

To all our customers

---

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

---

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

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

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

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**DESCRIPTION**

These are single-chip 16-bit microcomputers designed with high-performance CMOS silicon gate technology, being packaged in 42-pin plastic molded SSOP or shrink plastic molded DIP. These microcomputers support the 7900 Series instruction set, which are enhanced and expanded instruction set and are upper-compatible with the 7700/7751 Series instruction set.

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

Also, they are suitable for motor-control equipment since each of them includes the motor control circuit.

**DISTINCTIVE FEATURES**

- Number of basic machine instructions ..... 203
- Memory
  - [M37906M4C-XXXFP, M37906M4C-XXXSP]
  - [M37906M4H-XXXFP, M37906M4H-XXXSP] **(Note)**
  - ROM ..... 32 Kbytes
  - RAM ..... 1 Kbyte
  - [M37906M6C-XXXFP, M37906M6C-XXXSP]
  - ROM ..... 48 Kbytes
  - RAM ..... 3 Kbytes

[M37906M8C-XXXFP, M37906M8C-XXXSP]

- ROM ..... 60 Kbytes
- RAM ..... 3 Kbytes

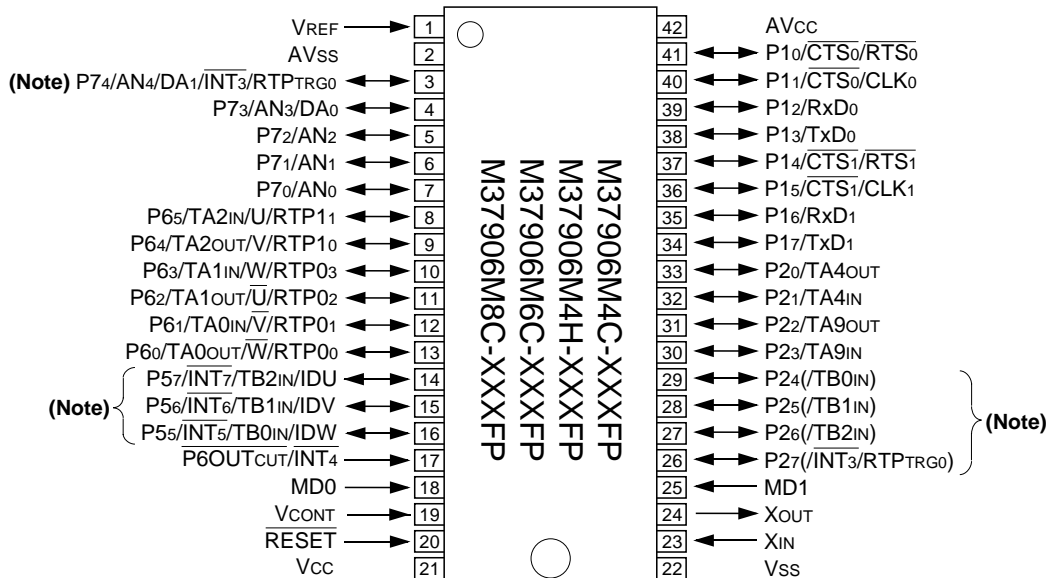
- Instruction execution time
  - The fastest instruction at 20 MHz frequency ..... 50 ns
- Single power supply ..... 5 V ± 0.5 V
- Interrupts ..... 5 external sources, 18 internal sources, 7 levels
- Multi-functional 16-bit timer ..... 10 + 3  
 (Three-phase motor drive waveform or Pulse motor drive waveform output is available.)
- Serial I/O (UART or Clock synchronous) ..... 2
- 10-bit A-D converter ..... 5-channel inputs
- 8-bit D-A converter ..... 2-channel outputs
- 12-bit watchdog timer
- Programmable input/output (ports P1, P2, P5, P6, P7) ..... 30

**Note:** The M37906M4C-XXXFP/SP and the M37906M4H-XXXFP/SP have the same functions, distinctive features, and electrical characteristics.

**APPLICATION**

- Control devices for office equipment such as copiers and facsimiles
- Control devices for industrial equipment such as communication and measuring instruments
- Control devices for equipment, requiring motor control, such as inverter air conditioners and general-purpose inverters

**M37906MxX-XXXFP PIN CONFIGURATION (TOP VIEW)**



**Note:** Allocation of these pins can be switched by software.

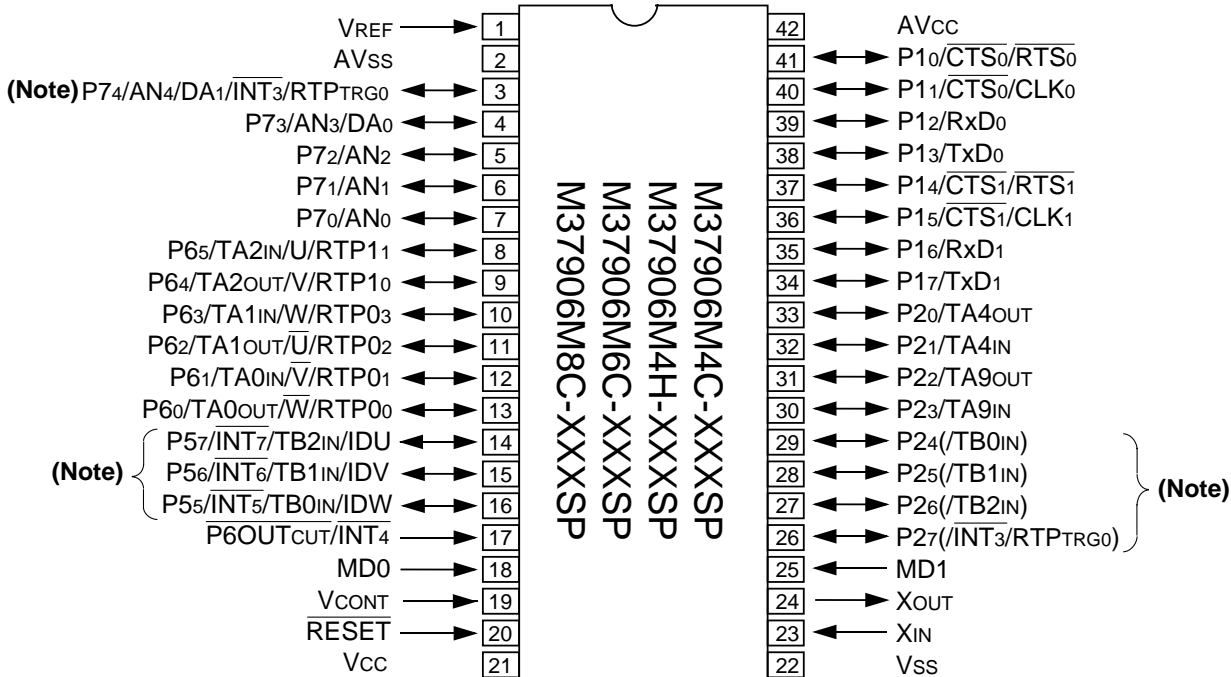
Outline 42P2R-E

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**M37906MxX-XXXSP PIN CONFIGURATION (TOP VIEW)**



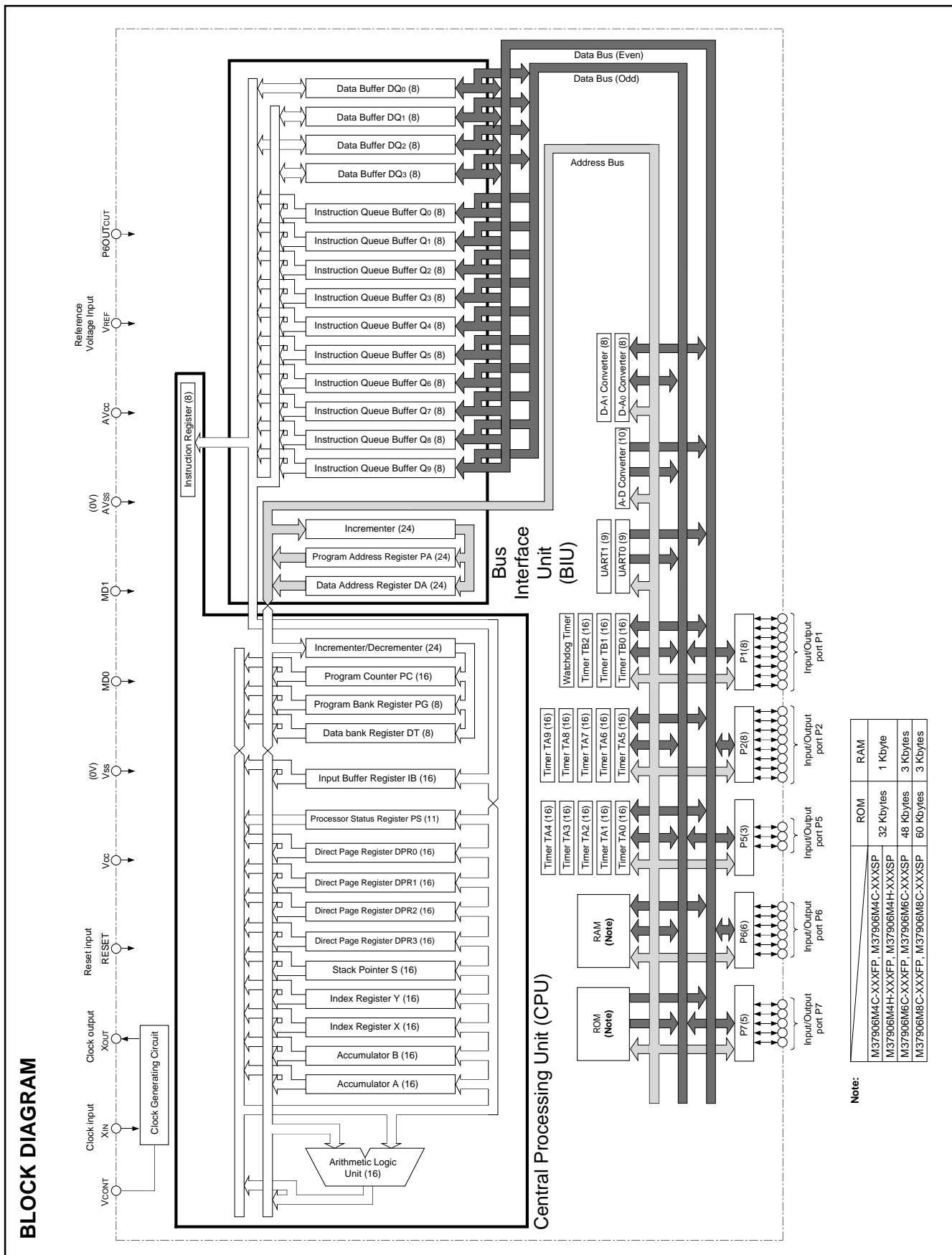
**Note:** Allocation of these pins can be switched by software.

**Outline 42P4B**

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER



**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**FUNCTIONS**

Parameter		Functions
Number of basic machine instructions		203
Instruction execution time		50 ns (the fastest instruction at $f(f_{sys}) = 20$ MHz)
External clock input frequency $f(XIN)$		20 MHz (Max.)
System clock input frequency $f(f_{sys})$		20 MHz (Max.)
Memory size	ROM	<b>(Note 1)</b>
	RAM	<b>(Note 1)</b>
Programmable input/output ports	P1, P2	8-bit X 2
	P5	3-bit X 1
	P6	6-bit X 1
	P7	5-bit X 1
Multi-functional timers	TA0–TA9	16-bit X 10
	TB0–TB2	16-bit X 3
Serial I/O	UART0 and UART1	(UART or Clock synchronous serial I/O) X 2
A-D converter		10-bit successive approximation method X 1 (5 channels)
D-A converter		8-bit X 2
Dead-time timer		8-bit X 3
Watchdog timer		12-bit X 1
Interrupts	Maskable interrupts	5 external sources, 18 internal sources. Each interrupt can be set to a priority level within the range of 0–7 by software.
	Non-maskable interrupts	3 internal sources
Clock generating circuit		Built-in (externally connected to a ceramic resonator or quartz-crystal oscillator).
PLL frequency multiplier		The following multiplication ratios are available: X 2, X 3, X 4
Power supply voltage		5 V±0.5 V
Power dissipation		125 mW (at $f(f_{sys}) = 20$ MHz, Typ., the PLL frequency multiplier is inactive)
Ports' input/output characteristics	Input/Output withstand voltage	5 V
	Output current	5 mA
Memory expansion		Not available (single-chip mode only).
Operating ambient temperature range		–20 to 85 °C
Device structure		CMOS high-performance silicon gate process
Package		<b>(Note 2)</b>

**Notes 1:**

ROM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes
	M37906M4H-XXXFP, M37906M4H-XXXSP	
	M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes
	M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes
RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	1 Kbyte
	M37906M4H-XXXFP, M37906M4H-XXXSP	
	M37906M6C-XXXFP, M37906M6C-XXXSP	3 Kbytes
	M37906M8C-XXXFP, M37906M8C-XXXSP	3 Kbytes

**2:**

Packages	M37906M4C-XXXFP, M37906M4H-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP	42-pin plastic molded SSOP (42P2R-E)
	M37906M4C-XXXSP, M37906M4H-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP	42-pin shrink plastic molded DIP (42P4B)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**PIN DESCRIPTION**

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.
$\overline{\text{RESET}}$	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 resonator or quartz-crystal oscillator between pins XIN and XOUT. When an external clock is used, the clock source should be connected to pin XIN, and pin XOUT should be left open.
XOUT	Clock output	Output	
VCONT	Filter circuit connection	—	When using the PLL frequency multiplier, connect this pin to the filter circuit. When not using, this pin should be left open.
AVcc, AVss	Analog power supply input	—	Power supply input pins for the A-D and D-A converters. 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 and D-A converters.
P10–P17	I/O port P1	I/O	Port P1 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 ar reset. These pins also function as I/O pins of UART0, 1.
P20–P27	I/O port P2	I/O	In addition to having the same functions as port P1, these pins function as I/O pins for timers A4 and A9. Also, they can be programmed to function as input pins for timers B0 to B2 and $\overline{\text{INT}}_3$ , or as trigger input pins in the pulse output port mode.
P55–P57	I/O port P5	I/O	In addition to having the same functions as port P1, these pins function as input pins for $\overline{\text{INT}}_5$ to $\overline{\text{INT}}_7$ . Also, they function as input pins for timers B0 to B2 and position data input pins in the three-phase waveform mode.
P60–P65	I/O port P6	I/O	In addition to having the same functions as port P1, these pins function as I/O pins for timers A0 to A2. Also, they function as motor drive waveform output pins.
P70–P74	I/O port P7	I/O	In addition to having the same functions as port P1, these pins function as input pins for the A-D converter. Also, P73 functions as an output pin for the D-A converter, P74 as an output pin for the D-A converter and an input pin for $\overline{\text{INT}}_3$ , and a trigger input pin in the pulse output port mode.
$\overline{\text{P6OUTCUT}}$	$\overline{\text{P6OUTCUT}}$ input	Input	This pin has the function to forcibly place port P6 pins in the input mode (port-out-put-cutoff function). Also, this pin functions as an input pin for $\overline{\text{INT}}_4$ , and as an input pin for the port-output-cutoff function in the motor drive waveform output mode.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**BASIC FUNCTION BLOCKS**

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

**MEMORY**

Figures 1 (1) through (3) show the memory maps. The address space is 64 Kbytes from addresses 016 through FFFF16. This ad-

dress space is called "bank 016".

The internal ROM and RAM are allocated as shown in Figures 1 (1) through (3).

Addresses FFB416 through FFFF16 contain the RESET and the interrupt vector addresses, and the interrupt vectors are stored there. For details, refer to the section on interrupts.

Allocated to addresses 016 through FF16 are peripheral devices such as I/O ports, A-D converter, D-A converter, serial I/O, timers, interrupt control registers, etc. Figures 2 and 3 show the location of SFRs.

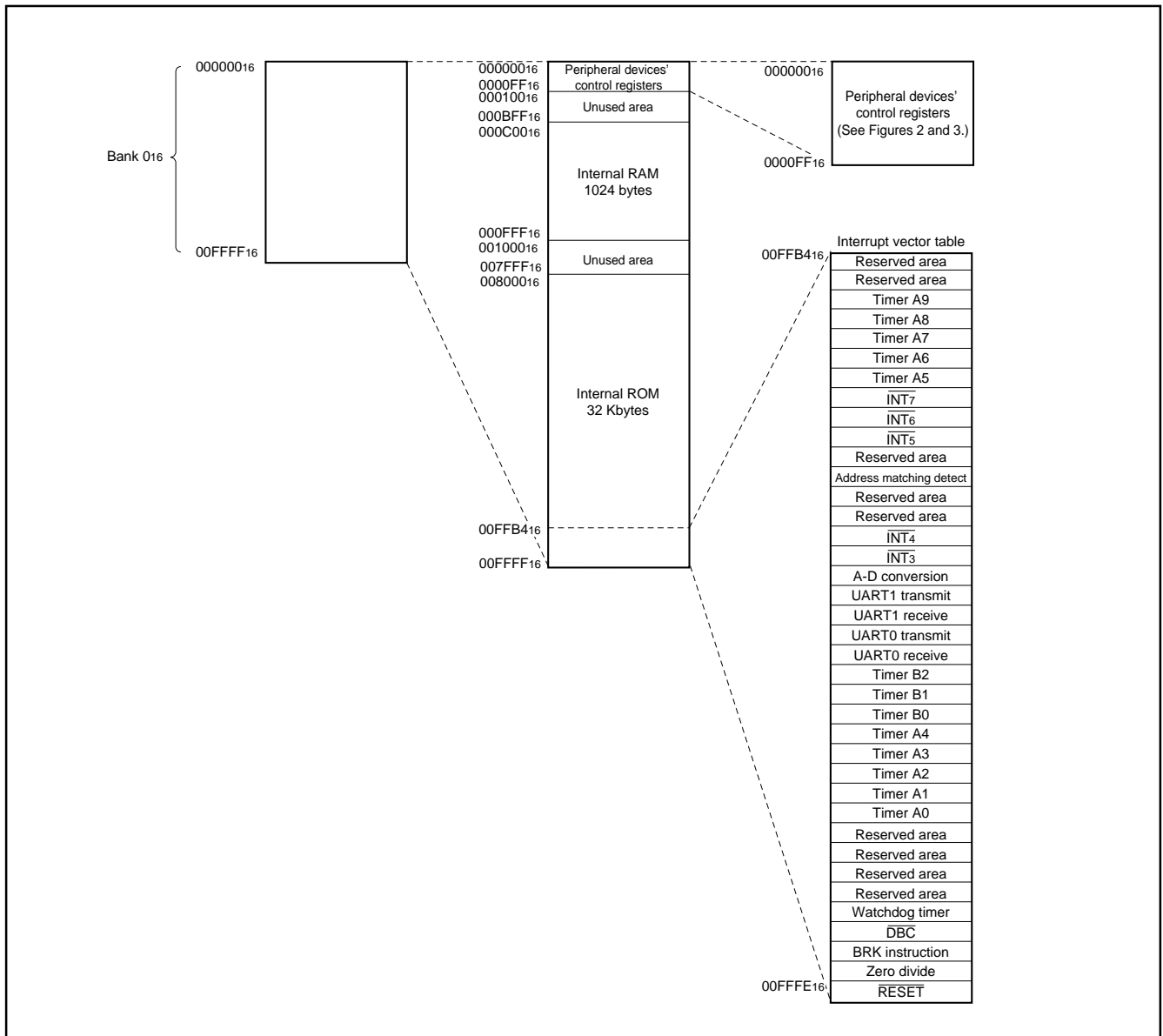


Fig. 1 (1) Memory map of M37906M4C-XXXFP/SP and M37906M4H-XXXFP/SP (Single-chip mode)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

