”, at each time when 1-transfer-unit data has been transferred. At this time, however, the DMA enable bit will not be cleared to “0” although the DMA request bit is cleared to “0” at each transfer of 1 transfer unit. Therefore, when the DMA request bit is set to “1” next, transfer is resumed at the point of interruption. When the transfer counter register’s value is “0” in the single transfer, or when both of the transfer counter register’s value and transfer block counter’s value are “0” in the array chain transfer, the DMA enable bit will be cleared to ”0” to terminate the whole DMA transfer operation.

Figure 74 shows an example of cycle steal transfer. When pin $\overline{\text{DMAREQ}}_i$’s input level changes from “H” to “L”, the DMA1 request bit will be set to “1” and the DMA controller will acquire the right to use bus and initiate DMA transfer. The DMA1 request bit is cleared to “0” when the channel 1 transfer cycle starts. Therefore, if there is no DMA transfer request from the other channels, the DMA controller returns the right to use bus to the CPU at the end of 1 transfer cycle. In the example shown in Figure 74, however, DMA0 transfer cycle execution continues because the channel 0’s request bit is set to “1”. When the DMA0 transfer cycle is terminated, the DMA request bits of all channels are cleared to “0” so that the DMA controller returns the right to use bus to the CPU. When the DMA1 request bit is set to “1”, only one cycle of transfer operation is performed. Even if the DMA1 request bit is cleared to “0” at this time, the DMA1 request bit is set to “1” again to perform continuous transfer, as long as pin $\overline{\text{DMAREQ}}_i$’s input level goes “L” before the end of the next transfer cycle. In the cycle steal transfer, the priorities of individual channels are detected at the end of each transfer cycle. Therefore, if the request is issued from channel 0, which has a higher priority than channel 1, channel 0 transfer is executed first. Furthermore, if a request to use bus which has a higher priority (for example, a refresh request from the DRAM controller) is generated, this request takes the precedence.



The above example applies on the following conditions :
- DMA request sources of DMA0 and DMA1: external source
- Channel priority : fixed (channel 0 > channel 1)

Fig. 74 Example of cycle steal transfer

Priority

Priorities are assigned to all DMA channels. Either the fixed or rotative priority can be selected. When bit 0 (priority select bit) of the DMAC control register is cleared to “0”, the fixed priority is selected. Note that the fixed priority is automatically selected after the reset removal. In the fixed priority, the channels are given fixed priorities and DMA transfer is executed in the order of priority. From high to low, the priorities are assigned to channels 0, 1, 2, and 3. As indicated in Figure 76, the priorities are detected at each cycle in the cycle steal transfer mode or when the first DMA request is accepted in the burst transfer mode.

When bit 0 of the DMAC control register is set to “1”, the rotative priority is selected. From high to low, the initial priorities are assigned to channels 0, 1, 2, and 3 as is the case with the fixed priority. When a DMA transfer for one channel is normally terminated with the rotative priority employed, the priorities are rotated in such a manner that the channel, for which transfer has just been completed, has the lowest priority. For example, when channel 0’s transfer is normally terminated as shown in Figure 75, the priorities are rotated upon completion of transfer so that the new priorities are, in decreasing order, channel 1, channel 2, channel 3, and channel 0. The priorities remain unchanged when DMA transfer is forcibly terminated by pin $\overline{TC}$’s input or the DMA enable bit clearance, etc.

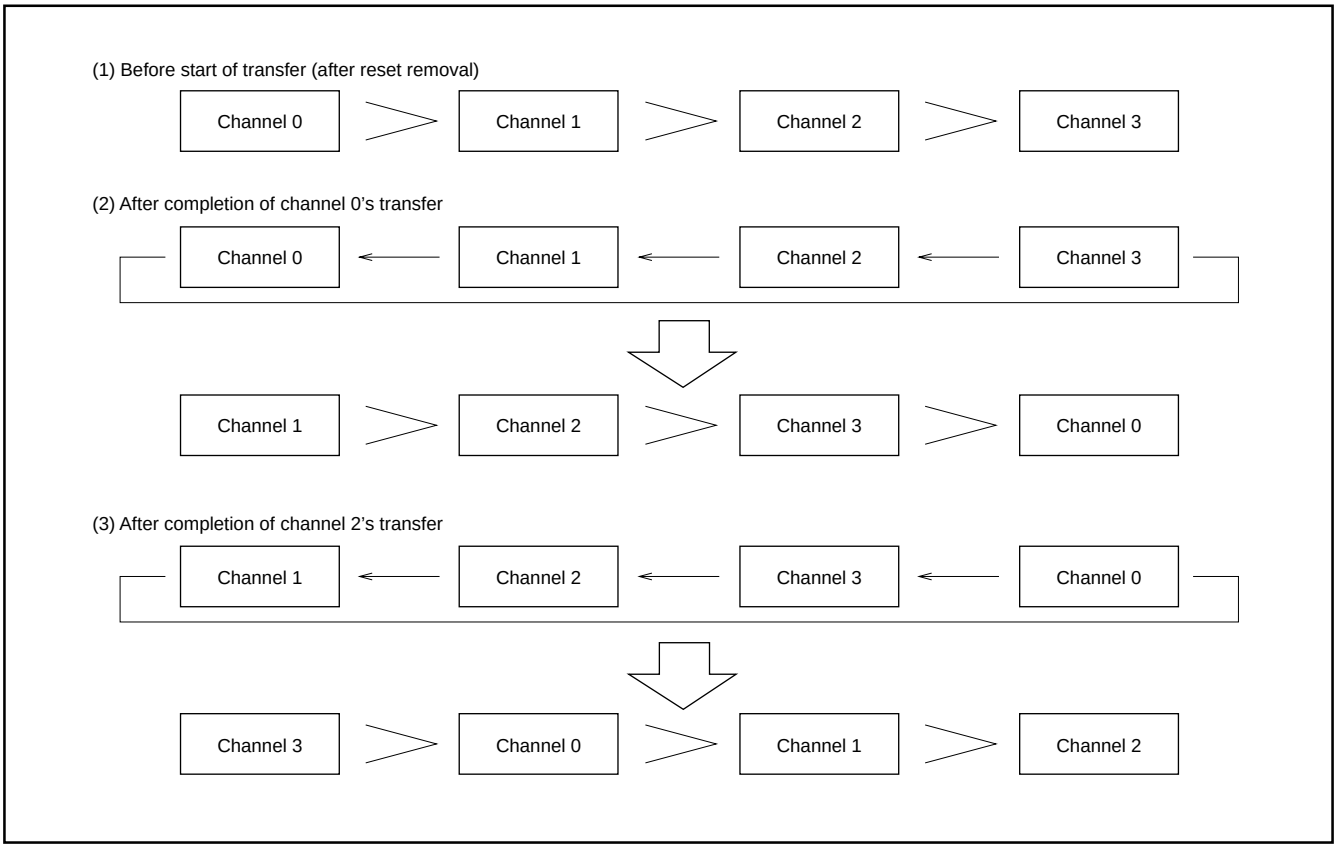
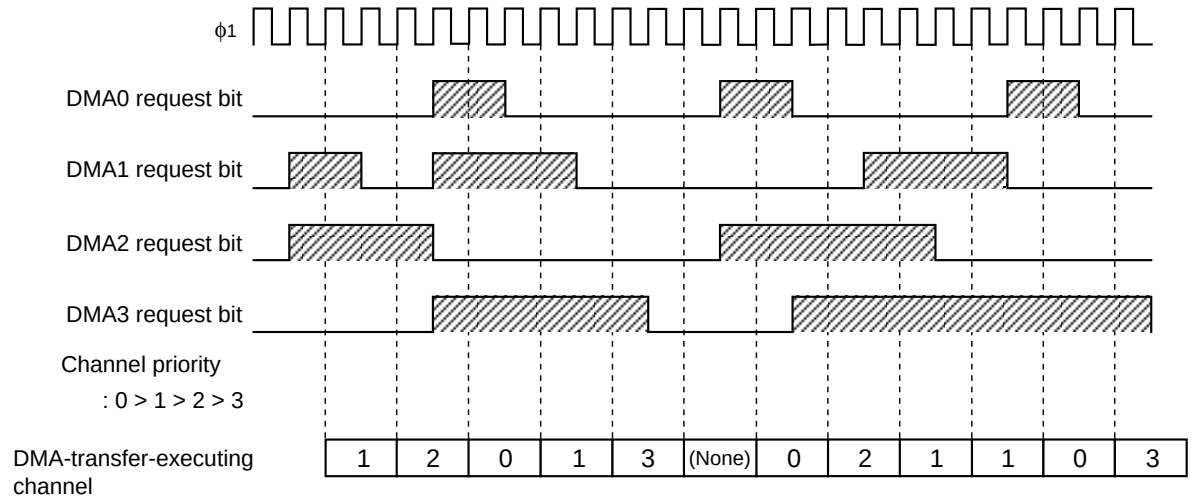
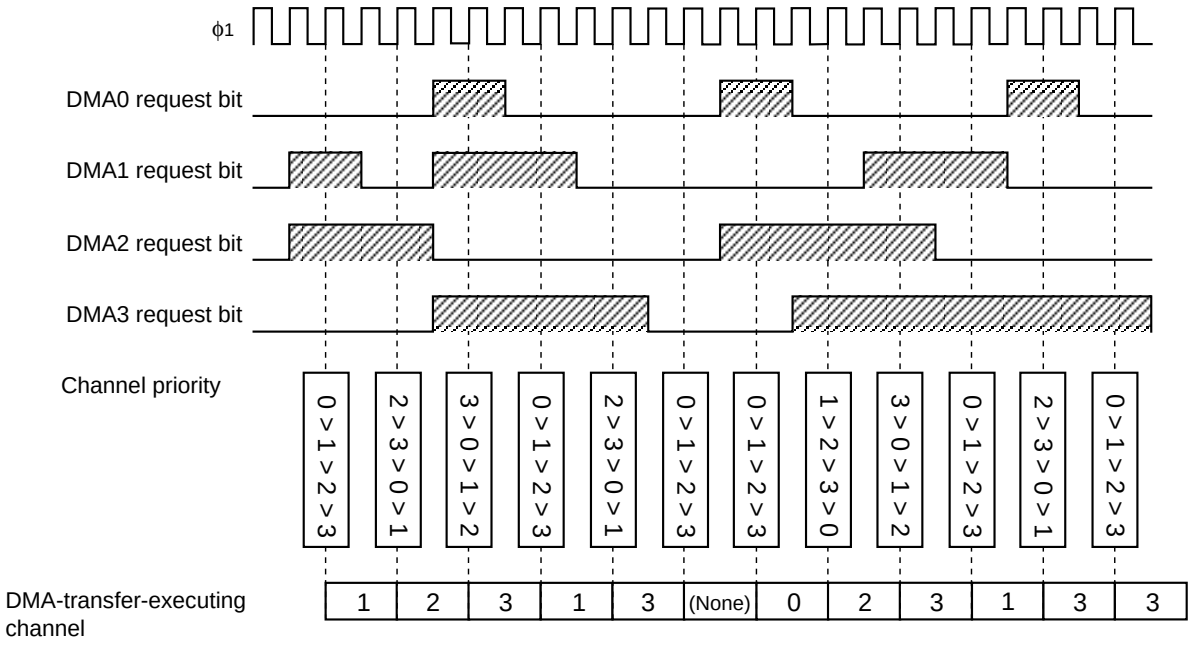


Fig. 75 Rotative priority

● Priority level: fixed



● Priority level: rotating



The above applies on the following conditions :
- No DRAM refresh request, no Hold request.
- All of DMAi enable bits are “1”.

Fig. 76 Example of channel priority detection

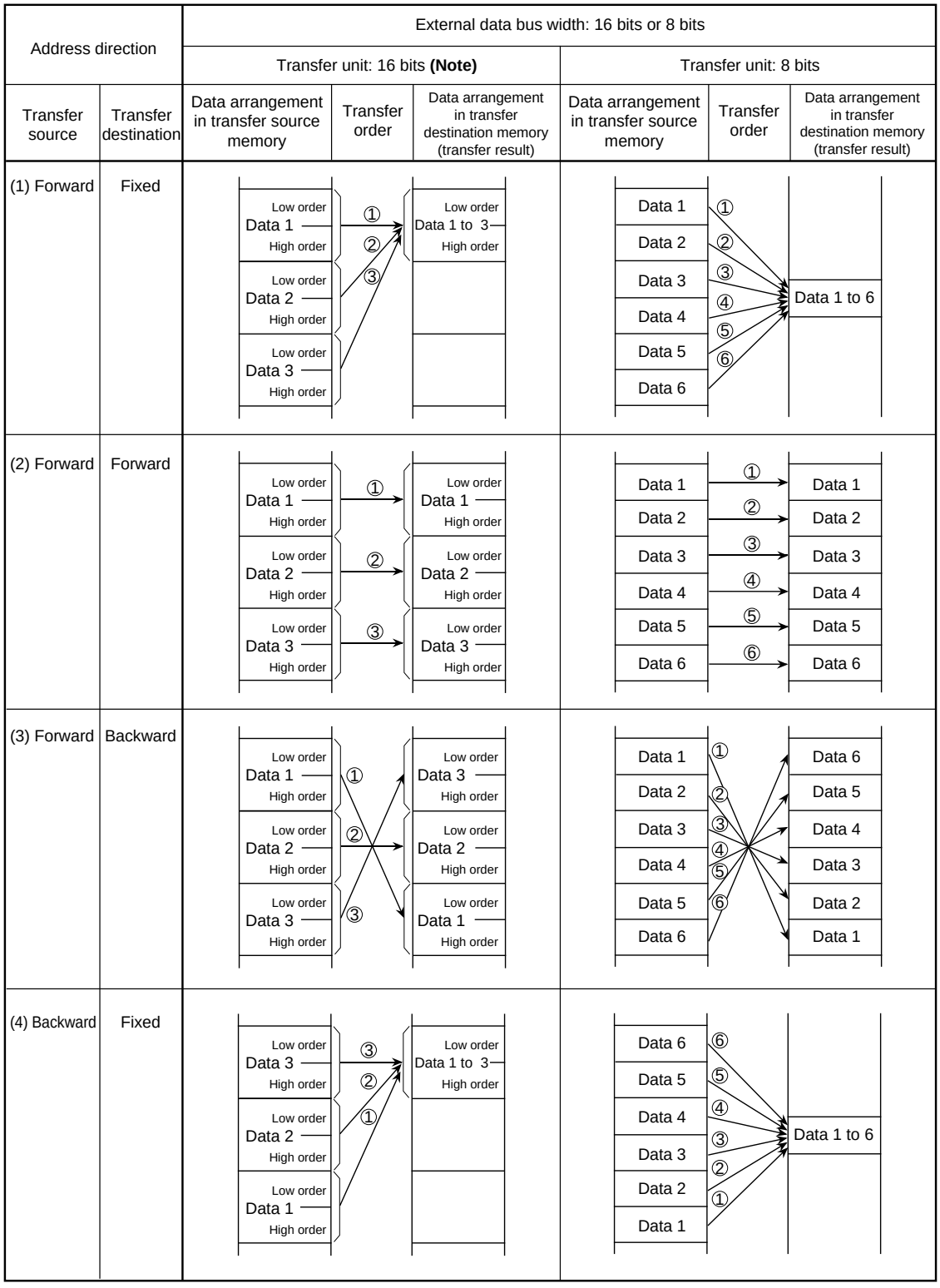
Transfer address direction

The address direction in DMA transfers can be designated independently for the transfer source and destination. These directions are available: “forward”, “backward”, and “fixed”. When the forward direction is selected, the address increments. When the backward direction is selected, the address decrements. When the fixed direction is selected, the address is fixed (2 bytes when 1 transfer unit consists of 16 bits, or 1 byte when 1 transfer unit consists of 8 bits) and does not change. Use bits 4 and 5 of the DMAi mode register L shown in Figure 69 to specify the transfer address direction for the transfer source. For the transfer destination, use bits 6 and 7.

Figure 77 shows an example of transfer address direction in the 2-bus cycle transfer (1 transfer unit = 16 bits). Figure 77-(1) shows an example when the transfer source address direction is “forward” and the destination addresses are “fixed”. In this setup, the transfer source memory’s data is called up in the forward address direction and written to the transfer destination memory’s fixed address by the “1 transfer unit”. Figure 77-(2) shows an example when both the transfer source and destination address directions are set to “forward” by using the DMAi mode register L. In this type of setup, data are transferred from the transfer source memory to the transfer destination memory in the sequence of ①, ②, ③, ... Figure 77-(3) shows an example when the transfer source address direction is “forward” and the destination address direction is “backward”. Figure 77-(4) shows an example when the transfer source address direction is “backward” and the destination address is “fixed”. In this setup, the transfer source memory’s data is written to the fixed transfer destination memory’s address by the “1 transfer unit” in the sequence of ①, ②, and ③....

As explained above, in 2-bus cycle transfer, three different address directions are selectable for each of the transfer source and destination. A total of nine different address direction combinations are available.

In 1-bus cycle transfer, the memory side’s address direction depends on the memory bits. For data transfer from memory to external I/O, therefore, use bits 4 and 5 (transfer-source-address-direction select bits) of the DMAi mode register L to determine the memory side’s (transfer source) address direction. This is not affected by bits 6 and 7 (transfer-destination-address-direction select bits). For data transfer from external I/O to memory, use bits 6 and 7 of the DMAi mode register L to determine the memory side’s (transfer destination) address direction. This is not affected by bits 4 and 5.



Note: The relationship of position between 16-bit data's high-order byte and its low-order byte is fixed, regardless of the address direction. (Data is transferred by the 16 bits.)

Fig. 77 Example of address directions and transfer results in 2-bus cycle transfer (1 transfer unit = 16 bits)

DMA continuous transfer
(1) Single transfer mode

In the single transfer mode, only the preselected number of bytes are transferred. As shown in Figure 69, first, set up the number of bits per 1 transfer unit, transfer method, transfer mode, and transfer address direction by using the DMAi mode registers L and H. Then, write the transfer source block's first transfer address (the block's lowest address in the forward or fixed transfer address direction, or the block's highest address in the backward address direction) into the source address register (hereinafter referred to as SAR). Further, write the destination block's first transfer address (the lowest address in the forward or fixed transfer address direction, or the highest address in the backward transfer address direction) into the destination address register (hereinafter referred to as DAR). Also write the desired number of bytes to be transferred, into the transfer counter register (hereinafter referred to as TCR). Write the value 1 or more into TCR. Each of SAR, DAR, and TCR consists of 24 bits, therefore, be sure to write into all these bits. The SAR, DAR, and TCR are located at the addresses shown in Figure 66. The next is to set a DMA source and others by the DMAi control register shown in Figure 68. Set up bit 0 (priority select bit) and bit 1 ($\overline{TC}$ pin validity bit) of the DMAC control register L shown in Figure 67, and finally set the DMAC control register H's DMA enable bit to "1" so as to make the DMA request acceptable.

When the contents of TCR are cleared to "0", the terminal count signal ($\overline{TC}$) is output, and at the same time, the interrupt request bit of the DMA interrupt control register is set to "1".

To forcedly terminate the DMA transfer, input "L" level to pin $\overline{TC}$ or write the value "0" to the DMA enable bit. At this time, the interrupt request bit of the DMA interrupt control register is not set to "1".

Figure 78 shows a timing diagram example in the single transfer mode on the following conditions:

- Transfer unit: 16 bits
- Transfer method: 2-bus cycle transfer
- Transfer mode: Burst transfer mode (edge sense)
- Transfer source address direction: Forward.
- Transfer destination address direction: Forward.
- Transfer source wait: 0 wait
- Transfer destination wait: 0 wait

As 2-bus cycle transfer mode is selected, a read operation is performed in the first bus cycle. First, the address written into the SAR is output to the address bus and then inputted into the incrementor/decrementor (hereinafter referred to as I/D). The I/D adds 1 or 2 to the inputted address and outputs the result back to the SAR. If one 16-bit transfer operation is not enough to complete the read operation, the read operation is performed within 2 bus cycles to achieve the purpose.

The operation is performed in the next bus cycle. First, the address written in the DAR is output to the address bus and then inputted into the I/D. The I/D adds 1 or 2 to the inputted address and outputs the result back to the DAR. If one 16-bit transfer operation is not enough to complete the write operation, the write operation is performed within 2 bus cycles to achieve the purpose. The operation performed so far is called the write cycle. The data stored in the BIU's data latch in the read cycle is output to the data bus in the write cycle and written into the destination memory or external I/O. The operations performed so far complete the transfer of 1 transfer unit. In the 2-bus cycle transfer, the read and write cycle combination is called the DMA transfer cycle. DMA transfer is executed by repeating the DMA transfer cycle.

In the 2-bus cycle transfer, the TCR varies in the read cycle. The remaining transfer bytes are read from the TCR in concurrence with address output from SAR in the read cycle and inputted into the decrementor (hereinafter referred to as D). The D subtracts 1 or 2 from the number of remaining bytes and outputs the result back to the TCR. In this manner, the contents of the TCR decrease each time when 1-transfer-unit data has been transferred. When the number of remaining bytes, which was read from the TCR, becomes "0", the DMA controller outputs the terminal count signal ($\overline{TC}$) to pin $\overline{TC}$, and at the same time, sets the interrupt request bit of the DMA interrupt control register to "1". At this time, the DMA enable bit is cleared to "0". As the burst transfer mode is selected in this example, the DMA request bit is also cleared to "0".

To forcedly terminate transfer, input "L" level to pin $\overline{TC}$ ($P4_2$) or write the value "0" to the DMA enable bit.

In the single transfer, the first values written in the SAR, DAR, and TCR are retained in the internal latches. Therefore, if DMA transfer is to be performed under the same conditions, it can be initiated simply by setting the DMA enable bit to "1".

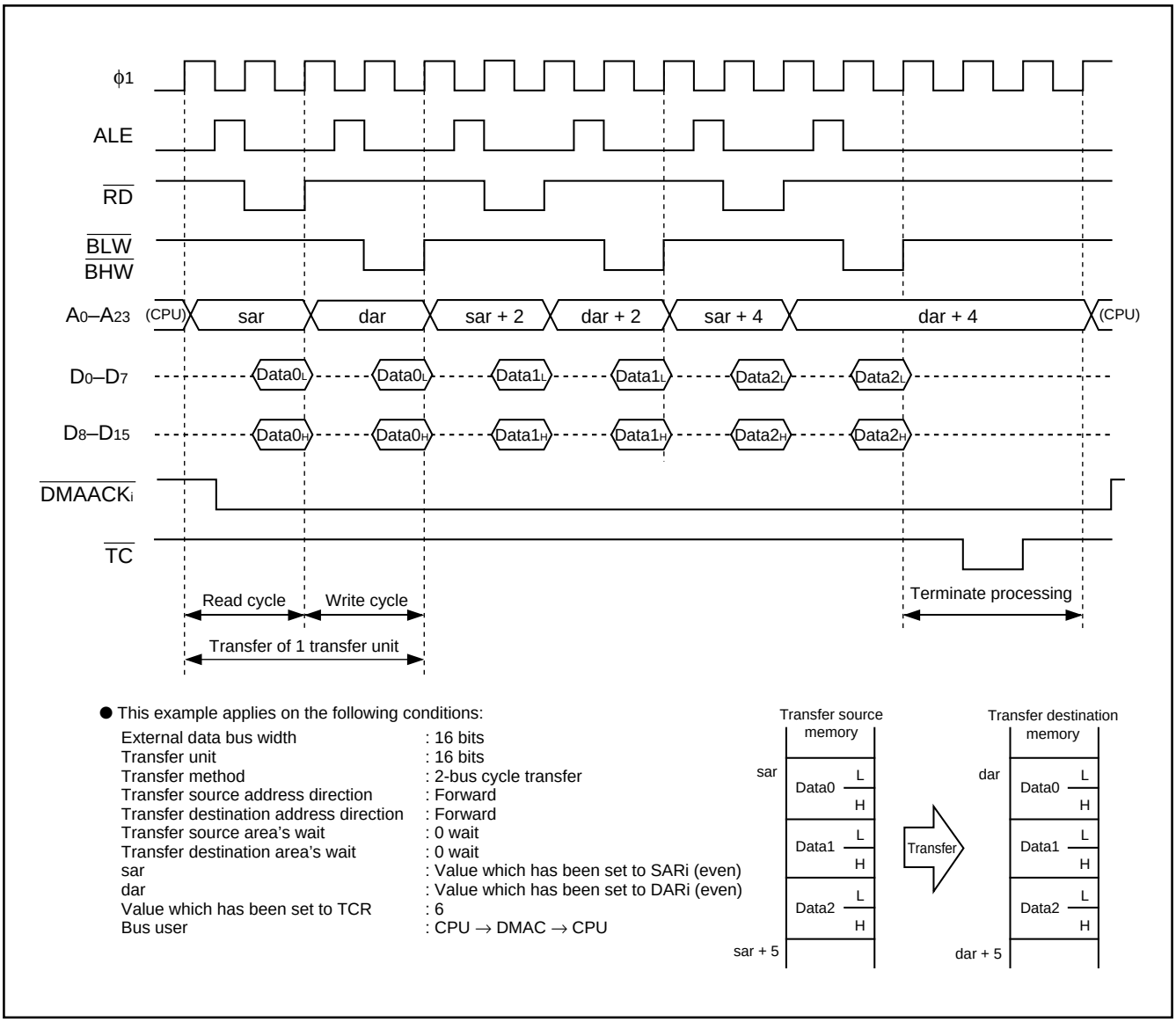


Fig. 78 Timing diagram example in single transfer mode (burst transfer mode)

(2) Repeat transfer mode

In the repeat transfer mode, the single transfer is repeated. First, set up the number of bits per 1 transfer unit, transfer method, transfer mode, and transfer address direction by using the DMAi mode register L. Next, write the transfer source block's transfer start address in the SAR and the transfer destination block's transfer start address in the DAR. Further, write the desired number of bytes to be transferred, into the TCR, and set up the DMAi control register and DMAC control register. The DMA request is now acceptable. When the DMA request occurs in this state, DMA transfer starts. Even when the number of remaining bytes, which was read from the TCR, becomes 0, the DMA enable bit is not cleared to "0". When the burst transfer mode is selected, the DMA request bit is not cleared to "0", also. When the cycle steal transfer mode is selected, the DMA request bit is cleared to "0" each time when 1-transfer-unit data has been transferred.

The values written in the SAR, DAR, and TCR first are retained in the internal latches. The contents of the latches are transferred to the SAR, DAR and TCR at the end of the last transfer cycle. Therefore, when the burst transfer mode is selected, the transfer operation is repeated starting with the values written first. When the cycle steal transfer mode is selected, these values are used as the initial values and transfer is performed each time the DMA request bit is set to "1".
To forcedly terminate transfer, input "L" level to the pin $\overline{TC}$ or write the value "0" to the DMA enable bit.
In the repeat transfer mode, $\overline{TC}$ signal output and the setting the interrupt request bit of the DMA interrupt control register to "1" are not performed.
Figure 78 shows the timing diagram example in the repeat transfer mode.

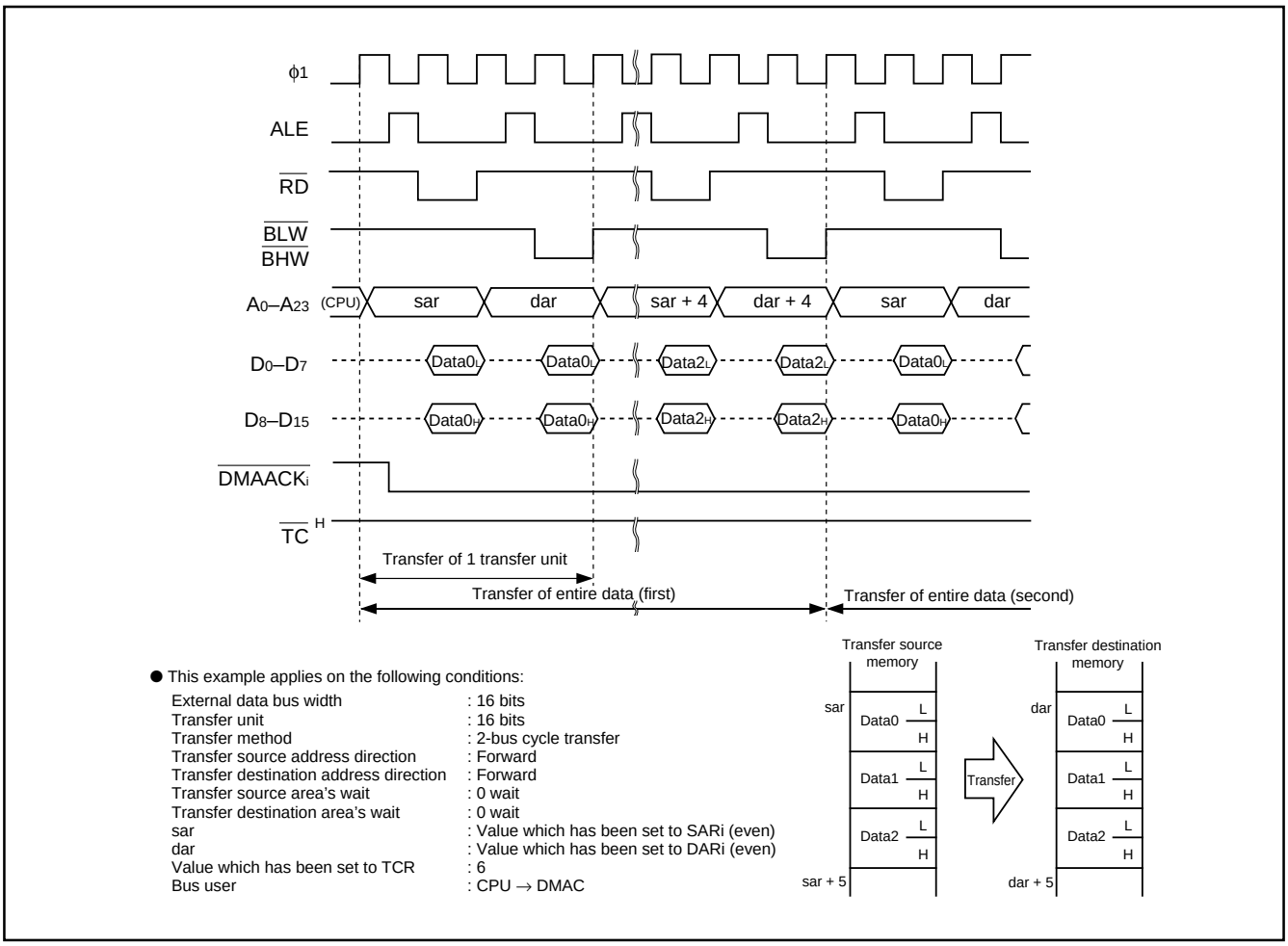


Fig. 79 Timing diagram example in repeat transfer mode (burst transfer mode)

(3) Array chain transfer mode

In the array chain transfer mode, one channel is used for the data transfer for two or more memory blocks.

Three parameters necessary for transfer, that is, the transfer source's transfer start address, transfer destination's transfer start address, and the number of transfer bytes, must be sequentially written into the transfer parameter memory. The transfer parameter memory can be located in an arbitrary position in the memory space. Figure 80 shows a transfer parameter memory map example in the array chain transfer mode. All of the transfer parameters of the memory blocks to be transferred must be written into the transfer parameter memory. The transfer parameter memory format is shown in Figure 81. For 1-bus cycle transfer, the external I/O side's parameters are not needed. For transfer from external memory to external I/O, for instance, consecutively write the transfer source start addresses and the number of transfer bytes only, as shown in Figure 82. As the transfer destination's transfer start addresses need not be written, it is possible to save the transfer time and transfer parameter memory area.

In the single and repeat transfer modes, the values written in the SAR, DAR, and TCR first are retained in the internal latches. In the array chain transfer and link array chain transfer modes, however, these latches perform different functions.

The SAR latch serves as the transfer parameter register (hereinafter referred to as TPR), which indicates the start address of the transfer parameter memory. The TCR latch serves as the transfer block counter (hereinafter referred to as TBC), which indicates the number of transfer blocks. In the array chain transfer and link array chain transfer modes, writing a value to an SAR address causes that value to be written in the TPR, and writing a value to a TCR address causes that value to be written in the TBC.

The array chain transfer operations are detailed below.

In the array chain transfer mode, also, first, set up the DMAi mode register, DMAi control register, and DMAC control register. Write the start address of the transfer parameter memory into the SAR. This value is then written into the TPR. Be sure that an even-numbered address is set to the start address. Nothing needs to be written into the DAR. Into the TCR, write the desired number of memory blocks to be transferred. This number is then written into the TBC. When the DMA enable bit is set to "1" after completion of the above setup, DMA transfer becomes enabled.

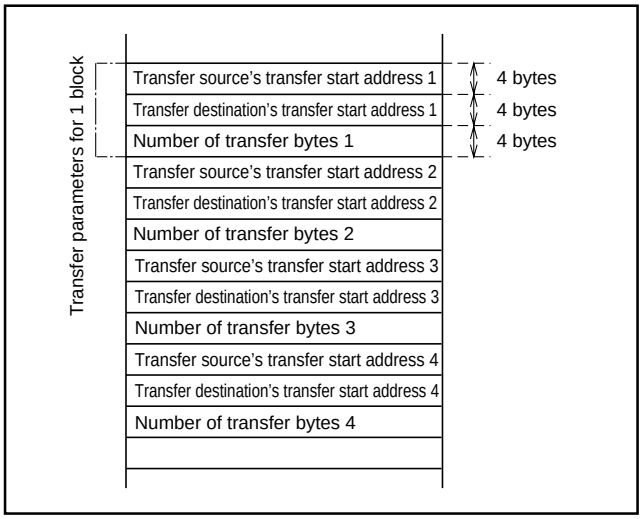


Fig. 80 Parameter memory map example in array chain transfer mode

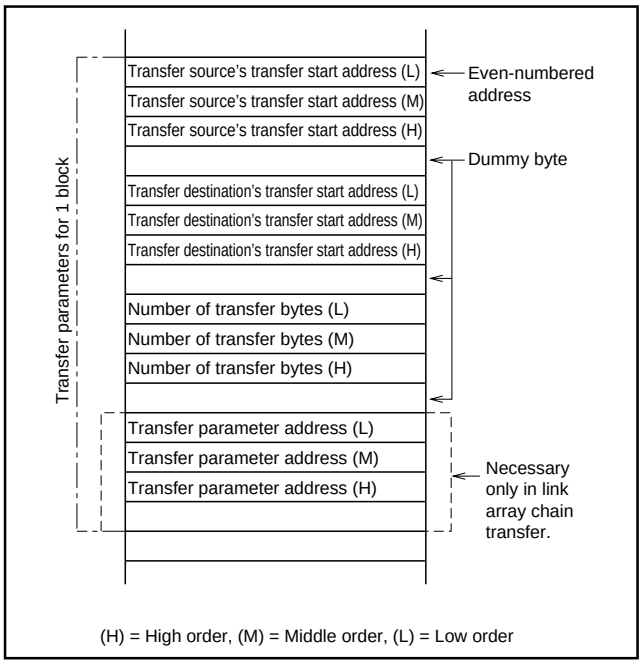


Fig. 81 Parameter memory format

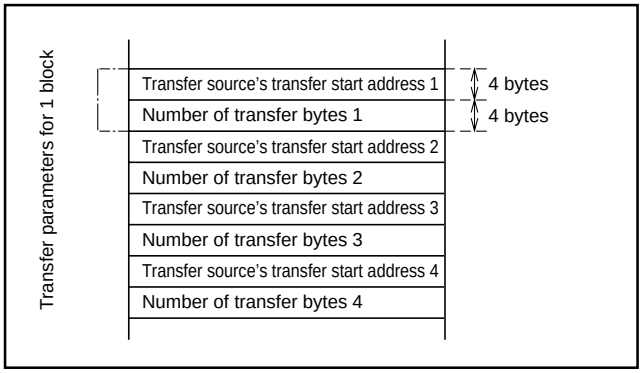


Fig. 82 Transfer parameter memory in 1 bus cycle transfer

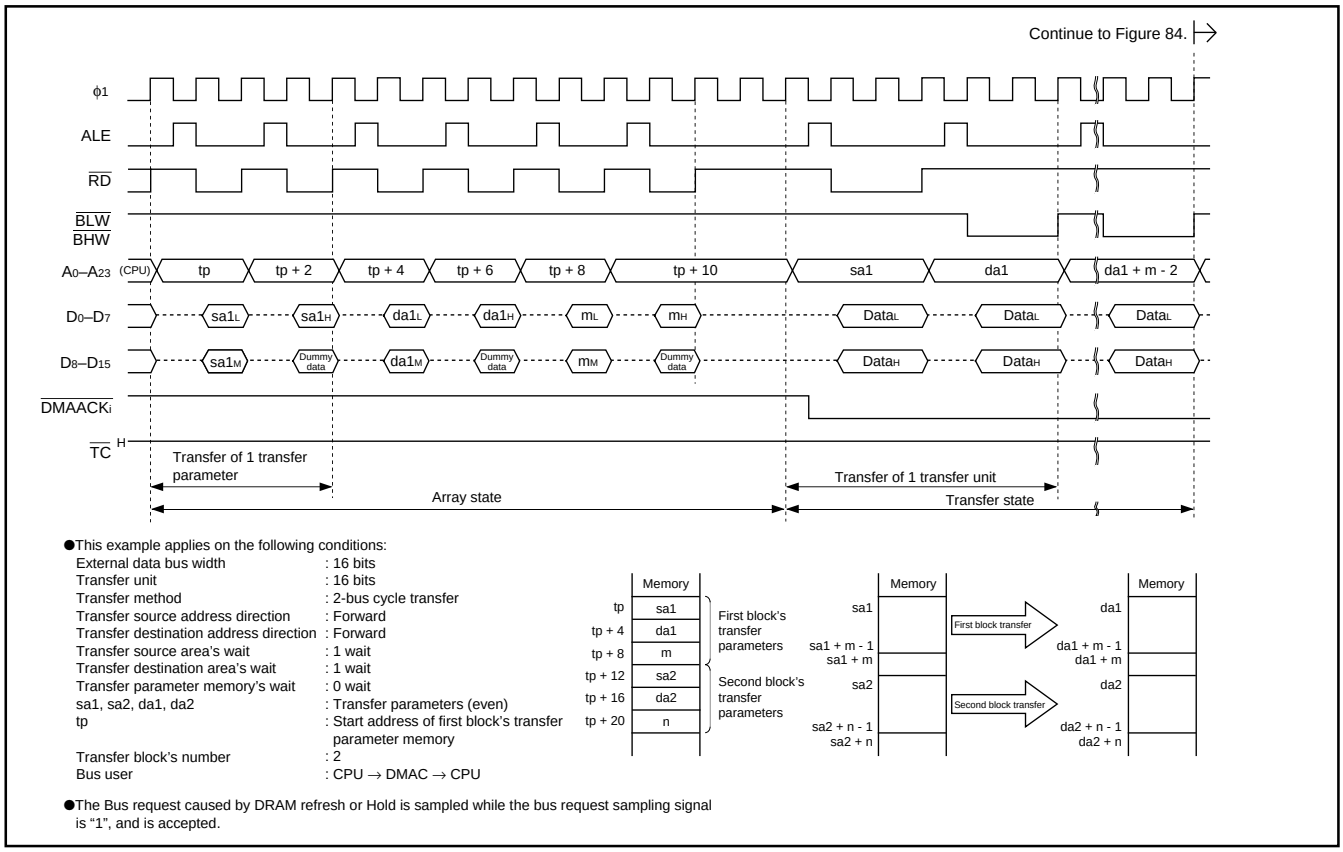
In the array chain transfer, the transfer parameters are first read from the transfer parameter memory and then written into the SAR, DAR, and TCR. This operations state is called the “array state”. Figures 83 and 84 show timing diagram examples in the array chain transfer mode (burst transfer mode). The DMA controller outputs the start address of the transfer parameter memory to the address bus, and sequentially stores the read data into the SAR, DAR, and TCR.

When the transfer parameters for 1 block are completely stored, the contents of the TBC are decremented by 1, and then, the first DMA transfer starts in accordance with the stored parameters. These operations for storing parameters are called “array state”.

In contrast to the array state, the state in which DMA transfer is active is called “transfer state”. In the transfer state, the same operations are performed as in the single transfer mode. Each time when 1-transfer-unit data has been transferred, the contents of the TCR are decremented by 1 in 8-bit transfer or by 2 in 16-bit transfer. Even when the contents of the TCR become 0, the DMA request bit and DMA enable bit are not cleared to “0” and the array state of the next block starts.

When the contents of the TBC are 0 at the start of the array state, the entire transfer operation is considered to be completed, and “L” level is output into pin $\overline{TC}$ to clear the DMA request bit and DMA enable bit and terminate array chain transfer. At the same time, the interrupt request bit of the DMA interrupt control register is set to “1”.

In the cycle steal transfer at the array chain transfer mode, one array state and transfer cycle of 1 transfer unit are made by one DMA request.



●This example applies on the following conditions:

External data bus width	: 16 bits
Transfer unit	: 16 bits
Transfer method	: 2-bus cycle transfer
Transfer source address direction	: Forward
Transfer destination address direction	: Forward
Transfer source area's wait	: 1 wait
Transfer destination area's wait	: 1 wait
Transfer parameter memory's wait	: 0 wait
sa1, sa2, da1, da2	: Transfer parameters (even)
tp	: Start address of first block's transfer parameter memory
Transfer block's number	: 2
Bus user	: CPU → DMAC → CPU

●The Bus request caused by DRAM refresh or Hold is sampled while the bus request sampling signal is "1", and is accepted.

Fig. 83 Timing diagram example in array chain transfer mode (burst transfer mode) (1)

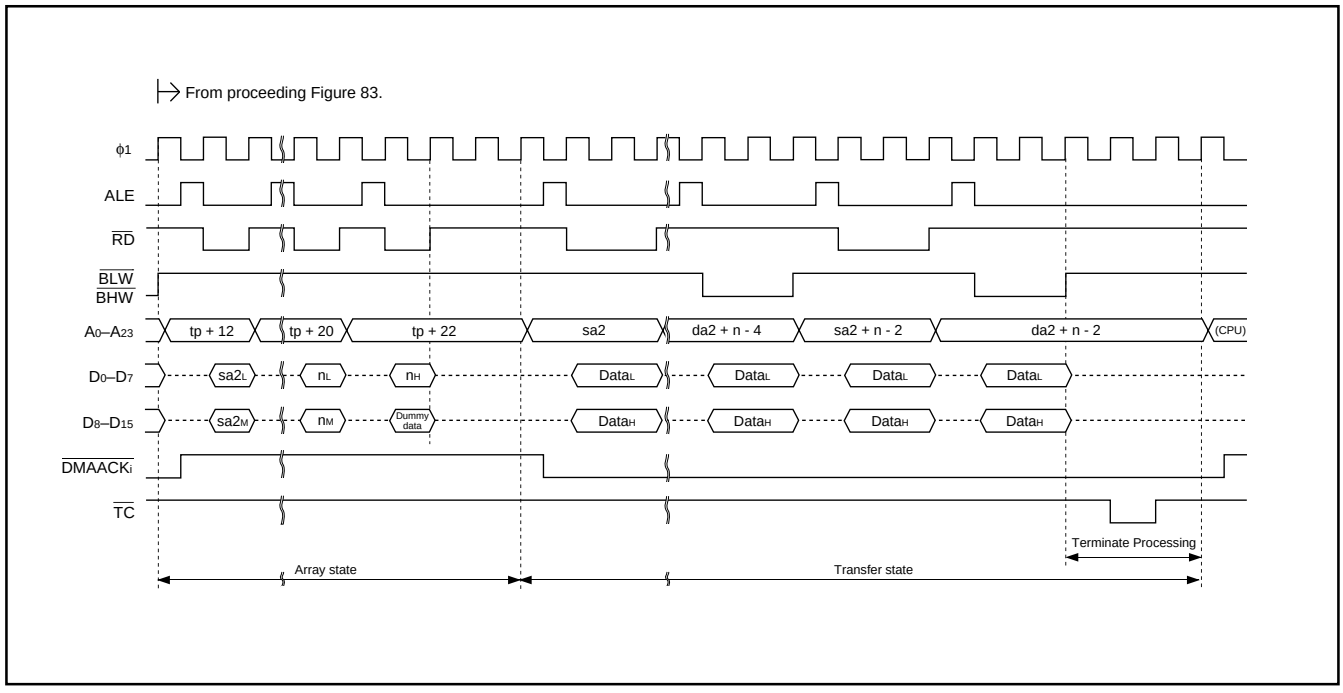


Fig. 84 Timing diagram example in array chain transfer mode (burst transfer mode) (2)

(4) Link array chain transfer mode

Figure 85 shows the perameter memory map in the link array chain transfer mode. As shown in this figure, not only the transfer source's transfer start address, transfer destination's transfer start address, and number of transfer bytes, but also the start address of the memory block which contains the next transfer parameters is stored. In the transfer parameter of the last block, be sure to set "000000_{16}" as the start address of the next transfer parameter. For 1-bus cycle transfer, the external I/O side's parameters are not needed.

In the link array chain transfer, also, the DMAi mode registers L and H, DMAi control register, and DMAC control registers L and H must be set up. Into the SAR, write the start address of the memory block that stores the parameters for the first transfer. This value is then written into the TPR. Be sure that an even-numbered address is set to the start address.

Nothing needs to be written in the DAR. Write the value 1 or more into the TCR. When the DMA enable bit is set to "1" after completion of the above setup, DMA transfer becomes enabled.

In the link array chain transfer, the transfer parameters are first read from the transfer parameter memory and then written into the SAR, DAR, and TCR. Further, the start address of the memory block that contains the next parameters has been written into the TPR. In the link array chain transfer mode, the state so far is referred to as the array state.

The DMA controller sequentially outputs the transfer parameters to the address bus, beginning with the start address of the memory block, storing the transfer parameters. The read data are sequentially stored into the SAR, DAR, and TCR, and then the start address of the memory block, containing the next parameters, is written into the TPR. A DMA transfer is made in accordance with the parameters read from the transfer parameter memory. The transfer state is the same as in the single transfer mode. The contents of the TCR are decremented by 1 or 2 each time when 1-transfer-unit data has been transferred.

Even when the contents of the TCR become 0, the DMA request bit and DMA enable bit are not cleared to "0" but the array state starts again. When the contents of the TPR are 0 at this time, however, "L" level is output into pin $\overline{TC}$ to clear the DMA request bit and DMA enable bit to "0" and terminate the link array chain transfer. At the same timing, the interrupt request bit of the DMA interrupt control register is set to "1".

In the cycle steal transfer at the link array chain transfer mode, one array state and the transfer cycle of 1 transfer unit are made by one DMA request.

Figures 86 and 87 show timing diagram examples in the link array chain transfer mode (burst transfer mode).

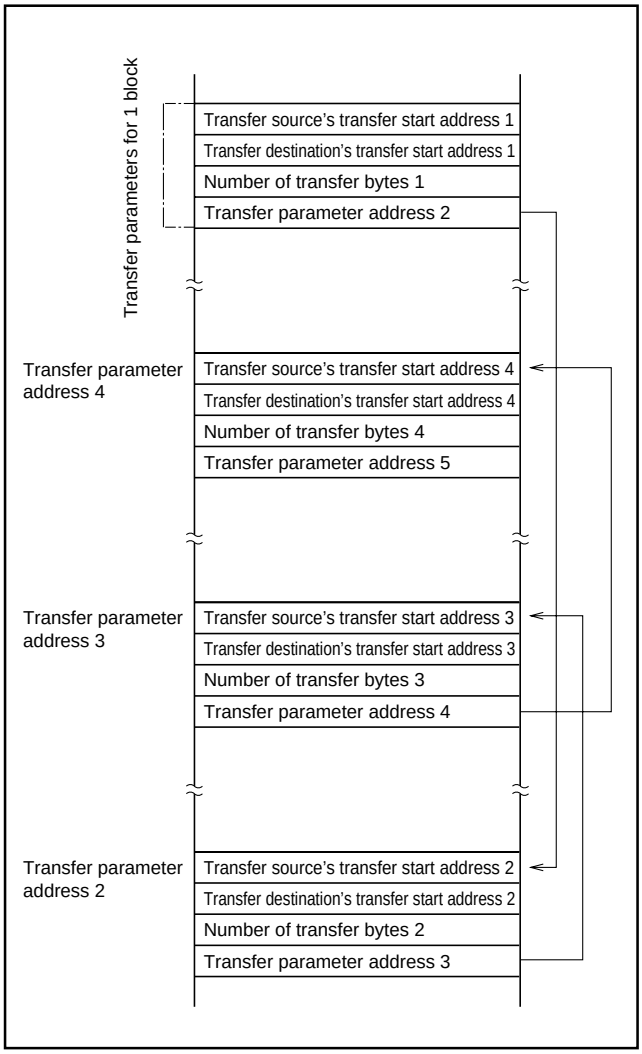


Fig. 85 Parameter memory map example in link array chain transfer mode

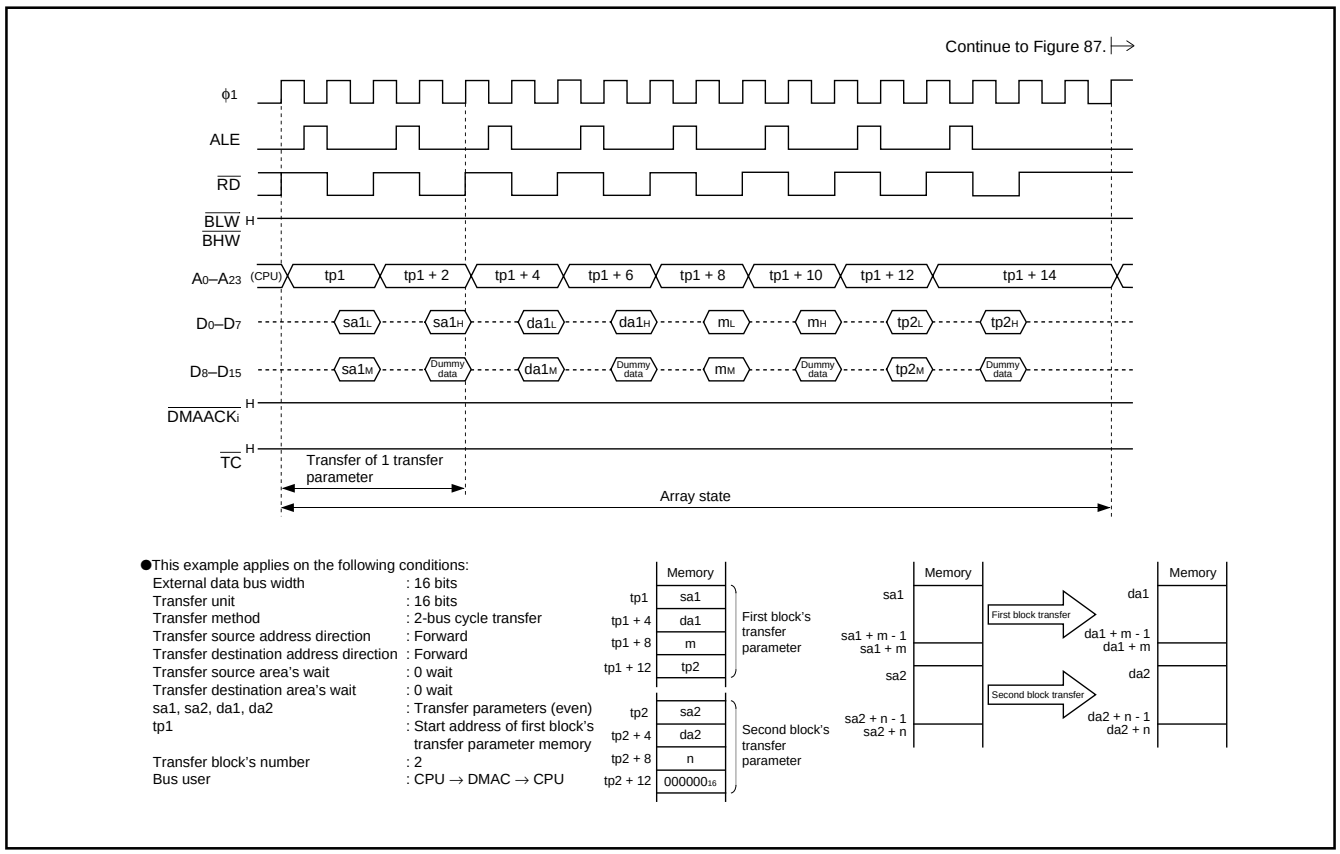


Fig. 86 Timing diagram example in link array chain transfer mode (burst transfer mode) (1)

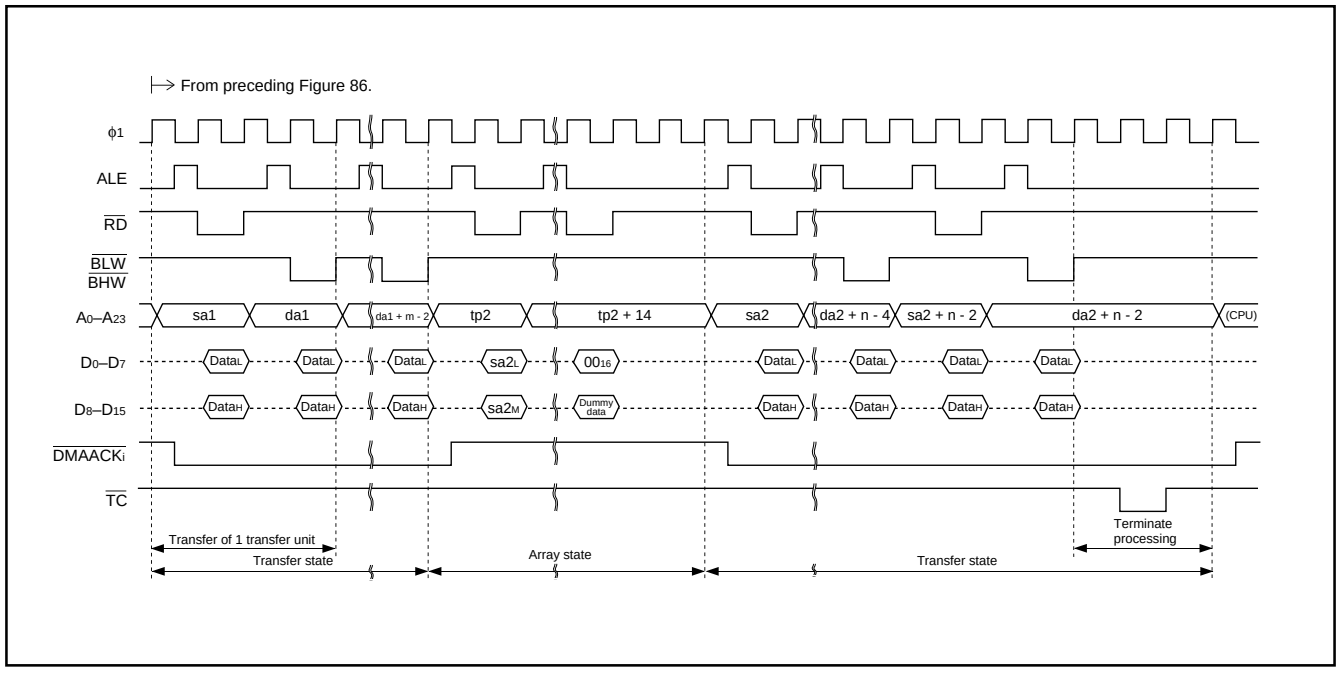


Fig. 87 Timing diagram example in link array chain transfer mode (burst transfer mode) (2)

DRAM CONTROLLER

The DRAM controller directly accesses the DRAM located in the external chip select area ($\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$).

Figure 88 shows the block diagram of the DRAM controller. Table 19 shows the functions of the DRAM-related signals, and Table 20 shows the relationship between the external data bus width and the multiplexed addresses.

The start address, block size, external data bus width, and DRAM space of the chip select area, which is to be accessed by the DRAM controller, are specified by the $\overline{CSj}$ control register L, $\overline{CSj}$ control register H, and the area $\overline{CSj}$ start address register in the chip select wait controller. For more details, refer to the section on the chip select wait controller.

Figure 89 shows the bit configuration of the $\overline{CSj}$ control register L with use of the DRAM controller. Bit 4 is the DRAM space designation bit. When bit 4 is set to "1", pins A_8/MA_0 to A_{16}/MA_8, A_{18}/MA_9, A_{20}/MA_{10}, A_{22}/MA_{11}, and $P9_4$ to $P9_6$, become the output pins for the DRAM control signals.

Figure 90 shows the bit configuration of the DRAM control register.

Bit 0 is the byte control select bit. When the device type of DRAM to be connected is 1$\overline{CAS}$/2$\overline{W}$, be sure to clear this bit to "0", and when the device type of DRAM to be connected is 2$\overline{CAS}$/1$\overline{W}$, be sure to set this bit to "1". When the external data bus width = 8 bits, however, be sure to clear this bit to "0". Table 21 shows the relationship between the byte control select bit and the pin functions. Each of Figures 91 and 92 shows an operating waveform example of the DRAM control signals, address bus, and data buses with 1$\overline{CAS}$/2$\overline{W}$ or 2$\overline{CAS}$/1$\overline{W}$ selected.

Bit 4 of the DRAM control register is the self-refresh operation select bit and controls the DRAM self-refresh operation in the stop mode; "0" disables the self-refresh operation in the stop mode, and "1" enables the self-refresh operation. Bit 7 is the refresh timer count start bit. The refresh timer starts counting when this bit is set to "1".

Figure 94 shows an operating waveform example of the DRAM control signals at refresh. This refreshing method, as shown in Figure 94, is the "$\overline{CAS}$ before $\overline{RAS}$ refresh". This method makes signal $\overline{CAS}$ falls before signal $\overline{RAS}$ falls.

The refresh interval is determined by the refresh timer (address $A9_{16}$). The refresh timer is an 8-bit timer performing a repetitive count with the reload register. The clock source is internal clock f_{32}. The refresh time issues a refresh request to the BIU each time when the refresh timer's count value reaches 00_{16}. Therefore, the relationship between the value to be loaded into the refresh timer, n (n = 01_{16} to FF_{16}), and DRAM refresh interval, m (μs), is as follows:

$$n = \{m \times f(X_{IN}) / 32\} - 1$$

Once the BIU accepts a refresh request, it performs the bus arbitration for the CPU and DMAC and outputs the refresh enable signal to the DRAM controller. Accordingly, the DRAM controller makes the refresh cycle ($\overline{CAS}$ before $\overline{RAS}$ refresh).

In the stop mode, since the refresh timer stops counting and the DRAM controller cannot perform the refresh operation ($\overline{CAS}$ before $\overline{RAS}$ refresh).

For DRAM supporting the self-refresh operation, by setting the self-refresh operation select bit to "1" before going into the stop mode, the self-refresh operation in the stop mode can be enabled.

Figure 95 shows an operating waveform example of the DRAM control signals at self-refresh.

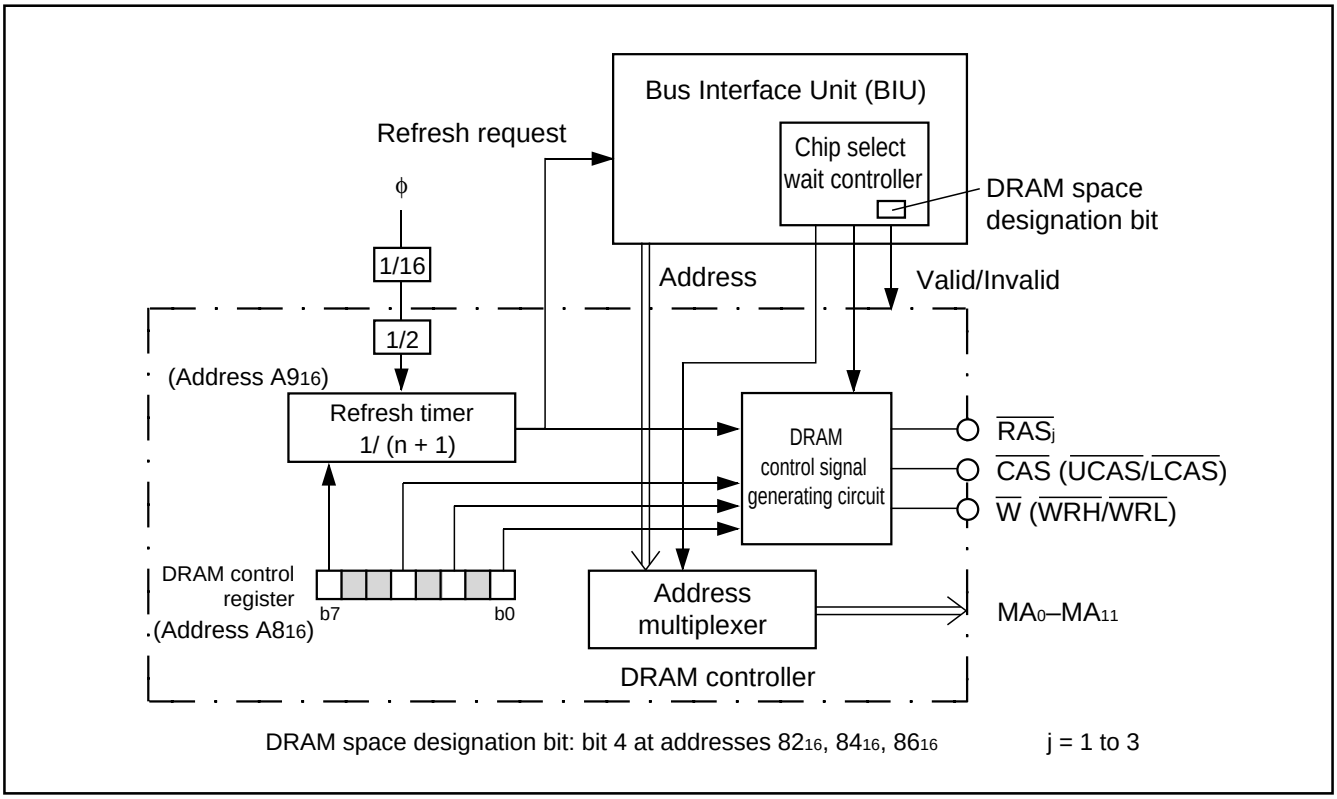


Fig. 88 Block diagram of DRAM controller

Table 19. Functions of DRAM-related signals

Signal		Functions	
		External data bus width = 16 bits	External data bus width = 8 bits
MA0–MA11		Multiplexed address output	
RASj (j = 1 to 3)		"L" when a row address is output.	
CAS	LCAS	"L" when a column address at an even-numbered address is output.	"L" when a row address is output.
	UCAS	"L" when a column address at an odd-numbered address is output.	"H" output (Fixed)
W	WRL	"L" when data at an even-numbered address is written.	"L" when data is written.
	WRH	"L" when data at an odd-numbered address is written.	"H" output (Fixed)

Table 20. Relationship between external data bus width and multiplexed addresses

Pin name			A8/MA0	A9/MA1	A10/MA2	A11/MA3	A12/MA4	A13/MA5	A14/MA6	A15/MA7	A16/MA8	A18/MA9	A20/MA10	A22/MA11
Output signal	External data bus width = 8 bits	Row address	A8	A9	A10	A11	A12	A13	A14	A15	A16	A18	A20	A22
		Column address	A0	A1	A2	A3	A4	A5	A6	A7	A17	A19	A21	A23
	External data bus width = 16 bits	Row address	A8	A9	A10	A11	A12	A13	A14	A15	A16	A18	A20	A22
		Column address	A0	A1	A2	A3	A4	A5	A6	A7	A8	A17	A19	A21

: These signals are not used.