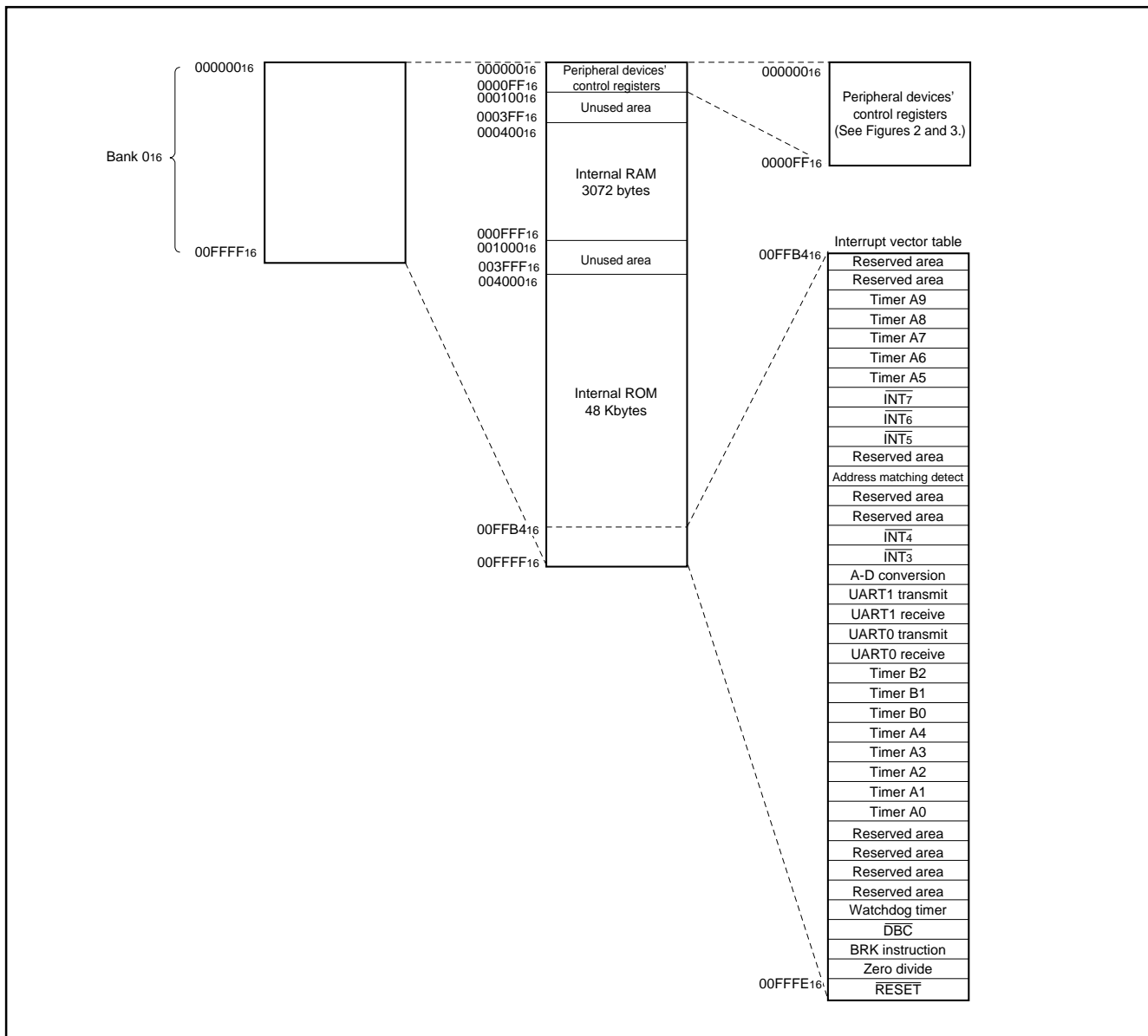


Fig. 1 (2) Memory map of M37906M6C-XXXFP/SP (Single-chip mode)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

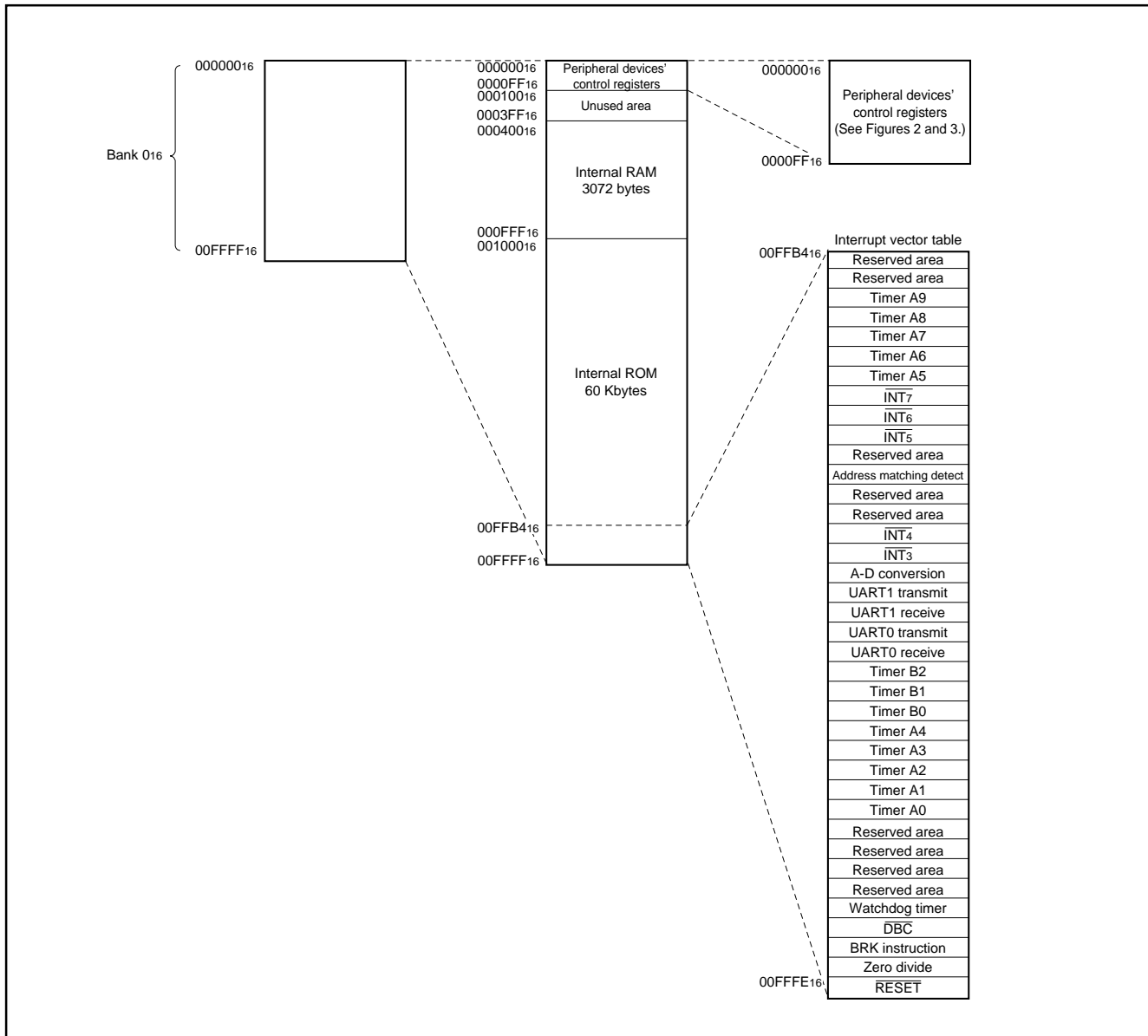


Fig. 1 (3) Memory map of M37906M8C-XXXFP/SP (Single-chip mode)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

Address (Hexadecimal notation)		Address (Hexadecimal notation)	
0000016	Reserved area (Note)	00004016	Count start register 0
00000116	Reserved area (Note)	00004116	Count start register 1
00000216	Reserved area (Note)	00004216	One-shot start register 0
00000316	Port P1 register	00004316	One-shot start register 1
00000416	Reserved area (Note)	00004416	Up-down register 0
00000516	Port P1 direction register	00004516	Timer A clock division select register
00000616	Port P2 register	00004616	Timer A0 register
00000716	Reserved area (Note)	00004716	
00000816	Port P2 direction register	00004816	Timer A1 register
00000916	Reserved area (Note)	00004916	
00000A16	Reserved area (Note)	00004A16	Timer A2 register
00000B16	Port P5 register	00004B16	
00000C16	Reserved area (Note)	00004C16	Timer A3 register
00000D16	Port P5 direction register	00004D16	
00000E16	Port P6 register	00004E16	Timer A4 register
00000F16	Port P7 register	00004F16	
00001016	Port P6 direction register	00005016	Timer B0 register
00001116	Port P7 direction register	00005116	
00001216	Reserved area (Note)	00005216	Timer B1 register
00001316		00005316	
00001416	Reserved area (Note)	00005416	Timer B2 register
00001516		00005516	
00001616	Reserved area (Note)	00005616	Timer A0 mode register
00001716	Reserved area (Note)	00005716	Timer A1 mode register
00001816	Reserved area (Note)	00005816	Timer A2 mode register
00001916	Reserved area (Note)	00005916	Timer A3 mode register
00001A16		00005A16	Timer A4 mode register
00001B16		00005B16	Timer B0 mode register
00001C16		00005C16	Timer B1 mode register
00001D16		00005D16	Timer B2 mode register
00001E16	A-D control register 0	00005E16	Processor mode register 0
00001F16	A-D control register 1	00005F16	Processor mode register 1
00002016	A-D register 0	00006016	Watchdog timer register
00002116		00006116	Watchdog timer frequency select register
00002216	A-D register 1	00006216	Particular function select register 0
00002316		00006316	Particular function select register 1
00002416	A-D register 2	00006416	Particular function select register 2
00002516		00006516	Reserved area (Note)
00002616	A-D register 3	00006616	Debug control register 0
00002716		00006716	Debug control register 1
00002816	A-D register 4	00006816	
00002916		00006916	Address comparison register 0
00002A16	Reserved area (Note)	00006A16	
00002B16	Reserved area (Note)	00006B16	
00002C16	Reserved area (Note)	00006C16	Address comparison register 1
00002D16	Reserved area (Note)	00006D16	
00002E16	Reserved area (Note)	00006E16	INT <sub>3</sub> interrupt control register
00002F16	Reserved area (Note)	00006F16	INT <sub>4</sub> interrupt control register
00003016	UART0 transmit/receive mode register	00007016	A-D conversion interrupt control register
00003116	UART0 baud rate register (BRG0)	00007116	UART0 transmit interrupt control register
00003216	UART0 transmit buffer register	00007216	UART0 receive interrupt control register
00003316		00007316	UART1 transmit interrupt control register
00003416	UART0 transmit/receive control register 0	00007416	UART1 receive interrupt control register
00003516	UART0 transmit/receive control register 1	00007516	Timer A0 interrupt control register
00003616	UART0 receive buffer register	00007616	Timer A1 interrupt control register
00003716		00007716	Timer A2 interrupt control register
00003816	UART1 transmit/receive mode register	00007816	Timer A3 interrupt control register
00003916	UART1 baud rate register (BRG1)	00007916	Timer A4 interrupt control register
00003A16	UART1 transmit buffer register	00007A16	Timer B0 interrupt control register
00003B16		00007B16	Timer B1 interrupt control register
00003C16	UART1 transmit/receive control register 0	00007C16	Timer B2 interrupt control register
00003D16	UART1 transmit/receive control register 1	00007D16	Reserved area (Note)
00003E16	UART1 receive buffer register	00007E16	Reserved area (Note)
00003F16		00007F16	Reserved area (Note)

Note: Do not write to this address.

Fig. 2 Location of SFRs (1)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

Address (Hexadecimal notation)		Address (Hexadecimal notation)	
000080 <sub>16</sub>	Reserved area (Note)	0000C0 <sub>16</sub>	
000081 <sub>16</sub>	Reserved area (Note)	0000C1 <sub>16</sub>	
000082 <sub>16</sub>	Reserved area (Note)	0000C2 <sub>16</sub>	
000083 <sub>16</sub>	Reserved area (Note)	0000C3 <sub>16</sub>	
000084 <sub>16</sub>	Reserved area (Note)	0000C4 <sub>16</sub>	Up-down register 1
000085 <sub>16</sub>	Reserved area (Note)	0000C5 <sub>16</sub>	
000086 <sub>16</sub>	Reserved area (Note)	0000C6 <sub>16</sub>	Timer A5 register
000087 <sub>16</sub>	Reserved area (Note)	0000C7 <sub>16</sub>	
000088 <sub>16</sub>		0000C8 <sub>16</sub>	Timer A6 register
000089 <sub>16</sub>		0000C9 <sub>16</sub>	
00008A <sub>16</sub>	Reserved area (Note)	0000CA <sub>16</sub>	Timer A7 register
00008B <sub>16</sub>		0000CB <sub>16</sub>	
00008C <sub>16</sub>	Reserved area (Note)	0000CC <sub>16</sub>	Timer A8 register
00008D <sub>16</sub>		0000CD <sub>16</sub>	
00008E <sub>16</sub>	Reserved area (Note)	0000CE <sub>16</sub>	Timer A9 register
00008F <sub>16</sub>		0000CF <sub>16</sub>	
000090 <sub>16</sub>	Reserved area (Note)	0000D0 <sub>16</sub>	Timer A0 <sub>1</sub> register
000091 <sub>16</sub>		0000D1 <sub>16</sub>	
000092 <sub>16</sub>	Reserved area (Note)	0000D2 <sub>16</sub>	Timer A1 <sub>1</sub> register
000093 <sub>16</sub>		0000D3 <sub>16</sub>	
000094 <sub>16</sub>		0000D4 <sub>16</sub>	Timer A2 <sub>1</sub> register
000095 <sub>16</sub>	External interrupt input read-out register	0000D5 <sub>16</sub>	
000096 <sub>16</sub>	D-A control register	0000D6 <sub>16</sub>	Timer A5 mode register
000097 <sub>16</sub>		0000D7 <sub>16</sub>	Timer A6 mode register
000098 <sub>16</sub>	D-A register 0	0000D8 <sub>16</sub>	Timer A7 mode register
000099 <sub>16</sub>	D-A register 1	0000D9 <sub>16</sub>	Timer A8 mode register
00009A <sub>16</sub>		0000DA <sub>16</sub>	Timer A9 mode register
00009B <sub>16</sub>		0000DB <sub>16</sub>	Reserved area (Note)
00009C <sub>16</sub>		0000DC <sub>16</sub>	Comparator function select register 0
00009D <sub>16</sub>		0000DD <sub>16</sub>	Reserved area (Note)
00009E <sub>16</sub>		0000DE <sub>16</sub>	Comparator result register 0
00009F <sub>16</sub>		0000DF <sub>16</sub>	Reserved area (Note)
0000A0 <sub>16</sub>	Reserved area (Note)	0000E0 <sub>16</sub>	Reserved area (Note)
0000A1 <sub>16</sub>		0000E1 <sub>16</sub>	Reserved area (Note)
0000A2 <sub>16</sub>	Reserved area (Note)	0000E2 <sub>16</sub>	Reserved area (Note)
0000A3 <sub>16</sub>		0000E3 <sub>16</sub>	Reserved area (Note)
0000A4 <sub>16</sub>	Reserved area (Note)	0000E4 <sub>16</sub>	Reserved area (Note)
0000A5 <sub>16</sub>		0000E5 <sub>16</sub>	Reserved area (Note)
0000A6 <sub>16</sub>	Waveform output mode register	0000E6 <sub>16</sub>	Reserved area (Note)
0000A7 <sub>16</sub>	Dead-time timer	0000E7 <sub>16</sub>	Reserved area (Note)
0000A8 <sub>16</sub>	Three-phase output data register 0	0000E8 <sub>16</sub>	Reserved area (Note)
0000A9 <sub>16</sub>	Three-phase output data register 1	0000E9 <sub>16</sub>	Reserved area (Note)
0000AA <sub>16</sub>	Position-data-retain function control register	0000EA <sub>16</sub>	Reserved area (Note)
0000AB <sub>16</sub>		0000EB <sub>16</sub>	Reserved area (Note)
0000AC <sub>16</sub>	Serial I/O pin control register	0000EC <sub>16</sub>	Reserved area (Note)
0000AD <sub>16</sub>		0000ED <sub>16</sub>	Reserved area (Note)
0000AE <sub>16</sub>	Port P2 pin function control register	0000EE <sub>16</sub>	Reserved area (Note)
0000AF <sub>16</sub>		0000EF <sub>16</sub>	Reserved area (Note)
0000B0 <sub>16</sub>	Reserved area (Note)	0000F0 <sub>16</sub>	
0000B1 <sub>16</sub>	Reserved area (Note)	0000F1 <sub>16</sub>	Reserved area (Note)
0000B2 <sub>16</sub>	Reserved area (Note)	0000F2 <sub>16</sub>	Reserved area (Note)
0000B3 <sub>16</sub>	Reserved area (Note)	0000F3 <sub>16</sub>	
0000B4 <sub>16</sub>	Reserved area (Note)	0000F4 <sub>16</sub>	
0000B5 <sub>16</sub>	Reserved area (Note)	0000F5 <sub>16</sub>	Timer A5 interrupt control register
0000B6 <sub>16</sub>	Reserved area (Note)	0000F6 <sub>16</sub>	Timer A6 interrupt control register
0000B7 <sub>16</sub>	Reserved area (Note)	0000F7 <sub>16</sub>	Timer A7 interrupt control register
0000B8 <sub>16</sub>	Reserved area (Note)	0000F8 <sub>16</sub>	Timer A8 interrupt control register
0000B9 <sub>16</sub>		0000F9 <sub>16</sub>	Timer A9 interrupt control register
0000BA <sub>16</sub>	Reserved area (Note)	0000FA <sub>16</sub>	
0000BB <sub>16</sub>	Reserved area (Note)	0000FB <sub>16</sub>	
0000BC <sub>16</sub>	Clock control register 0	0000FC <sub>16</sub>	
0000BD <sub>16</sub>	Reserved area (Note)	0000FD <sub>16</sub>	INT <sub>5</sub> interrupt control register
0000BE <sub>16</sub>	Reserved area (Note)	0000FE <sub>16</sub>	INT <sub>6</sub> interrupt control register
0000BF <sub>16</sub>	Reserved area (Note)	0000FF <sub>16</sub>	INT <sub>7</sub> interrupt control register

**Note:** Do not write to this address.

Fig. 3 Location of SFRs (2)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**CENTRAL PROCESSING UNIT (CPU)**

The CPU has 13 registers and is shown in Figure 4. 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).

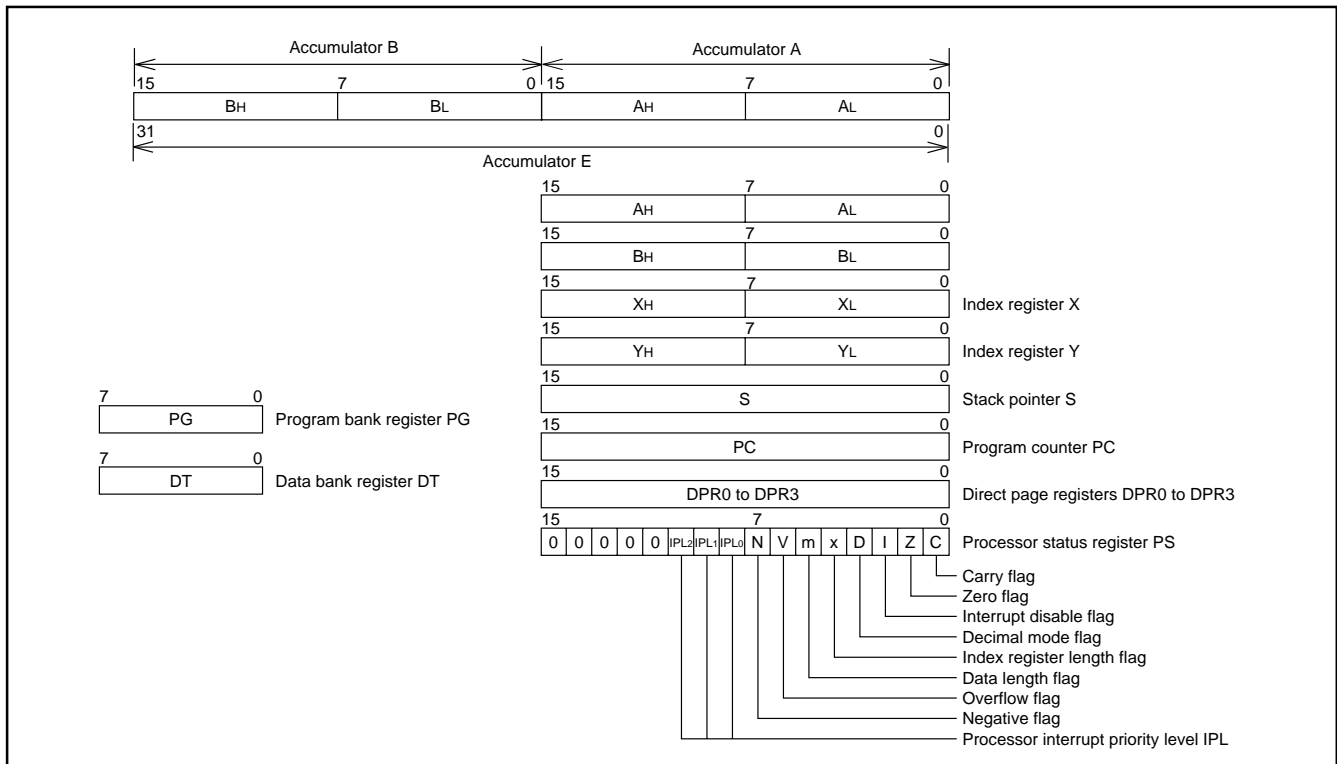


Fig. 4 Register structure

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

### 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 through 3 (DPR0 through 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 and software interrupts 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.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

### 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".

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**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<sub>16</sub> to FF<sub>16</sub>) 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<sub>16</sub> (addresses 0<sub>16</sub> to FFFF<sub>16</sub>). The direct page and direct addressing modes have been provided for the effective access to bank 0<sub>16</sub>. 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<sub>16</sub>", 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<sub>16</sub>", 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".