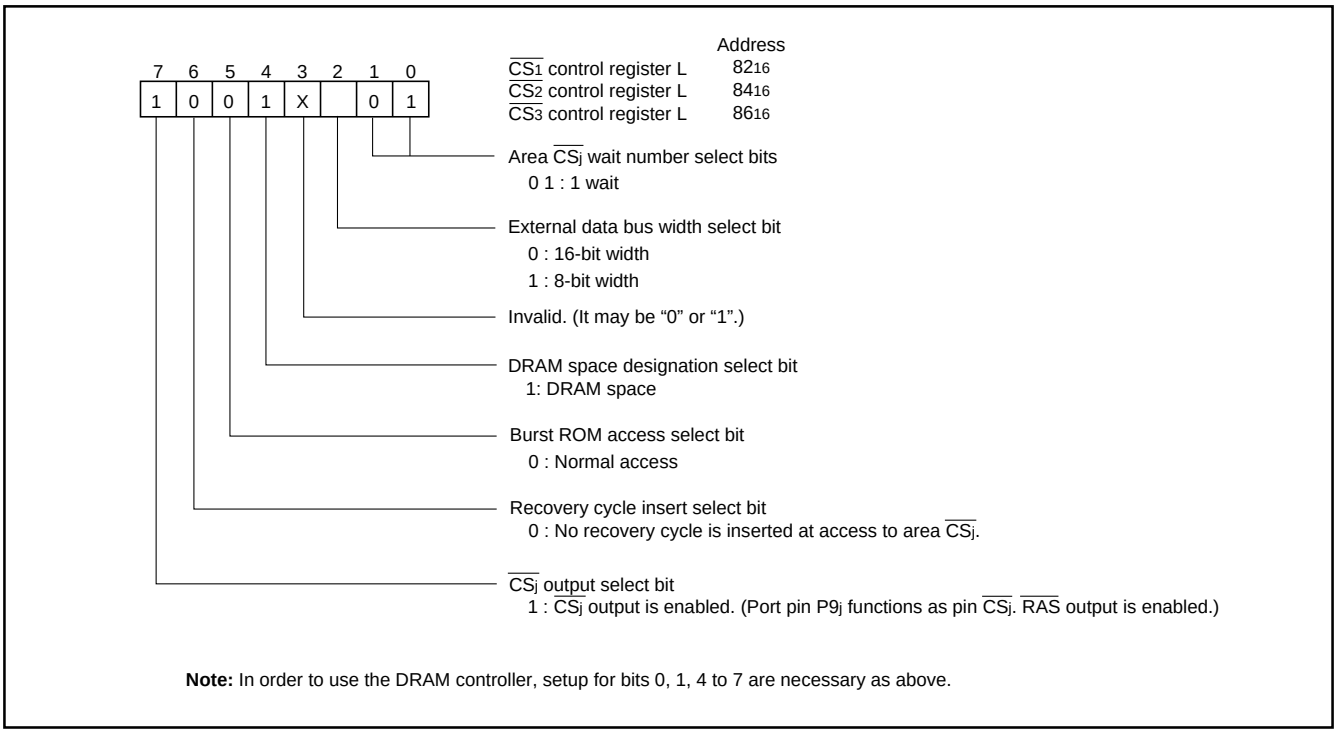


Fig. 89 Bit configuration of $\overline{CS_j}$ control register L with use of DRAM controller (j = 1 to 3)

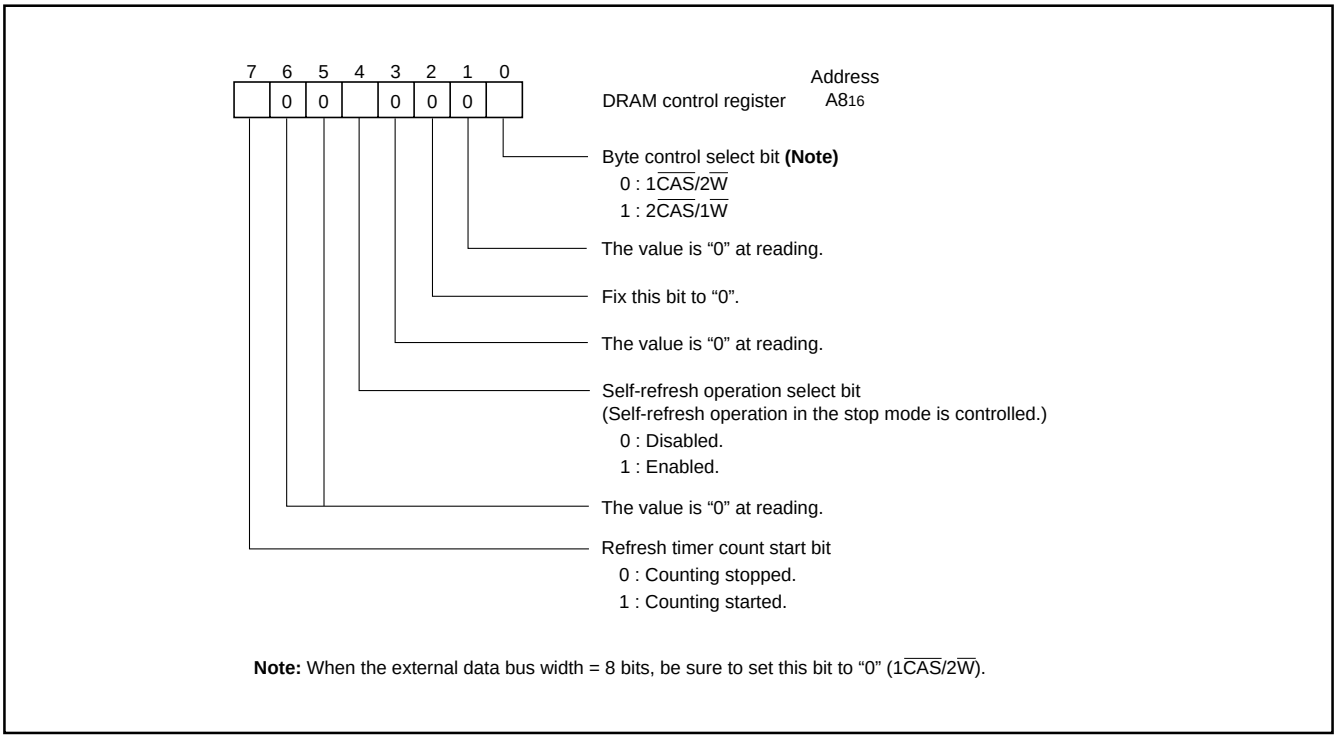
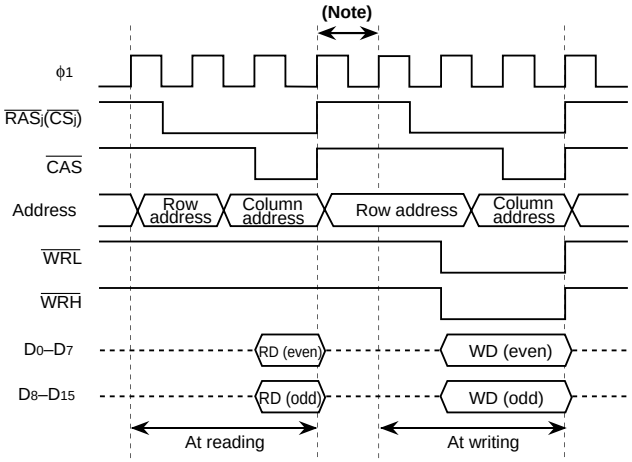


Fig. 90 Bit configuration of DRAM control register

Table 21. Relationship between byte control select bit and pin functions

Pin	Byte control select bit	
	0 (1CAS/2W)	1 (2CAS/1W)
P94	CAS	W
P95	WRL	LCAS
P96	WRH	UCAS

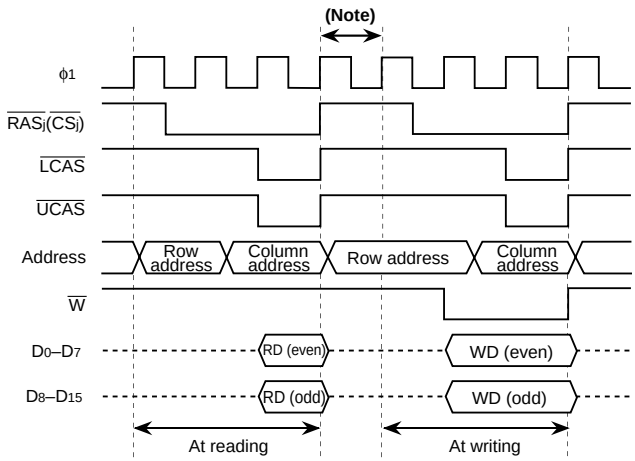
Operating waveform example when 16-bit data is accessed with
the external data bus width = 16 bits, starting at an even-numbered address



Note: When DRAM is continuously accessed with the fast page access OFF, 1 cycle of φ1 will be inserted between bus cycles.

Fig. 91 Operating waveform example of DRAM control signals, address bus, and data buses with 1CAS/2W selected

Operating waveform example when 16-bit data is accessed with
the external data bus width = 16 bits, starting at an even-numbered address



Note: When DRAM is continuously accessed with the fast page access OFF, 1 cycle of φ1 will be inserted between bus cycles.

Fig. 92 Operating waveform example of DRAM control signals, address bus, and data buses with 2CAS/1W selected

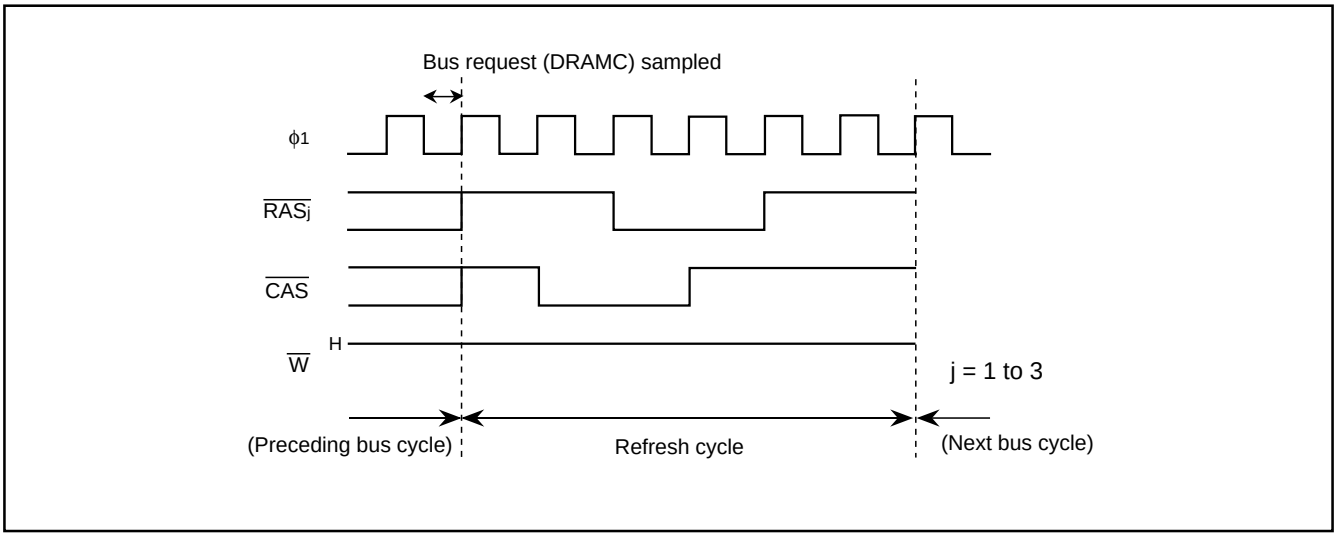


Fig. 94 Operating waveform example of DRAM control signals at $\overline{CAS}$ before $\overline{RAS}$ refresh

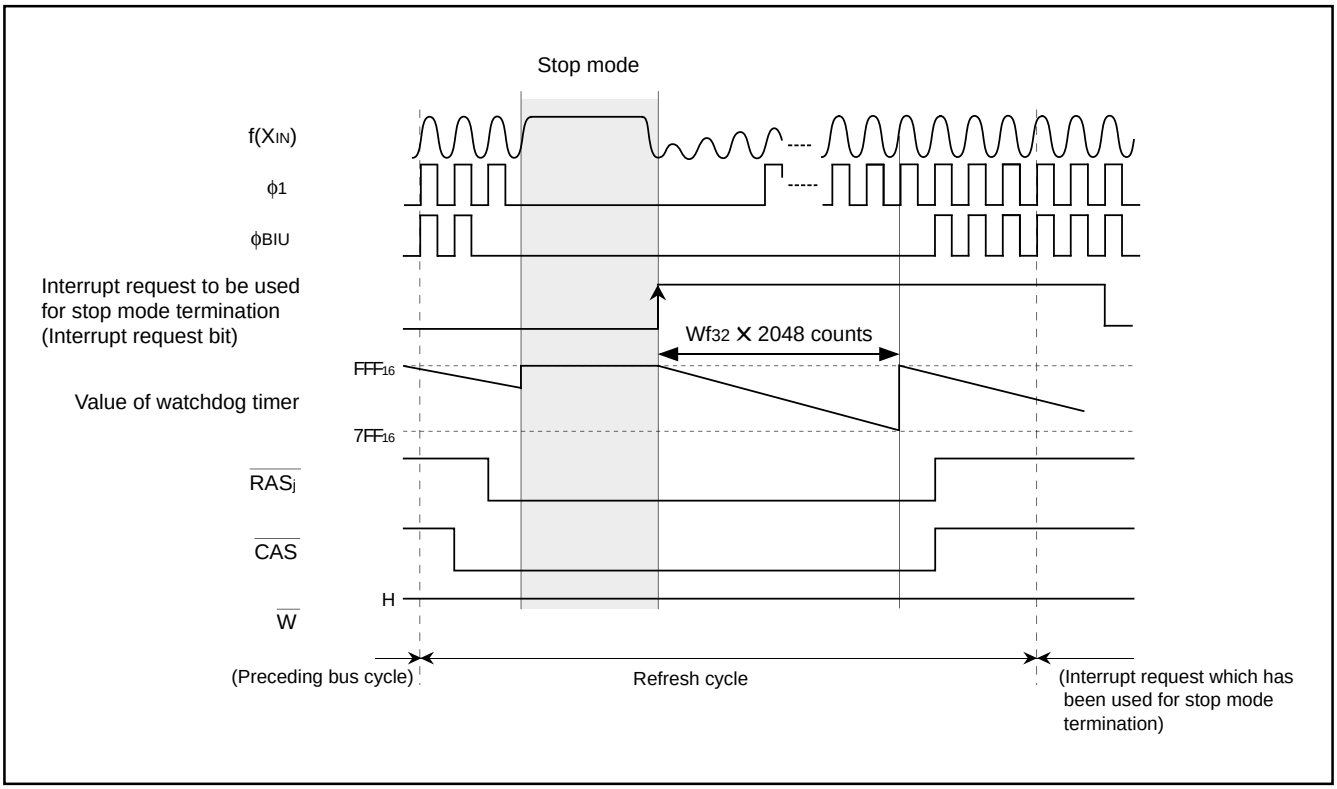


Fig. 95 Operating waveform example of DRAM control signals at self-refresh

REAL-TIME OUTPUT

Each of these microcomputers is equipped with the 8-bit real-time output function.

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 96.) 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 97 and 98 show the pulse output data register 0/1 (address $A2_{16}$/$A4_{16}$) bit configuration 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.

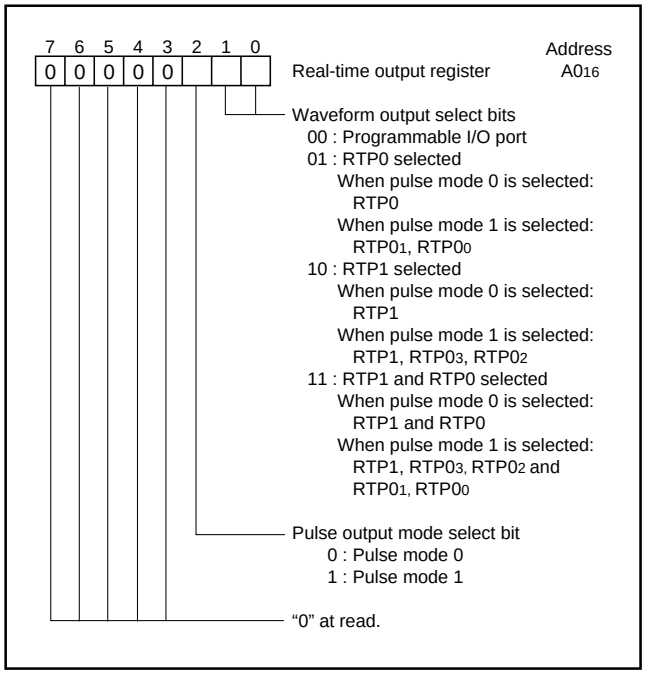


Fig. 96 Bit configuration of real-time output control register

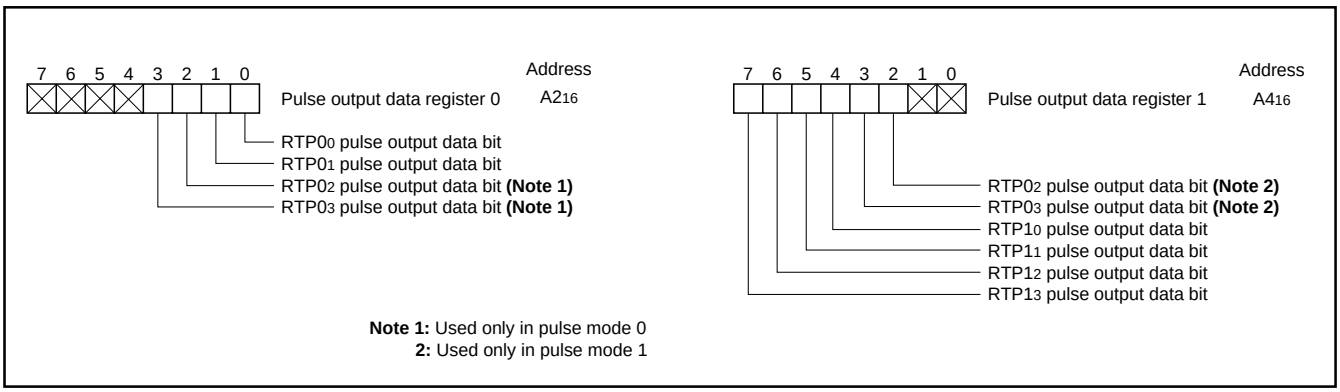


Fig. 97 Bit configuration of pulse output data register

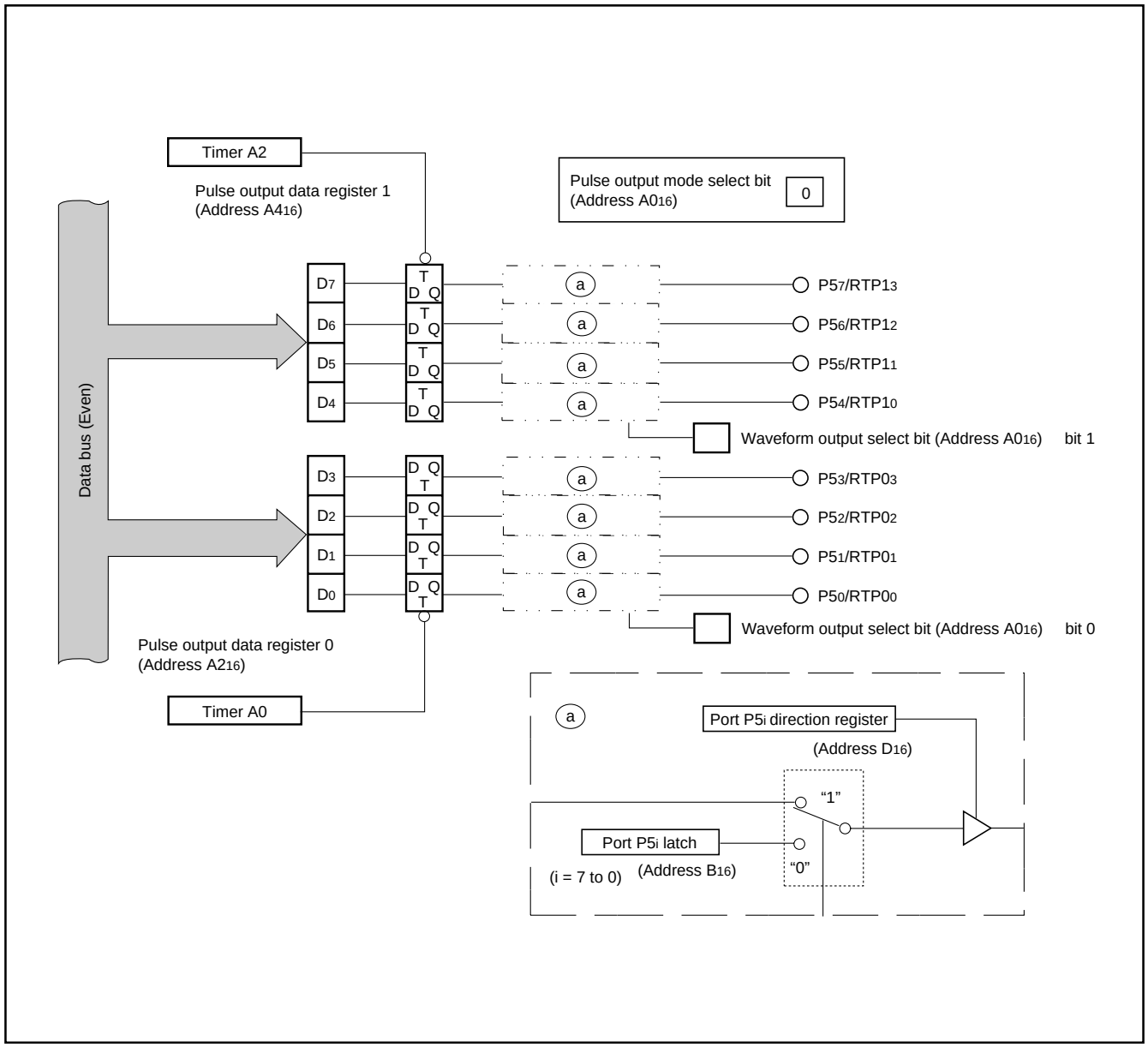


Fig. 98 Real-time output structure in pulse mode 0

(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).

Figure 99 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".

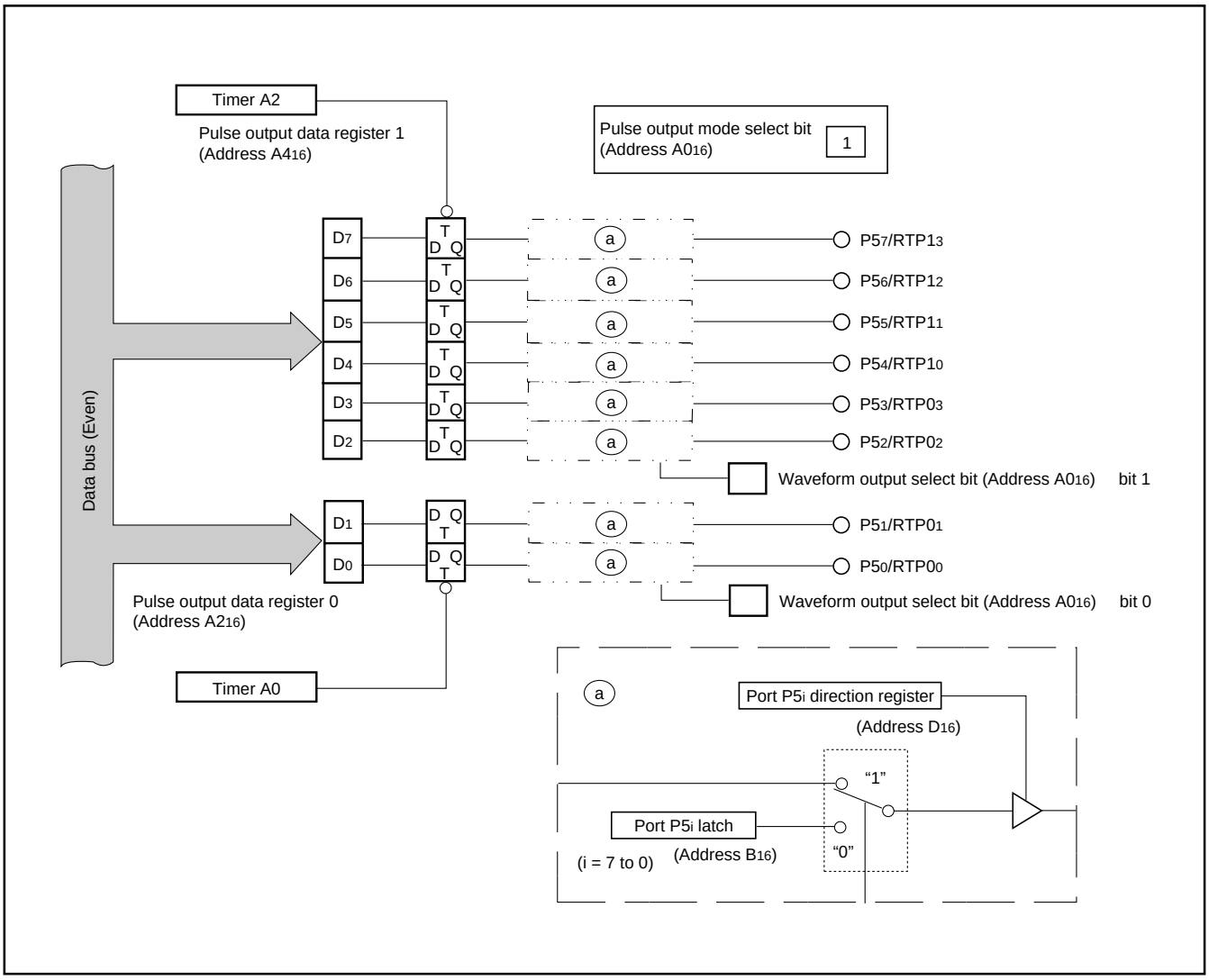


Fig. 99 Real-time output structure in pulse mode 1

Table 22 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 22. 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 0B16: Port P5
Address A216: Pulse output data register 0
Address A416: Pulse output data register 1

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software runaway and others. Figure 100 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. The watchdog timer frequency select register (bit 0 = watchdog timer frequency select bit) shown in Figure 101 selects which clock is to be counted.

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

FFF_{16} is set in the watchdog timer when "L" level voltage is applied to pin $\overline{RESET}$, 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.

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

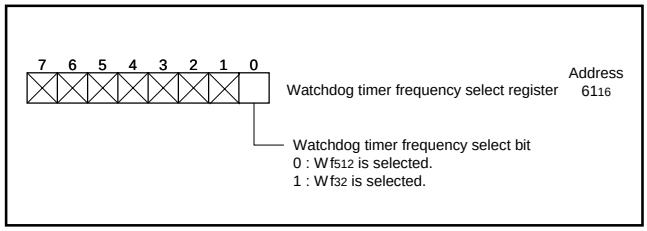


Fig. 101 Bit configuration of watchdog timer frequency select register

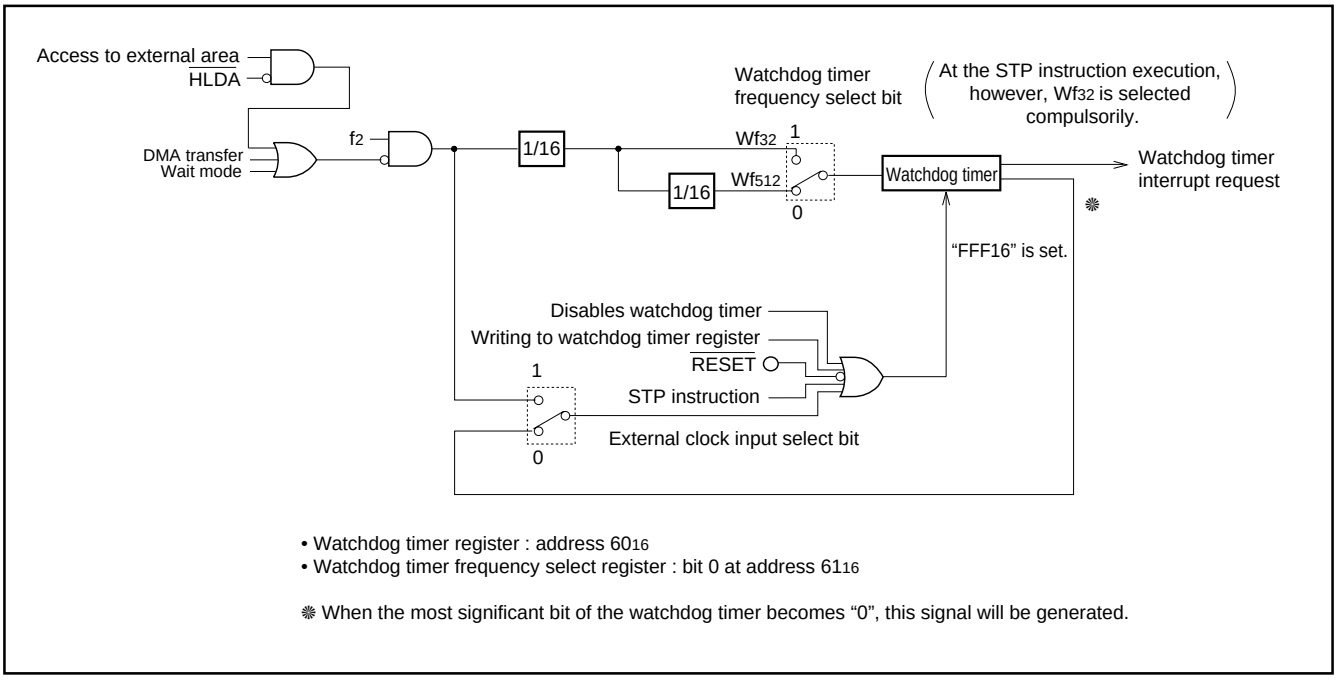


Fig. 100 Block diagram of watchdog timer

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.)