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**BUS INTERFACE UNIT**

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

Figure 5 shows the BIU and the bus structure. The CPU and BIU are connected by a dedicated bus, and any transfer between the CPU and BIU is controlled by this dedicated bus.

On the other hand, data transfer between the BIU and internal pe-

ripheral 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.

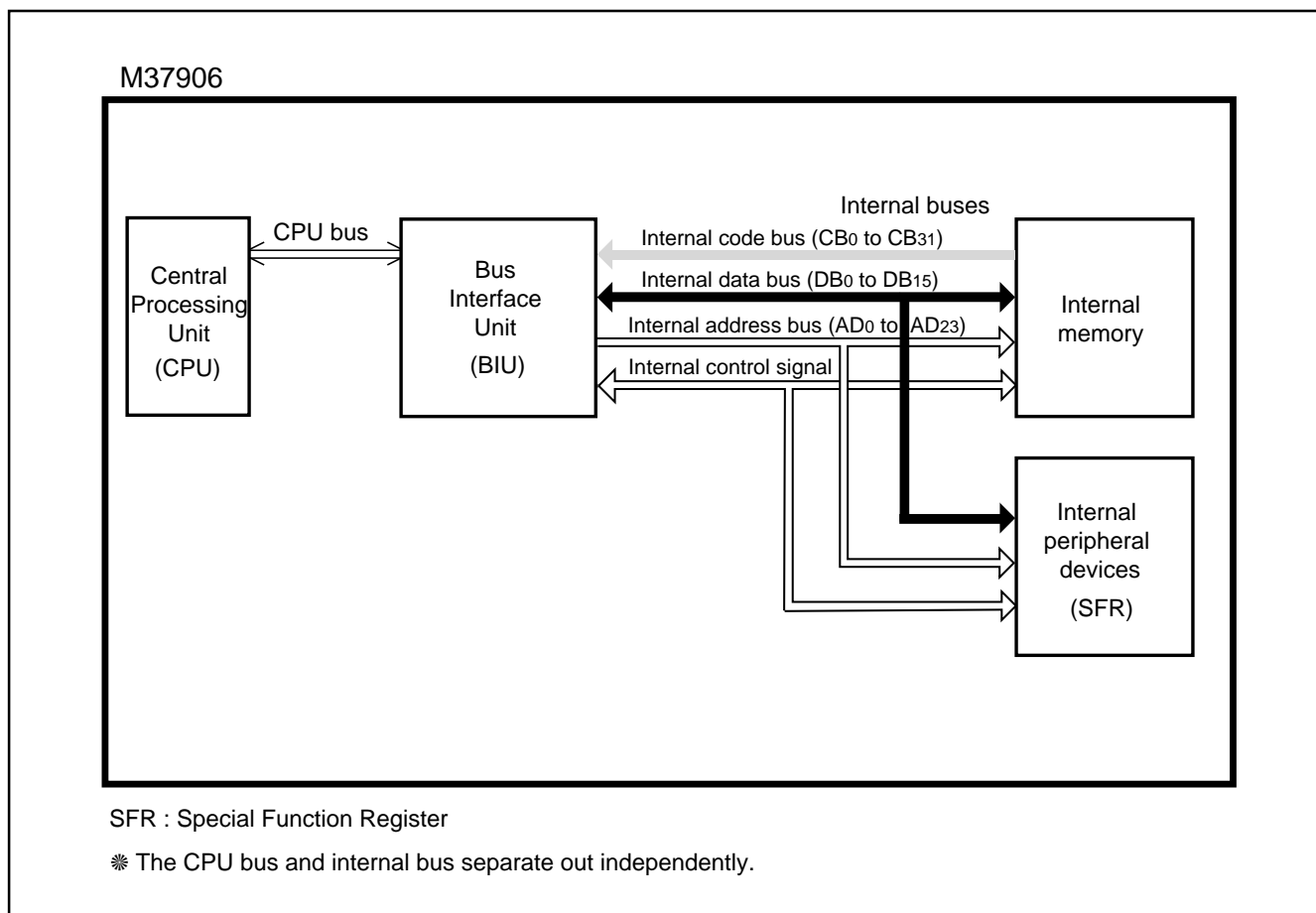


Fig. 5 BIU and bus structure

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**BIU structure**

The BIU consists of four registers shown in Figure 6. 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 or internal peripheral devices by the BIU; or temporarily stores data which is to be written to internal memory or internal peripheral devices by the CPU. Consists of 32 bits.

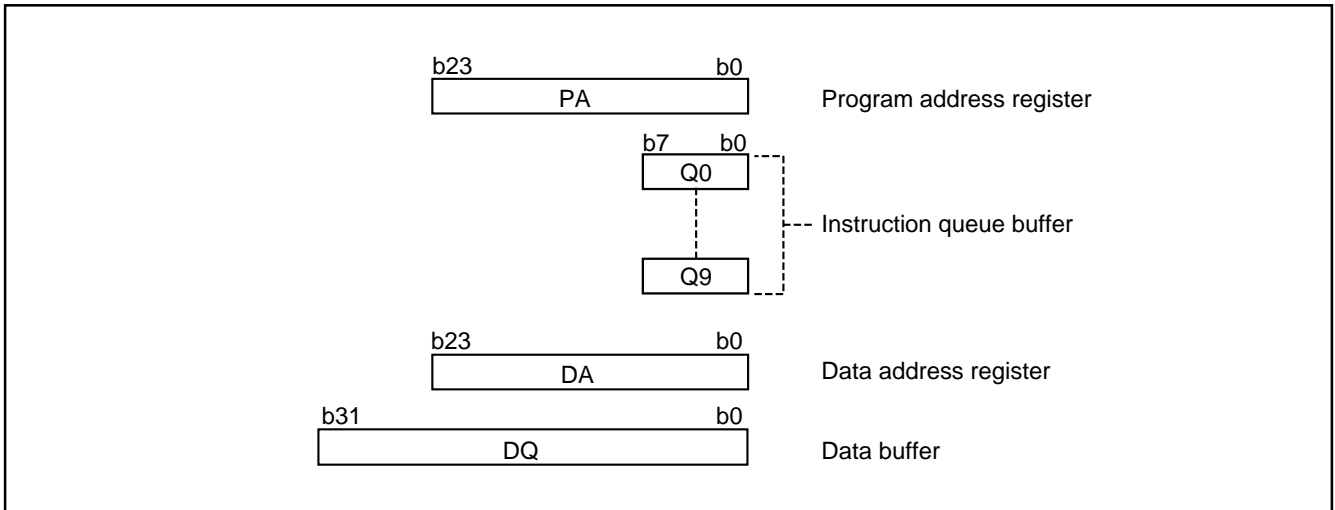


Fig. 6 Register structure of BIU

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**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 or internal peripheral devices, 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 or internal peripheral devices, 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 or internal peripheral devices. This operation is called "bus cycle". The bus cycle is affected by the length of data to be transferred (byte, word, or double-word) at data access.

Table 2. Store addresses for instructions to be prefetched

	Low-order 3 bits of store address for instruction		
	AD2 (A2)	AD1 (A1)	AD0 (A0)
Even address	X	X	0
4-byte boundary	X	0	0
8-byte boundary	0	0	0

X: 0 or 1

Figures 7 and 8 show the bus cycle waveform examples for instruction prefetch and data access.

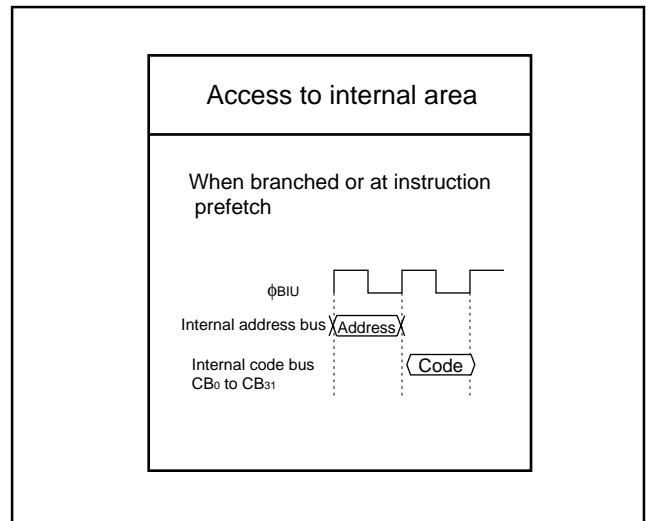


Fig. 7 Bus cycle waveform example for instruction prefetch

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

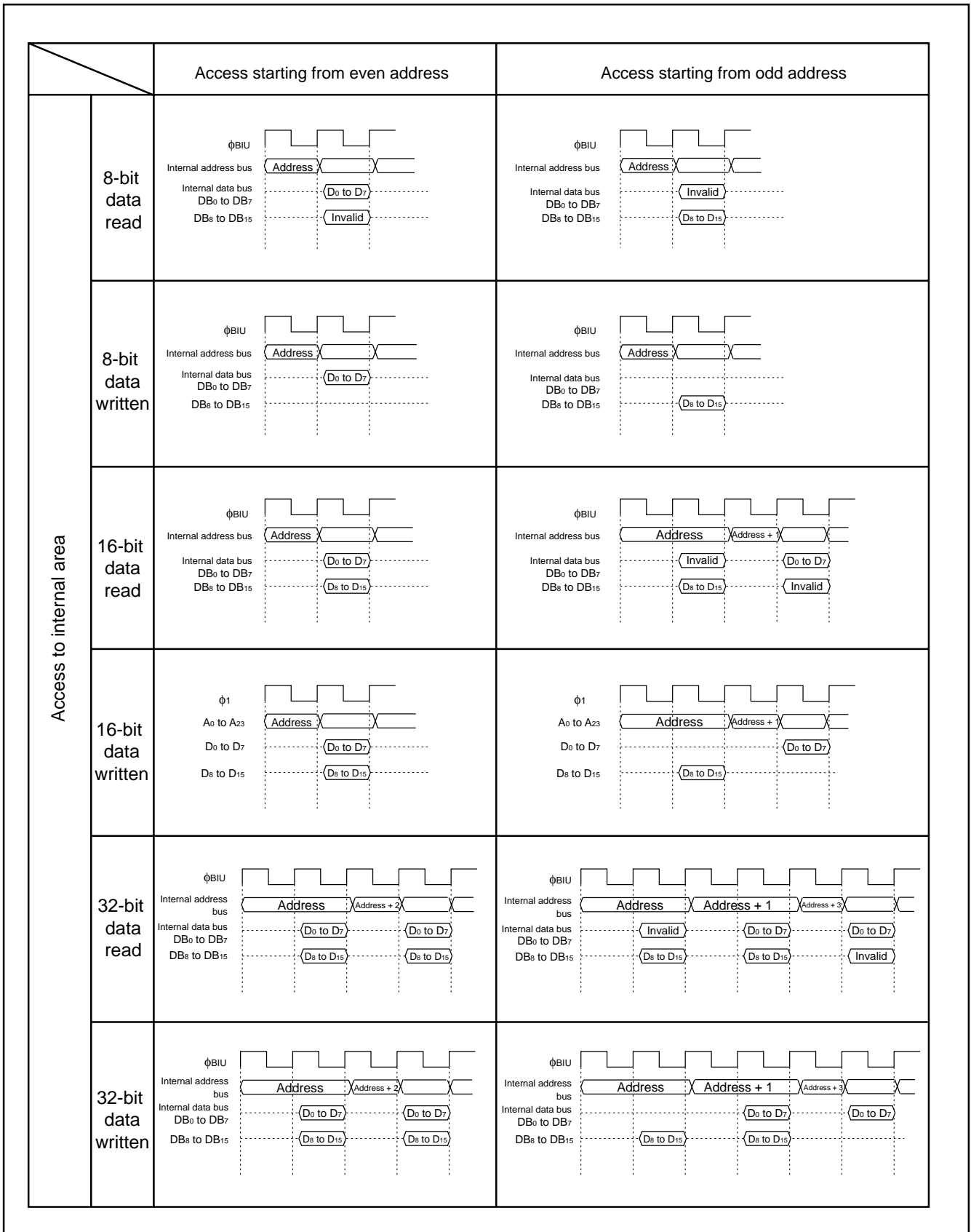


Fig. 8 Bus cycle waveform example for data access (access to internal area)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

● **Number of bus cycles**

Figure 9 shows the bus cycle waveform at access to the internal area. Bit 7 of the processor mode register 1 (address 5F16), which is shown in Figure 10, selects the number of bus cycles for the internal

ROM: 3φ or 2φ. (This bit 7 is the internal ROM bus cycle select bit.)  
 The internal RAM, SFRs (internal peripheral devices' control registers) are always accessed with 1 bus cycle = 2φ.

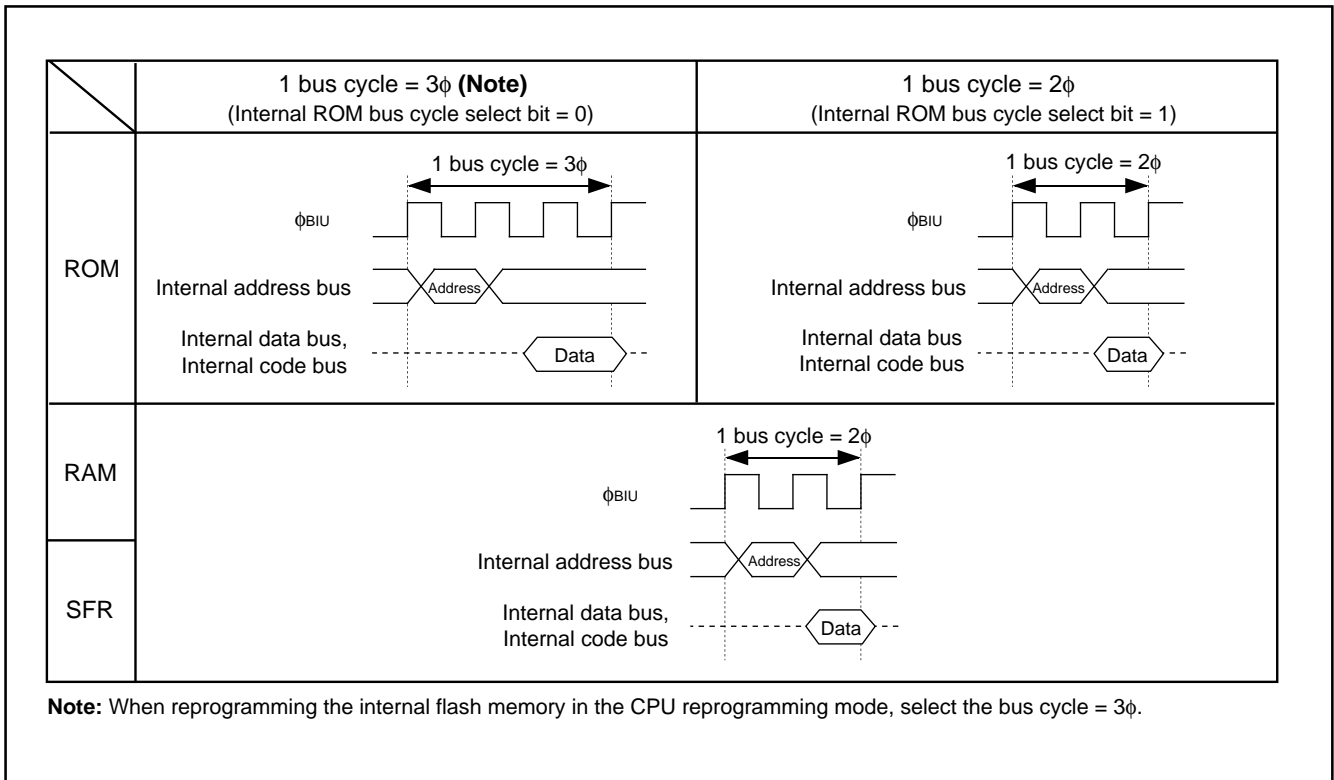


Fig. 9 Bus cycle waveform at access to internal area

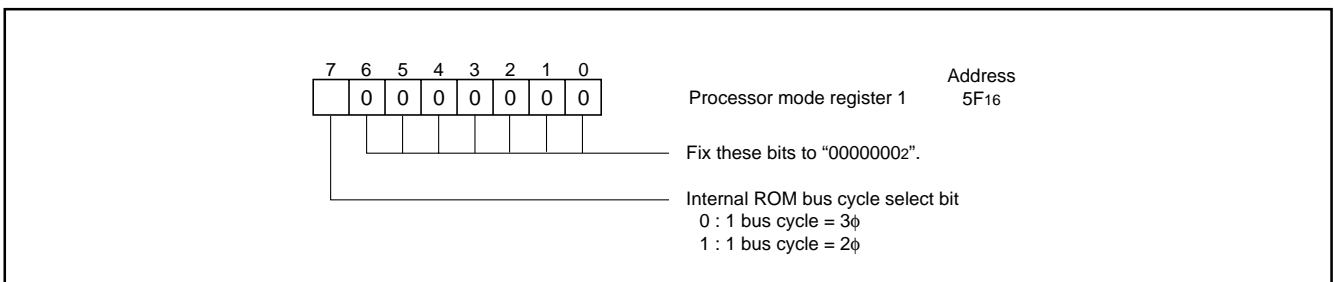


Fig. 10 Bit configuration of processor mode register 1

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**PROCESSOR MODES**

This microcomputer is dedicated to the single-chip mode. Therefore, be sure to connect pin MD0 to Vss, and be sure to fix the processor mode bits (bits 1 and 0 of the processor mode register 0, address 5E16), which is shown in Figure 11, to "002".

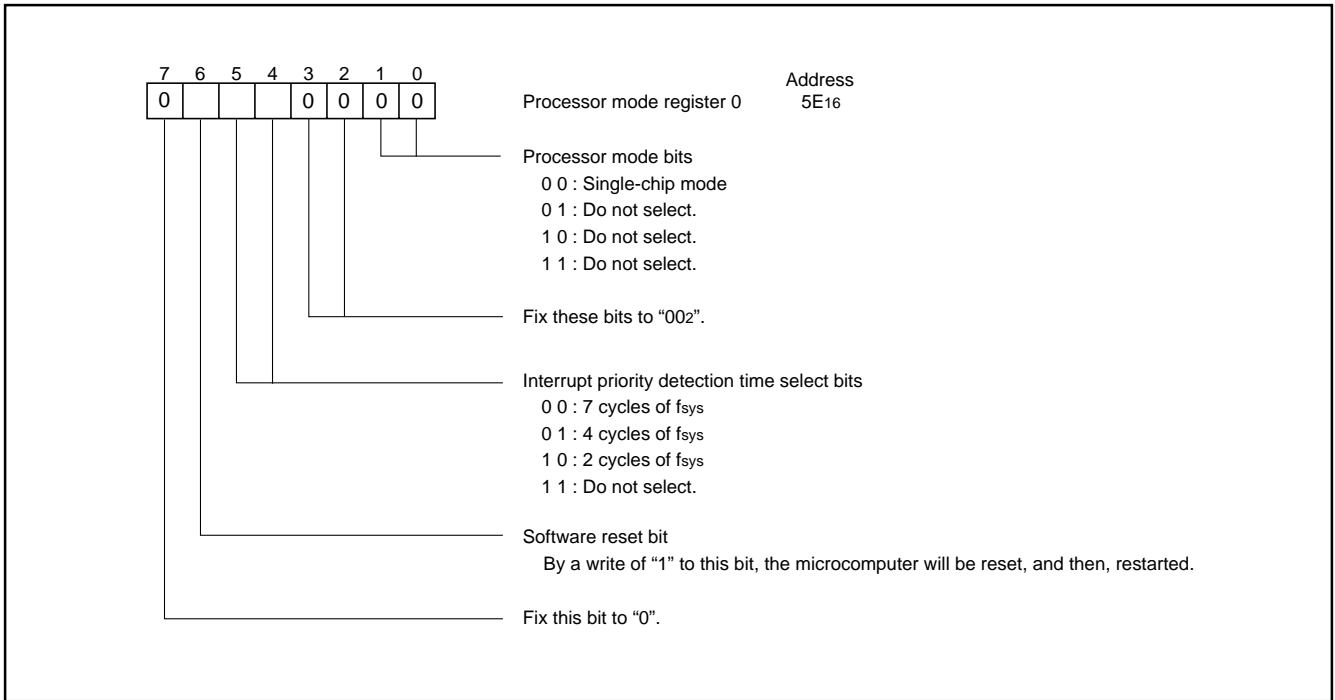


Fig. 11 Bit configuration of processor mode register 0

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**INTERRUPTS**

Table 3 shows the interrupt sources and the corresponding interrupt vector addresses. Reset is also handled as an interrupt source 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, and address matching detection all have interrupt control registers. Table 4 shows the addresses of the interrupt control registers and Figure 13 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 except for that of a watchdog timer interrupt can be cleared by software.

An  $\overline{INT}_i$  ( $i = 3$  to  $7$ ) interrupt request is generated by an external input.

$\overline{INT}_i$  is an external interrupt; 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.

The position where pin  $\overline{INT}_3/\overline{RTPTRG0}$  is allocated can be selected by the pin  $\overline{INT}_3/\overline{RTPTRG0}$  select bit (bit 3 of the port P2 pin function control register; address  $AE_{16}$ ), as shown in Figure 17.

When using the following pins as external interrupt input pins, be sure to clear the direction registers of the corresponding multiplexed ports to "0": pins  $P2_7/\overline{INT}_3$ ,  $P5_5/\overline{INT}_5$ ,  $P5_6/\overline{INT}_6$ , and  $P5_7(P7_4)/\overline{INT}_7$ . When the external interrupt input read register (address  $95_{16}$ ), which is shown in Figure 12, is read out, the status of pins  $\overline{INT}_3$  through  $\overline{INT}_7$  can directly be read.

Timer and UART interrupts are described in the respective section. The priority of interrupts when multiple interrupt requests are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 14.

The hardware priority is fixed as the following:

reset > watchdog timer > other interrupts

Table 3. Interrupt sources and interrupt vector addresses

Interrupts	Vector addresses
Timer A9	00FFB8 <sub>16</sub> 00FFB9 <sub>16</sub>
Timer A8	00FFBA <sub>16</sub> 00FFBB <sub>16</sub>
Timer A7	00FFBC <sub>16</sub> 00FFBD <sub>16</sub>
Timer A6	00FFBE <sub>16</sub> 00FFBF <sub>16</sub>
Timer A5	00FFC0 <sub>16</sub> 00FFC1 <sub>16</sub>
$\overline{INT}_7$ external interrupt	00FFC2 <sub>16</sub> 00FFC3 <sub>16</sub>
$\overline{INT}_6$ external interrupt	00FFC4 <sub>16</sub> 00FFC5 <sub>16</sub>
$\overline{INT}_5$ external interrupt	00FFC6 <sub>16</sub> 00FFC7 <sub>16</sub>
Address matching detection interrupt	00FFCA <sub>16</sub> 00FFCB <sub>16</sub>
$\overline{INT}_4$ external interrupt	00FFD0 <sub>16</sub> 00FFD1 <sub>16</sub>
$\overline{INT}_3$ external interrupt	00FFD2 <sub>16</sub> 00FFD3 <sub>16</sub>
A-D conversion	00FFD4 <sub>16</sub> 00FFD5 <sub>16</sub>
UART1 transmit	00FFD6 <sub>16</sub> 00FFD7 <sub>16</sub>
UART1 receive	00FFD8 <sub>16</sub> 00FFD9 <sub>16</sub>
UART0 transmit	00FFDA <sub>16</sub> 00FFDB <sub>16</sub>
UART0 receive	00FFDC <sub>16</sub> 00FFDD <sub>16</sub>
Timer B2	00FFDE <sub>16</sub> 00FFDF <sub>16</sub>
Timer B1	00FFE0 <sub>16</sub> 00FFE1 <sub>16</sub>
Timer B0	00FFE2 <sub>16</sub> 00FFE3 <sub>16</sub>
Timer A4	00FFE4 <sub>16</sub> 00FFE5 <sub>16</sub>
Timer A3	00FFE6 <sub>16</sub> 00FFE7 <sub>16</sub>
Timer A2	00FFE8 <sub>16</sub> 00FFE9 <sub>16</sub>
Timer A1	00FFEA <sub>16</sub> 00FFEB <sub>16</sub>
Timer A0	00FFEC <sub>16</sub> 00FFED <sub>16</sub>
Watchdog timer	00FFF6 <sub>16</sub> 00FFF7 <sub>16</sub>
$\overline{DBC}$ (Do not select.)	00FFF8 <sub>16</sub> 00FFF9 <sub>16</sub>
Break instruction (Do not select.)	00FFFA <sub>16</sub> 00FFFB <sub>16</sub>
Zero divide	00FFFC <sub>16</sub> 00FFFD <sub>16</sub>
Reset	00FFFE <sub>16</sub> 00FFFF <sub>16</sub>

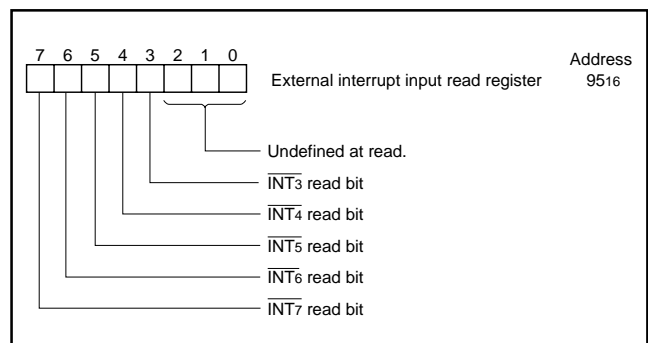


Fig. 12 Bit configuration of external interrupt input read register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

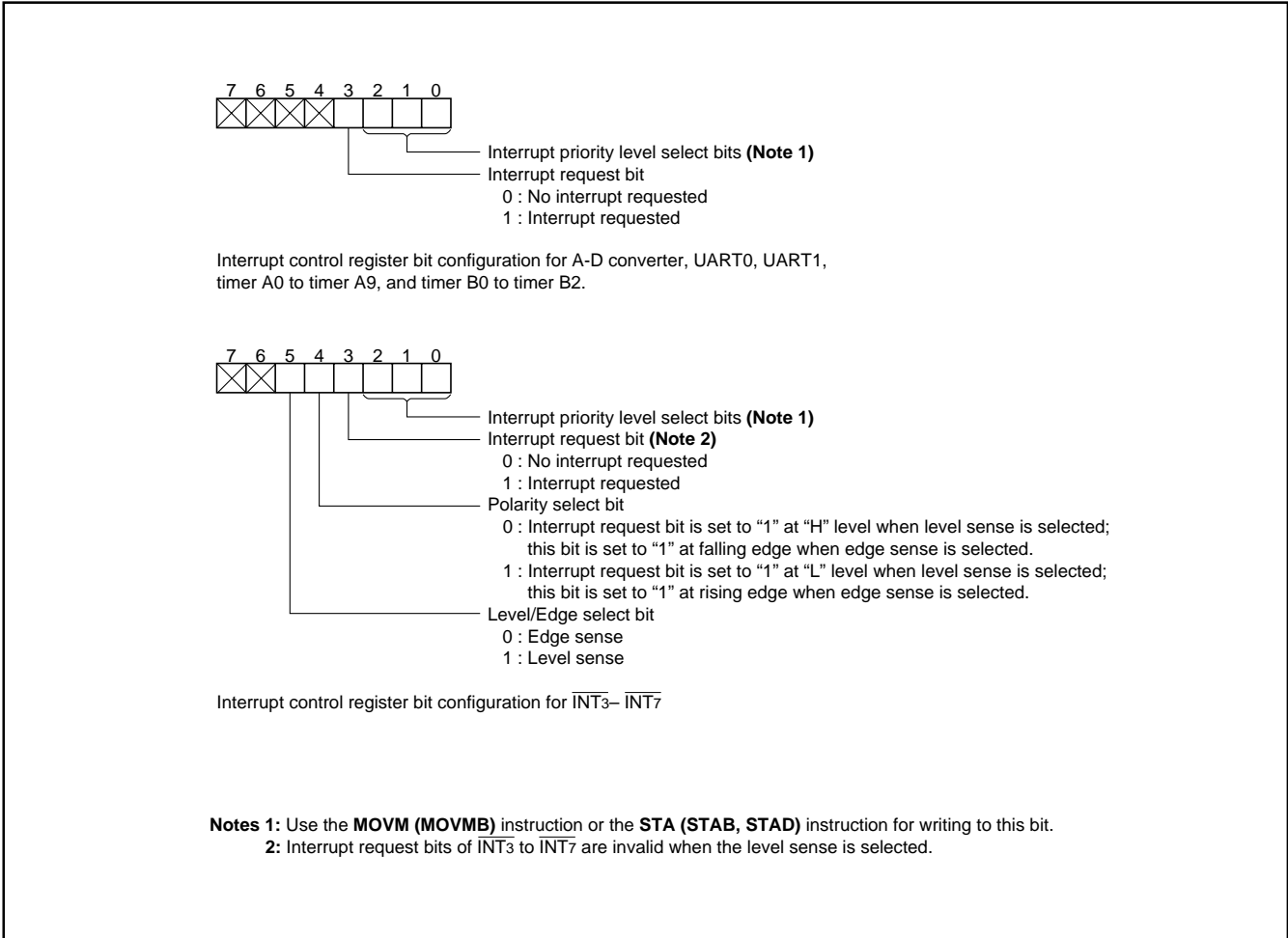


Fig. 13 Bit configuration of interrupt control register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

Table 4. Addresses of interrupt control registers

Interrupt control registers	Addresses
INT3 interrupt control register	00006E16
INT4 interrupt control register	00006F16
A-D interrupt control register	00007016
UART0 transmit interrupt control register	00007116
UART0 receive interrupt control register	00007216
UART1 transmit interrupt control register	00007316
UART1 receive interrupt control register	00007416
Timer A0 interrupt control register	00007516
Timer A1 interrupt control register	00007616
Timer A2 interrupt control register	00007716
Timer A3 interrupt control register	00007816
Timer A4 interrupt control register	00007916
Timer B0 interrupt control register	00007A16
Timer B1 interrupt control register	00007B16
Timer B2 interrupt control register	00007C16
Timer A5 interrupt control register	0000F516
Timer A6 interrupt control register	0000F616
Timer A7 interrupt control register	0000F716
Timer A8 interrupt control register	0000F816
Timer A9 interrupt control register	0000F916
INT5 interrupt control register	0000FD16
INT6 interrupt control register	0000FE16
INT7 interrupt control register	0000FF16

Interrupts caused by the address matching detection and when dividing by zero are software interrupts and are not included in Figure 14.

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 15 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 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, and address match detection interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 5.

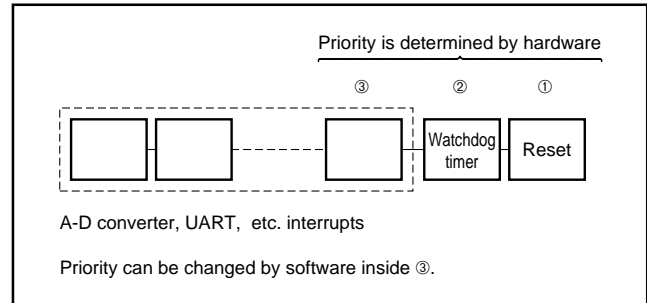


Fig. 14 Interrupt priority

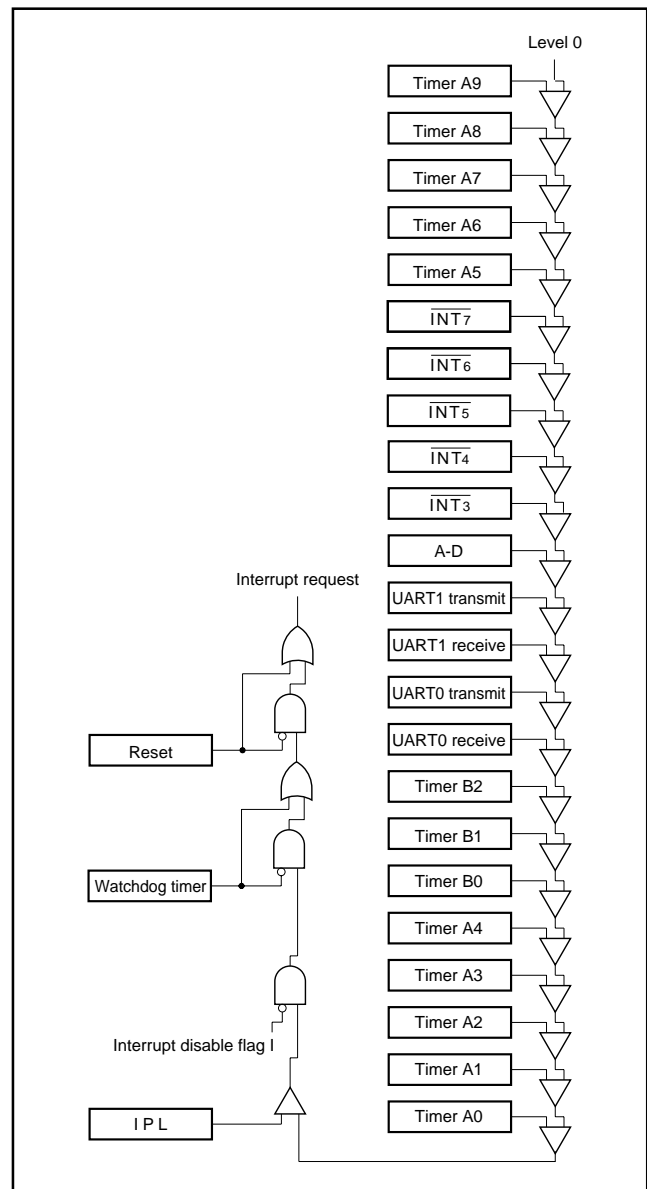


Fig. 15 Interrupt priority detection

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

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

As shown in Figure 16, 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 5E16) shown in Figure 11. Table 6 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register 0 is initialized to "0016." Therefore, the longest time is automatically set, however, the shortest time must be selected by software.

Table 5. Value loaded in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
Watchdog timer	7
Zero divide	Not change value of IPL.
Address matching detection	Not change value of IPL.

Table 6. Relationship between interrupt priority detection time select bit and number of cycles

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

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

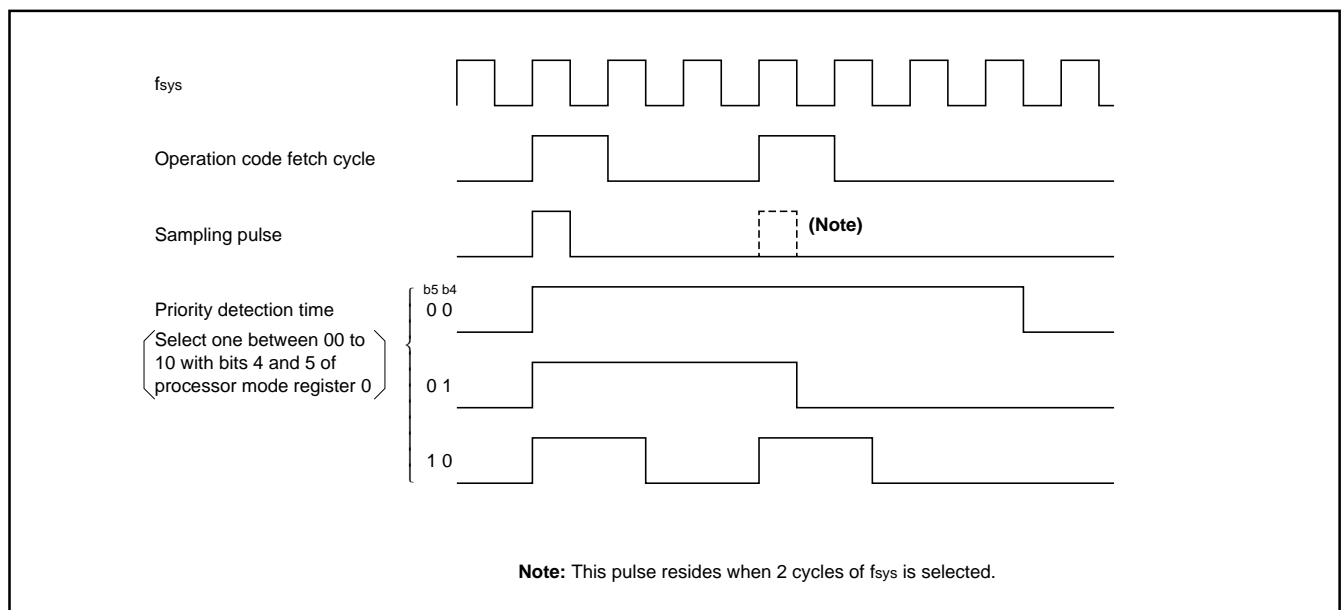


Fig. 16 Interrupt priority detection time

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

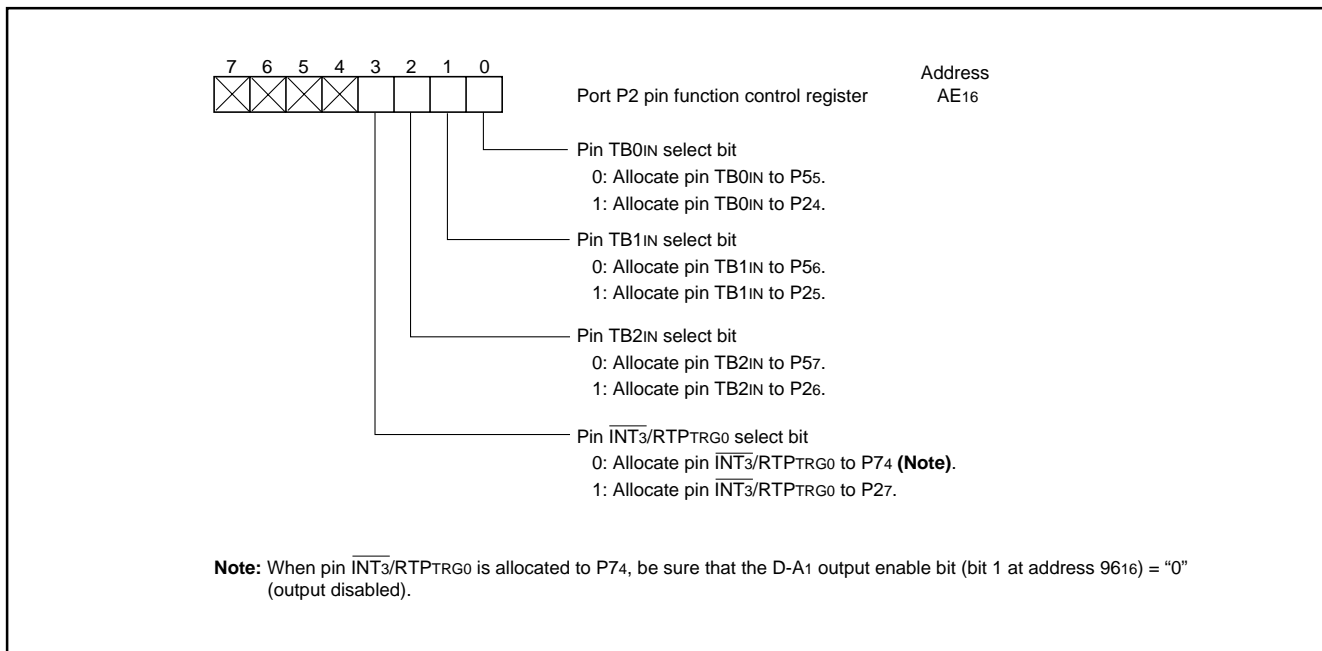


Fig. 17 Bit configuration of port P2 pin function control register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**TIMER**

There are eight 16-bit timers. They are divided by type into timer A (10) and timer B (3).