Stop and Wait mode

The stop (hereafter called STP) and the wait (hereafter called WIT) 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 102 and 103 show the bit configurations of the particular function select registers 0, 1. 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. After reset is removed, since the above bit is cleared to "0", 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 23 lists the microcomputer's operation in the STP and WIT modes.

STP mode

The execution of the STP instruction stops the oscillation circuit. It also stops clock source ϕ, $\phi 1$, ϕ_{BIU}, ϕ_{CPU}, and divide clocks $f_{1(\phi)}$ to f_{4096}, Wf_{32} and Wf_{512} in the "L" state. In the watchdog timer, "FFF$_{16}$" is automatically set, and regardless the contents of watchdog timer frequency select bit (bit 0 at address 61$_{16}$), the count source of the watchdog timer becomes Wf_{32}. This setting is terminated by clearance of the most significant bit of the watchdog timer or reset, and the count source is back to the one which was selected with the watchdog timer frequency select bit.

In the STP mode, the A-D converter, DMA controller, DRAM controller (Reflesh timer is also stopped.), and watchdog timer, which use divide clocks $f_{1(\phi)}$ 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.

The STP mode is terminated by acceptance of an interrupt request or reset, and the oscillation restarts. Supply of clock source ϕ, $\phi 1$, divide clocks $f_{1(\phi)}$ to f_{4096}, Wf_{32} and Wf_{512} is also restarted.

When the oscillation is restarted by the interrupt request acceptance, ϕ_{BIU} and ϕ_{CPU} are stopped at "L" level until the most significant bit of the watchdog timer, which is counted down with divide clock Wf_{32}, is cleared to "0". Note that, when the oscillation is restarted, supply of ϕ_{BIU} and ϕ_{CPU} starts immediately after the oscillation restarts. Therefore, the reset input must be raised to "H" after the enough oscillation-stabilizing time has elapsed.

The system where a stable clock is input from the external to pin X_{IN} is equipped with the mode where an instruction can be executed immediately after the STP mode termination. For details, refer to the section on "Stop of oscillation circuit" of the power saving function.

WIT mode

When the WIT instruction is executed with the internal clock stop select bit at WIT (bit 3 of the particular function select register 1 in Figure 103) = "0", ϕ_{BIU}, ϕ_{CPU}, and divide clocks Wf_{32} and Wf_{512} are stopped in the "L" state. However, the oscillation circuit, clock source ϕ, $\phi 1$, and divide clocks $f_{1(\phi)}$ to f_{4096} remain operating. Therefore, BIU, CPU, and DMA controller are stopped, whereas timers A and B, serial I/O, and the A-D converter, which use the divide clocks $f_{1(\phi)}$ to f_{4096}, are still operating. Because the reflesh timer of the DRAM con-

Table 23. Microcomputer's operation in STP and WIT modes

Instruction	Internal clock stop select bit at WIT	Operations in WIT and STP modes				
		Oscillation circuit	ϕ, $\phi 1$, $f_{1(\phi)}$ to f_{4096}	Wf_{32}, Wf_{512}	ϕ_{BIU}, ϕ_{CPU}	Peripheral devices using $f_{1(\phi)}$ to f_{4096}, Wf_{32}, Wf_{512}
STP	—	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, DMA controller: Stopped. DRAM controller: Stopped. (Reflesh timer is also stopped.) (Watchdog timer: Stopped.)
WIT	"0"	Active	Active	Stopped ("L")	Stopped ("L")	Timers A, B, Serial I/O, A-D converter: Operation is enabled. DRAM controller: Reflesh timer is operated. DMA controller: Stopped. (Watchdog timer: Stopped.)
	"1"	Active	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, DMA controller: Stopped. DRAM controller: Reflesh timer is operated. (Watchdog timer: Stopped.)

troller is operating, DRAM reflesh is performed. Note that the watchdog timer is stopped.

On the other hand, when the WIT instruction is executed with the internal clock stop select bit at WIT = "1", the oscillation circuit is operating, while ϕ_{BIU}, ϕ_{CPU}, and divide clocks $f_{1(\phi)}$ to f_{4096} stop operating. As a result, the A-D converter, DMA controller, and watchdog timer, which use divide clocks $f_{1(\phi)}$ to f_{4096}, Wf_{32} and Wf_{512}, are stopped. Because the reflesh timer of the DRAM controller is operating, DRAM reflesh is performed. 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 internal 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 is operating in the WIT mode, an interrupt processing can be executed just after the WIT mode termination.

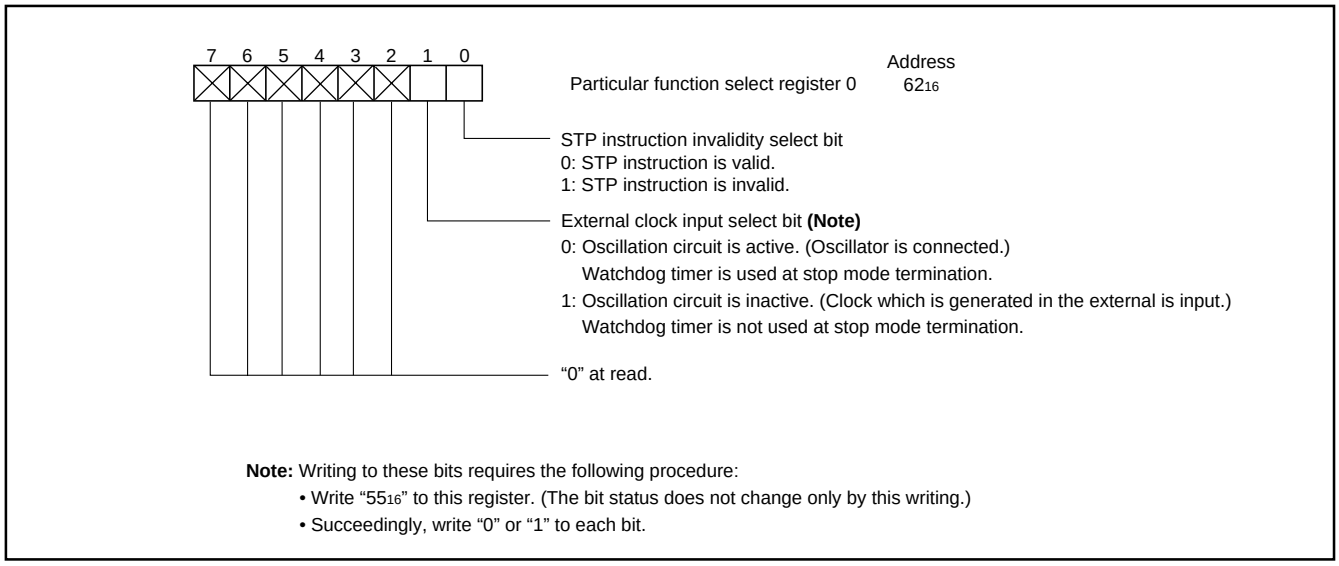


Fig. 102 Bit configuration of particular function select register 0

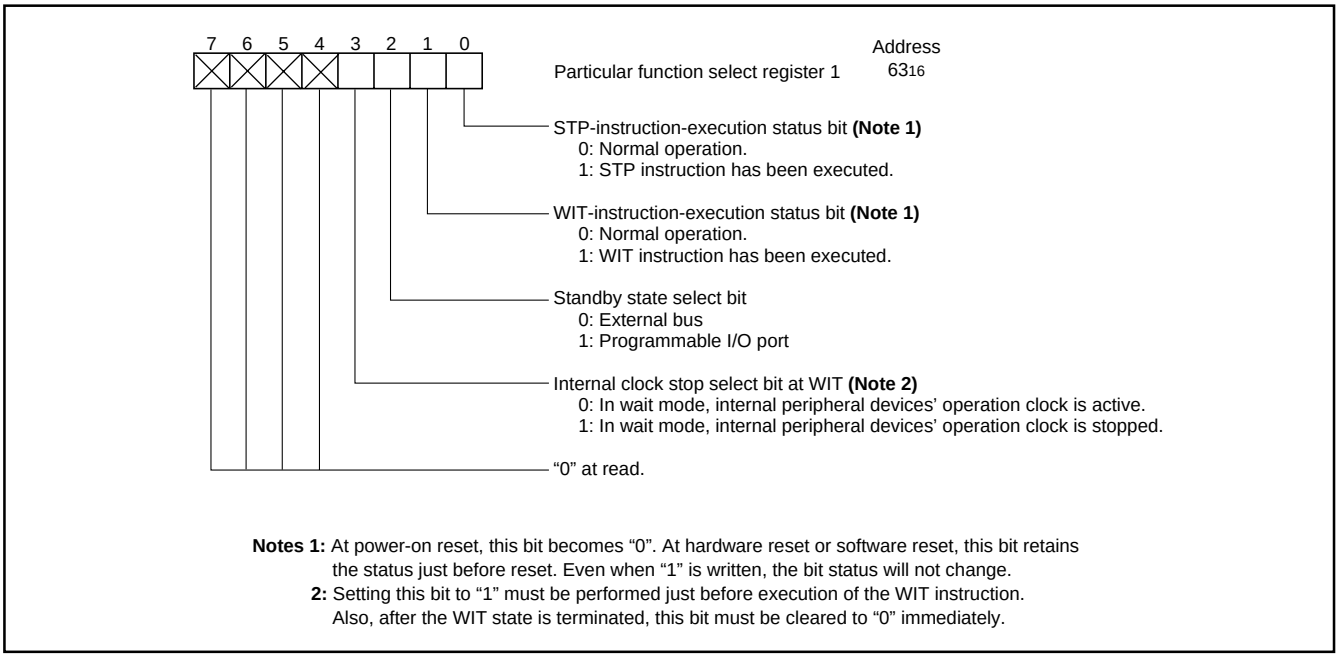


Fig. 103 Bit configuration of particular function select register 1

POWER SAVING FUNCTION

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

(1) Bus fixation in STP and WIT modes

By setting the standby state select bit (bit 2 of the particular function select register 1) to "1", in the stop or wait state, 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 saved. Table 24 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 state. At termination of the stop or wait mode, the original functions of external buses and bus control signals become valid.

Table 24. 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 $P10_7$ **(Note 2)**, $P11_0$ to $P11_7$ **(Note 2)**, $P0_0$ to $P0_7$ **(Note 2)**
D_0 to D_7, D_8 to D_{15}	D_0 to D_7, D_8 to D_{15} **(Note 1)**	$P1_0$ to $P1_7$ **(Note 2)**, $P2_0$ to $P2_7$
$\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$	$\overline{RD}$, $\overline{BLW}$, $\overline{BHW}$ **(Note 1)**	$P3_1$, $P3_2$ **(Note 2)**, $P3_3$
$\overline{CS_0}$	$\overline{CS_0}$	$P9_0$ **(Note 2)**

Notes 1: 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: Pin functions of port pins P0, P1, $P3_1$, $P3_2$, $P9_0$, P10, P11 are not shown in the pin configuration. However, relationship with corresponding bus signals and ports is listed in Table 24. For the addresses of these port's registers and direction registers, refer to the location of the perpheral devices' control register (Figures 4 and 5).

(2) Stop of internal clock in wait mode

In the WIT mode, if the internal peripheral devices need not to be operated, be sure to set the internal clock stop select bit at WIT (bit 3 of the particular function select register 1) to "1". As a result, the clock source for each internal peripheral device is stopped, and the power dissipation of the microcomputer can be saved.

For details, refer to the section on the Stop and Wait modes.

(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 circuit 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 the **STP** mode is terminated by an interrupt request occurrence, the watchdog timer is not used. Therefore, an instruction can be executed just after the termination of the **STP** mode. For details, refer to the section on the clock generating circuit and stop and wait modes.

(4) Disconnection from pin V_{REF}

When not using the A-D converter, by setting the V_{REF} connection select bit (bit 6 of the A-D control register 1) to "1", the 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 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 the A-D conversion starts a period of 1 µs or more has elapsed. For details, refer to the section on the A-D converter.

	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$_{16}$)···	××××0000
Port P4 direction register	(0C$_{16}$)···	×××00000
Port P5 direction register	(0D$_{16}$)···	00$_{16}$
Port P6 direction register	(10$_{16}$)···	×0000000
Port P7 direction register	(11$_{16}$)···	××××0000
Port P8 direction register	(14$_{16}$)···	×0000000
Port P9 direction register	(15$_{16}$)···	×0000000
Port P10 direction register	(18$_{16}$)···	00$_{16}$
Port P11 direction register	(19$_{16}$)···	00$_{16}$
Port P12 direction register	(1C$_{16}$)···	××××××000
A-D control register 0	(1E$_{16}$)···	00000???
A-D control register 1	(1F$_{16}$)···	×0000001
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$_{16}$)···	00001000
UART 1 Transmit/Receive control register 0	(3C$_{16}$)···	00001000
UART 0 Transmit/Receive control register 1	(35$_{16}$)···	00000010
UART 1 Transmit/Receive control register 1	(3D$_{16}$)···	00000010
Count start register	(40$_{16}$)···	00$_{16}$
One-shot start register	(42$_{16}$)···	0××00000
Up-down register	(44$_{16}$)···	00000000
Timer A clock division select register	(45$_{16}$)···	××××××00
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$_{16}$)···	00?×0000
Timer B1 mode register	(5C$_{16}$)···	00?×0000
Timer B2 mode register	(5D$_{16}$)···	00?×0000

	Address	
Processor mode register 0	(5E$_{16}$)···	(Note 2)00010(Note 2)0
Processor mode register 1	(5F$_{16}$)···	00(Note 2)1(Note 2)00
Watchdog timer	(60$_{16}$)···	FFF$_{16}$
Watchdog timer frequency select register	(61$_{16}$)···	×××××××0
Particular function select register 0	(62$_{16}$)···	××××××00
Particular function select register 1	(63$_{16}$)···	××××00(Note 3)
Debug control register 0	(66$_{16}$)···	1(Note 3)
Debug control register 1	(67$_{16}$)···	000(Note 3)000(Note 3)
$\overline{INT_3}$ interrupt control register	(6E$_{16}$)···	××××0000
$\overline{INT_4}$ interrupt control register	(6F$_{16}$)···	××××0000
A-D conversion interrupt control register	(70$_{16}$)···	××××?000
UART 0 transmit interrupt control register	(71$_{16}$)···	××××0000
UART 0 receive interrupt control register	(72$_{16}$)···	××××0000
UART 1 transmit interrupt control register	(73$_{16}$)···	××××0000
UART 1 receive interrupt control register	(74$_{16}$)···	××××0000
Timer A0 interrupt control register	(75$_{16}$)···	××××0000
Timer A1 interrupt control register	(76$_{16}$)···	××××0000
Timer A2 interrupt control register	(77$_{16}$)···	××××0000
Timer A3 interrupt control register	(78$_{16}$)···	××××0000
Timer A4 interrupt control register	(79$_{16}$)···	××××0000
Timer B0 interrupt control register	(7A$_{16}$)···	××××0000
Timer B1 interrupt control register	(7B$_{16}$)···	××××0000
Timer B2 interrupt control register	(7C$_{16}$)···	××××0000
$\overline{INT_0}$ interrupt control register	(7D$_{16}$)···	××000000
$\overline{INT_1}$ interrupt control register	(7E$_{16}$)···	××000000
$\overline{INT_2}$ interrupt control register	(7F$_{16}$)···	××000000
Processor status register PS		000??0001??
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: The status just after reset depends on the voltage level applied to pin MD0.
3: At power-on reset, these bits are clear to "0". At hardware or software reset, on the other hand, these bits retain the state just before reset.

Fig. 104 Microcomputer internal register's status just after reset (1)

	Address	b7	b6	b5	b4	b3	b2	b1	b0
$\overline{CS_0}$ control register L	(80_{16})···	(Note 2)	1	0	X	0	(Note 3)	1	0
$\overline{CS_0}$ control register H	(81_{16})···	X	X	X	X	X	0	0	1
$\overline{CS_1}$ control register L	(82_{16})···	0	1	0	0	0	(Note 3)	1	0
$\overline{CS_1}$ control register H	(83_{16})···	0	X	X	X	X	0	0	0
$\overline{CS_2}$ control register L	(84_{16})···	0	1	0	0	0	(Note 3)	1	0
$\overline{CS_2}$ control register H	(85_{16})···	0	X	X	X	X	0	0	0
$\overline{CS_3}$ control register L	(86_{16})···	0	1	0	0	0	(Note 3)	1	0
$\overline{CS_3}$ control register H	(87_{16})···	X	X	X	X	X	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
Flash memory control register	($9E_{16}$)···	X	X	0	0	0	0	0	1
Real-time output control register	($A0_{16}$)···	0	0	0	0	0	0	0	0
DRAM control register	($A8_{16}$)···	0	0	0	0	0	0	0	0
$\overline{CTS}$/$\overline{RTS}$ separate select register	(AC_{16})···	0	0	0	0	0	0	0	0
DMAC control register L	($B0_{16}$)···	0	0	0	0	0	0	0	0

	Address	b7	b6	b5	b4	b3	b2	b1	b0
DMAC control register H	($B1_{16}$)···	0	0	0	0	0	0	0	0
DMA0 interrupt control register	($B2_{16}$)···	X	X	X	X	0	0	0	0
DMA1 interrupt control register	($B3_{16}$)···	X	X	X	X	0	0	0	0
DMA2 interrupt control register	($B4_{16}$)···	X	X	X	X	0	0	0	0
DMA3 interrupt control register	($B5_{16}$)···	X	X	X	X	0	0	0	0
DMA0 mode register L	(CC_{16})···	0	0	0	0	0	0	0	0
DMA0 mode register H	(CD_{16})···	0	0	0	0	0	0	0	0
DMA0 control register	(CE_{16})···	0	0	0	0	0	0	0	0
DMA1 mode register L	(DC_{16})···	0	0	0	0	0	0	0	0
DMA1 mode register H	(DD_{16})···	0	0	0	0	0	0	0	0
DMA1 control register	(DE_{16})···	0	0	0	0	0	0	0	0
DMA2 mode register L	(EC_{16})···	0	0	0	0	0	0	0	0
DMA2 mode register H	(ED_{16})···	0	0	0	0	0	0	0	0
DMA2 control register	(EE_{16})···	0	0	0	0	0	0	0	0
DMA3 mode register L	(FC_{16})···	0	0	0	0	0	0	0	0
DMA3 mode register H	(FD_{16})···	0	0	0	0	0	0	0	0
DMA3 control register	(FE_{16})···	0	0	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: The status just after reset depends on the voltage level applied to pin MD0.
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. 105 Microcomputer internal register's status just after reset (2)

RESET CIRCUIT

While the power source voltage satisfies the recommended operating condition, reset state is removed if pin $\overline{RESET}$'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 104 and 105 show the microcomputer internal register's status just after reset, and Figure 106 shows an operation example of the reset circuit. Apply "L" level voltage to pin $\overline{RESET}$ 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.

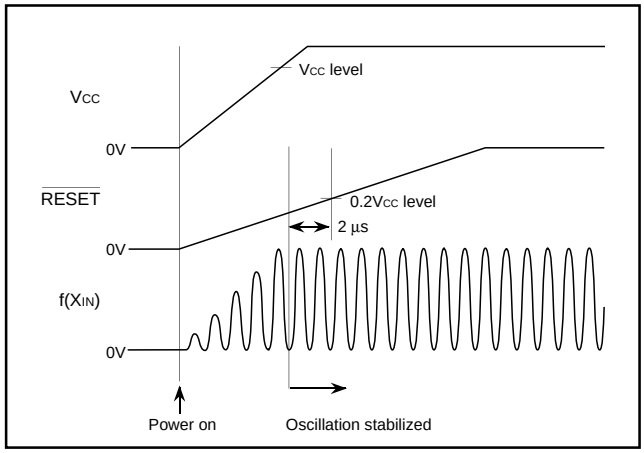


Fig. 106 Operation example of reset circuit (Note that proper evaluation is necessary in the system development stage.)

INPUT/OUTPUT PINS

Each of ports P0 to P12 has an 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”.

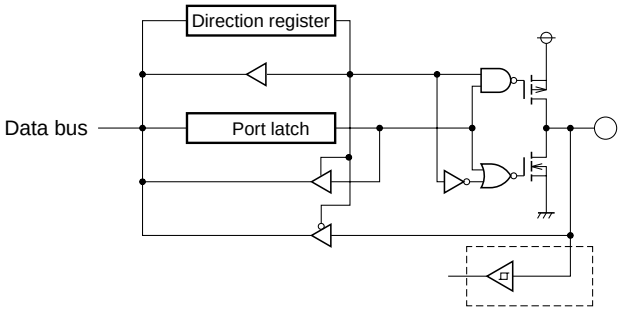
When a pin is programmed as an output pin, the data written to its port latch is output to the output pin. When a pin is programmed as an output pin, the contents of the port latch are read out instead of the value of the pin. Accordingly, a previously output value can be read out correctly even when 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 placed in the flooting state, and the value input to the pin can be read out correctly. When a pin is programmed as an input pin, the data can be written only in the port latch, and the pin remains floating.

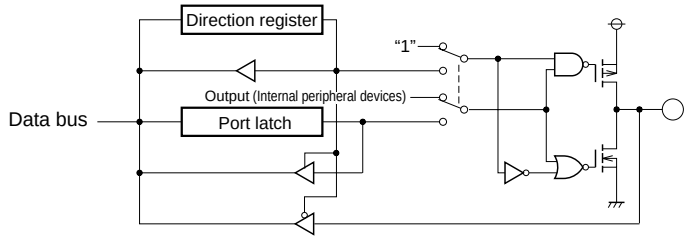
Each of Figures 107 and 108 shows the block diagram for each port pin.

[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$_0$/$\overline{RDY}$, P4$_3$/$\overline{HOLD}$,
P6$_1$/TA1$_{IN}$/$\overline{DMAREQ_0}$,
P6$_3$/TA3$_{IN}$/$\overline{DMAREQ_1}$,
P6$_5$/TA4$_{IN}$/$\overline{DMAREQ_2}$, P6$_6$/$\overline{DMAREQ_3}$,
P8$_1$/RxD$_1$, P8$_5$/RxD$_0$, P12$_0$/$\overline{INT_0}$/TB0$_{IN}$,
P12$_1$/$\overline{INT_1}$/TB1$_{IN}$, P12$_2$/$\overline{INT_2}$/TB2$_{IN}$

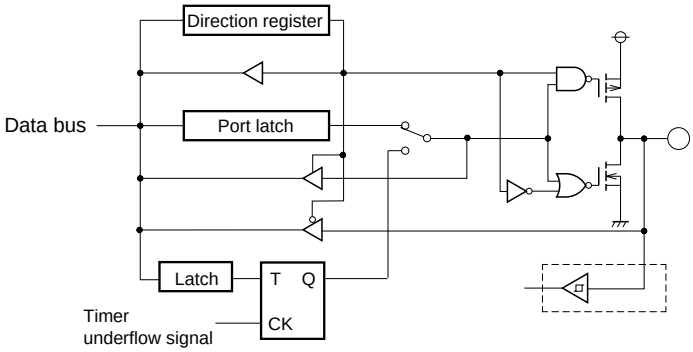


P4$_0$/ALE, P4$_1$/ϕ_1, P4$_4$/$\overline{HLDA}$,
P6$_0$/TA1$_{OUT}$/$\overline{DMAACK_0}$,
P6$_2$/TA3$_{OUT}$/$\overline{DMAACK_1}$,
P6$_4$/TA4$_{OUT}$/$\overline{DMAACK_2}$,
P8$_0$/TxD$_1$, P8$_4$/TxD$_0$, P9$_0$/$\overline{CS_0}$,
P9$_1$/$\overline{CS_1}$/$\overline{RAS_1}$, P9$_2$/$\overline{CS_2}$/$\overline{RAS_2}$,
P9$_3$/$\overline{CS_3}$/$\overline{RAS_3}$, P9$_4$/$\overline{CAS}$/$\overline{W}$,
P9$_5$/$\overline{WRL}$/$\overline{LCAS}$, P9$_6$/$\overline{WRH}$/$\overline{UCAS}$



[Inside dotted-line not included]
P5$_2$/RTP0$_2$, P5$_3$/RTP0$_3$, P5$_4$/RTP1$_0$,
P5$_5$/RTP1$_1$

[Inside dotted-line included]
P5$_1$/TA0$_{IN}$/RTP0$_1$, P5$_7$/TA2$_{IN}$/RTP1$_3$



P5$_0$/TA0$_{OUT}$/RTP0$_0$, P5$_6$/TA2$_{OUT}$/RTP1$_2$

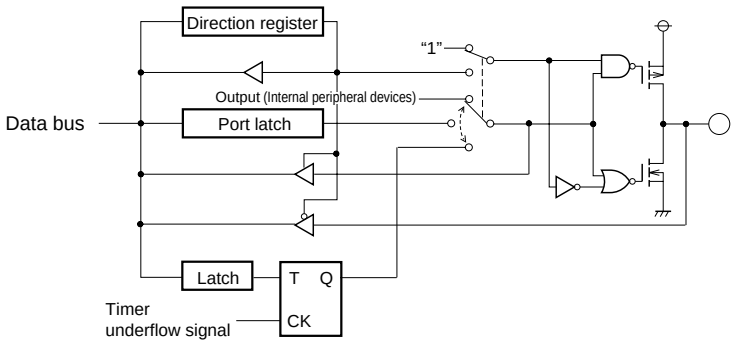
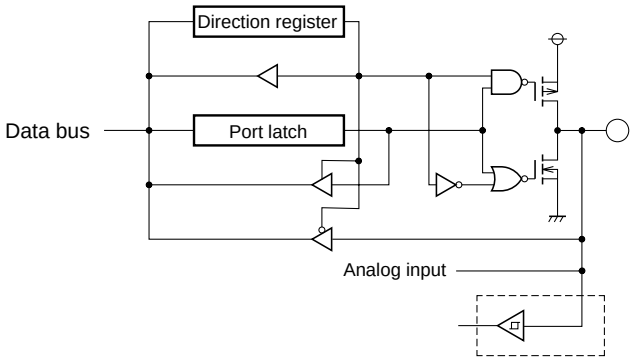


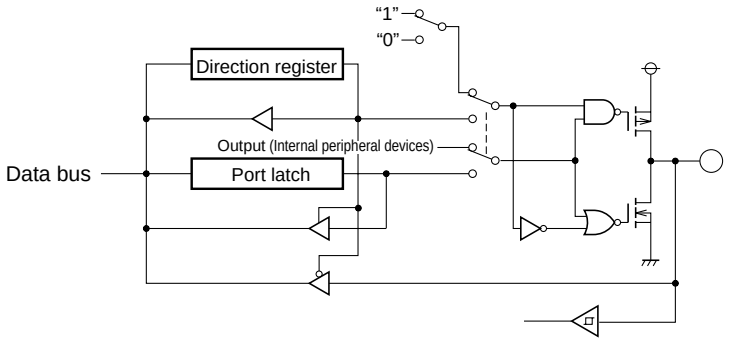
Fig. 107 Block diagram for each port pin (1)

[Inside dotted-line not included]
P7$_0$/AN$_0$, P7$_1$/AN$_1$

[Inside dotted-line included]
P7$_2$/AN$_2$/$\overline{\text{INT}_3}$, P7$_3$/AN$_3$/$\overline{\text{AD}_{\text{TRG}}}$/$\overline{\text{INT}_4}$



P8$_2$/$\overline{\text{CTS}_0}$/CLK$_1$, P8$_3$/$\overline{\text{CTS}_0}$/$\overline{\text{RTS}_0}$,
P8$_6$/CLK$_0$



P4$_2$/$\overline{\text{TC}}$

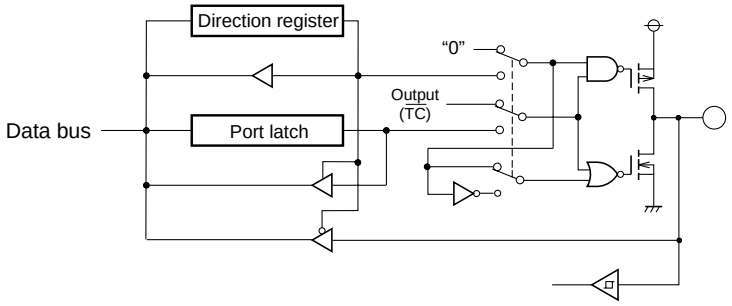


Fig. 108 Block diagram for each port pin (2)

CLOCK GENERATING CIRCUIT

In the clock generating circuit, the basic clock which is used to operate the CPU and each internal peripheral device is made by a clock input from pin XIN. Figure 111 shows the block diagram of the clock generating circuit. The clock which is input from the external clock input pin, XIN, generates the following;

- ϕ, which is used to operate the microcomputer
- ϕBIU, which is used to operate the BIU
- ϕCPU, which is used to operate the CPU

ϕ is stopped only when the external clock input is disabled by STP instruction. ϕBIU is stopped when to STP or WIT instruction is executed. Also, ϕCPU is stopped when STP or WIT instruction is executed or when a CPU wait request is issued by the BIU.

$f_{1(\phi)}$ is the basic clock for internal peripheral devices. This basic clock is divided furthermore in the divide circuit, as shown in Figure 111, and some frequency types are generated. Serial I/O communication and timer B can use any of four clocks (f_2, f_{16}, f_{64}, f_{512}), respectively. Timer A can use any of six clocks (f_2, f_{16}, f_{64}, f_{512}, and $f_{1(\phi)}$ and f_{4096}). "f_2" indicates that this clock is $f_{1(\phi)}$ divided by 2. For operation of the watchdog timer, refer to section on the watchdog timer.

When the STP instruction is executed, ϕ, ϕBIU, and ϕCPU stop at the "L" state.

The STP mode is terminated by acceptance of an interrupt, and the oscillation is started. Simultaneously, supply of ϕ is started.

When the watchdog timer starts to count down with Wf_{32} and the most significant bit of the watchdog timer is cleared to "0", supply of ϕBIU and ϕCPU is restarted. The count source of the watchdog timer is back to the count source which was selected before execution of the STP instruction, and the generated interrupt request is accepted.