The timer I/O pins are multiplexed with I/O pins for ports P2, 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 18 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 9). Each of these modes is described below.

Figure 19 shows the bit configuration of the timer A clock division select register. Timers A0 to A9 use the count source which has been

selected by bits 0 and 1 of this register.

Each of timers A3, and A5 through A8 has no its own I/O pins. Therefore, these timers are available only in the timer mode.

**(1) Timer mode [00]**

Figure 20 shows the bit configuration of the timer Ai mode register in the 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 7.)

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

Figure 21 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<sub>16</sub>. At the same time, the contents of the reload register is transferred to the counter and count is continued.

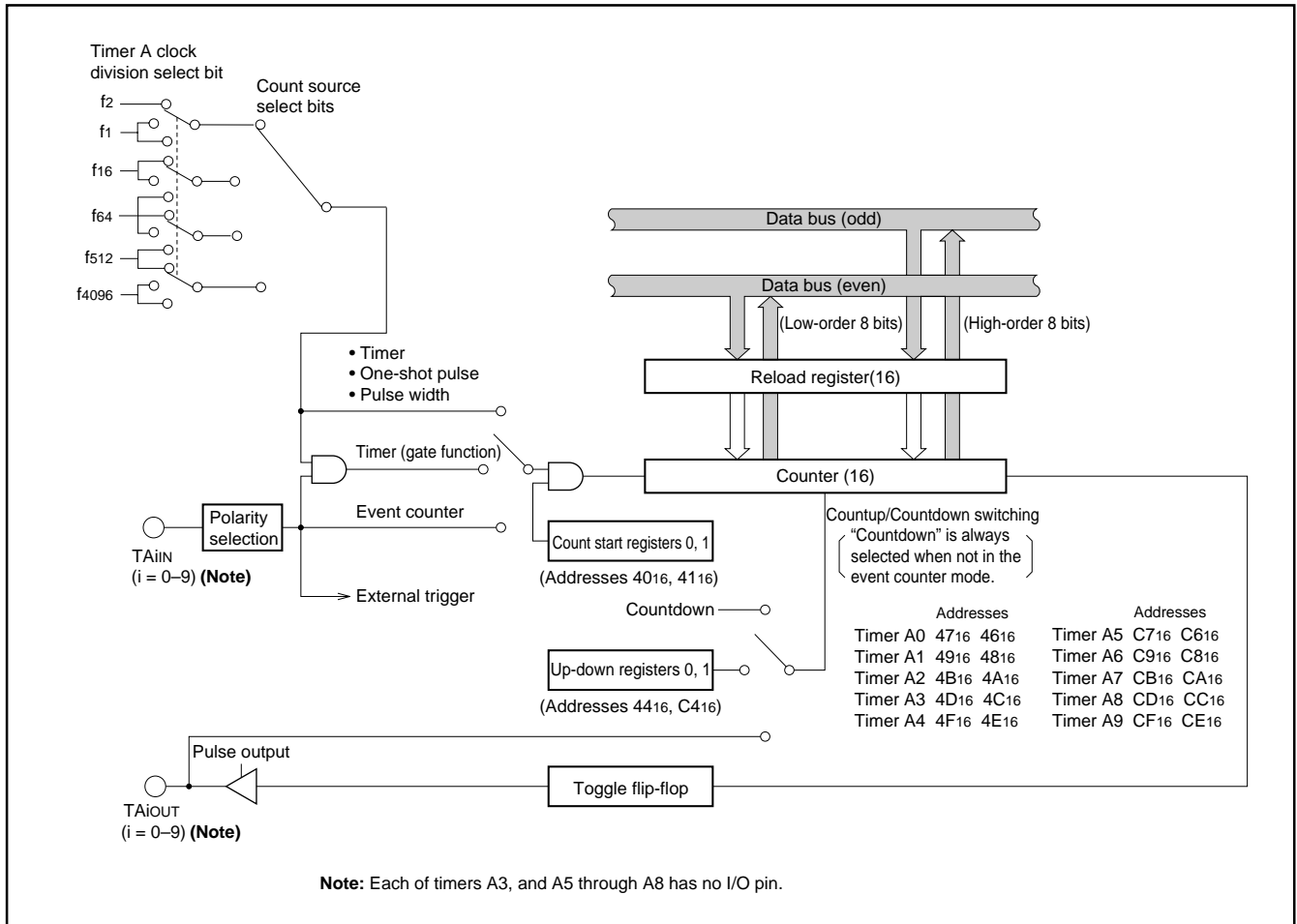


Fig. 18 Block diagram of timer A

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

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<sub>16</sub>. 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 22. 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).

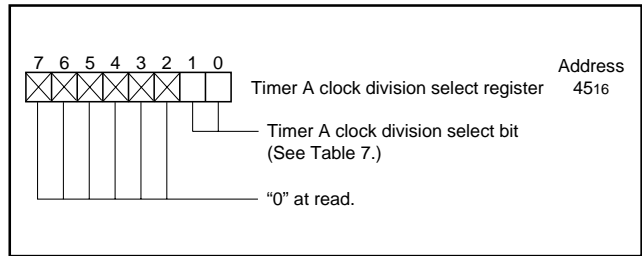


Fig. 19 Bit configuration of timer A clock division select register

Table 7. 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 <sub>16</sub> to 5A <sub>16</sub> ) (bits 7 and 6 at addresses D6 <sub>16</sub> to DA <sub>16</sub> )	Timer A clock division select bits (bits 1 and 0 at address 45 <sub>16</sub> )			
	00	01	10	11
0 0	f <sub>2</sub>	f <sub>1</sub>	f <sub>1</sub>	Do not select.
0 1	f <sub>16</sub>	f <sub>16</sub>	f <sub>64</sub>	
1 0	f <sub>64</sub>	f <sub>64</sub>	f <sub>512</sub>	
1 1	f <sub>512</sub>	f <sub>4096</sub>	f <sub>4096</sub>	

**Note:** Timers A0 to A9 use the same clock, which is selected by the timer A clock division select bits.

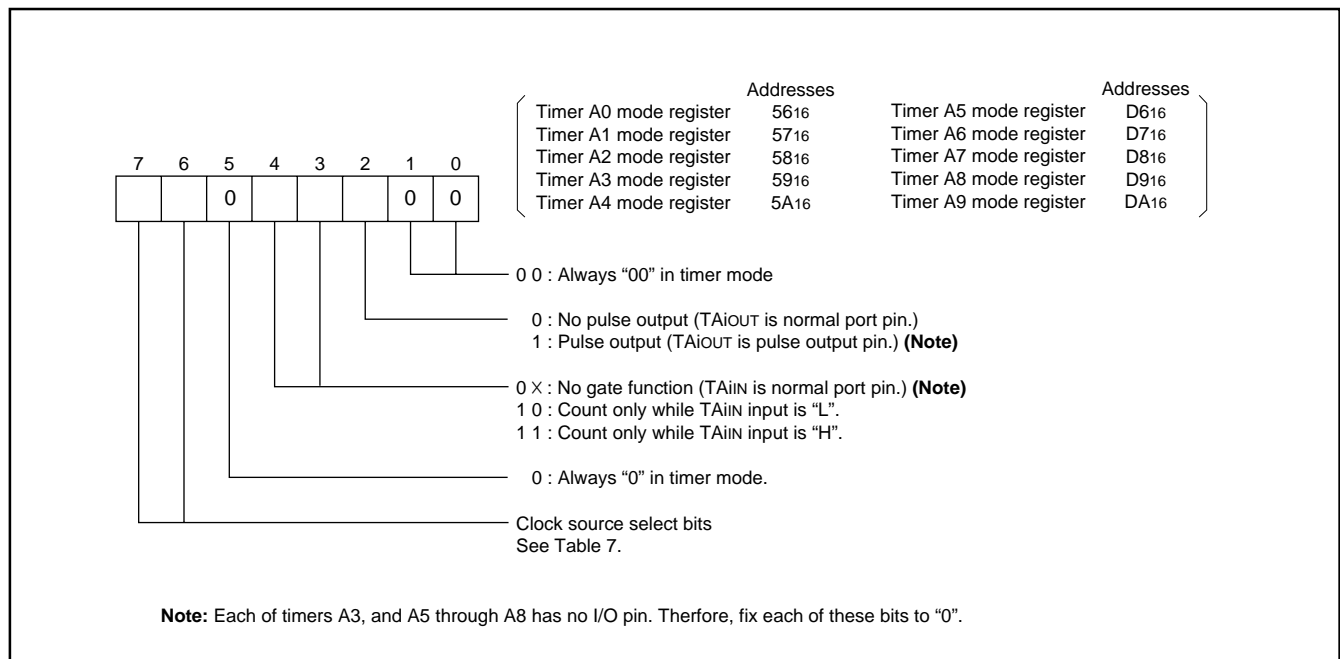


Fig. 20 Bit configuration of timer Ai mode register in timer mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

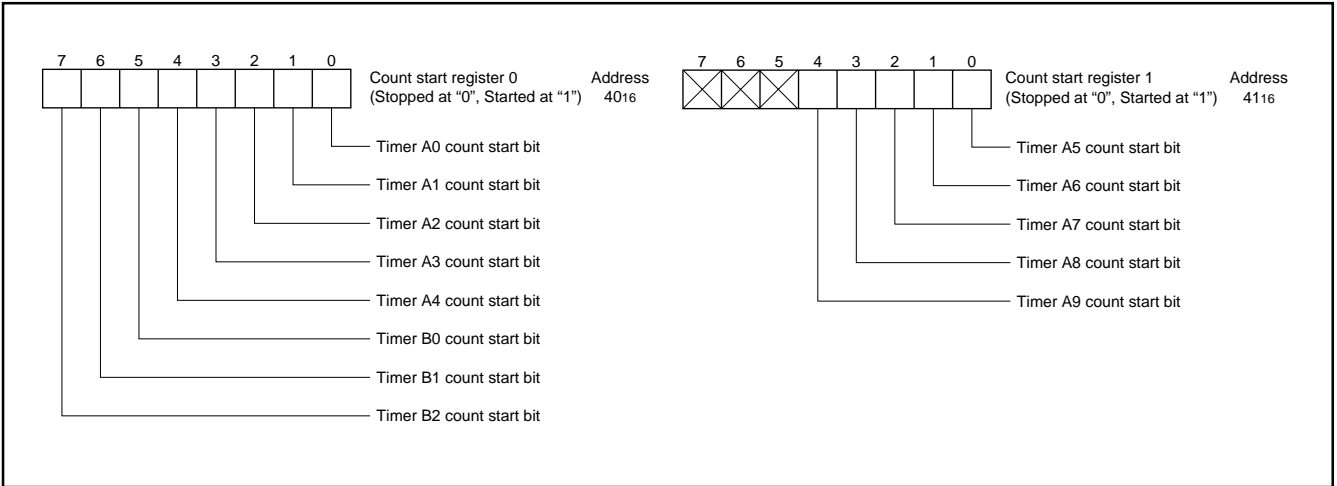


Fig. 21 Bit configuration of count start register

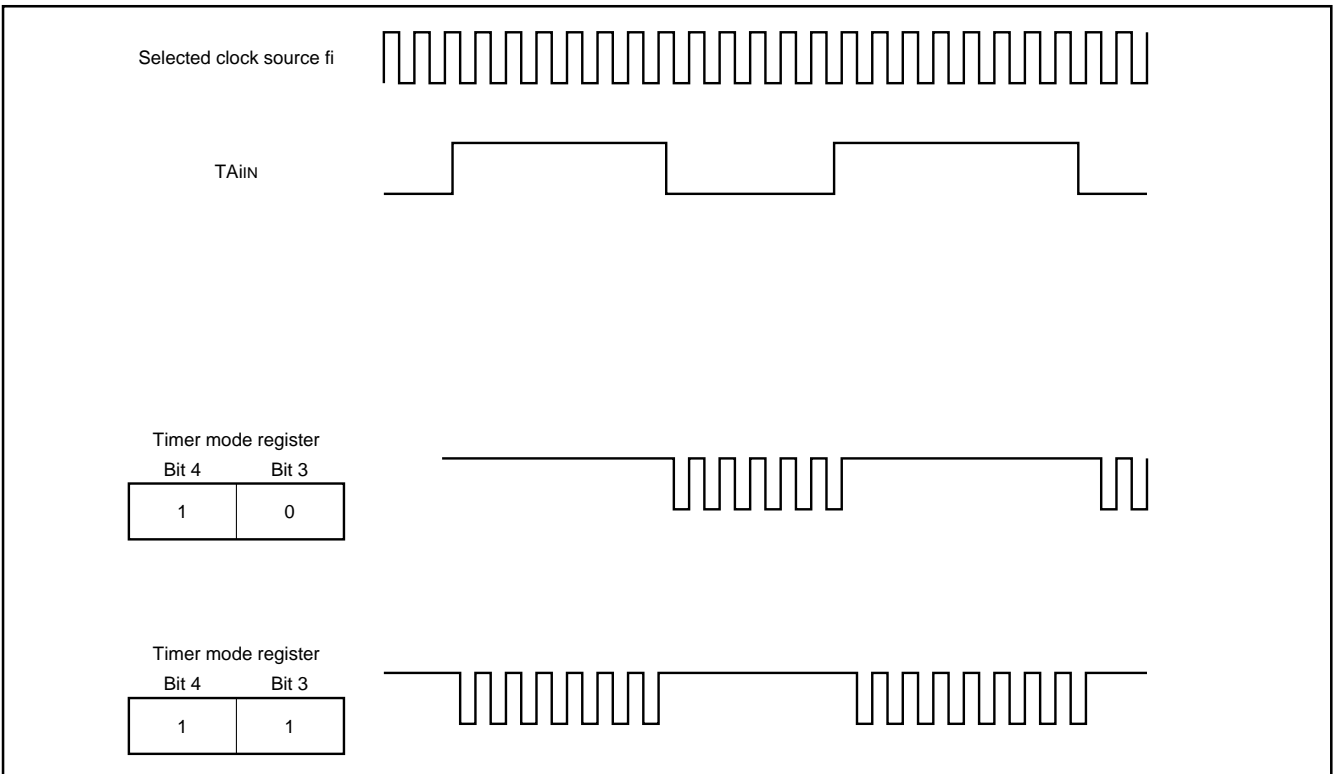


Fig. 22 Count waveform when gate function is available

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**(2) Event counter mode [01]**

Figure 23 shows the bit configuration of the timer Ai mode register in the 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 21 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 24 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<sub>16</sub> (decrement count) or FFFF<sub>16</sub> (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<sub>16</sub> (decrement

count) or FFFF<sub>16</sub> (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.

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, A4, or A9. There are two types of two-phase pulse processing operations. One uses timer A2, and the other uses timers A4 and A9. In both processing operations, two pulses described above are input to the TAJOUT (j = 2, 4, 9) pin and TAJIN pin respectively.

When timer A2 is used, as shown in Figure 25, the count is incremented when a rising edge is input to the TA2IN pin after the level of TA2OUT pin changes from "L" to "H", and when the falling edge is input, the count is decremented.

For timers A4 and A9, as shown in Figure 26, when a phase-related pulse with a rising edge input to the TAKIN pin is input after the level of TAKOUT (k = 4, 9) pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the TAKOUT pin and TAKIN pin.

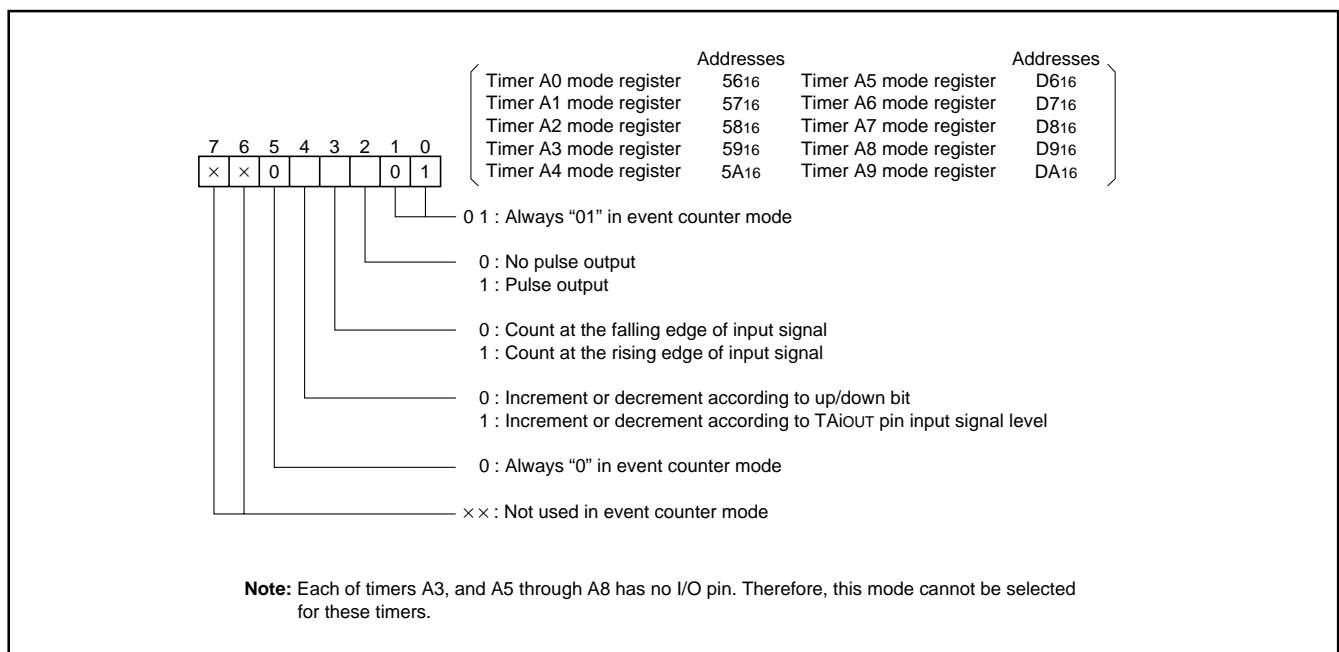


Fig. 23 Bit configuration of timer Ai mode register in event counter mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

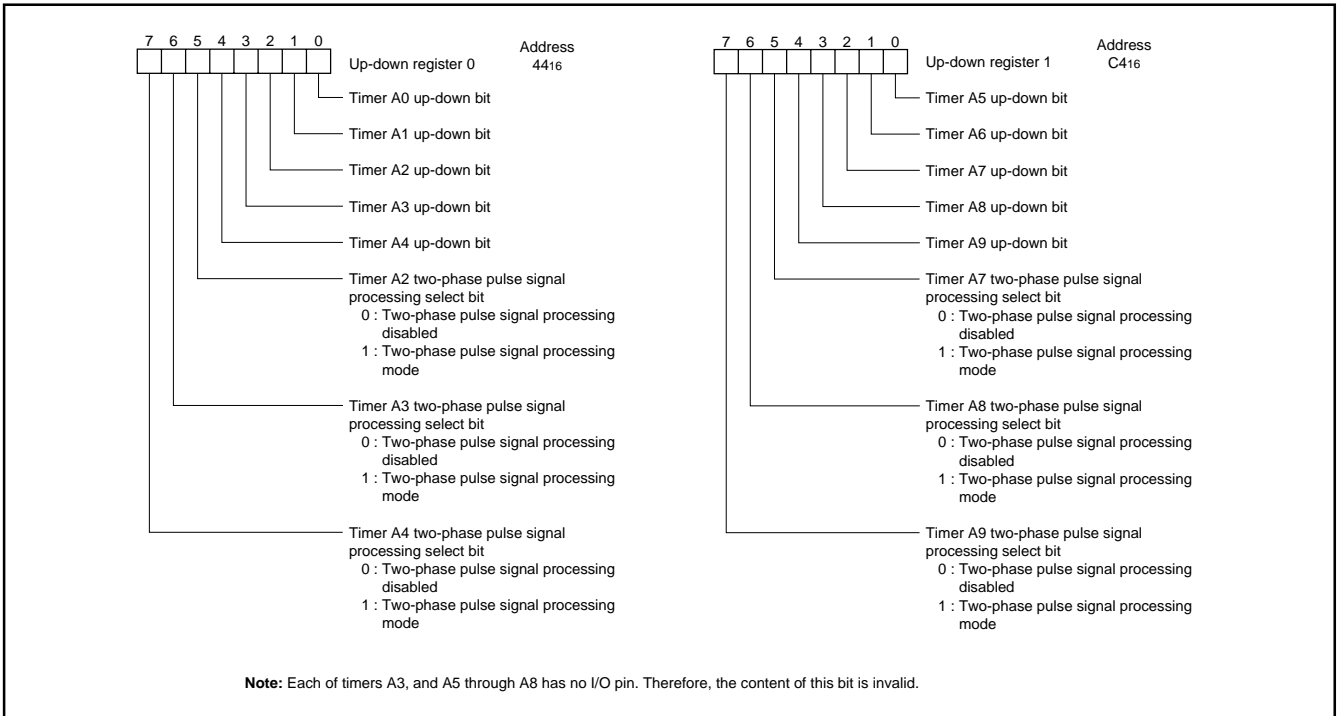


Fig. 24 Bit configuration of up-down register

When a phase-related pulse with a falling edge input to the T<sub>Ak</sub>OUT pin is input after the level of T<sub>Ak</sub>IN pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the T<sub>Ak</sub>IN pin and T<sub>Ak</sub>OUT pin. When performing this two-phase pulse signal processing, timer A<sub>j</sub> 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 27.) Note that bits 5 and 7 of the up-down register 0 (address 4416) are the two-phase pulse signal processing select bits for timers A2 and A4 respectively. Also, bit 7 of the up-down register 1 (address C416) is the two-phase pulse signal processing select bit for timer A9. 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.