When the WIT instruction is executed, ϕBIU and ϕCPU stops at the "L" state. However, ϕ is not stopped. Immediately after the interrupt request is accepted, ϕBIU and ϕCPU start their operations.

In order to terminate the STP or WIT mode by a non-maskable interrupt, it is necessary to make the interrupt request acceptable before execution of the STP or WIT instruction. For setting method, refer to the section on interrupts.

Figure 109 shows a circuit example with an external ceramic resonator or a quartz crystal oscillator. The constants such as capacitance etc. depend on a resonator. Therefore, for these constants, adopt the oscillator/resonator manufacturer's recommended values. Figure 110 shows a circuit e example with an external clock source.

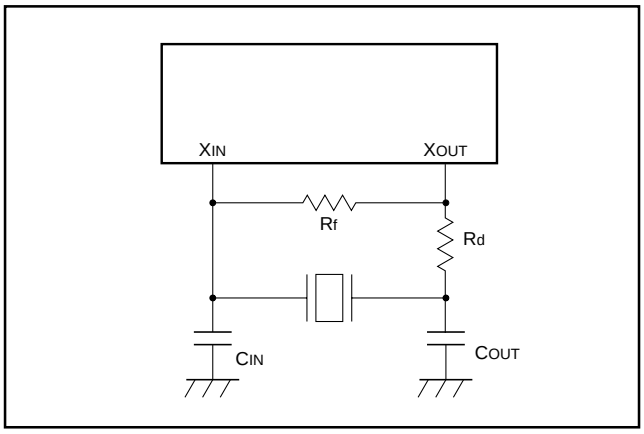


Fig. 109 Circuit example with external ceramic resonator or quartz crystal oscillator

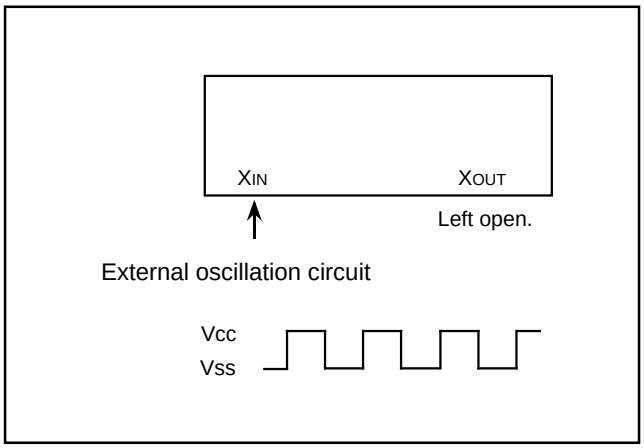


Fig. 110 Circuit example with external clock source

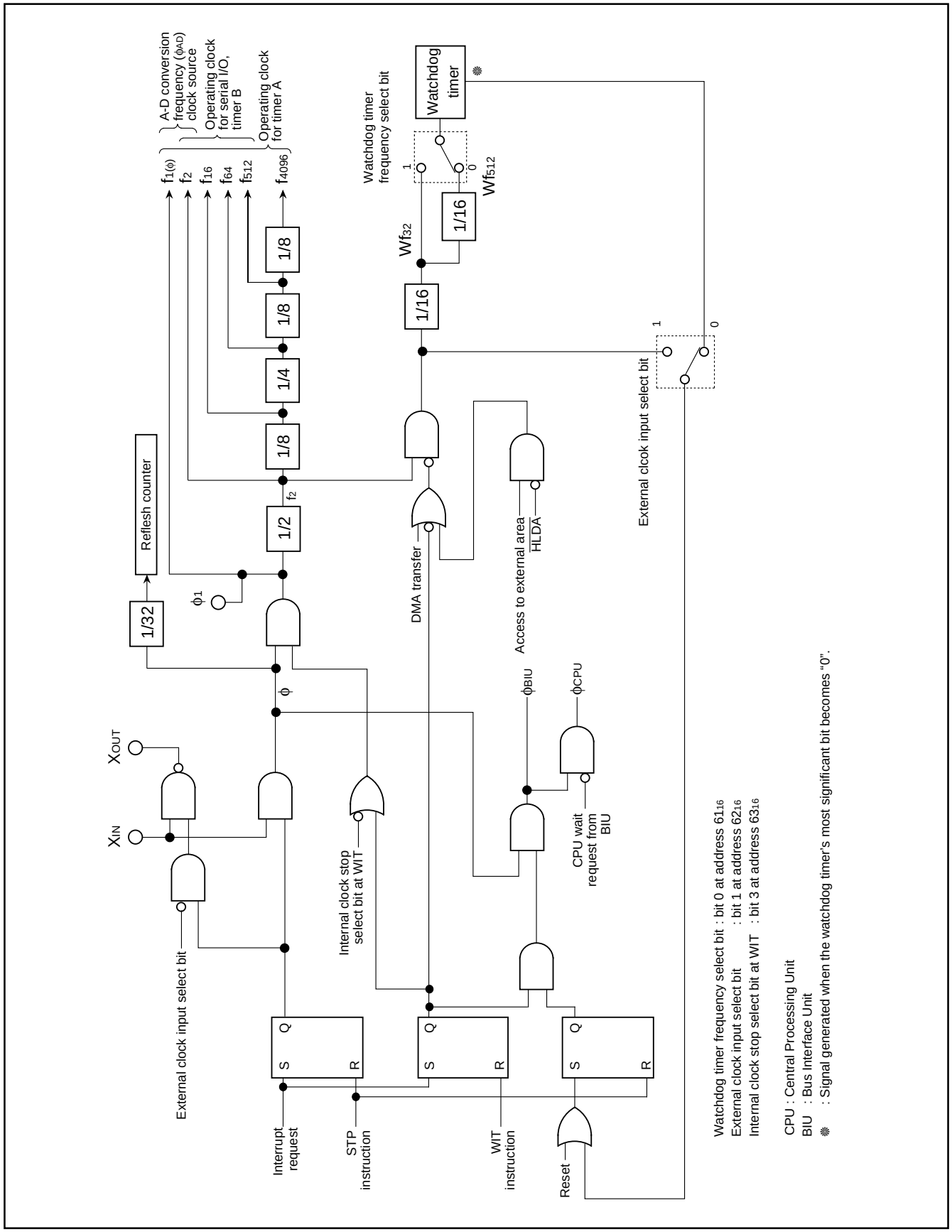


Fig. 111 Block diagram of clock generating circuit

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.

Figure 112 shows the block diagram of the debug function. Figures 113 and 114 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.

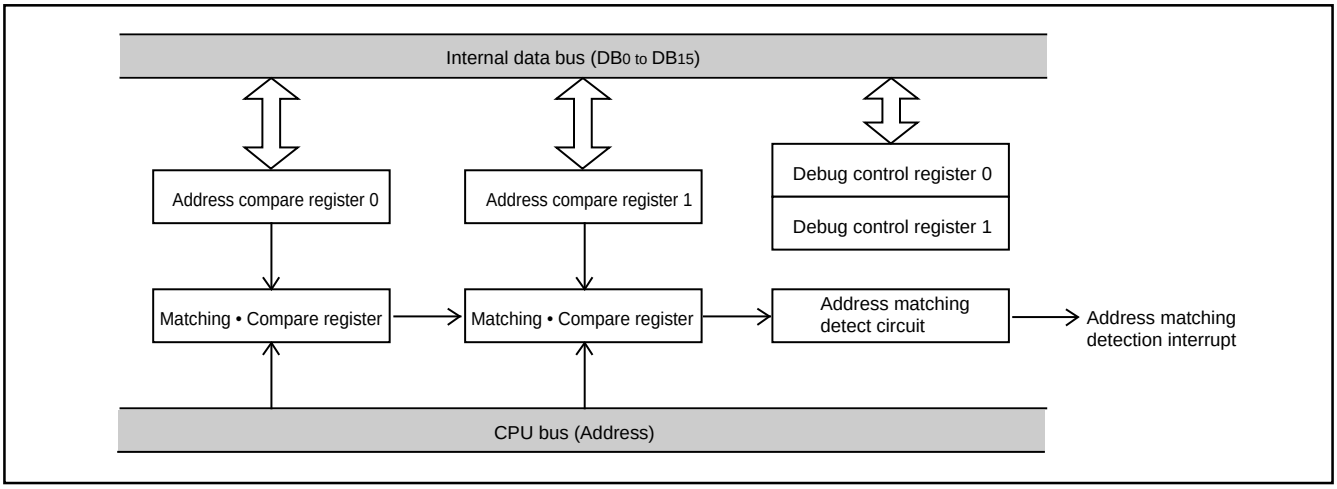
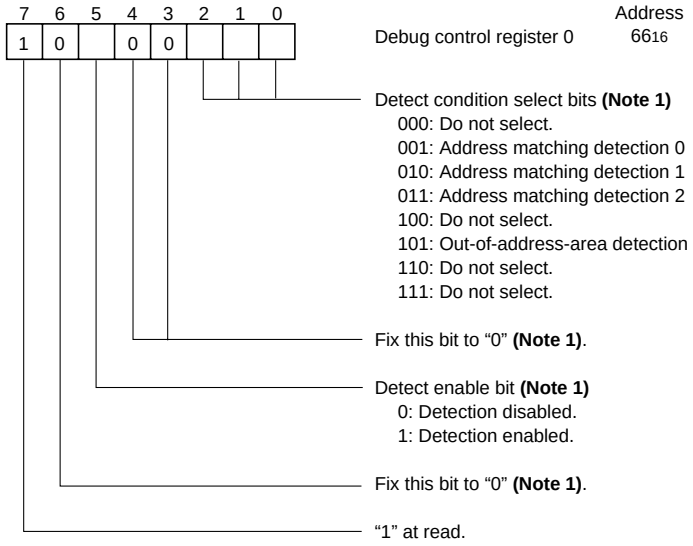
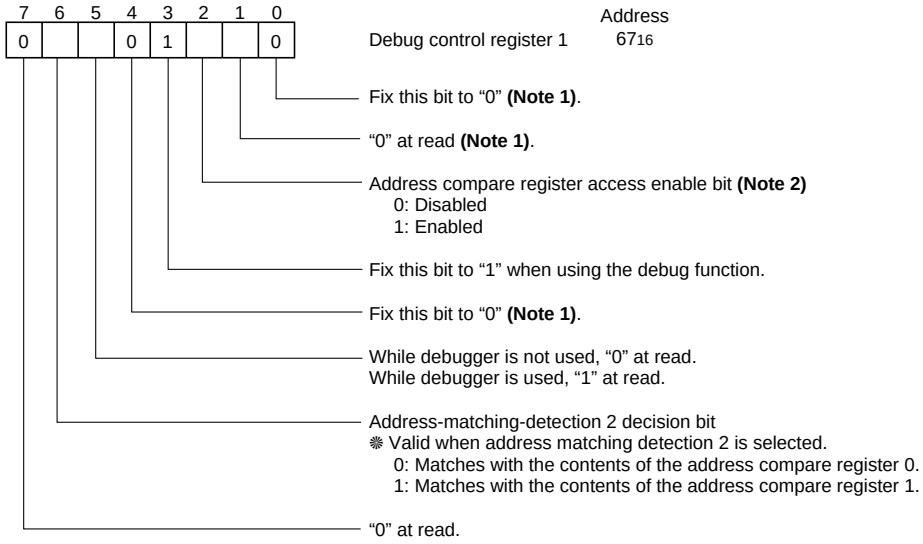


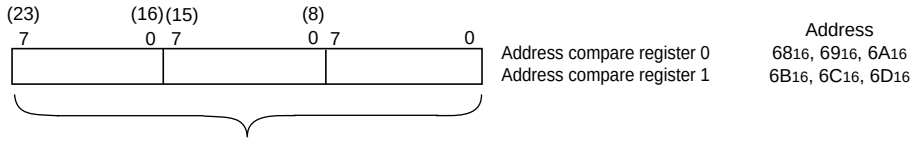
Fig. 112 Block diagram of debug function





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 6816 to 6D16) is accessed. And then, clear this bit to "0" with the instruction just after the access.

Fig. 113 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. 114 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 116 and 117, 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.

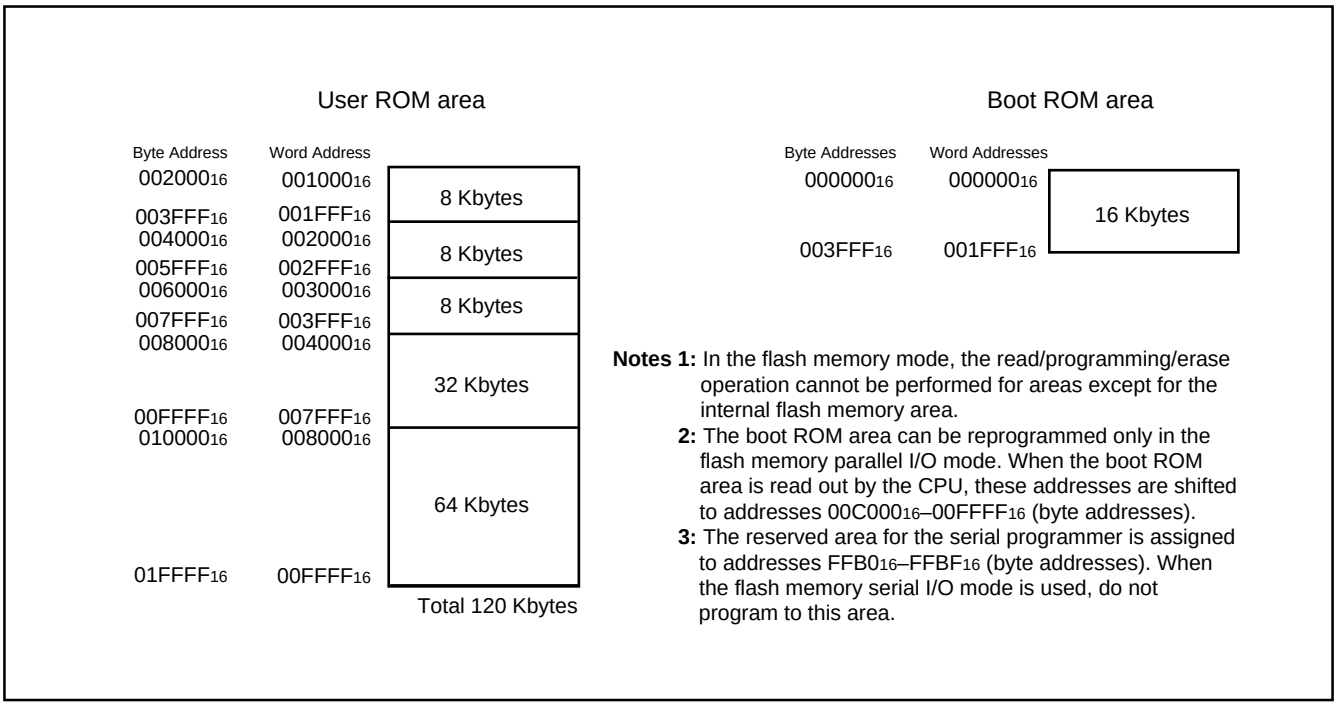


Fig. 116 M37920FCCGP, M37920FCCHP: block configuration of internal flash memory

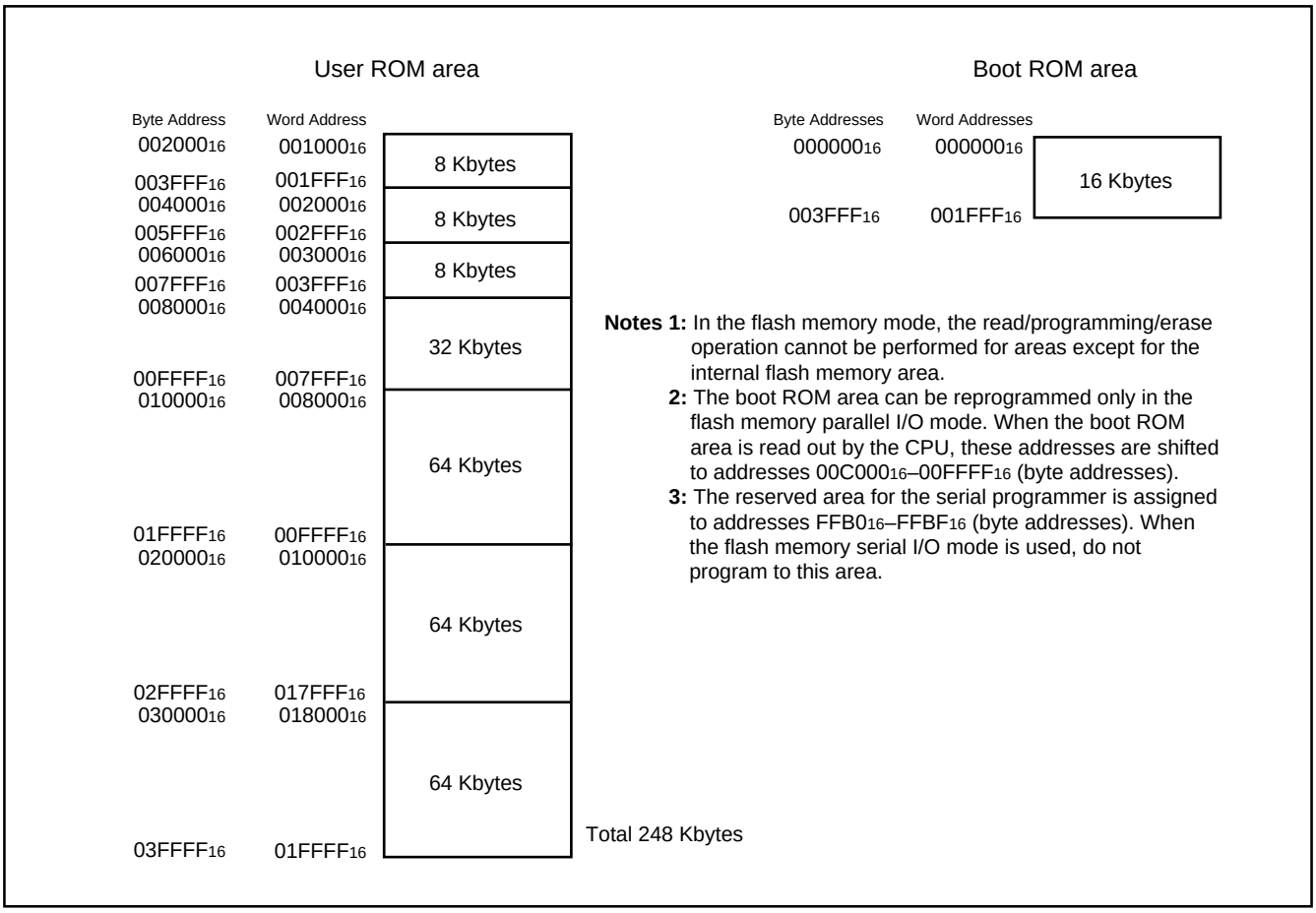


Fig. 117 M37920FGCGP, M37920FGCHP: block configuration of internal flash memory

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 25 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 25. 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).

PIN DESCRIPTION (FLASH MEMORY SERIAL I/O 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	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.
XIN	Clock input	Input	Connect a ceramic resonator between the XIN and XOUT pins, or input an external clock from the XIN pin with the XOUT pin left open.
XOUT	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.)
AVcc, AVss	Analog supply input	—	Connect AVcc to Vcc, and AVss to Vss.
VREF	Reference voltage input	Input	Input an arbitrary level within the range of VSS–VCC. (This is not used in the flash memory serial I/O mode.)
P00–P07	Input port P0	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P10–P17	Input port P1	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P20–P27	Input port P2	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P30–P33	Input port P3	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P40, P41	Input port P4	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P42	SDA I/O	I/O	This is an I/O pin for serial data. Connect this pin to VCC via a resistor (about 1 kΩ).
P43	BUSY output	Output	This is an output pin for the BUSY signal.
P44	SCLK input	Input	This is an input pin for a serial clock.
P50–P57	Input port P5	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P60–P66	Input port P6	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P70–P73	Input port P7	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P80–P86	Input port P8	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P90–P96	Input port P9	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P100–P107	Input port P10	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P110–P117	Input port P11	Input	Input “H” or “L”, or leave them open. (This is not used in the flash memory serial I/O mode.)
P120–P122	Input port P12	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.

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.

Figures 118 and 119 show the pin connections 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.

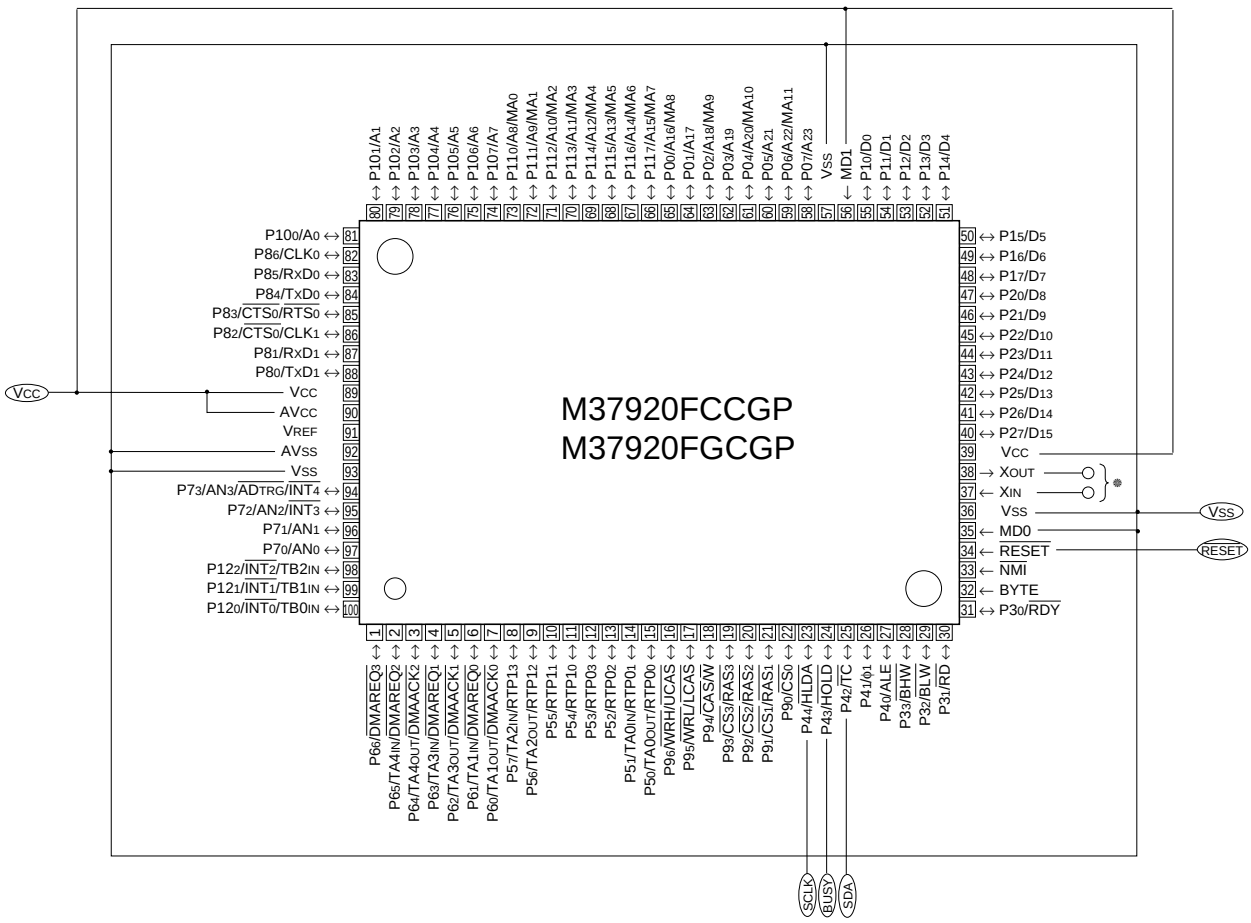
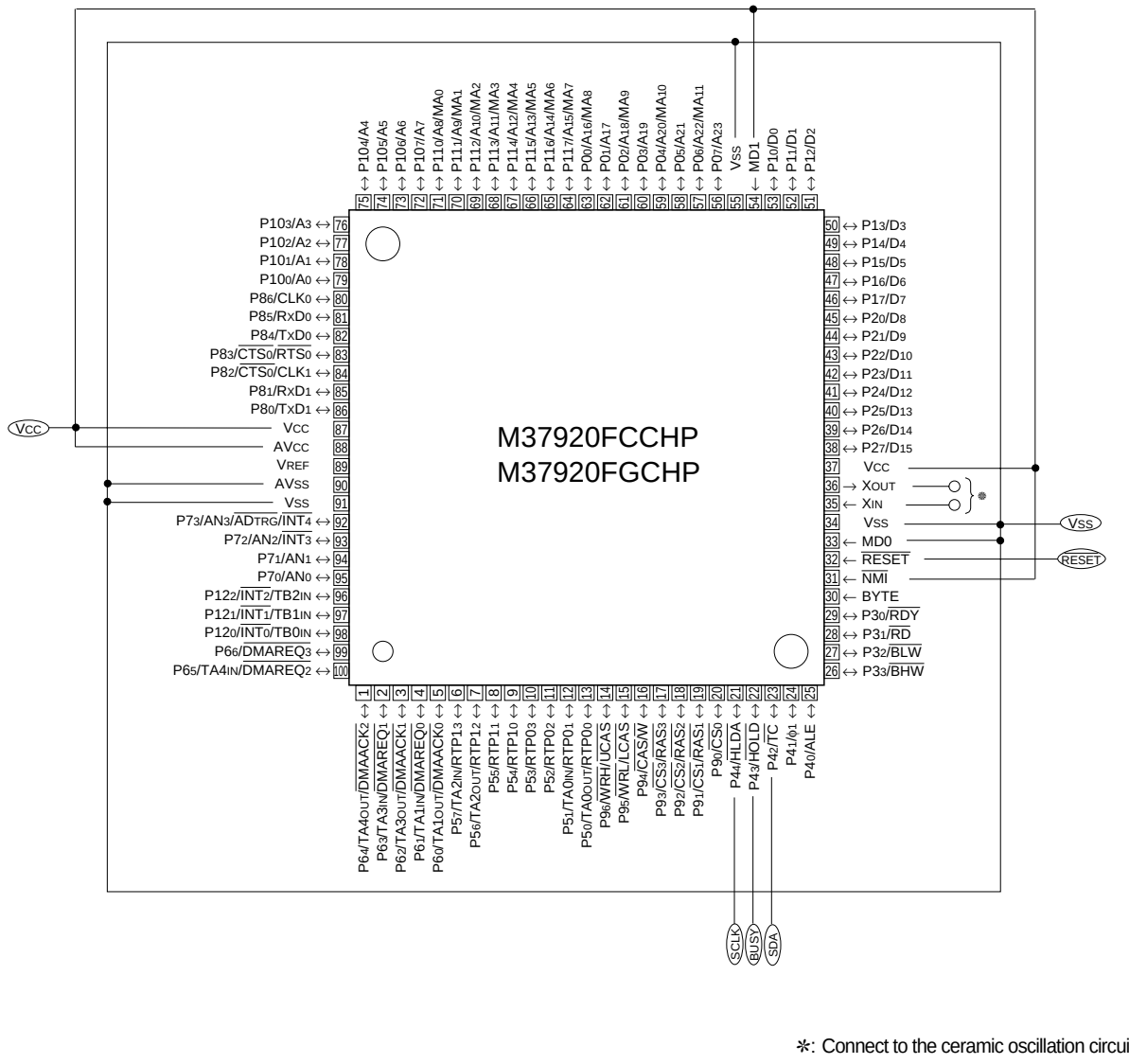


Fig.118 Pin connection of M37920FxCGP in flash memory serial I/O mode



Output 100P6Q-A

Fig.119 Pin connection of M37920FxCHP 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

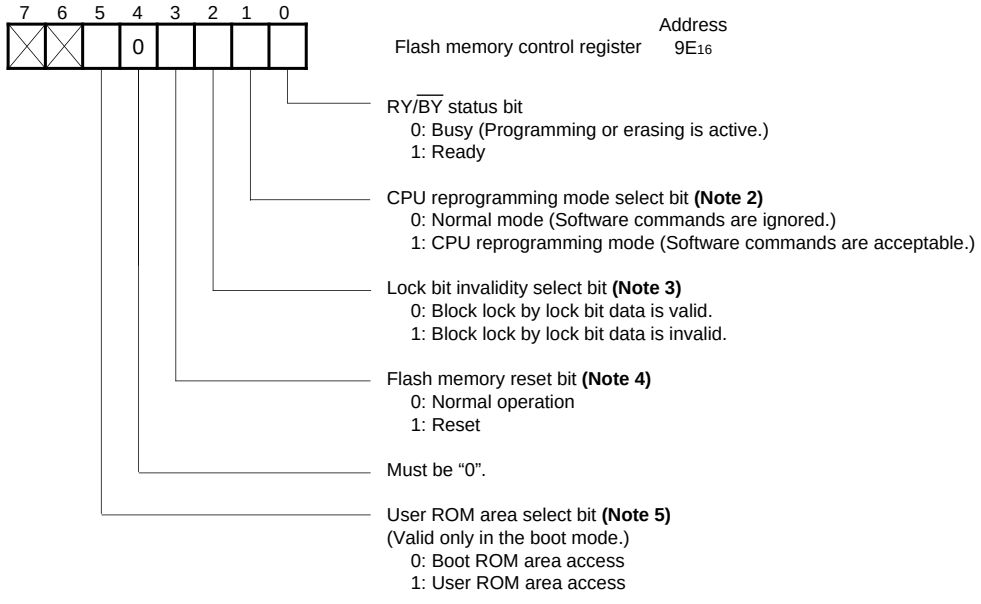
The user-original reprogramming control software for the CPU reprogramming mode must be stored into the user ROM area or the boot ROM area in the flash memory parallel I/O mode in advance. (If this program has been stored into the boot ROM area, the flash memory serial I/O mode will become unavailable).

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.



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 the CPU reprogramming mode select bit (bit 1) = “1”.
5: Writing to bit 5 must be performed by the user-original reprogramming control software in the internal RAM.

Fig. 120 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 120, 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 120 shows the bit configuration of the flash memory control register.

Bit 0 (the RY/$\overline{BY}$ 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”.

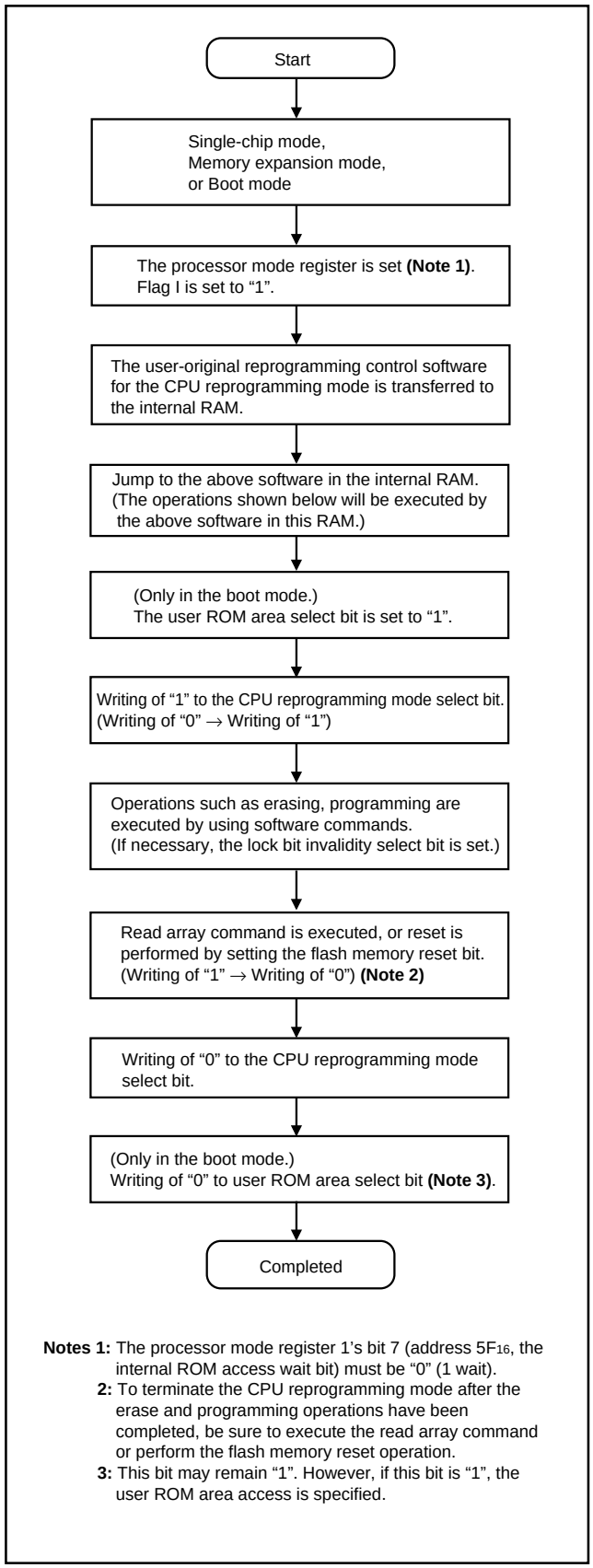
Bit 3 (the flash memory reset bit) resets the control circuit of the internal flash memory and is used when the CPU reprogramming mode is terminated or when an abnormal access to the flash memory happens. Writing of “1” to bit 3 with the CPU reprogramming mode select bit = “1” preforms the reset operation. 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 121 shows the CPU reprogramming mode set/termination flowchart, and be sure to follow this flowchart. As shown in Note 1 of Figure 121, 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 $\overline{NMI}$ interrupt or a watchdog timer interrupt request is generated in the CPU reprogramming mode, when an input to the $\overline{RESET}$ 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 $\overline{NMI}$ pin to “H” to avoid an $\overline{NMI}$ interrupt request occurrence. In the CPU reprogramming mode, be sure not to use the **STP** and **WIT** instructions.



Notes 1: The processor mode register 1's bit 7 (address $5F_{16}$, the internal ROM access wait bit) must be "0" (1 wait).
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. 121 CPU reprogramming mode set/termination flowchart

Software Commands

Table 26 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 26. 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/$\overline{BY}$ 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/$\overline{BY}$ 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 122 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/$\overline{BY}$ 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/$\overline{BY}$ 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 123 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.

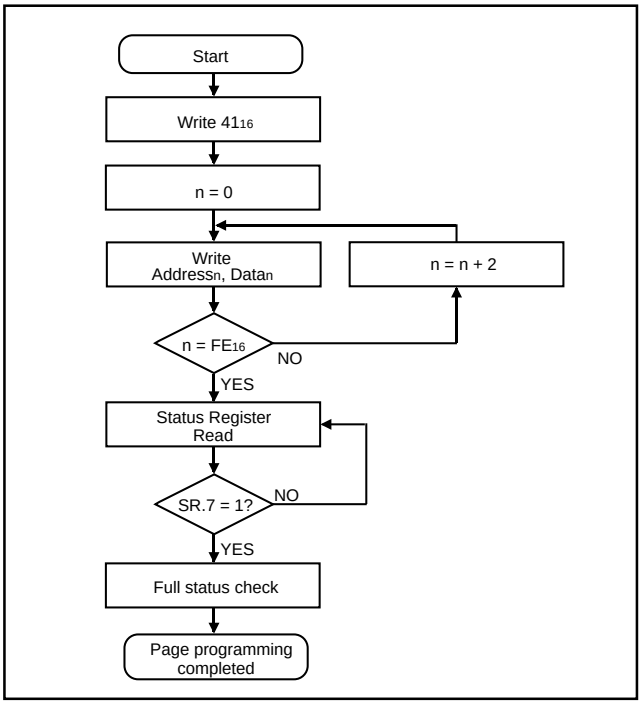


Fig.122 Page programming flowchart

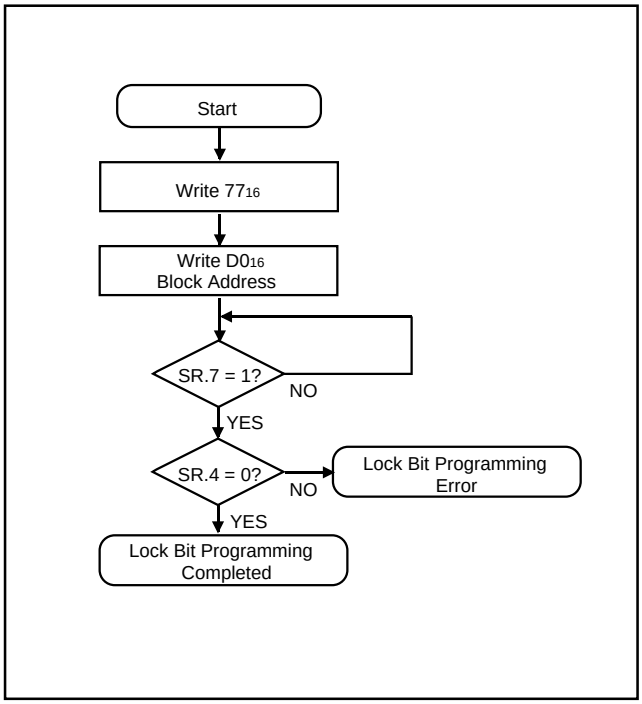


Fig.124 Lock bit programming flowchart

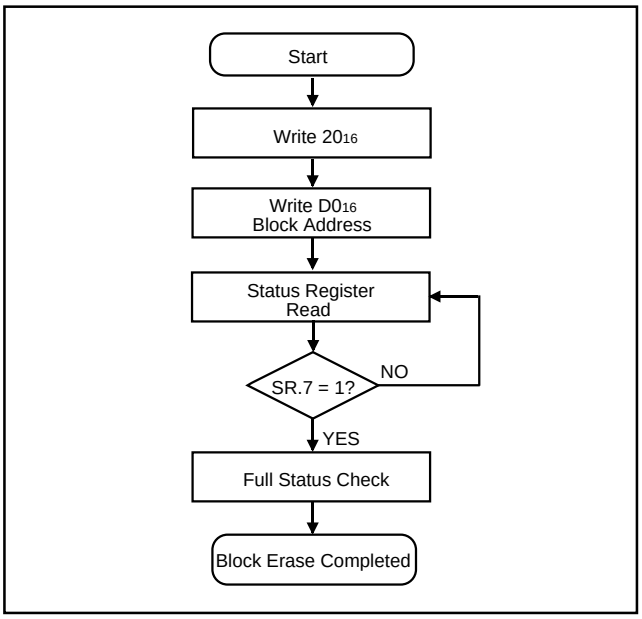


Fig.123 Block erase flowchart

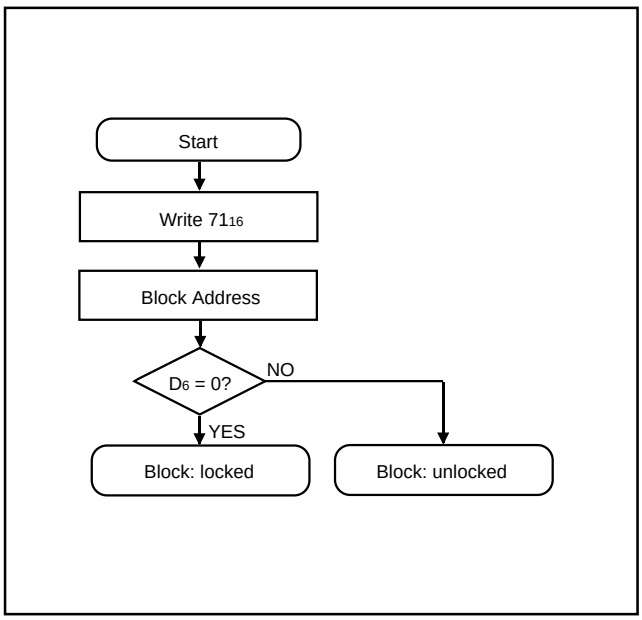


Fig.125 Read lock bit status flowchart

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/$\overline{BY}$ 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 124 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 125 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 27 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.

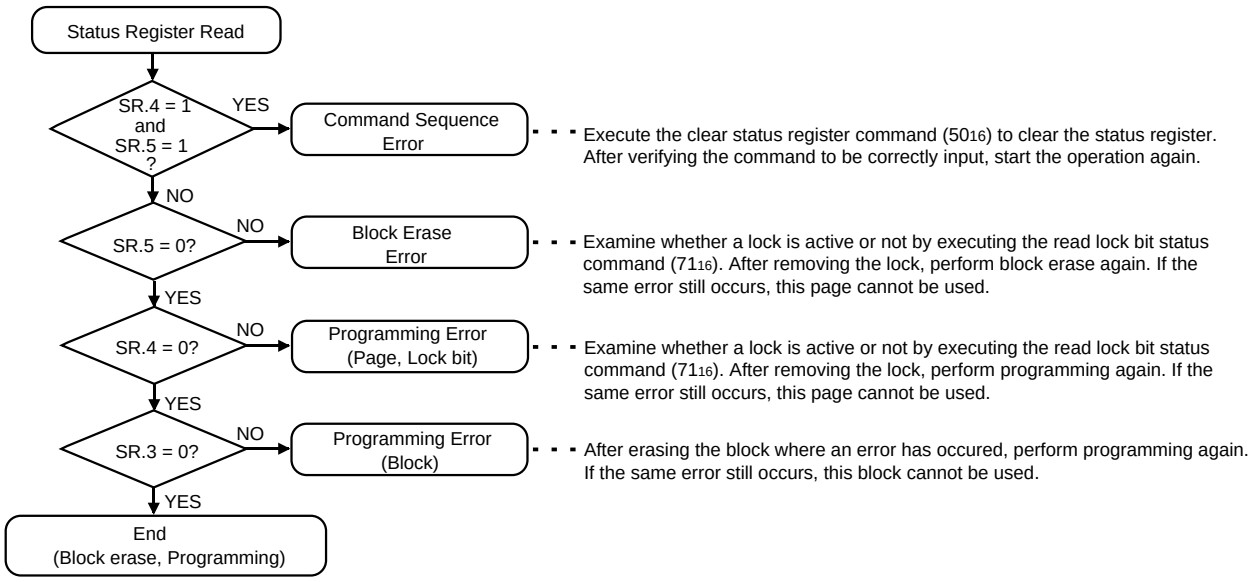
Full Status Check

The full status check reports the results of the erase or programming operation.

Figure 126 shows the full status check flowchart and actions to be taken if an error has occurred.

Table 27. 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	—	—



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.126 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(X_{IN}) = 20 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(X_{IN}) indicates the system clcok (X_{IN}) frequency.

AC Electrical Characteristics (V_{CC} = 5 V ± 0.5 V, Ta = 0 to 60 C°, f(X_{IN}) = 20 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(X_{IN}) indicates the system clock (X_{IN}) 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 D_0–D_7, $D_8/P2_0$–$D_{15}/P2_7$, $P3_0$, $P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_1$–$P9_6$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, $P12_0$–$P12_2$, V_{REF}, X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1, $\overline{NMI}$	–0.3 to V_{CC}+0.3	V
V_O	Output voltage D_0–D_7, $D_8/P2_0$–$D_{15}/P2_7$, $P3_0$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_1$–$P9_6$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, $P12_0$–$P12_2$, X_{OUT}	–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	300	mW
T_{opr}	Operating ambient temperature	–20 to 85	°C
T_{stg}	Storage temerature	–40 to 150	°C

RECOMMENDED OPERATING CONDITIONS (V_{CC} = 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	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	$P0_0$–$P0_7$, $P3_0$–$P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_1$–$P9_6$, $P10_1$–$P10_7$, $P11_1$–$P11_7$, $P12_0$–$P12_2$, X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1, $\overline{NMI}$	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (In single-chip mode)	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltag	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (In memory expansion and microprocessor modes)	0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage	$P0_0$–$P0_7$, $P3_0$–$P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_1$–$P9_6$, $P10_1$–$P10_7$, $P11_1$–$P11_7$, $P12_0$–$P12_2$, X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1, $\overline{NMI}$	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (In single-chip mode)	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage	$P1_0$–$P1_7$, $P2_0$–$P2_7$ (In memory expansion and microprocessor modes)	0		0.16V_{CC}	V
$I_{OH\ (peak)}$	High-level peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_0$–$P9_6$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, $P12_0$–$P12_2$			–10	mA
$I_{OH\ (avg)}$	High-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_0$–$P9_6$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, $P12_0$–$P12_2$			–5	mA
$I_{OL\ (peak)}$	Low-level peak output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_0$–$P9_6$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, $P12_0$–$P12_2$			10	mA
$I_{OL\ (avg)}$	Low-level average output current	$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_3$, $P4_0$–$P4_4$, $P5_0$–$P5_7$, $P6_0$–$P6_6$, $P7_0$–$P7_3$, $P8_0$–$P8_6$, $P9_0$–$P9_6$, $P10_0$–$P10_7$, $P11_0$–$P11_7$, $P12_0$–$P12_2$			5	mA
f(X_{IN})	External clock input frequency				20	MHz

Notes 1: Average output current is the average value of an interval of 100 ms.

2: 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, P9, and P12 must be 80 mA or less, the sum of $I_{OH(peak)}$ for ports P3–P7, P9, and P12 must be 80 mA or less.

DC ELECTRICAL CHARACTERISTICS (Vcc = 5 V, Vss = 0 V, Ta = –20 to 85 °C, f(XIN) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VOH	High-level output voltage P00–P07, P10–P17, P20–P27, P30, P40–P44, P50–P57, P60–P66, P70–P73, P80–P86, P90–P93, P100–P107, P110–P117, P120–P122	IOH = –10 mA	3			V
VOH	High-level output voltage P00–P07, P10–P17, P20–P27, P40, P44, P90–P93, P100–P107, P110–P117	IOH = –400 μA	4.7			V
VOH	High-level output voltage P31–P33, P94–P96	IOH = –10 mA	3.4			V
		IOH = –400 μA	4.8			
VOL	Low-level output voltage P00–P07, P10–P17, P20–P27, P30, P40–P44, P50–P57, P60–P66, P70–P73, P80–P86, P90–P93, P100–P107, P110–P117, P120–P122	IOL = 10 mA			2	V
VOL	Low-level output voltage P00–P07, P10–P17, P20–P27, P40, P44, P90–P93, P100–P107, P110–P117	IOL = 2 mA			0.45	V
VOL	Low-level output voltage P31–P33, P94–P96	IOL = 10 mA			1.6	V
		IOL = 2 mA			0.4	
VT+ —VT–	Hysteresis $\overline{TA0IN}$–$\overline{TA4IN}$, TB0IN–$\overline{TB2IN}$, $\overline{INT0}$–$\overline{INT4}$, $\overline{DMAREQ0}$–$\overline{DMAREQ3}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CLK0}$, $\overline{CLK1}$, RxD0, RxD1, $\overline{NMI}$, $\overline{RDY}$, $\overline{HOLD}$		0.4		1	V
VT+ —VT–	Hysteresis $\overline{RESET}$		0.5		1.5	V
VT+ —VT–	Hysteresis XIN		0.1		0.3	V
IIH	High-level input current P00–P07, P10–P17, P20–P27, P30–P33, P40–P44, P50–P57, P60–P66, P70–P73, P80–P86, P91–P96, P100–P107, P110–P117, P120–P122, XIN, $\overline{RESET}$, BYTE, MD0, MD1, $\overline{NMI}$	VI = 5.0 V			5	μA
IIL	Low-level input current P00–P07, P10–P17, P20–P27, P30–P33, P40–P44, P50–P57, P60–P66, P70–P73, P80–P86, P91–P96, P100–P107, P110–P117, P120–P122, XIN, $\overline{RESET}$, BYTE, MD0, MD1, $\overline{NMI}$	VI = 0 V			–5	μA
VRAM	RAM hold voltage	When clock is stoped.	2			V
ICC	Power source current	At reset in microprocessor mode, output-only pins are open, and the other pins are connected to Vss. f(XIN) = 20 MHz.		25	50	mA
		Ta = 25 °C when clcock is stopped.			1	μA
		Ta = 80 °C when clcock 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(X_{IN}) ≤ 20 MHz	10-bit resolution mode	5.9		*µ*s
			8-bit resolution mode	2.45 **(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}) = $f_{1(\phi)}$.

PERIPHERAL DEVICE INPUT/OUTPUT TIMING

(V_{CC} = 5 V ± 0.5 V, V_{CC} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz unless otherwise noted)

✻ For limits depending on f(X_{IN}), their calculation formulas are shown below. Also, the values at f(X_{IN}) = 20 MHz are shown in ( ).

Timer A input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi$_{IN}$ input cycle time	80		ns
$t_{w(TAH)}$	TAi$_{IN}$ input high-level pulse width	40		ns
$t_{w(TAL)}$	TAi$_{IN}$ input low-level pulse width	40		ns

Timer A input (Gating input in timer mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TAi$_{IN}$ input cycle time	f(X_{IN}) ≤ 20 MHz	$\frac{16 \times 10^9}{f(X_{IN})}$ (800)		ns
$t_{w(TAH)}$	TAi$_{IN}$ input high-level pulse width	f(X_{IN}) ≤ 20 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns
$t_{w(TAL)}$	TAi$_{IN}$ input low-level pulse width	f(X_{IN}) ≤ 20 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns

Note : The TAi$_{IN}$ input cycle time requires 4 or more cycles of a count source. The TAi$_{IN}$ input high-level pulse width and the TAi$_{IN}$ 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(X_{IN}) ≤ 20 MHz.

Timer A input (External trigger input in one-shot pulse mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TA)}$	TAi$_{IN}$ input cycle time	f(X_{IN}) ≤ 20 MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns
$t_{w(TAH)}$	TAi$_{IN}$ input high-level pulse width		80		ns
$t_{w(TAL)}$	TAi$_{IN}$ input low-level pulse width		80		ns

Timer A input (External trigger input in pulse width modulation mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAi$_{IN}$ input high-level pulse width	80		ns
$t_{w(TAL)}$	TAi$_{IN}$ input low-level pulse width	80		ns

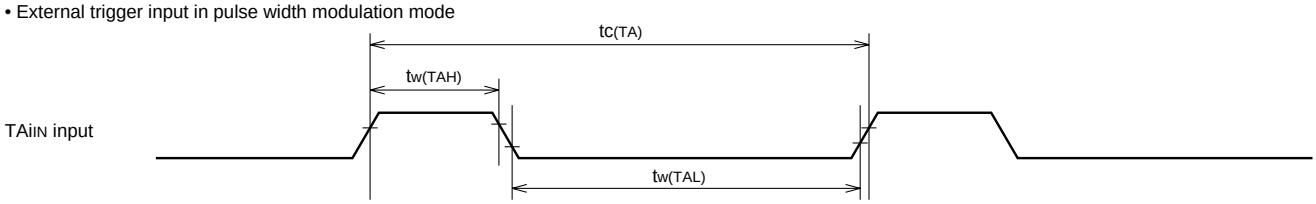
Timer A input (Up-down input and Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(UP)}$	TAi$_{OUT}$ input cycle time	2000		ns
$t_{w(UPH)}$	TAi$_{OUT}$ input high-level pulse width	1000		ns
$t_{w(UPL)}$	TAi$_{OUT}$ input low-level pulse width	1000		ns
$t_{su(UP\text{-}T_{IN})}$	TAi$_{OUT}$ input setup time	400		ns
$t_{h(T_{IN}\text{-}UP)}$	TAi$_{OUT}$ input hold time	400		ns

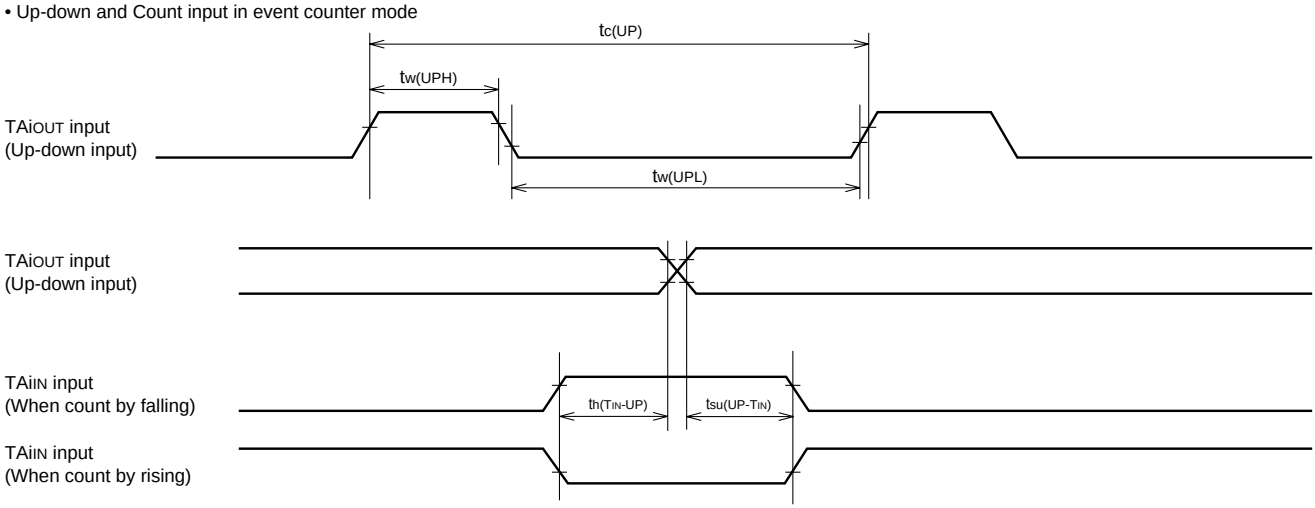
Timer A input (Two-phase pulse input in event counter mode)

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

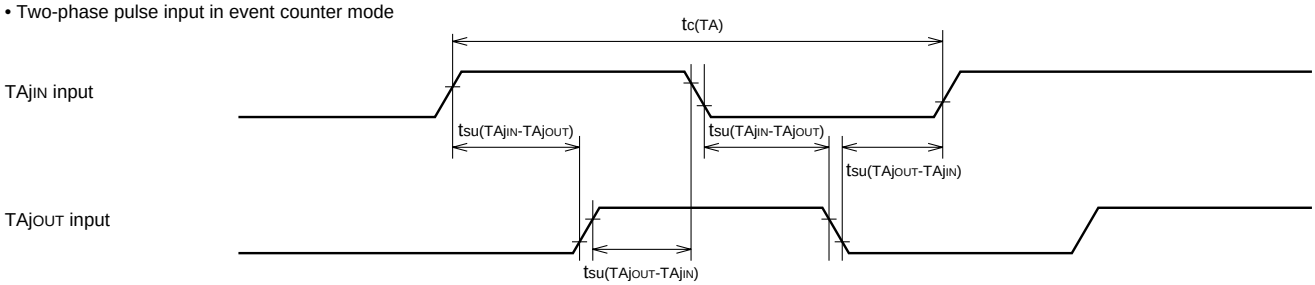
- Gating input in timer mode
- Count input in event counter mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode



- 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} = 1.0 V, V_{IH} = 4.0 V

Timer B input (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (one edge count)	80		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (one edge count)	40		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (one edge count)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (both edge count)	160		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width (both edge count)	80		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width (both edge count)	80		ns

Timer B input (Pulse period measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	$f(X_{IN}) \le 20$ MHz	$\frac{16 \times 10^9}{f(X_{IN})}$ (800)		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width	$f(X_{IN}) \le 20$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width	$f(X_{IN}) \le 20$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns

Note: The TBiIN input cycle time requires 4 or more cycles of a count source. The TBiIN input high-level pulse width and the TBiIN 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 $f(X_{IN}) \le 20$ MHz.

Timer B input (Pulse width measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	$f(X_{IN}) \le 20$ MHz	$\frac{16 \times 10^9}{f(X_{IN})}$ (800)		ns
$t_{w(TBH)}$	TBiIN input high-level pulse width	$f(X_{IN}) \le 20$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns
$t_{w(TBL)}$	TBiIN input low-level pulse width	$f(X_{IN}) \le 20$ MHz	$\frac{8 \times 10^9}{f(X_{IN})}$ (400)		ns

Note: The TBiIN input cycle time requires 4 or more cycles of a count source. The TBiIN input high-level pulse width and the TBiIN 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 $f(X_{IN}) \le 20$ MHz.

A-D trigger input

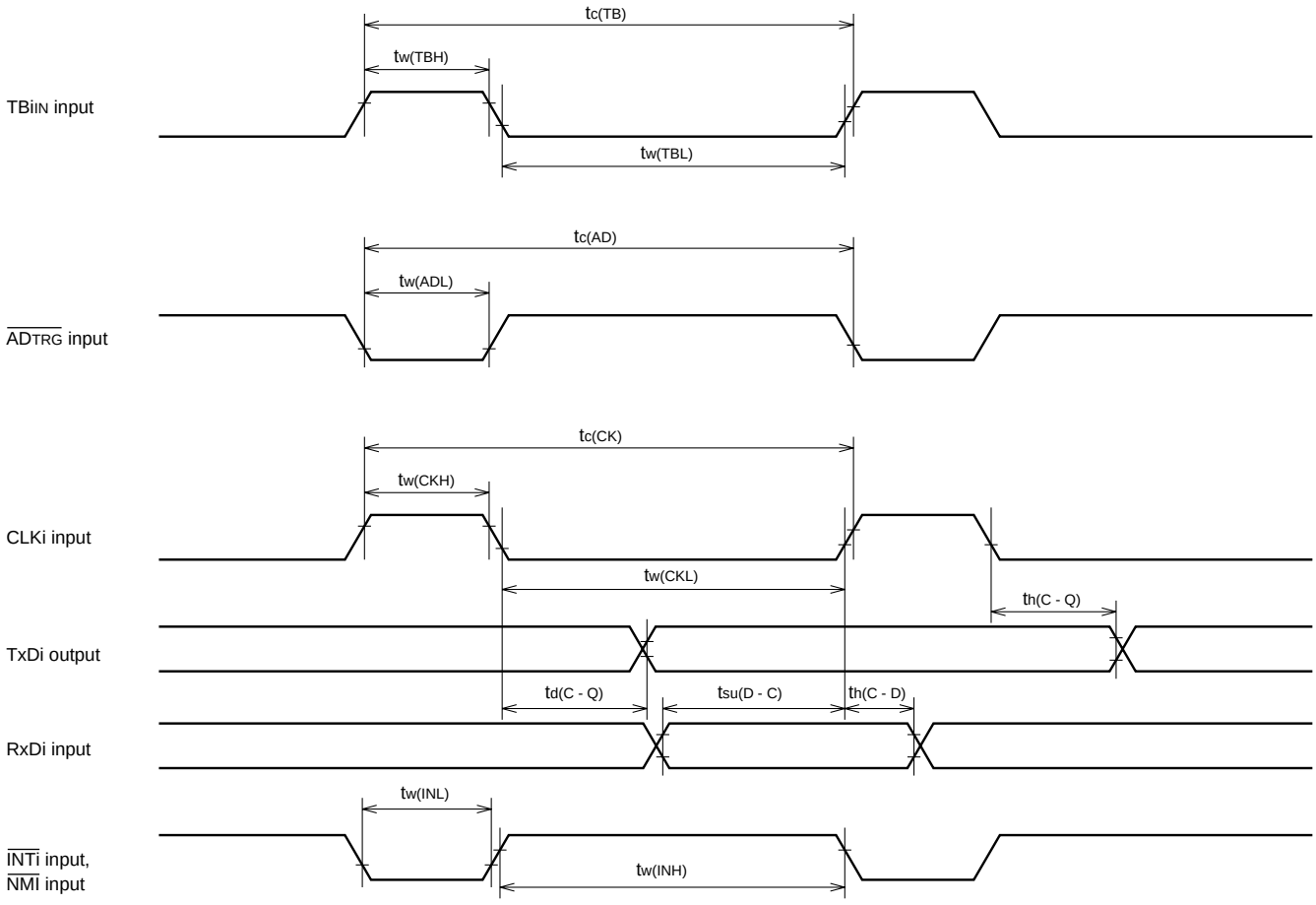
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (minimum allowable trigger)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input low-level pulse width	125		ns

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input high-level pulse width	100		ns
$t_{w(CKL)}$	CLKi input low-level pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	20		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

External interrupt ($\overline{INT}_i$) input, $\overline{NMI}$ input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INT}_i$ input/$\overline{NMI}$ input high-level pulse width	250		ns
$t_{w(INL)}$	$\overline{INT}_i$ input/$\overline{NMI}$ 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} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 50 pF

READY, HOLD TIMING

Timing requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

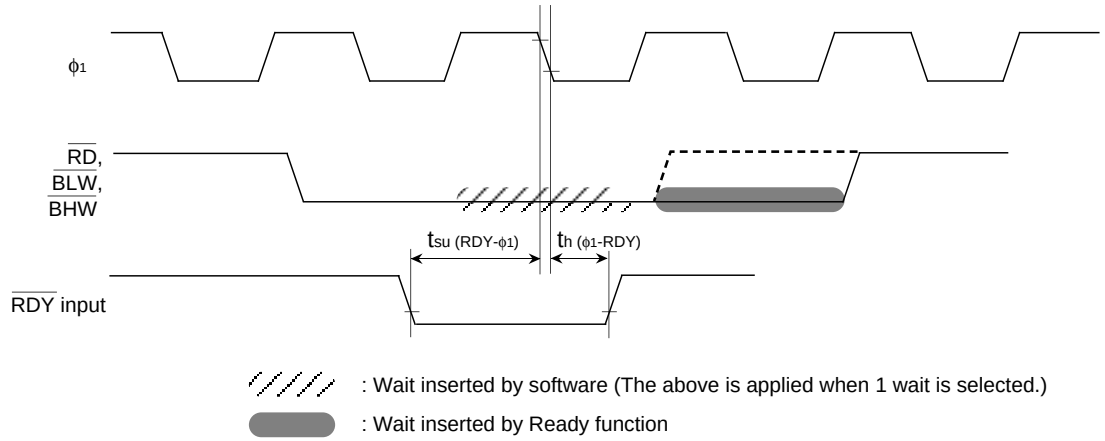
Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(RDY-\phi1)}$	$\overline{RDY}$ input setup time	40		ns
$t_{su(HOLD-\phi1)}$	$\overline{HOLD}$ input setup time	40		ns
$t_{h(\phi1-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(\phi1-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns

Switching characteristics (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{d(\phi1-HLDAL)}$	$\overline{HLDA}$ output delay time		20	ns
$t_{d(RDH-HLDAL)}$	$\overline{HLDA}$ low-level output delay time after read	tc –15 **(Note)**		ns
$t_{d(BXWH-HLDAL)}$	$\overline{HLDA}$ low-level output delay time after write	tc –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: tc = 1/f(X_{IN}).