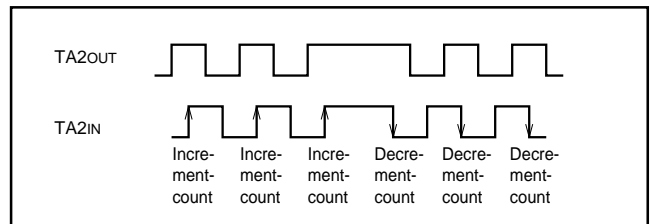


Fig. 25 Two-phase pulse processing operation of timer A2

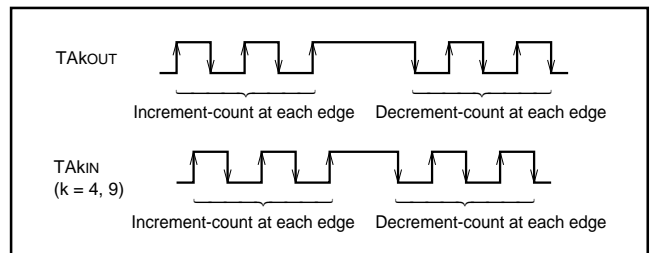


Fig. 26 Two-phase pulse processing operation of timers A4 and A9

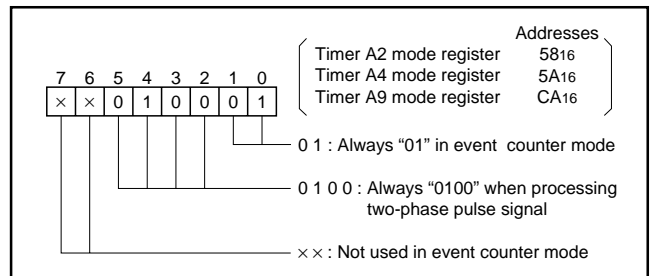


Fig. 27 Bit configuration of timer A<sub>j</sub> mode register when performing two-phase pulse signal processing in event counter mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**(3) One-shot pulse mode [10]**

Figure 28 shows the bit configuration of the timer Ai mode register in the 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 29 shows the bit configuration of the one-shot start register. As shown in Figure 30, 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 7.)

If the contents of the counter is not 0000<sub>16</sub>, the TAIOUT pin goes "H" when a trigger signal is received. The count direction is decrement. When the counter reaches 0001<sub>16</sub>, 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 31, 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.

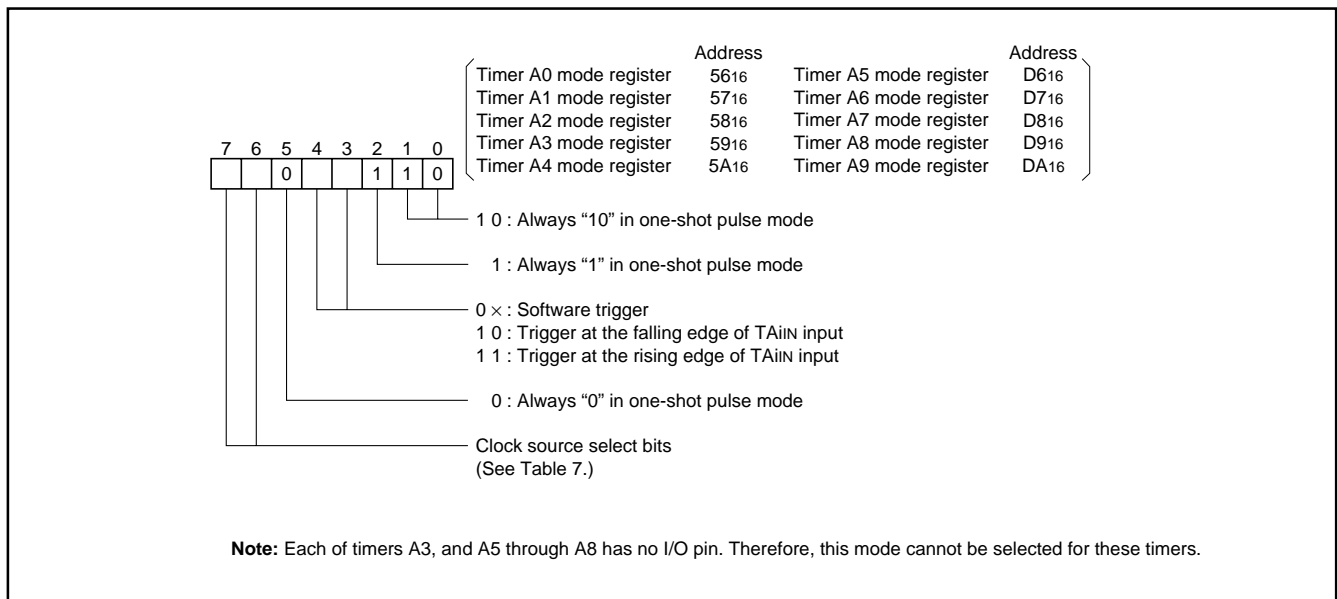


Fig. 28 Bit configuration of timer Ai mode register in one-shot pulse mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

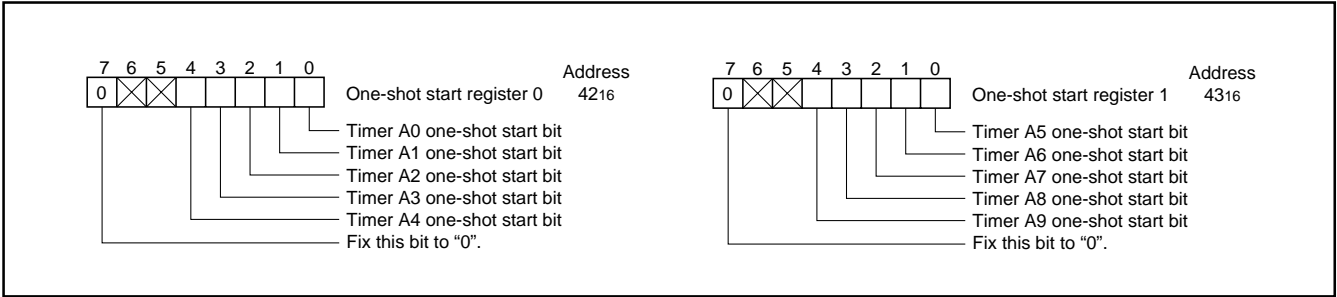


Fig. 29 Bit configuration of one-shot start register

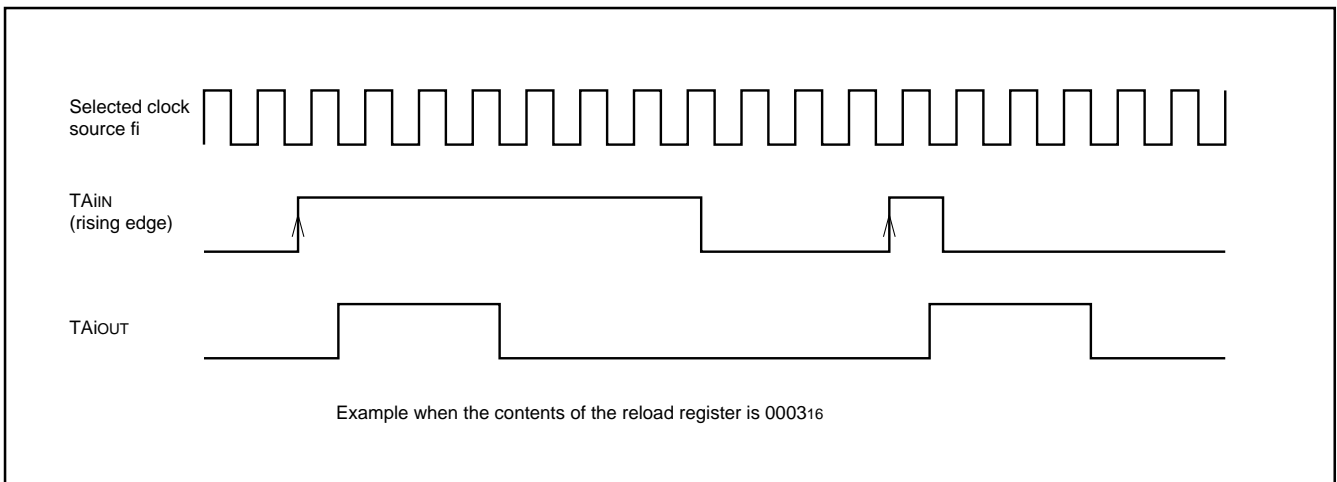


Fig. 30 Pulse output example when external rising edge is selected

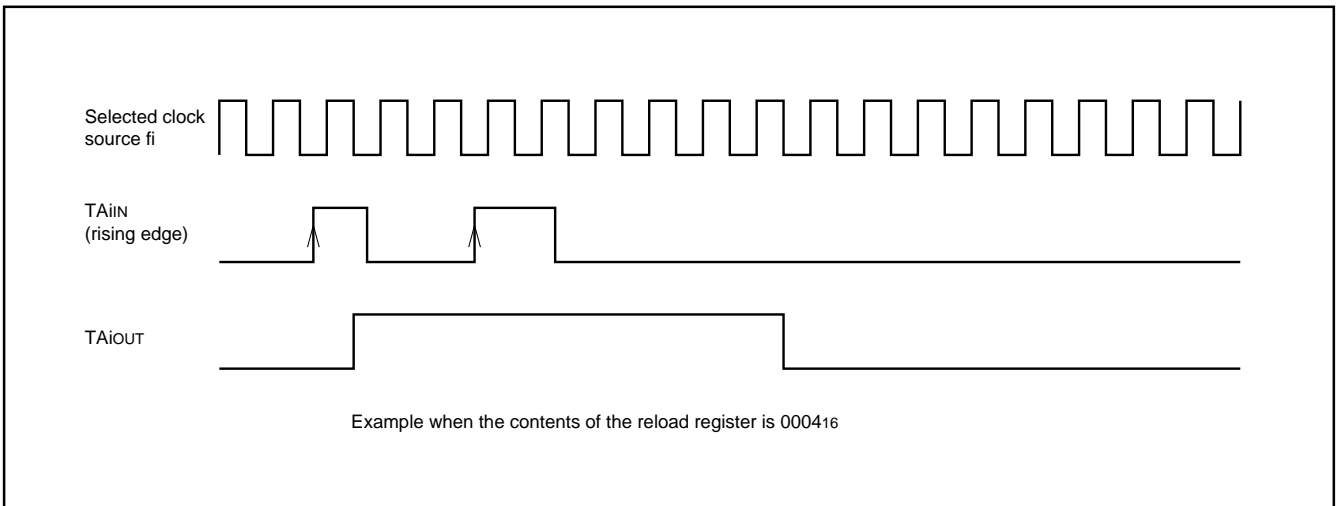


Fig. 31 Example when trigger is re-issued during pulse output

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**(4) Pulse width modulation mode [11]**

Figure 32 shows the bit configuration of the timer Ai mode register in the 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 33 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 7.) A pulse is generated when the counter reaches 0000<sub>16</sub> as shown in Figure 34. At the same time, the contents of the reload register is transferred to the counter and count is continued.

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

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

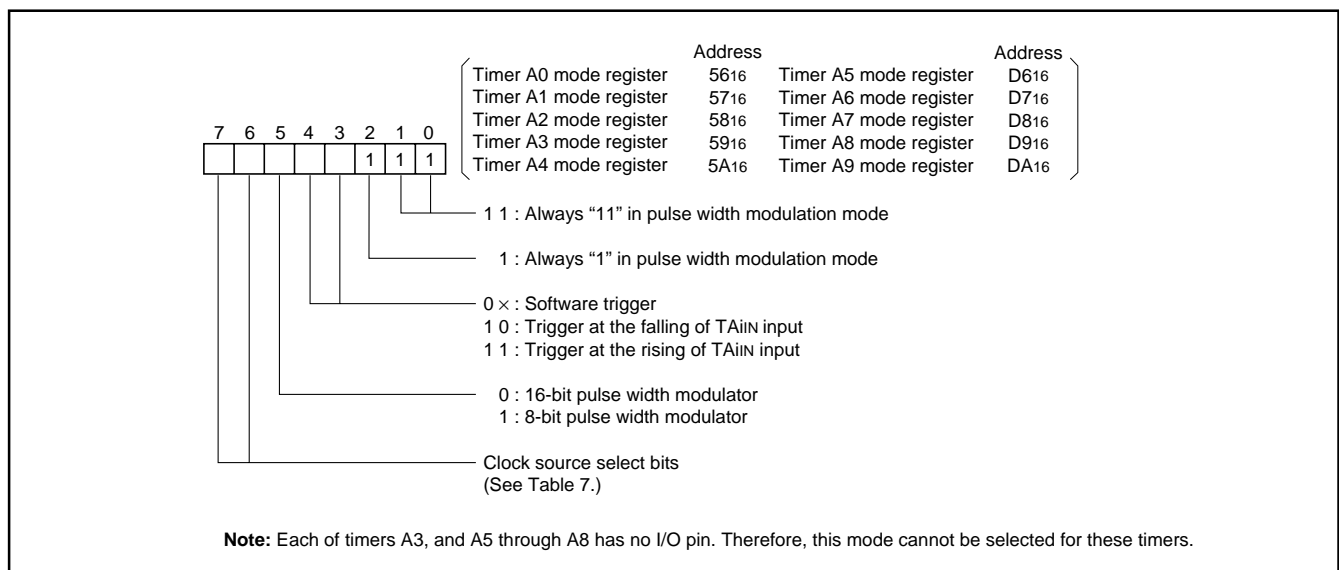


Fig. 32 Bit configuration of timer Ai mode register in pulse width modulation mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

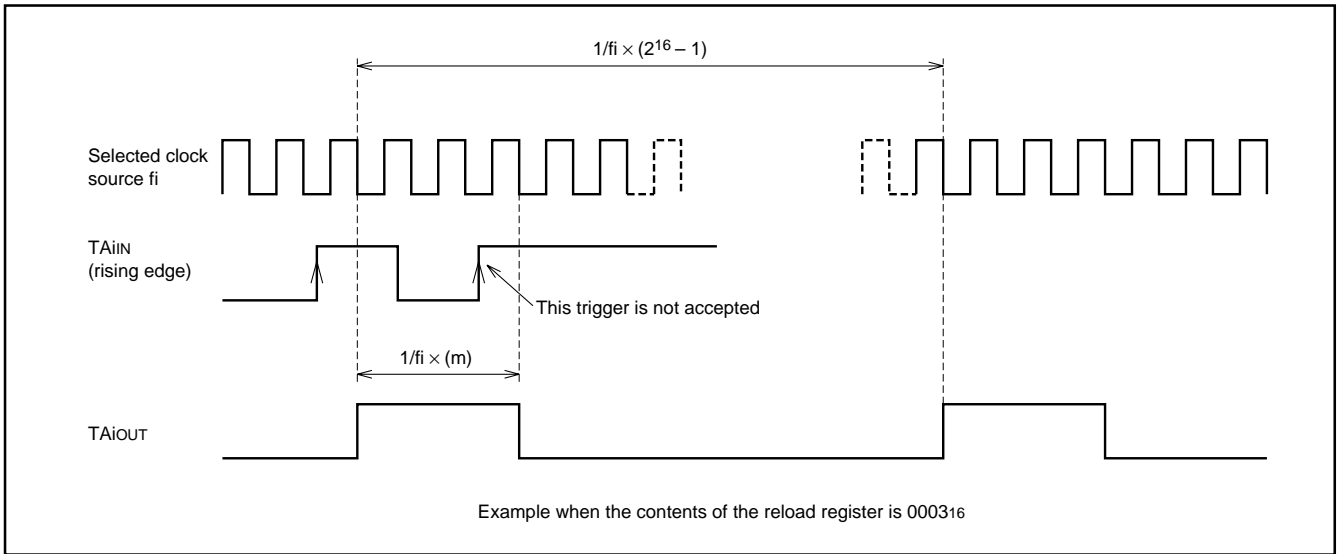


Fig. 33 16-bit length pulse width modulator output pulse example

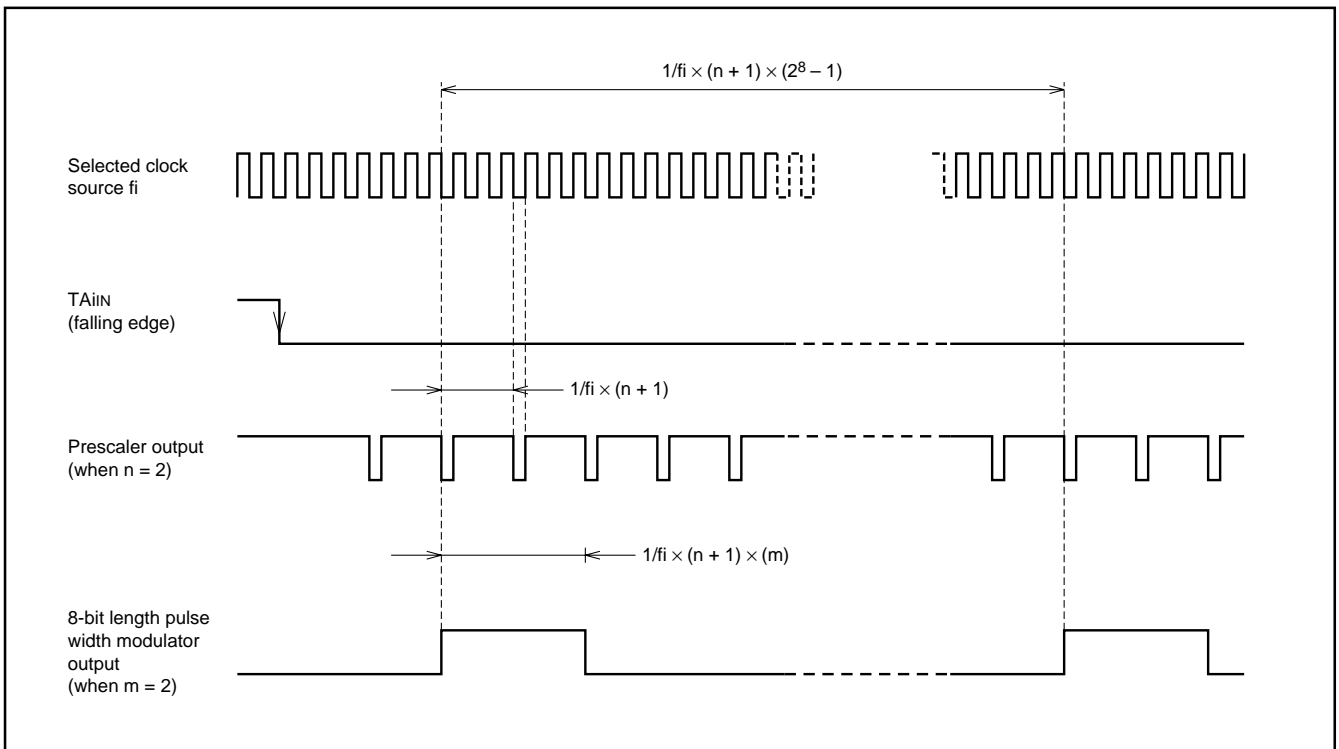


Fig. 34 8-bit length pulse width modulator output pulse example

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**TIMER B**

Figure 35 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 36 shows the bit configuration of the timer Bi mode register in the 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 21, the timer Bi's count start bits are allocated at

the same address to which some of the timer Ai's count start bits are allocated. (In other words, they are allocated in the count start register 0.) 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<sub>16</sub>. 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.