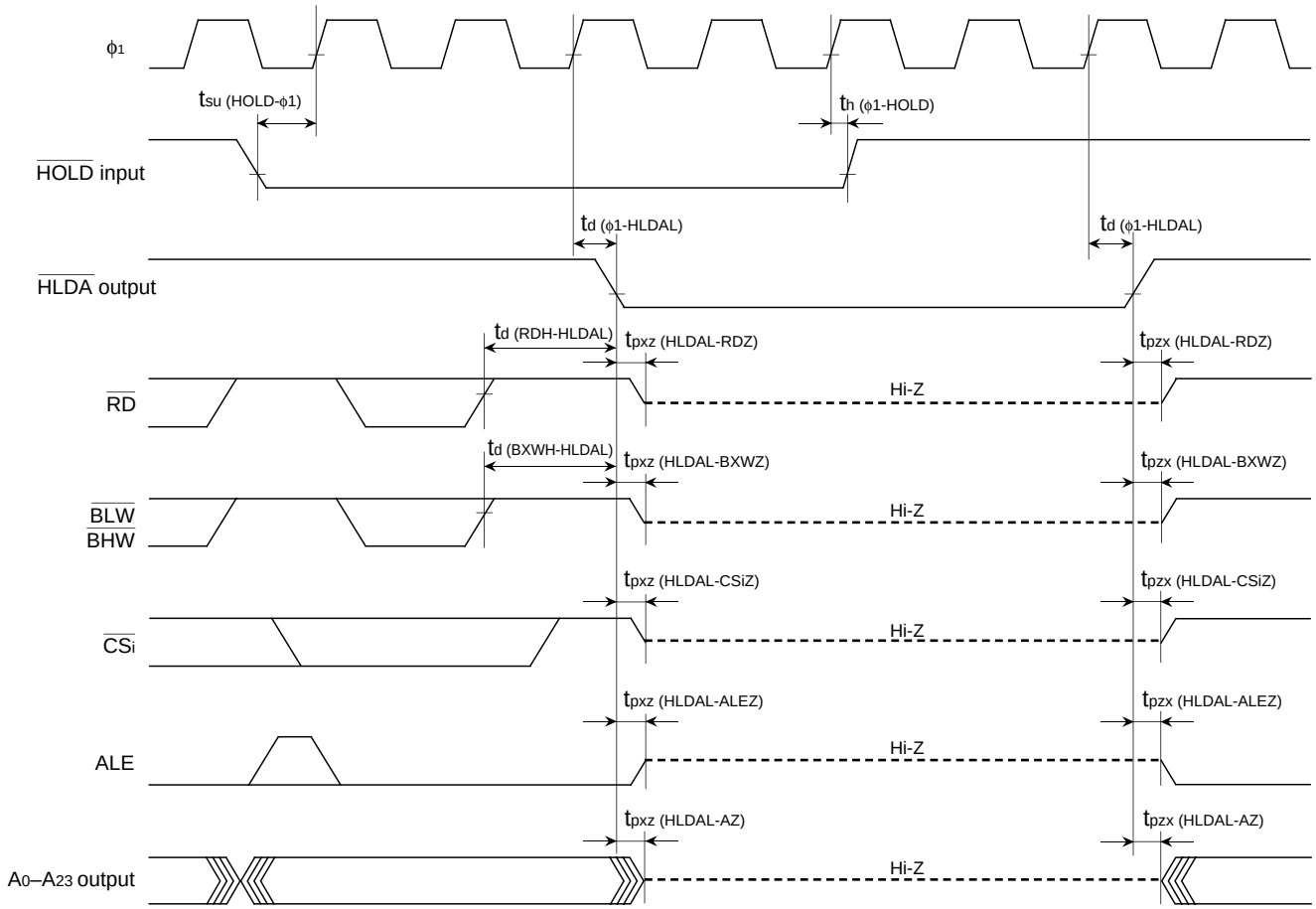
$\overline{\text{RDY}}$ input



: Wait inserted by software (The above is applied when 1 wait is selected.)

: 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} = 1.0 V, V_{IH} = 4.0 V
- $\overline{\text{HLDA}}$ output : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 50 pF

External bus timing

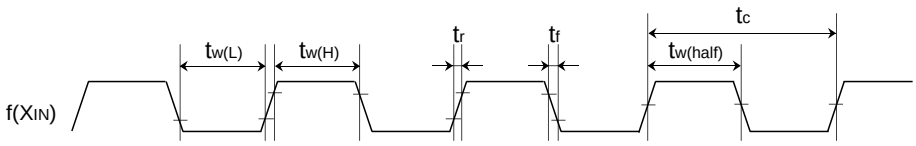
For limits depending on f(X_{IN}), their calculation formulas are shown below.

W = 0 (0 wait)
W = 1 (1 wait)
W = 2 (2 wait)
tc = 1/f(X_{IN}).

Timing Requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits				Unit
		When 0/1/2 wait is selected		When ALE expansion wait is selected		
		Min.	Max.	Min.	Max.	
t_c	External clock input cycle time	50		50		ns
$t_{w(half)}$	External clock input pulse width with half input-volage	0.45 tc	0.55tc	0.45tc	0.55tc	ns
$t_{w(H)}$	External clock input high-level pulse width	0.5tc-8		0.5tc-8		ns
$t_{w(L)}$	External clock input low-level pulse width	0.5tc-8		0.5tc-8		ns
t_r	External clock input rise time		8		8	ns
t_f	External clock input fall time		8		8	ns
$t_{a(A-D)}$	Address access time		(2 + W)tc-45		4tc-45	ns
$t_{a(CSiL-D)}$	Chip select access time		(1.5 + W)tc-35		3.5tc–35	ns
$t_{a(RDL-D)}$	Read access time		(1 + W)tc-30		2tc-30	ns
$t_{su(D-RDL)}$	Read data setup time	15		15		ns
$t_{h(RDH-D)}$	Data input hold time after read	0		0		ns
$t_{a(BA-D)}$	Address access time at burst ROM access		(1 + W)tc-35		2tc-35	ns
$t_{h(BA-D)}$	Data hold time after address at burst ROM access	0		0		ns

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, $t_{w(half)}$)

Switching characteristics (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

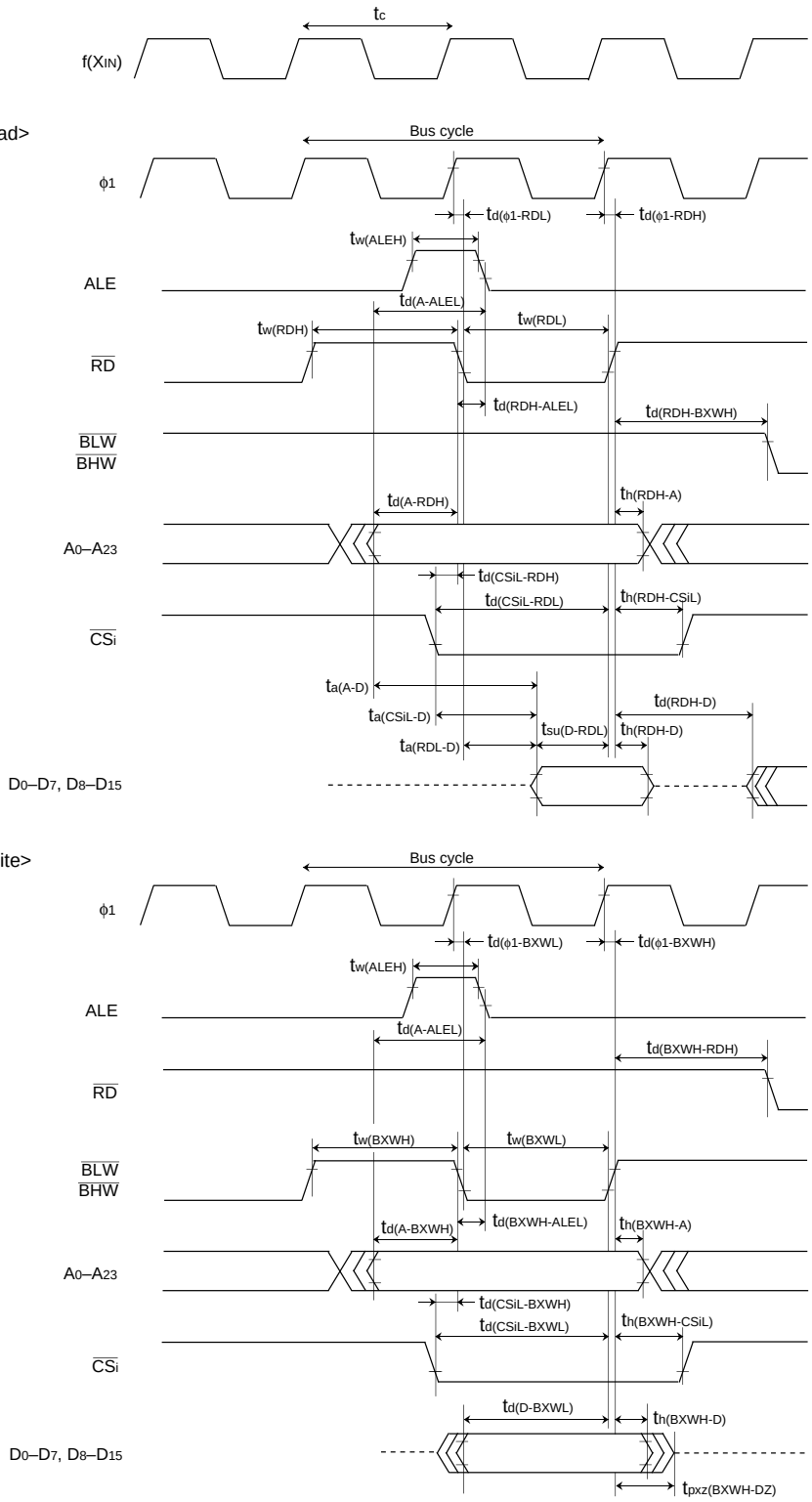
Symbol	Parameter	Limits				Unit
		When 0/1/2 wait is selected		When ALE expansion wait is selected		
		Min.	Max.	Min.	Max.	
td(φ1-RDL)	Read low-level output delay time	–10	15	–10	15	ns
td(φ1-RDH)	Read high-level output delay time	–10	10	–10	10	ns
td(φ1-BXWL)	Write low-level output delay time	–10	15	–10	15	ns
td(φ1-BXWH)	Write high-level output delay time	–10	10	–10	10	ns
tw(ALEH)	ALE pulse width	0.5tc-20		tc-20		ns
td(A-ALEL)	ALE completion delay time after address stabilization	tc-30		1.5tc-30		ns
tw(RDL)	Read output pulse width	(1+W)tc-15		2tc-15		ns
tw(RDH)	Read output high-level width **(Note 1)**	tc-15		2tc-15		ns
td(RDH-BXWH)	Write disable valid time after read **(Note 2)**	tc-15		tc-15		ns
td(A-RDH)	Address valid time before read	tc-30		2tc-30		ns
th(RDH-A)	Address hold time after read **(Note 3)**	8		8		ns
td(RDH-ALEL)	ALE completion delay time after read start		20			ns
td(ALEL-RDH)	Read disable valid time after ALE completion			0.5tc-20		ns
td(CSiL-RDH)	Chip select valid time before read	0.5tc-20		1.5tc-20		ns
td(CSiL-RDL)	Chip select output valid time before read completion	(1.5 + W)tc-20		3.5tc-20		ns
th(RDH-CSiL)	Chip select hold time after read	0.5tc-20		0.5tc-20		ns
td(RDH-D)	Next write cycle data output delay time after read **(Note 2)**	tc-15		tc-15		ns
tw(BXWL)	Write output pulse width	(1 + W)tc-15		2tc-15		ns
tw(BXWH)	Write output high-level width **(Note 1)**	tc-15		2tc-15		ns
td(BXWH-RDH)	Read disable valid time after write **(Note 2)**	tc-15		tc-15		ns
td(A-BXWH)	Address valid time before write	tc-30		2tc-30		ns
th(BXWH-A)	Address hold time after write **(Note 3)**	8		8		ns
td(BXWH-ALEL)	ALE completion delay time after write start		20			ns
td(ALEL-BXWH)	Write disable valid time after ALE completion			0.5tc-20		ns
td(CSiL-BXWH)	Chip select valid time before write	0.5tc-20		1.5tc-20		ns
td(CSiL-BXWL)	Chip select output valid time before write completion	(1.5 + W)tc-20		3.5tc-20		ns
th(BXWH-CSiL)	Chip select hold time after write	0.5tc-20		0.5tc-20		ns
td(D-BXWL)	Data output valid time before write completion	(1 + W)tc-20		2tc-20		ns
th(BXWH-D)	Data hold time after write	0.5tc-10		0.5tc-10		ns
tpxz(BXWH-DZ)	Floating start delay time after write		0.5tc+10		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).

2: When accessing the area where the recovery cycle insertion is selected, this parameter is extended by tc (ns).

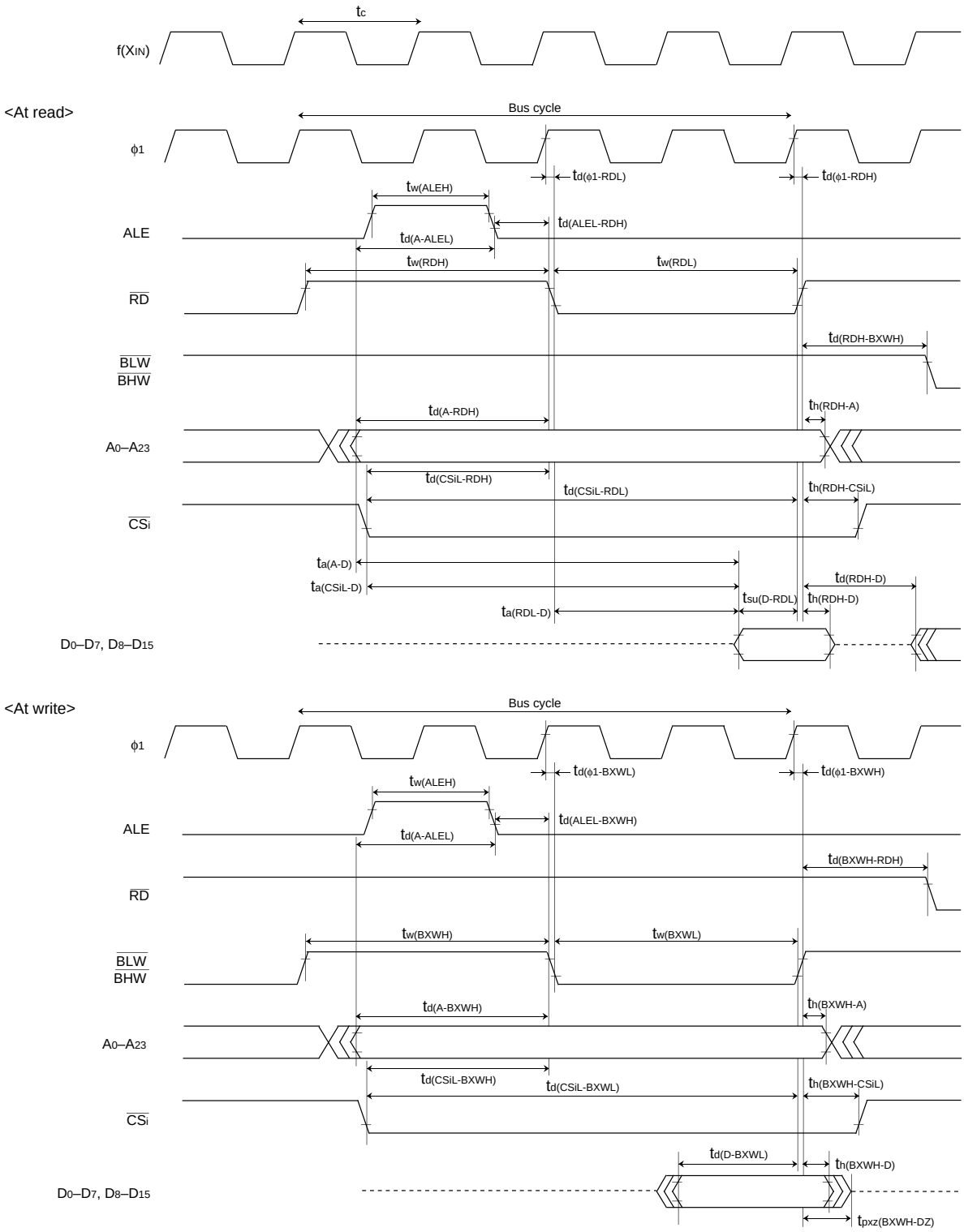
3: When accessing the area where the recovery cycle insertion is selected, this parameter is extended by tc (ns). However, except for the case at instruction prefetch.

Normal access : 0/1/2 wait



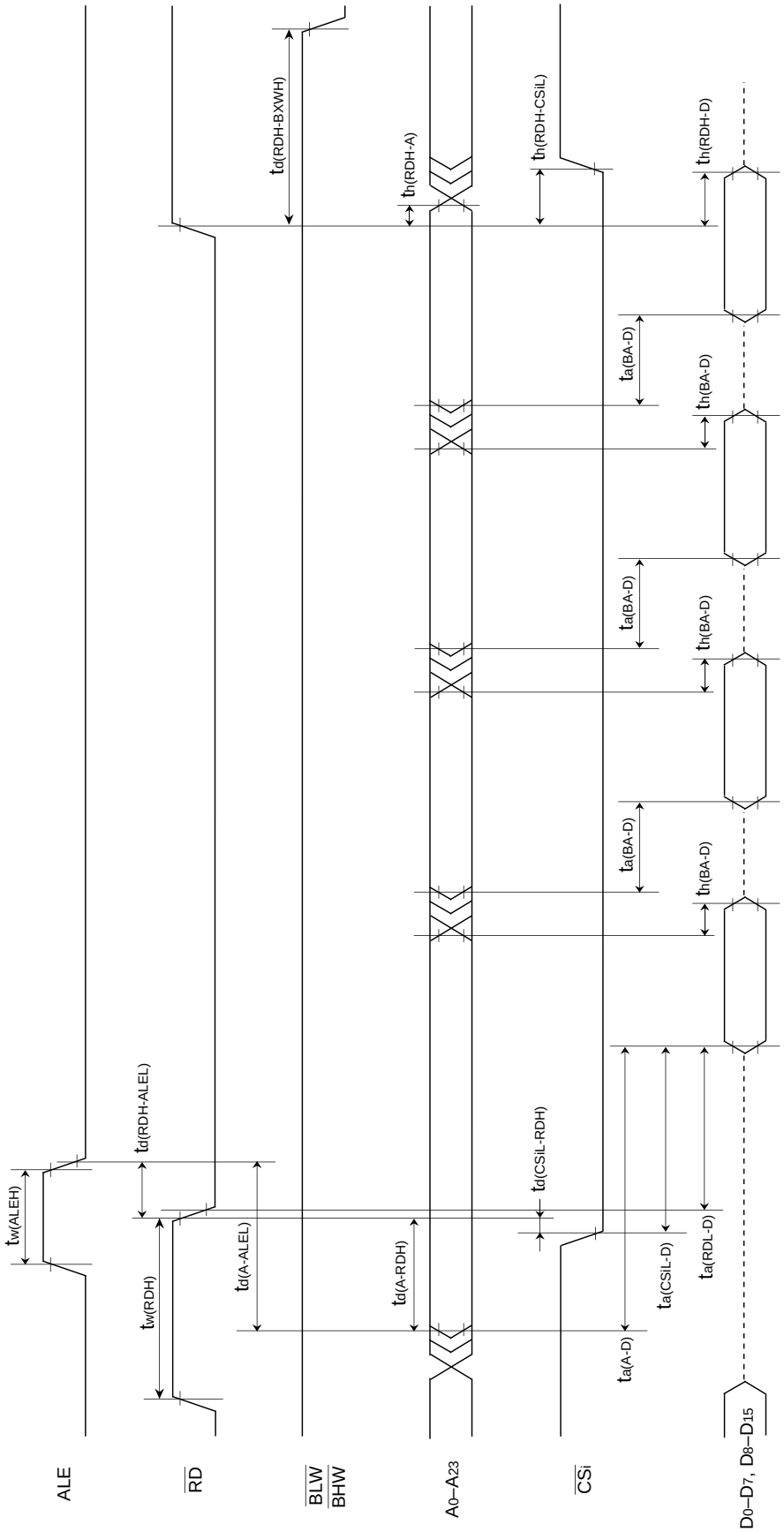
Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.5 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{CS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{CS_i}$)

Normal access : ALE expansion wait



Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.5 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 : 0/1/2 wait at instruction prefetch



Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.5 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{CSi}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{CSi}$)

DRAM access

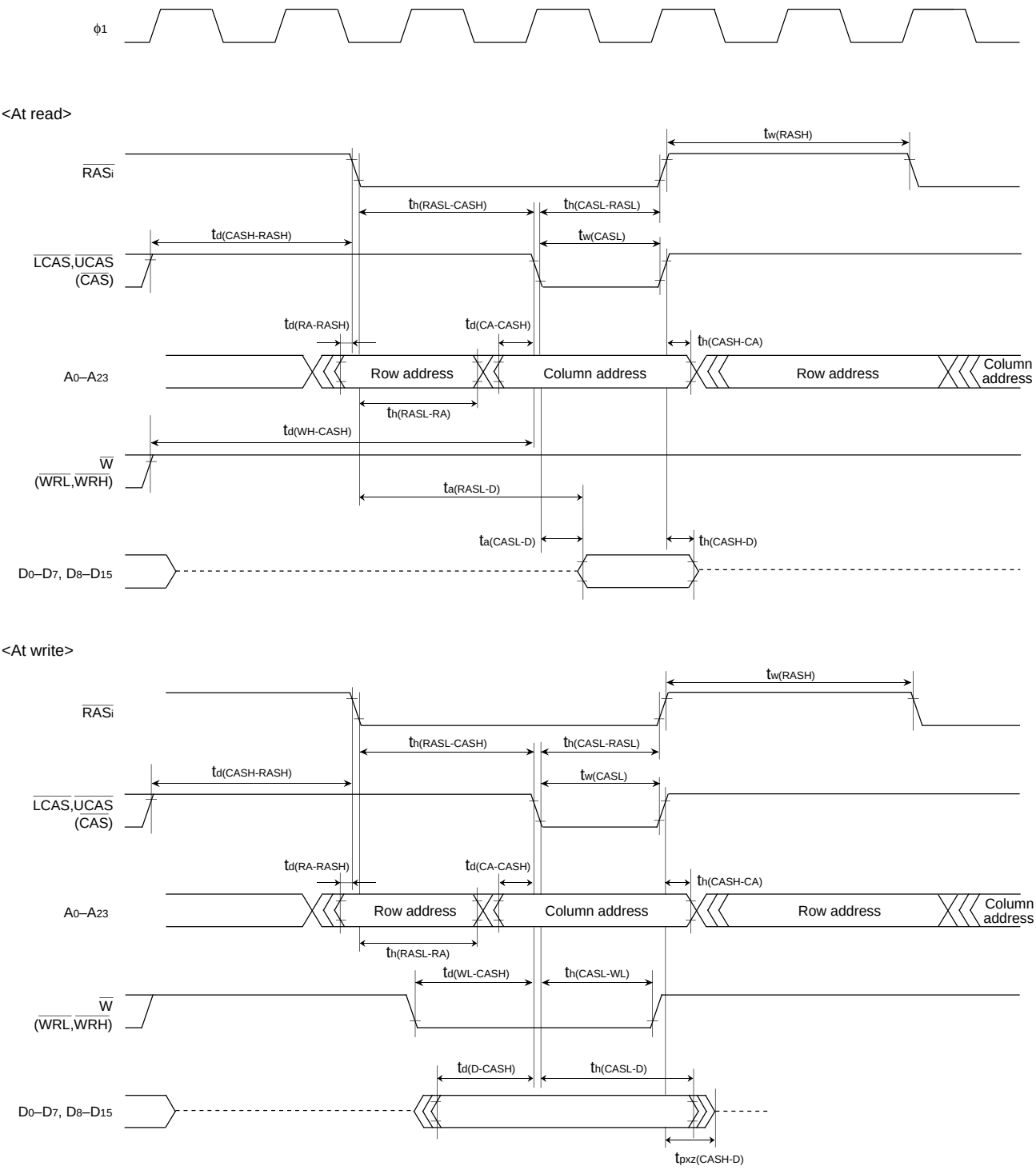
Timing Requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = 0 to 70 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{a(RASL-D)}$	$\overline{RAS}$ access time		2.5tc-35	ns
$t_{a(CASL-D)}$	$\overline{CAS}$ access time		tc-30	ns
$t_{h(CASH-D)}$	Data input hold time after $\overline{CAS}$	0		ns

Switching characteristics (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = 0 to 70 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(RASH)}$	$\overline{RAS}$ high-level pulse width	1.5tc-20		ns
$t_{d(CASH-RASH)}$	$\overline{CAS}$ high-level valid time before $\overline{RAS}$	1.5tc-20		ns
$t_{h(RASL-CASH)}$	$\overline{CAS}$ high-level hold time after $\overline{RAS}$'s low level	1.5tc-20		ns
$t_{h(CASL-RASL)}$	$\overline{RAS}$ hold time after $\overline{CAS}$'s low level	tc-15		ns
$t_{w(CASL)}$	$\overline{CAS}$ low-level pulse width	tc-15		ns
$t_{d(RA-RASH)}$	Row address valid time before $\overline{RAS}$	0.5tc-25		ns
$t_{h(RASL-RA)}$	Row address hold time after $\overline{RAS}$'s low level	tc-40		ns
$t_{d(CA-CASH)}$	Column address valid time before $\overline{CAS}$	0.5tc-20		ns
$t_{h(CASH-CA)}$	Column address hold time after $\overline{CAS}$'s high level	0		ns
$t_{d(WH-CASH)}$	$\overline{W}$ high-level valid time before $\overline{CAS}$	3tc-15		ns
$t_{d(WL-CASH)}$	$\overline{W}$ low-level valid time before $\overline{CAS}$	tc-15		ns
$t_{h(CASL-WL)}$	$\overline{W}$ hold time after $\overline{CAS}$'s low level	tc-15		ns
$t_{d(D-CASH)}$	Data output valid time before $\overline{CAS}$	tc-20		ns
$t_{h(CASL-D)}$	Data output hold time after $\overline{CAS}$'s low level	1.5tc-15		ns
$t_{pxz(CASH-D)}$	Floating start delay time after $\overline{CAS}$	0.5tc+10		ns
$t_{pxz(WH-D)}$	Floating start delay time after write		0.5tc+10	ns

DRAM access



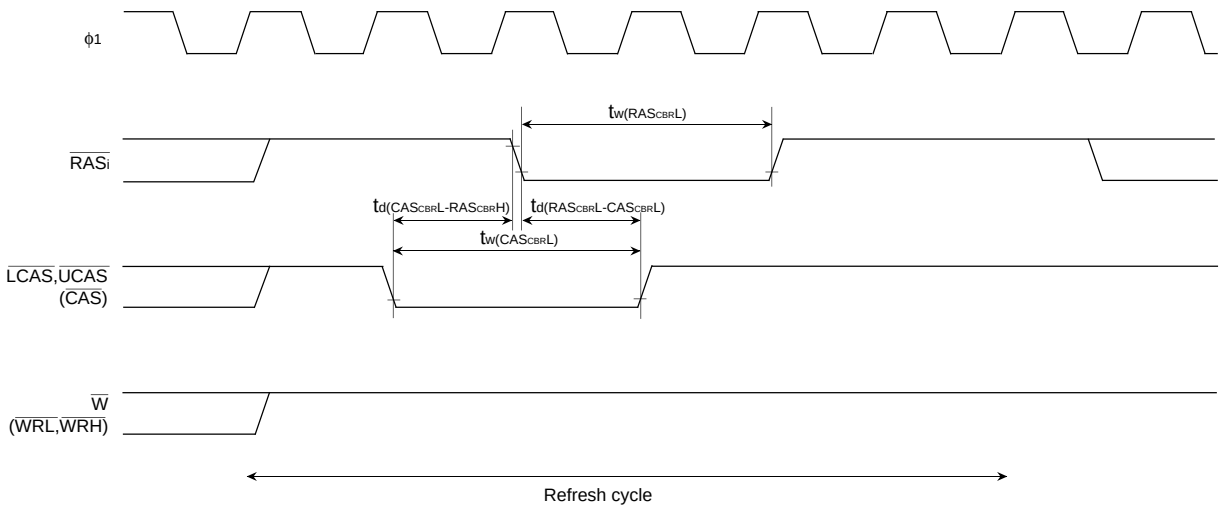
Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = 0 to 70 °C
- Input timing voltage : V_{IL}=0.8 V, V_{IH}=2.5 V
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{RAS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{RAS_i}$)