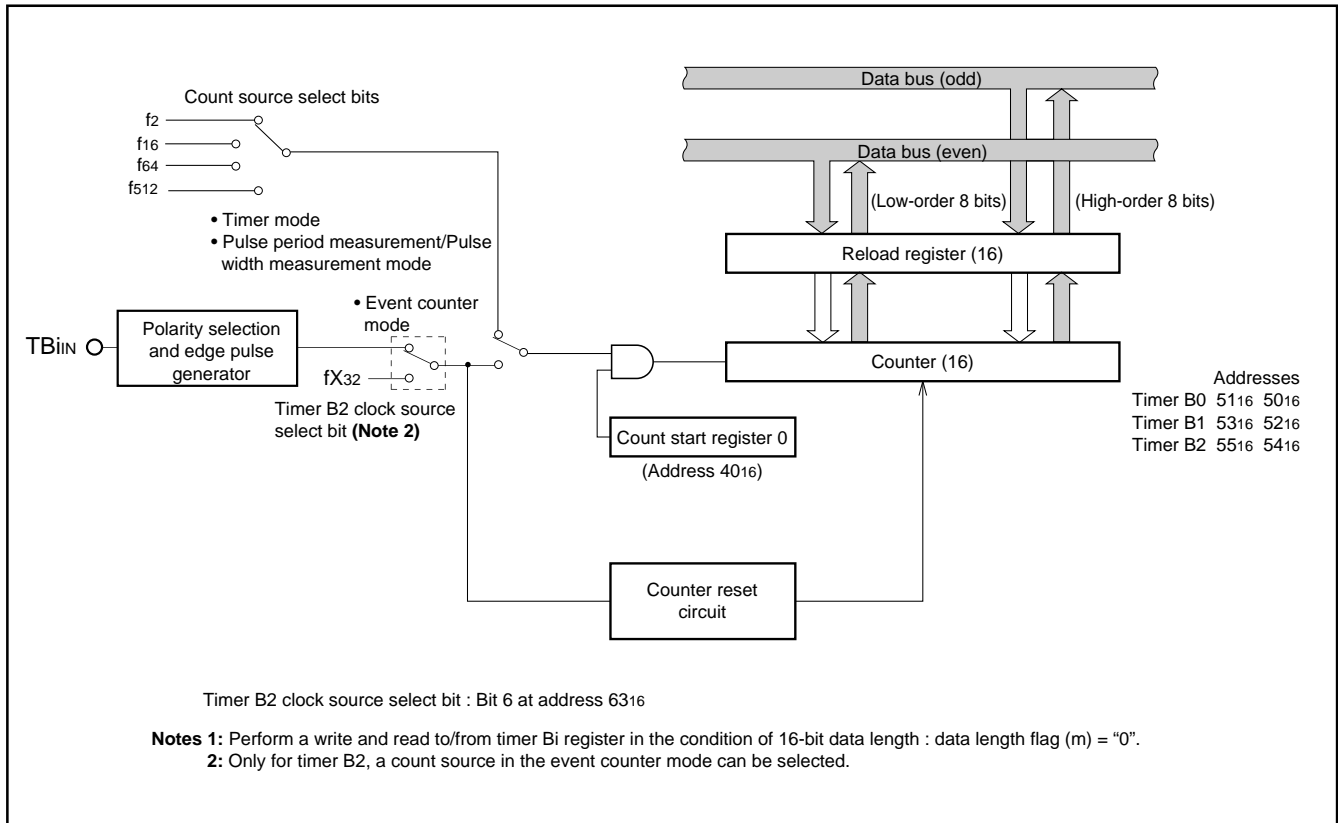


Fig. 35 Block diagram of timer B

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**(2) Event counter mode [01]**

Figure 37 shows the bit configuration of the timer Bi mode register in the 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.

Only for timer B2, when the timer B2 clock source select bit of the particular function select register 1 (bit 6 at address 63<sub>16</sub>) = "1" in the event counter mode, fX<sub>32</sub> (**Note**) can be selected. (When this bit is "0", an input signal from pin TB2IN becomes the count source as described above.) For the bit configuration of the particular function select register 1, refer to the section on the standby function.

Also, the pin position where pin TBiIN is to be allocated can be selected by the pin TBiIN select bit (bit 0, 1, or 2 at address AE<sub>16</sub>; the port P2 pin function control register).

**Note:** fX<sub>32</sub> = f(X<sub>IN</sub>)/32

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

Figure 38 shows the bit configuration of the timer Bi mode register in the pulse period measurement/pulse width measurement mode.

In the 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 the 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 39, 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 40.

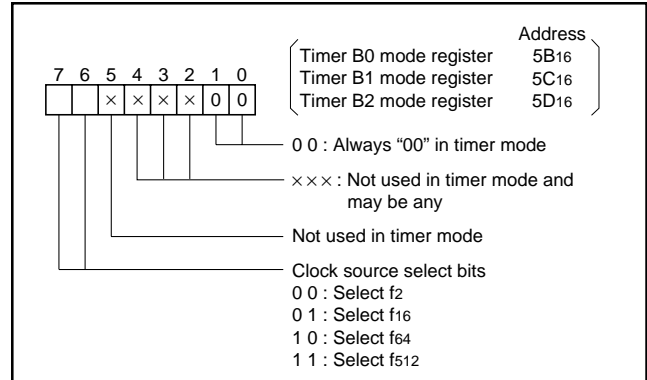


Fig. 36 Bit configuration of timer Bi mode register in timer mode

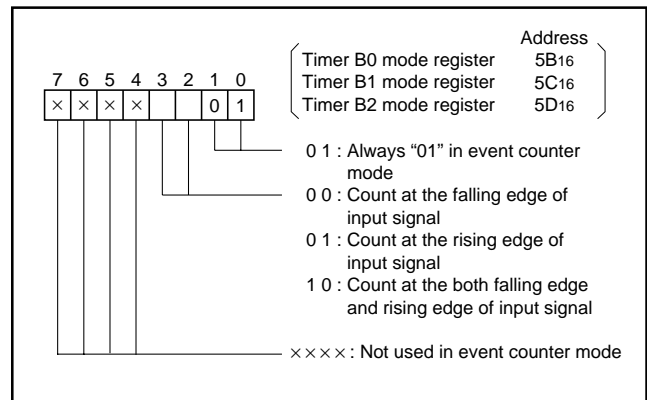


Fig. 37 Bit configuration of timer Bi mode register in event counter mode

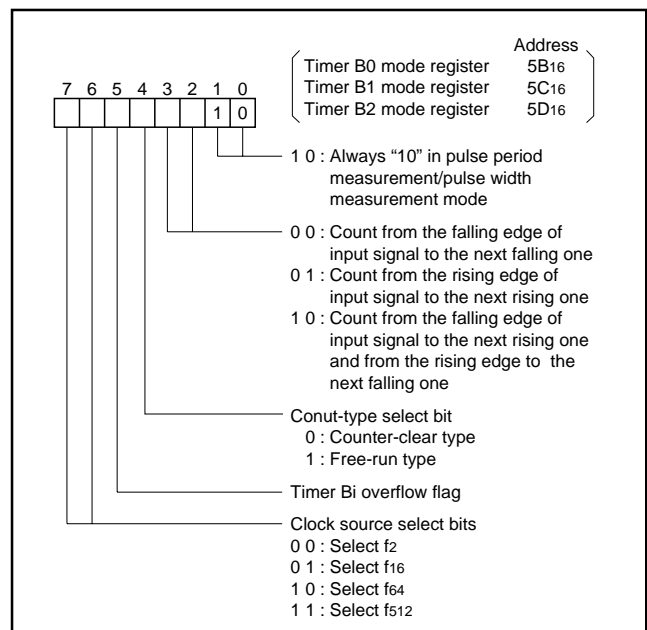


Fig. 38 Bit configuration of timer Bi mode register in pulse period measurement/pulse width measurement mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

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<sub>16</sub>, 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.  
 In these modes, the count type can be selected by the count-type select bit (bit 4 of the timer Bi mode register). When this bit = "1", the free-run type is selected; in this case, even when a valid edge is input to pin TBIiN, the contents of the counter will not be cleared to "0000<sub>16</sub>", and counting will continue. However, when a valid edge is input, an interrupt-requesting signal will be generated.

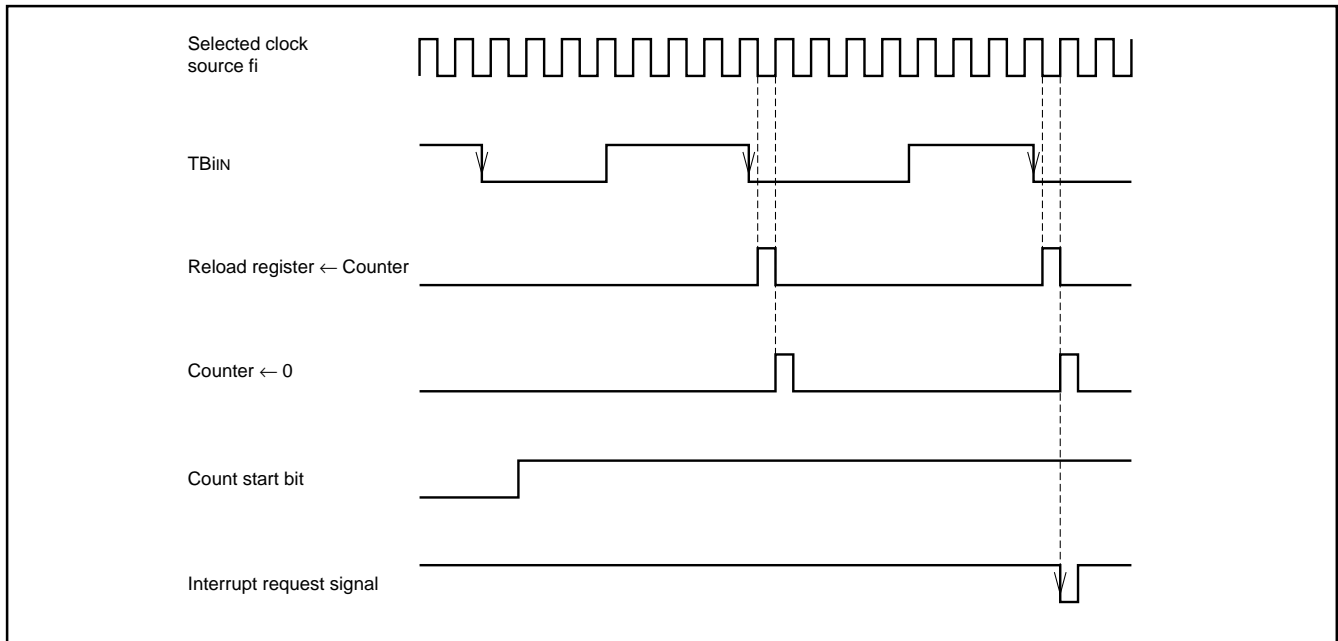


Fig. 39 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

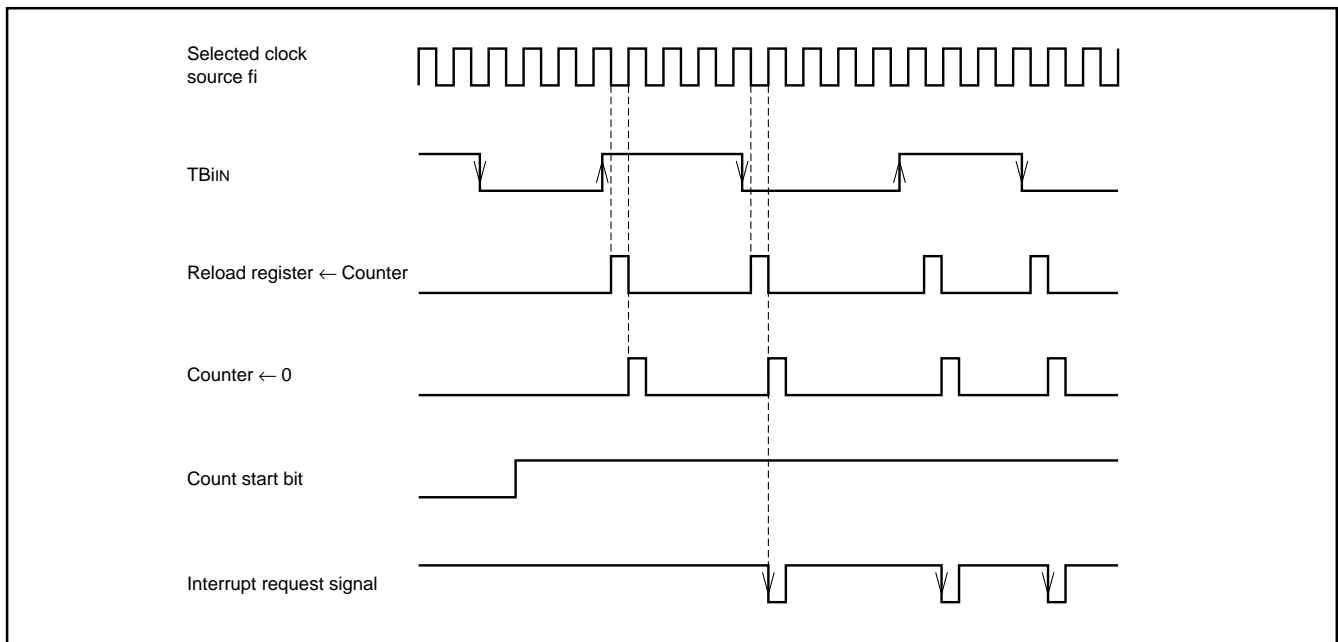


Fig. 40 Pulse width measurement mode operation

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**TIMER FUNCTION FOR MOTOR CONTROL**

Three-phase motor drive waveform or pulse motor drive waveform can be output by using plural internal timers As. These modes are explained below.

**Three-phase motor drive waveform output mode (three-phase waveform mode)**

Three-phase waveform mode using timers A0, A1, A2 and A3 is selected by setting the waveform output select bits of the waveform output mode register (bits 2 through 0 at address A616, Figure 41) to "100".

There are two types of the three-phase waveform mode: three-phase mode 0 and three-phase mode 1. Bit 4 of the waveform output mode register selects either mode. In the three-phase waveform mode, set the corresponding timer mode registers of timers A0, A1, and A2 to select the one-shot pulse mode with the rising edge of an external trigger valid; set the timer mode register of timer A3 to select the timer mode.

Figure 43 shows the block diagram in the three-phase waveform mode. The three-phase waveform mode outputs six waveforms, positive waveforms (U, V, W phases) and negative waveforms ( $\bar{U}$ ,  $\bar{V}$ ,  $\bar{W}$  phases), from the respective ports with "L" level active.

Timer A2 controls U and  $\bar{U}$  phases; timer A1 does V and  $\bar{V}$  phases and timer A0 does W and  $\bar{W}$  phases. Timer A3 controls these one-shot pulses' periods of timers A2, A1 and A0.

In the waveform output, a short circuit prevention time can be set to prevent "L" level of positive waveform outputs (U, V, W phases) from overlapping with "L" level of their negative waveform outputs ( $\bar{U}$ ,  $\bar{V}$ ,  $\bar{W}$  phases). The short circuit prevention time can be set with three 8-bit dead-time timers, sharing one reload register. The dead-time timer operates as a one-shot timer. It's start trigger is selected from the following two types: both the rising and falling edges of timers A0 to A2's one-shot pulses or their falling edges. Additionally, bit 6 of the waveform output mode register (address A616) controls this selection. When that bit is "0", both the rising and falling edges are selected; when that bit is "1", the falling edges are selected.

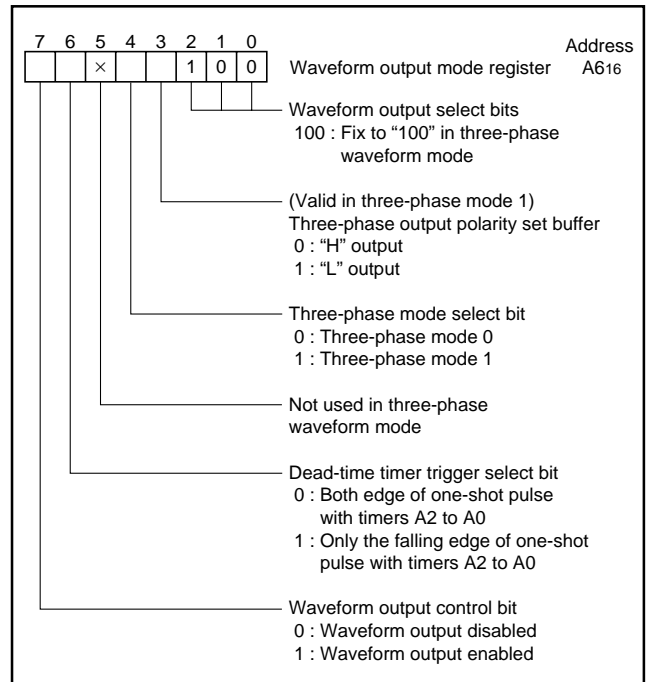


Fig. 41 Bit configuration of waveform output mode register in three-phase waveform mode

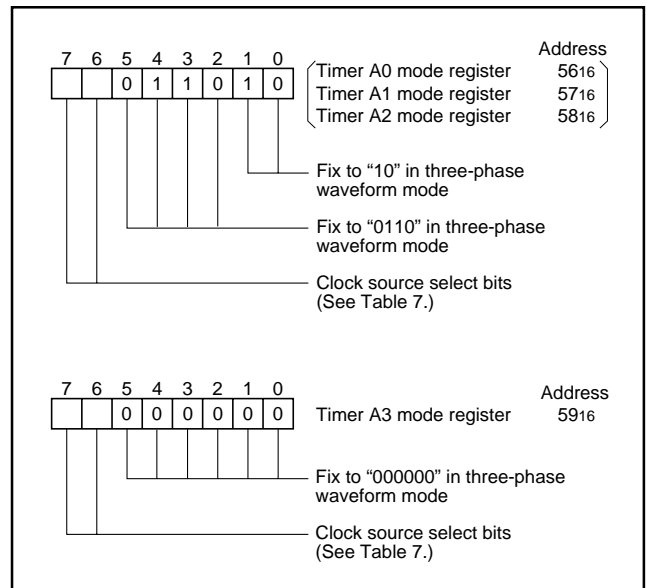


Fig. 42 Bit configuration of timer A0, A1, A2, mode register and timer A3 mode register in three-phase waveform mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

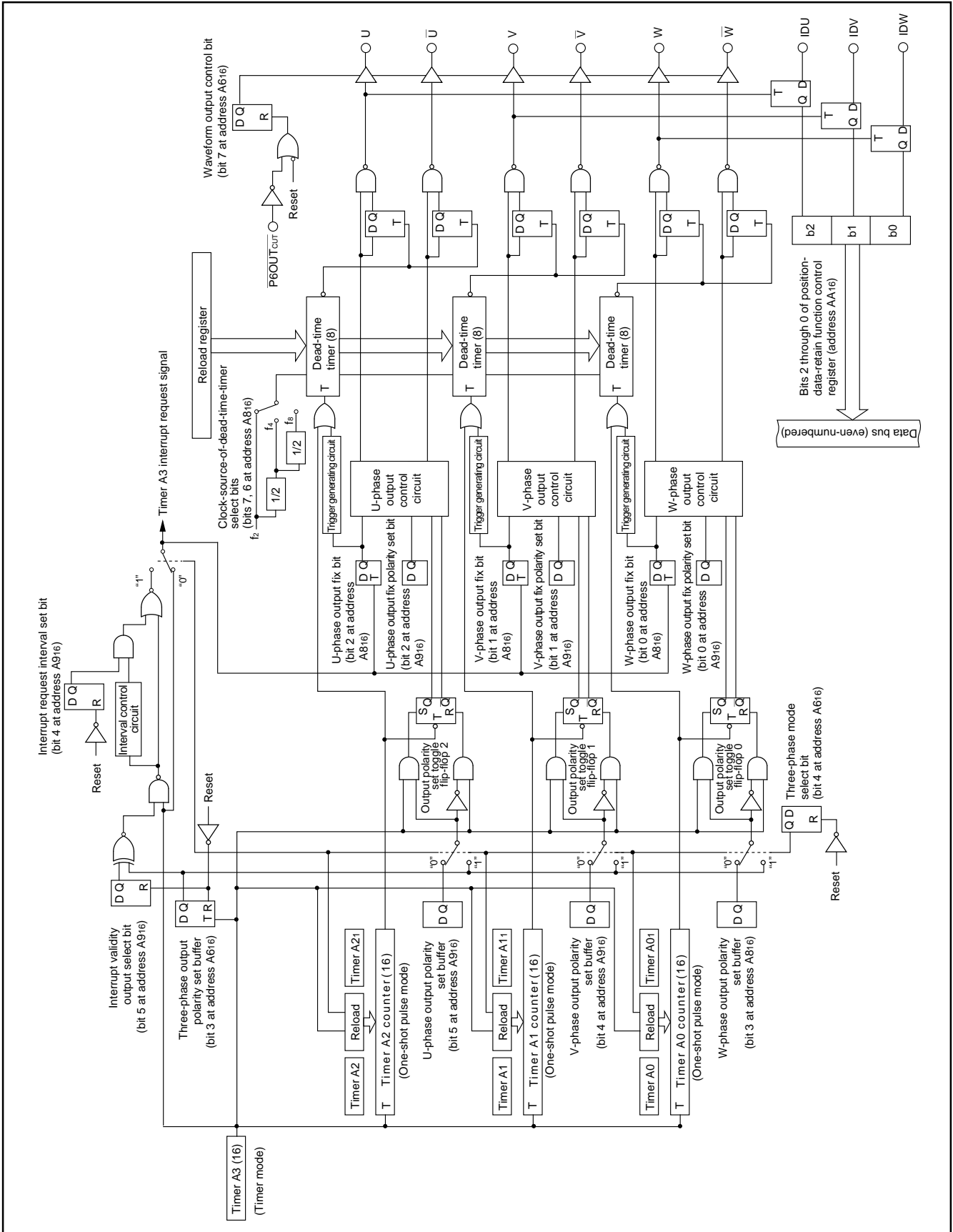


Fig. 43 Block diagram in three-phase waveform mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

When writing data to the dead-time timer (address A716), the data is written to the reload register shared by three dead-time timers. When the dead-time timers catch the start trigger from the respective timers, the reload register contents are transferred to its counter and the dead-time timer decrements with the clock source selected by bits 6 and 7 of three-phase output data register 0 (address A816). Additionally, this timer can accept another trigger before completion of the preceding trigger operation. In this case, after transferring the reload register contents to the dead-time timer at acceptance of the trigger, the value is decremented.