DRAM refresh

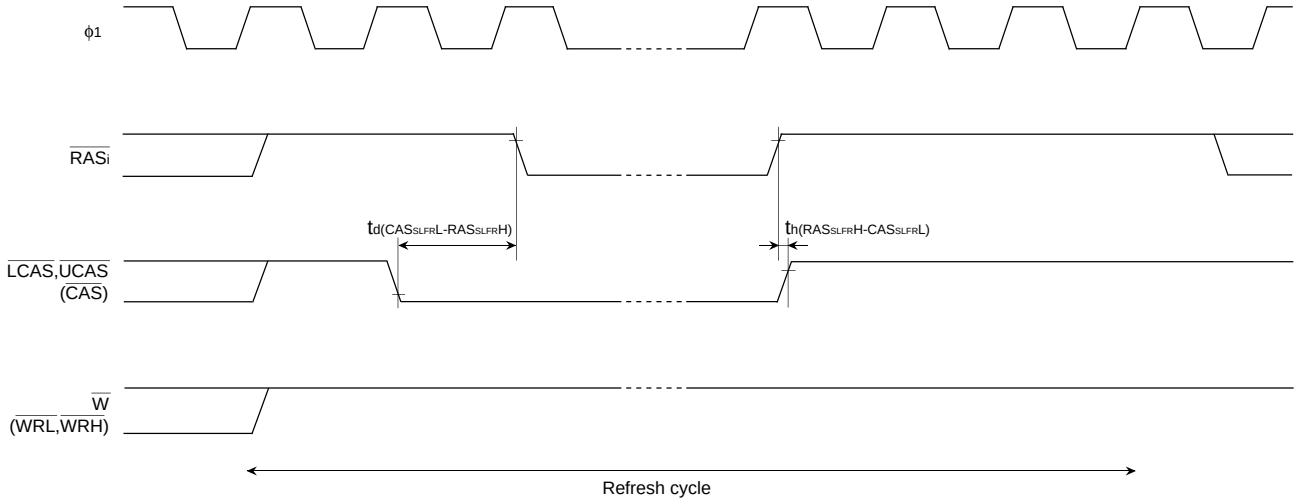
Switching characteristics (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = 0 to 70 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(RAS_{CBR}L)}$	$\overline{RAS}$ low-level pulse width (At $\overline{CAS}$ before $\overline{RAS}$ refresh)	2tc–15		ns
$t_{w(CAS_{CBR}L)}$	$\overline{CAS}$ low-level pulse width (At $\overline{CAS}$ before $\overline{RAS}$ refresh)	2tc–15		ns
$t_{d(CAS_{CBR}L\text{-}RAS_{CBR}H)}$	$\overline{RAS}$ high-level valid time after $\overline{CAS}$'s low level start (At $\overline{CAS}$ before $\overline{RAS}$ refresh)	tc–15		ns
$t_{d(RAS_{CBR}L\text{-}CAS_{CBR}L)}$	$\overline{CAS}$ low-level valid time after $\overline{RAS}$'s low level start (At $\overline{CAS}$ before $\overline{RAS}$ refresh)	tc–15		ns
$t_{d(CAS_{SLFR}L\text{-}RAS_{SLFR}H)}$	$\overline{RAS}$ high-level valid time after $\overline{CAS}$'s low level start (At selfrefresh)	tc–15		ns
$t_{h(RAS_{SLFR}H\text{-}CAS_{SLFR}L)}$	$\overline{CAS}$ low-level hold time after $\overline{RAS}$'s high level (At selfrefresh)	–15	15	ns

DRAM refresh : $\overline{CAS}$ before $\overline{RAS}$ refresh



DRAM refresh : selfrefresh



Test conditions

- V_{CC} = 5 V ± 0.5 V, Ta = 0 to 70 °C
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=15 pF ($\overline{RAS_i}$)
- Output timing voltage: V_{OL}=0.8 V, V_{OH}=2.0 V, C_L=50 pF (except for $\overline{RAS_i}$)

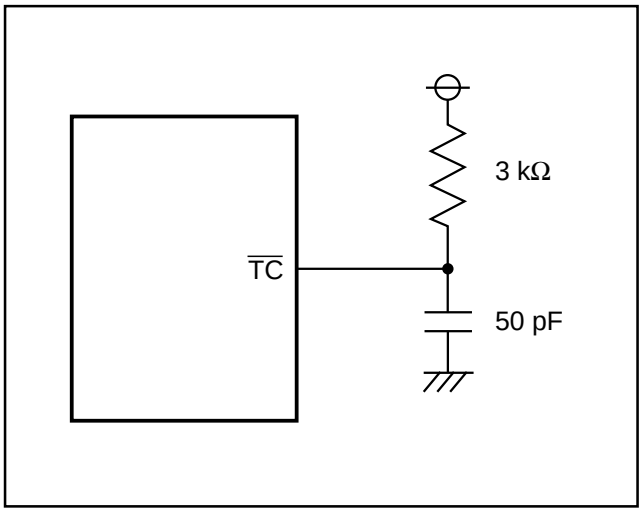
DMA transfer timing

Timing Requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{su(TC_{IN}L-\phi1)}$	$\overline{TC}$ input setup time	40		ns
$t_{w(TC_{IN}L)}$	$\overline{TC}$ input pulse width	tc + 20		ns
$t_{su(DRQL-\phi1)}$	$\overline{DMAREQ_i}$ input setup time	40		ns
$t_{w(DRQL)}$	$\overline{DMAREQ_i}$ input pulse width	tc		ns

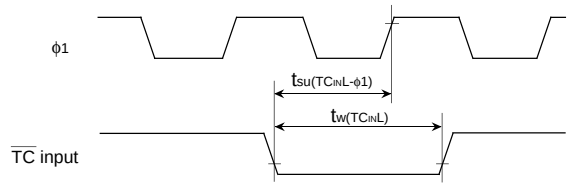
Switching characteristics (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0 V, T_a = –20 to 85 °C, f(X_{IN}) = 20 MHz, unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{w(TCL)}$	$\overline{TC}$ output pulse width	tc-20		ns
$t_{d(RDH-TCL)}$	$\overline{TC}$ output start delay time after read	tc-15		ns
$t_{d(BXWH-TCL)}$	$\overline{TC}$ output start delay time after write	tc-15		ns
$t_{d(TCL-DMAACKL)}$	$\overline{DMAACK}$ low-level output valid time after $\overline{TC}$ output start	2.5tc-20		ns

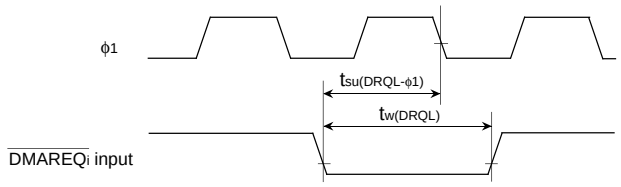


Test circuit for $\overline{TC}$ output

• $\overline{TC}$ input

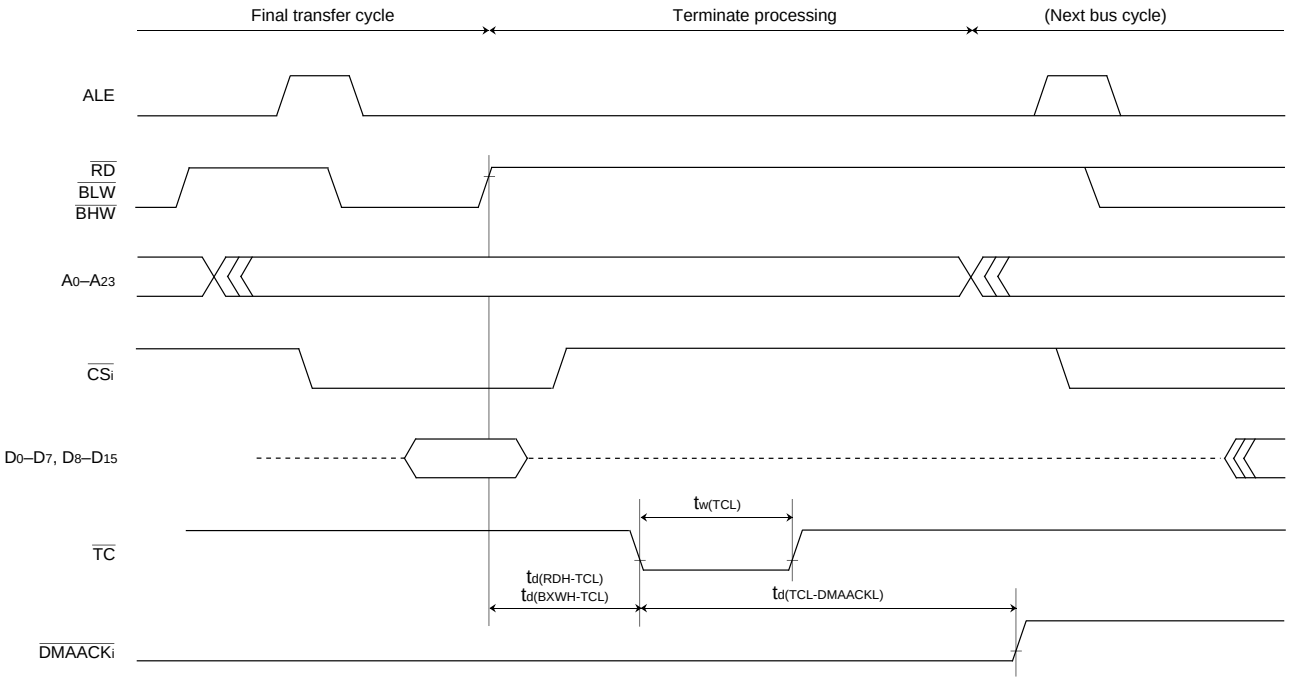


• $\overline{DMAREQ_i}$ input



Test conditions
- Vcc = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : VIL = 1.0 V, VIH = 4.0 V
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 50 pF

• Transfer terminate timing



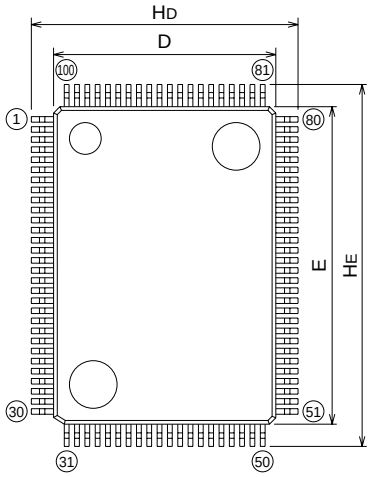
Test conditions
- Vcc = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Output timing voltage : VOL = 0.8 V, VOH = 2.0 V, CL = 50 pF

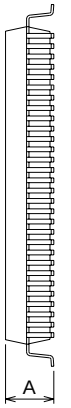
PACKAGE OUTLINE

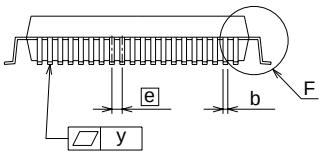
100P6S-A

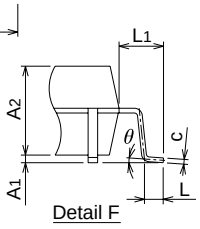
Plastic 100pin 14×20mm body QFP

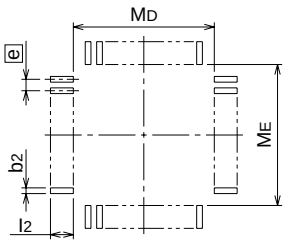
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	–	1.58	Alloy 42











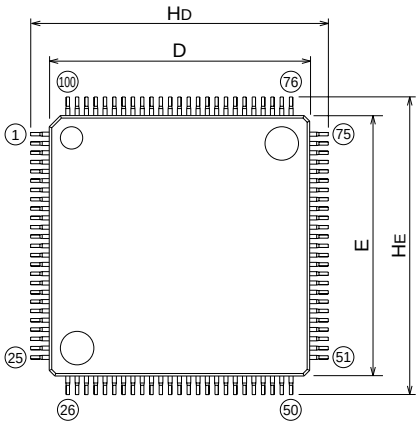
Recommended Mount Pad

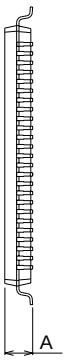
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.65	–
H_D	16.5	16.8	17.1
H_E	22.5	22.8	23.1
L	0.4	0.6	0.8
L_1	–	1.4	–
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.35	–
l_2	1.3	–	–
M_D	–	14.6	–
M_E	–	20.6	–

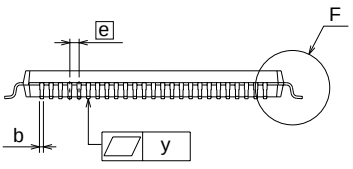
100P6Q-A

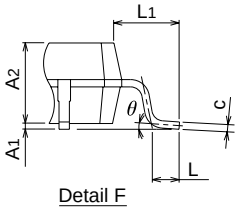
Plastic 100pin 14×14mm body LQFP

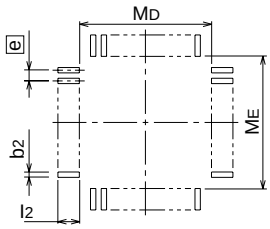
EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	–		Cu Alloy











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
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	–

M37920FCCGP, M37920FCCHP
M37920FGCGP, M37920FGCHP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER FLASH MEMORY VERSION

New publication, effective Jun., 2000.
Specifications subject to change without notice.

MITSUBISHI ELECTRIC

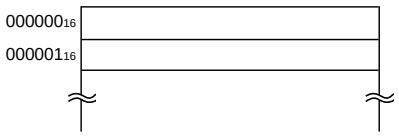
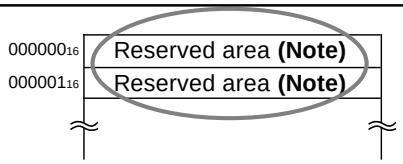
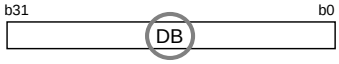
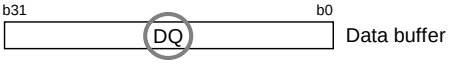
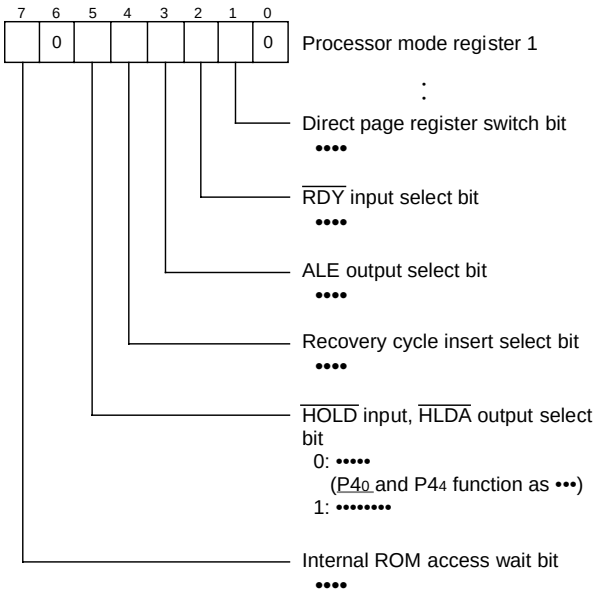
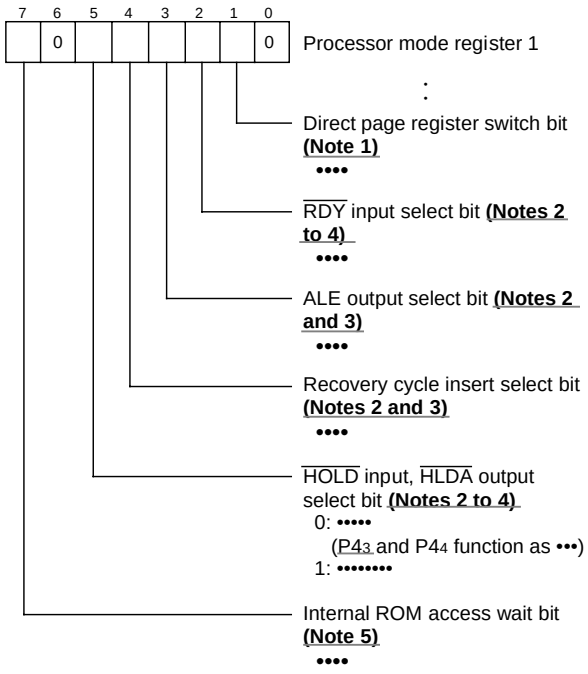
Revision History	M37920FCCGP/HP, M37920FGCGP/HP Datasheet

Rev. No.	Revision Description	Rev. date
1.0	First Edition	990602
2.0	Refer to **Corrections and Supplementary Explanation for "M37920FxCGP/HP Datasheet (REV.A)"**.	000628

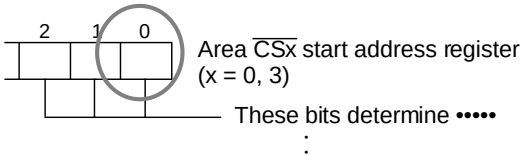
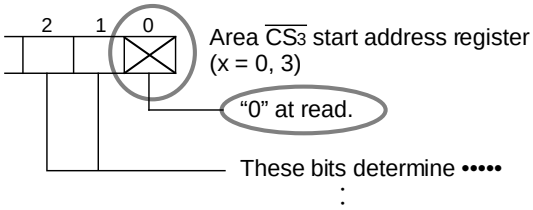
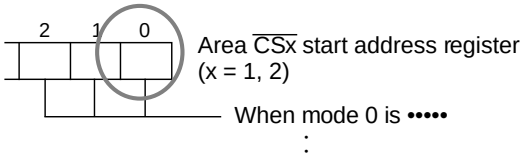
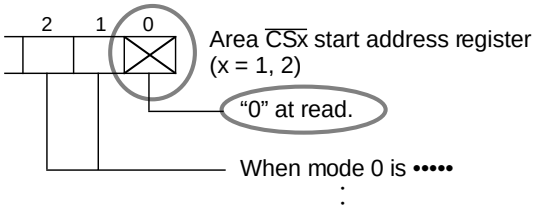
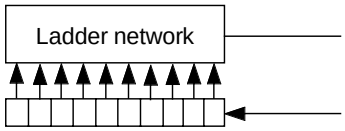
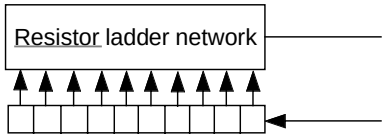
Corrections and Supplementary Explanation for M37920FxC Datasheet (REV.A) NO.1

<table>
<tr><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td>All pages,
Header</td><td>M37920F8CGP, M37920F8CHP, M37920FCCGP,
M37920FCCHP, M37920FGCGP, M37920FGCHP</td><td>M37920FCCGP, M37920FCCHP, M37920FGCGP,
M37920FGCHP</td></tr>
<tr><td>Page 1
DESCRIPTION;
Lines 11, 12</td><td>These microcomputers include the 4-channel DMA controller and the DRAM controller with enhanced fast page mode.</td><td>These microcomputers include the 4-channel DMA controller and the DRAM controller.</td></tr>
<tr><td>Page 1
DISTINCTIVE FEATURES ;
Memory</td><td>[M37920F8CGP, M37920F8CHP]
Flash memory (User ROM area) .....60 Kbytes
RAM.....2048 bytes</td><td>(Deleted)</td></tr>
<tr><td>Page 1
DISTINCTIVE FEATURES ;
Interrupts</td><td>Interrupts .....6 external sources, 17 internal sources, 7 levels</td><td>Interrupts .....6 external sources, 20 internal sources, 7 levels</td></tr>
<tr><td>Page 2,
PIN CONFIGURATION
Page 120,
Fig. 118,
Pin connection of M37920FxCGP in flash memory</td><td>(Type)
M37920F8CGP
M37920FCCGP
M37920FGCGP</td><td>(Type)
M37920FCCGP
M37920FGCGP</td></tr>
<tr><td>Page 3,
PIN CONFIGURATION
Page 121,
Fig. 119,
Pin connection of M37920FxCHP in flash memory</td><td>(Type)
M37920F8CHP
M37920FCCHP
M37920FGCHP</td><td>(Type)
M37920FCCHP
M37920FGCHP</td></tr>
<tr><td>Page 4,
BLOCK DIAGRAM Note:</td><td>Note:
<table><tr><th></th><th>Flash memory</th><th>RAM</th></tr><tr><td>M37920F8CGP,M37920F8CHP</td><td>60 Kbytes</td><td>2048 bytes</td></tr><tr><td>M37920FCCGP,M37920FCCHP</td><td>120 Kbytes</td><td>4096 bytes</td></tr><tr><td>M37920FGCGP,M37920FGCHP</td><td>248 Kbytes</td><td>6144 bytes</td></tr></table></td><td>Note:
<table><tr><th></th><th>Flash memory</th><th>RAM</th></tr><tr><td>M37920FCCGP,M37920FCCHP</td><td>120 Kbytes</td><td>4096 bytes</td></tr><tr><td>M37920FGCGP,M37920FGCHP</td><td>248 Kbytes</td><td>6144 bytes</td></tr></table></td></tr>
<tr><td>Page 5,
DRAM controller</td><td>1 channel
Supports fast page access mode.
Incorporates 8-bit refresh timer.
Supports CAS before RAS refresh method ••••••</td><td>1 channel
Incorporates 8-bit refresh timer.
Supports CAS before RAS refresh method ••••••</td></tr>
<tr><td>Page 5,
Chip-select wait control</td><td>Chip select area × 4 ($\overline{CS_0}$–$\overline{CS_3}$). A wait number and bus width can be set for each chip select area.</td><td>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.</td></tr>
<tr><td>Page 5,
Interrupts</td><td>6 external types, 17 internal types. Each interrupt ••••••</td><td>6 external types, 20 internal types. Each interrupt ••••••</td></tr>
<tr><td>Page 5,
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>Page 5,
Note:</td><td>Note:
<table><tr><td>Flash memory (User ROM area)</td><td>M37920F8CGP,M37920F8CHP</td><td>60 Kbytes</td></tr><tr><td></td><td>M37920FCCGP,M37920FCCHP</td><td>120 Kbytes</td></tr><tr><td></td><td>M37920FGCGP,M37920FGCHP</td><td>248 Kbytes</td></tr><tr><td>RAM</td><td>M37920F8CGP,M37920F8CHP</td><td>2048 bytes</td></tr><tr><td></td><td>M37920FCCGP,M37920FCCHP</td><td>4096 bytes</td></tr><tr><td></td><td>M37920FGCGP,M37920FGCHP</td><td>6144 bytes</td></tr></table></td><td>Note:
<table><tr><td>Flash memory (User ROM area)</td><td>M37920FCCGP,M37920FCCHP</td><td>120 Kbytes</td></tr><tr><td></td><td>M37920FGCGP,M37920FGCHP</td><td>248 Kbytes</td></tr><tr><td>RAM</td><td>M37920FCCGP,M37920FCCHP</td><td>4096 bytes</td></tr><tr><td></td><td>M37920FGCGP,M37920FGCHP</td><td>6144 bytes</td></tr></table></td></tr>
<tr><td>Page 6,
Note</td><td>Note:
<table><tr><td>User ROM area</td><td>M37920F8CGP, M37920F8CHP</td><td>4 blocks ••••••</td></tr><tr><td></td><td>M37920FCCGP, M37920FCCHP</td><td>5 blocks ••••••</td></tr><tr><td></td><td>M37920FGCGP, M37920FGCHP</td><td>7 blocks ••••••</td></tr></table></td><td>Note:
<table><tr><td>User ROM area</td><td>M37920FCCGP, M37920FCCHP</td><td>5 blocks ••••••</td></tr><tr><td></td><td>M37920FGCGP, M37920FGCHP</td><td>7 blocks ••••••</td></tr></table></td></tr>
</table>

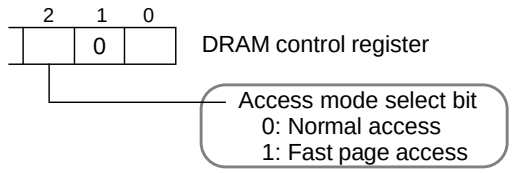
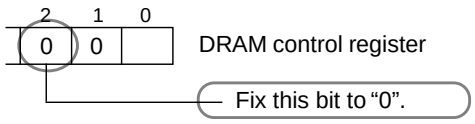
Corrections and Supplementary Explanation for M37920FxC Datasheet (REV.A) NO.2

Page	Error	Correction
———	Memory map of M37920F8CGP and M37920F8CHP (Single-chip mode)	(Deleted)
Page 9, Fig. 1	Fig. 2. Memory map of M37920FCCGP and M37920FCCHP (Single-chip mode)	Fig. 1. Memory map of M37920FCCGP and M37920FCCHP (Single-chip mode)
Page 10, Fig. 2	Fig. 3. Memory map of M37920FGCGP and M37920FGCHP (Single-chip mode)	Fig. 2. Memory map of M37920FGCGP and M37920FGCHP (Single-chip mode)
Page 11, Fig. 4; address 00_{16}, 01_{16}		
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. 8		
The final page of Selection of "processor mode"	(this contents is published for one page) ALE is an address latch enable signal. •••••••• •••••••• **Note:** The chip select wait controller, ••••••••• by the chip select wait controller.	(Deleted)
Page 28, Fig. 12		

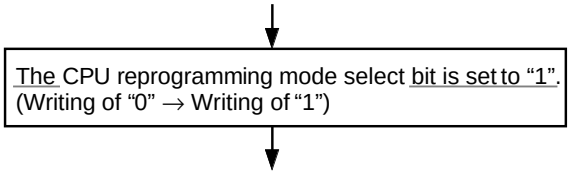
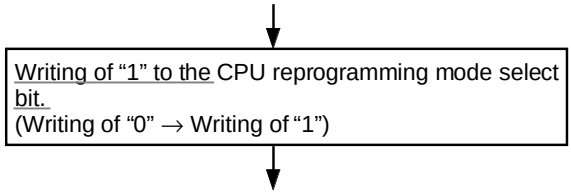
Corrections and Supplementary Explanation for M37920FxC Datasheet (REV.A) NO.3

Page	Error	Correction
Page 28, Fig. 12	———————	**Notes 1:** After reset, this bit's contents can be switched only once. During the software execution, be sure not to switch this bit's contents. **2:** In the single-chip mode, these bits' functions are disabled regardless of these bits' contents. **3:** 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. **4:** 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. **5:** 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 35, Fig. 17, Area $\overline{CSx}$ start address register (x = 0 to 3)		
		
Page 51, Left column, Lines 14, 17, 22	••••• timer Ai start bit •••••	••••• count start bit •••••
Page 61, Interrupt request at completion of reception	(Line 9) ••••• control register 0 (UART receive interrupt mode select bit) •••••	(Line 9) ••••• control register 0 (UARTk receive interrupt mode select bit) •••••
	(Lines 18, 20) ••••• UARTi receive interrupt mode select bit •••••	(Lines 18, 20) ••••• UARTk receive interrupt mode select bit •••••
Page 67, Fig. 63		
Page 68, V_{REF} connection	(Lines 2, 3) ••••••• the ladder network or not ••••••••••	(Lines 2, 3) ••••••• the resistor ladder network or not ••••••••••
	(Line 7) the ladder network can be cut off by disconnecting ladder network •••••••	(Line 7) the resistor ladder network can be cut off by disconnecting resistor ladder network •••••••

Corrections and Supplementary Explanation for M37920FxC Datasheet (REV.A) NO.4

Page	Error	Correction
Page 92, Left column, Lines 28, 29	••••••• selected. Bit 2 of the DRAM control register is the access mode select bit. When bit 2 is "0", normal access is selected, when it is "1", fast page access is selected. When the fast page access is selected, the access supporting the fast page access mode of DRAM is performed, if the range which can be specified with the same row address is continuously accessed. If the row address changes during the fast page access, the new row addresses will be output again after that fast page access is terminated. Then, the fast page access will restart. Figure 93 shows an operating waveform example of the DRAM control signals and address bus in the fast page access. Bit 4 of the DRAM •••••••	••••••• selected. Bit 4 of the DRAM •••••••
Page 95, Fig. 90		
Page 99, 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$_0$, RTP0$_3$, and RTP0$_2$ 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 •••••••••
———	Operating waveform example of DRAM control signals and address bus in fast page access	(Deleted)
———	M37920F8CGP, M37920F8CHP : block configuration of internal flash memory	(Deleted)
Page 117, 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 •••••••••
Page 118	The reset input pin. Input "H" after "L" is input.	The reset input pin.
$\overline{\text{RESET}}$; [Function]		
P4$_2$; [Function]	This is an I/O pin for serial data.	This is an I/O pin for serial data. Connect this pin to V_{CC} via a resistor (about 1 kΩ).
P4$_4$; [Function]	This is an input pin for a serial clock. Connect this pin to V_{CC} via a resistor (about 1 kΩ).	This is an input pin for a serial clock.
$\overline{\text{NMI}}$; [Function]	Input "H".	Input "H", or leave this pin open.
Page 122, Boot mode	(Lines 22, 23) program the user ROM area.	(Lines 22, 23) program the user ROM area. After reset removal, be sure not to change the status at pins MD0 and MD1.
Page 122, Fig. 120, Note 4	**4:** Valid only when bit 1 = "1". Set bit 3 to "1" (reset), and then clear to "0".	**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 the CPU reprogramming mode select bit (bit 1) = "1".

Corrections and Supplementary Explanation for M37920FxC Datasheet (REV.A) NO.5

Page	Error	Correction
Page 123, Left column, Lines 15 to 19	Therefore, a software 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 ••••••	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 ••••••
Page 123, Right column, After line 24	request occurrence.	request occurrence. In the CPU reprogramming mode, be sure not to use the **STP** and **WIT** instructions.
Page 124, Fig. 121		
Page 124, Software Commands	(Lines 6, 7) ••••••• (D_8–D_{15}) is ignored.	(Lines 6, 7) ••••••• (D_8–D_{15}) is ignored. (Except for the write data at the 2nd cycle of a page programming command.)
Page 124, Page program Command	(Titlle) Page Program Command (41_{16})	(Titlle) Page Programming Command (41_{16})
	(After line 20) ••••••• mode is maintained.	(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.
Page 125, Page program Command	(Lines 4 to 7) •••••••, the same way as bit 7 of the status register. Reading out the •••••••	(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 •••••••
Page 125, Block Erase Command	(Lines 20 to 23) •••••••, the same way as bit 7 of the status register. Reading out the •••••••	(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 •••••••
Page 127, Erase All Unlocked Block Command	(Lines 9 to 11) ••••••• is also reported by a read of the status register. When the lock bit •••••••	(Lines 9 to 11) ••••••• is also reported by a read of the status register. During the automatic erase execution (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 •••••••
Page 127, Data Protect Function (Block Lock)	(After line 20) lock bit is terminated.	(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).

Corrections and Supplementary Explanation for M37920FxC Datasheet (REV.A) NO.6

<table>
<tr><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td>Page 130,
ABSOLUTE MAXIMUM RATINGS;
T_{opr}</td><td>

Operating temperature	••••••

</td><td>

Operating ambient temperature	•••••

</td></tr>
<tr><td>Page 144,
Switching characteristics</td><td>

Symbol	Parameter	Limits		Unit
		Min.	Max.	
≈	≈	≈	≈	≈
$t_{pxz(CASH-D)}$	Floating start delay time after $\overline{CAS}$	0.5tc+10		ns
$t_{d(CAF-CASH)}$	Column address valid time before $\overline{CAS}$ (When fast page access ON is selected)	tc–40		ns
$t_{d(WFL-CASH)}$	$\overline{W}$ low-level valid time before $\overline{CAS}$ (When fast page access ON is selected)	0.5tc–20		ns
$t_{d(DF-CASH)}$	Data output valid time before $\overline{CAS}$ (When fast page access ON is selected)	0.5tc–20		ns
$t_{pxz(WH-D)}$	Floating start delay time after write		0.5tc+10	ns

</td><td>

Symbol	Parameter	Limits		Unit
		Min.	Max.	
≈	≈	≈	≈	≈
$t_{pxz(CASH-D)}$	Floating start delay time after $\overline{CAS}$	0.5tc+10		ns
$t_{pxz(WH-D)}$	Floating start delay time after write		0.5tc+10	ns

</td></tr>
<tr><td>Page 145,
DRAM access</td><td>(Title)
DRAM access : fast page access OFF</td><td>(Title)
DRAM access</td></tr>
<tr><td>———</td><td>Timing of DRAM access : fast page access OFF</td><td>(Deleted)</td></tr>
</table>