The dead-time timer operates as a one-shot pulse timer. Accordingly, this timer starts pulse output when the trigger is caught, and finishes pulse output and stops operation when its contents become "0016", and waits next trigger.

In the three-phase waveform mode, setting bit 7 of the waveform output mode register (address A616) to "1" makes positive waveforms (U, V, W phases) and their negative waveforms ( $\bar{U}$ ,  $\bar{V}$ ,  $\bar{W}$  phases) output from the respective ports. When this bit is "0", their ports are floating. This bit is cleared to "0" by inputting a falling edge to pin  $\bar{P6OUTCUT}$ , by reset, or by executing instructions. Additionally, setting bits 2 through 0 of the three-phase output data register 0 (address A816) to "1" makes the corresponding waveform outputs fixed. Whether the outputs are fixed to "H" or "L" is selected by bits 2 through 0 of the three-phase output data register 1 (address A916). Clearing these bits to "0" makes the corresponding waveform outputs fixed to "H"; setting these bits to "1" makes the outputs fixed to "L".

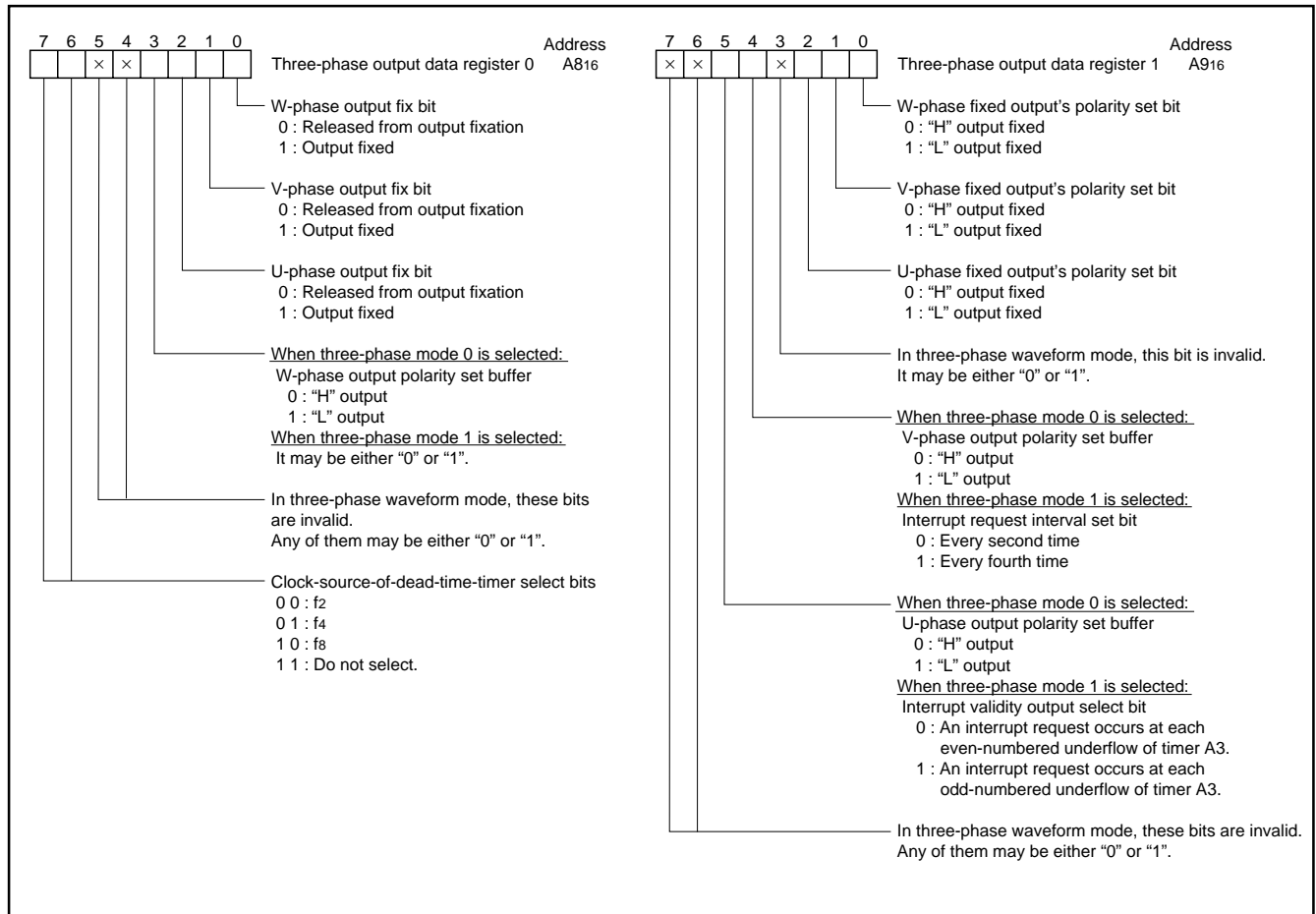


Fig. 44 Bit configuration of three-phase output data registers 1 and 0 in three-phase waveform mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**Three-phase mode 0**

In selecting three-phase waveform mode, three-phase mode 0 is selected by setting bit 4 of the waveform output mode register (address A616) to "0".

The output polarity of three-phase waveform depends on the output polarity set toggle flip-flop. The positive waveform of the three-phase waveform is "H" output when the toggle flip-flop is "0"; it is "L" output when the toggle flip-flop is "1". (Three-phase waveform is output as a negative waveform.)

Each output polarity set toggle flip-flop has the output polarity set buffer shown in Figure 44. When the timer A3's counter contents become 000016, the contents of output polarity set buffer are set into the output polarity set toggle flip-flop. After that, the polarity of the contents of output polarity set toggle flip-flop are reversed each time completion of one-shot pulse of timer (timers A2 to A0) corresponding to each phase.

Figure 45 shows an example of U-phase waveform and the output operation is explained. Three-phase mode 0 becomes valid when writing "0" to the U-phase output polarity set buffer (bit 5 at address A916) and actuating the timer A3. When the counter of timer A3 becomes 000016, the timer A3 interrupt request signal occurs and the timer A2 simultaneously starts one-shot pulse output. At this time, the contents of U-phase output polarity set buffer, "0" in this case, are set into the output polarity set toggle flip-flop 2.

When the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 is reversed from "0" to "1". Simultaneously, the one-shot pulse of the 8-bit dead-time timer is output for ensuring time not to overlap "L" levels of U phase waveform and its negative  $\bar{U}$  phase waveform.

The U-phase waveform output keeps "H" level from the start until the one-shot pulse output of the dead-time timer is completed, even if the contents of output polarity set toggle flip-flop 2 are reversed from "0" to "1" owing to the timer A2's one-shot pulse output. When the one-shot pulse output of the dead-time timer is completed, "1" of output polarity set toggle flip-flop 2 which has been reversed becomes valid and the U phase waveform changes to "L" level.

Then, write "1" to the U-phase output polarity set buffer (bit 5 at address A916) before the counter of timer A3 becomes 000016.

After that, when the counter of timer A3 becomes 000016, the timer A2 starts one-shot pulse output. Simultaneously, the contents of U-phase output polarity set buffer, "1" in this case, are set into the output polarity set toggle flip-flop 2 and the U phase waveform remains "L" level.

When the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 is reversed from "1" to "0". Simultaneously, the one-shot pulse output of the dead-time timer starts.

When the contents of output polarity set toggle flip-flop 2 are reversed from "1" to "0", the U-phase waveform changes its output level from "L" to "H" without waiting for completion of the one-shot pulse output of the dead-time timer.

U-phase waveform is generated by repeating the operation above. The way to generate  $\bar{U}$ -phase waveform, which is the negative phase of U-phase, is the same as that for U-phase waveform except that the contents of output polarity set toggle flip-flop 2 are treated as the reversed signal from the case of U-phase waveform.

In this way, U-phase waveform and  $\bar{U}$ -phase waveform, having the negative phase of U-phase, are output from the pins so that their "L" levels do not overlap each other. The width of "L" level can be also modified by changing the value of timer A2 or A3.

V-, W-phase waveform and  $\bar{V}$ -,  $\bar{W}$ -phase waveform, having their negative phase, are similarly output according to the corresponding timer operation.

The explanation above is an example of three-phase waveform generating due to an triangular wave modulation. Three-phase waveform due to a saw-tooth-wave modulation can also be generated by fixing each beginning level of phases.

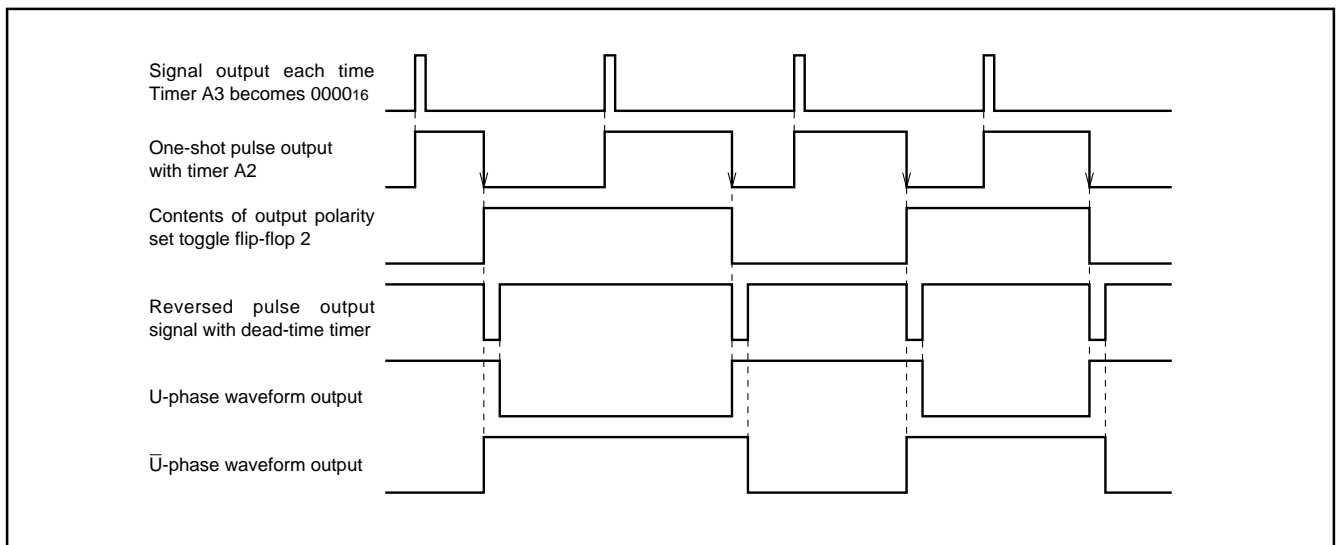


Fig. 45 U-phase waveform output example in three-phase mode 0 (triangular wave modulation)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

### Three-phase mode 1

In selecting three-phase waveform mode, three-phase mode 1 is selected by setting bit 4 of the waveform output mode register (address A6<sub>16</sub>) to "1".

In this mode, each of timers A0 to A2 can have two timer registers and the contents of those registers are alternately reloaded into the counter each time the counter of timer A3 becomes 0000<sub>16</sub>.

The interrupt request normally occurs when the counter of timer A3 becomes 0000<sub>16</sub>. However, this occurrence interval can be switched between "every second time" and "every fourth time." Bit 4 of the pulse output data register 1 (address A9<sub>16</sub>) selects that.

Additionally, an even-numbered or odd-numbered timer A3's underflow can be used as the occurrence factor of timer A3 interrupt request. Bit 5 of the three-phase output data register 1 (address A9<sub>16</sub>) selects that.

When the timer A3's counter contents become 0000<sub>16</sub>, the contents of three-phase output polarity set buffer are set into the output polarity set toggle flip-flop on which the output polarity of three-phase waveform depends. The contents of three-phase output polarity set buffer are reversed after that operation.

The polarity of the contents of output polarity set toggle flip-flop is reversed each time completion of one-shot pulse of timer (timers A2 to A0) corresponding to each phase.

Figure 46 shows an example of U-phase waveform and the output operation is explained.

Write "0" to the three-phase-output-polarity set buffer (bit 3 at address A6<sub>16</sub>). Clear the interrupt request interval set bit (bit 4 at address A9<sub>16</sub>) to "0" so that the timer A3 interrupt request occurs at every second time. Additionally, clear the interrupt validity output select bit (bit 5 at address A9<sub>16</sub>) so that the timer A3 interrupt request occurs at an each even-numbered underflow of timer A3.

After the procedure above, three-phase mode 1 starts operation when actuating timer A3.

When the counter of timer A3 becomes 0000<sub>16</sub>, the timer A3 interrupt request occurs and timer A2 simultaneously starts one-shot pulse output. At this time, the contents of three-phase output polarity set buffer, "0" in this case, are set into the output polarity set toggle flip-flop 2. The contents of three-phase output polarity set buffer are reversed from "0" to "1" after that operation.

When the timer A2 counter counts the value written into the timer A2 and the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 are reversed from "0" to "1". Simultaneously, the one-shot pulse of the 8-bit dead-time timer is output for ensuring time, so that "L" levels of U- and  $\bar{U}$ -phase waveforms do not overlap.

The U-phase waveform output keeps "H" level from the start until the one-shot pulse output of the dead-time timer is completed, even if the contents of output polarity set toggle flip-flop 2 are reversed from "0" to "1" owing to the timer A2's one-shot pulse output.

When the one-shot pulse output of the dead-time timer is completed, "1" of output polarity set toggle flip-flop 2 which has been reversed becomes valid and the U-phase waveform changes to "L" level.

Then, when the counter of timer A3 becomes 0000<sub>16</sub>, the timer A2 counter counts the value written into timer A2 and timer A2 starts one-shot pulse output. Simultaneously, the contents of three-phase output polarity set buffer are set into the output polarity set toggle flip-flop 2. However, the U-phase waveform remains "L" level, be-

cause the value is the same ("1").

The contents of three-phase output polarity set buffer are reversed from "1" to "0" after that operation.

When the one-shot pulse output of timer A2 is completed, the contents of output polarity set toggle flip-flop 2 is reversed from "1" to "0". Simultaneously, the one-shot pulse output of the dead-time timer starts.

When the contents of output polarity set toggle flip-flop 2 is reversed from "1" to "0", the U-phase waveform changes its output level from "L" to "H" without waiting for completion of the one-shot pulse output of the dead-time timer.

U-phase waveform is generated by repeating the operation above. The way to generate  $\bar{U}$ -phase waveform, which is the negative phase of U-phase, is the same as that for U-phase waveform except that the contents of output polarity set toggle flip-flop 2 is treated as the reversed signal from the case of U-phase waveform.

In this way, U-phase waveform and  $\bar{U}$ -phase waveform, having the negative phase of U-phase, are output from the pins so that their "L" levels do not overlap each other. The width of "L" level can be also modified by changing the value of timers A2, A21, or A3.

V-, W-phase waveform and  $\bar{V}$ -,  $\bar{W}$ -phase waveform, having their negative phase, are similarly output according to the corresponding timer operation.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

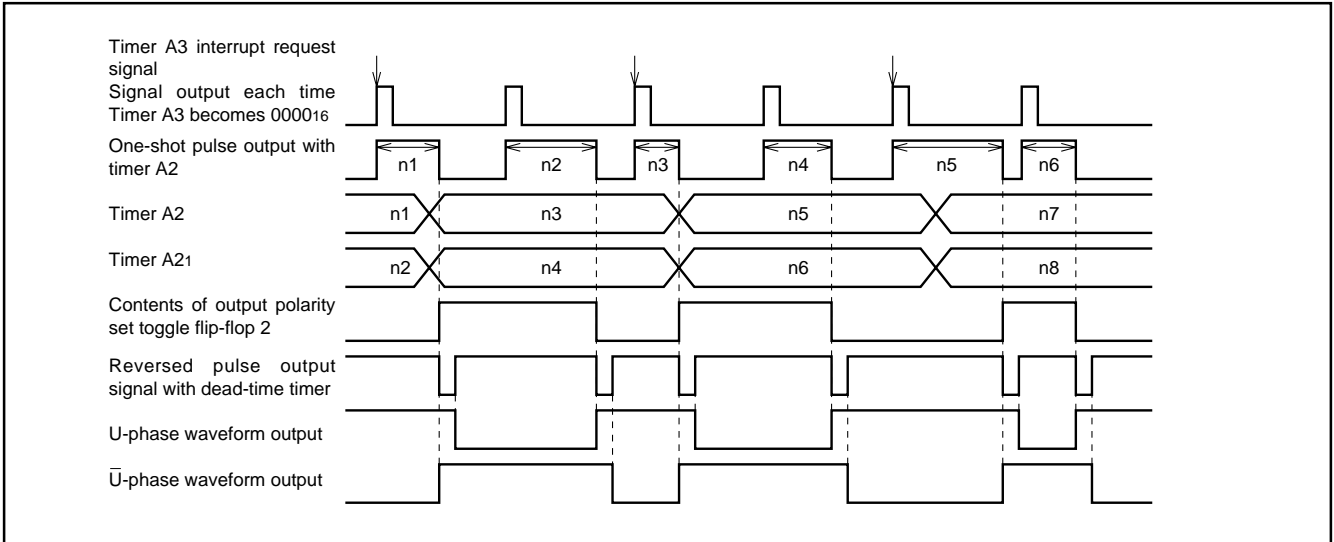


Fig. 46 U-phase waveform output example in three-phase mode 1 (triangular wave modulation)

**Position-data-retain function**

The three-phase waveform mode has the function to retain the input data of the corresponding pin (IDU, IDV, IDW) at an edge of a positive waveform (U, V, or W phase). Whether to retain the data at a falling edge or rising one is selected by bit 3 of the position-data-retain function control register (address AA16).

Retain data can be read out by bits 2 through 0 of the position-data-retain function control register (address AA16).

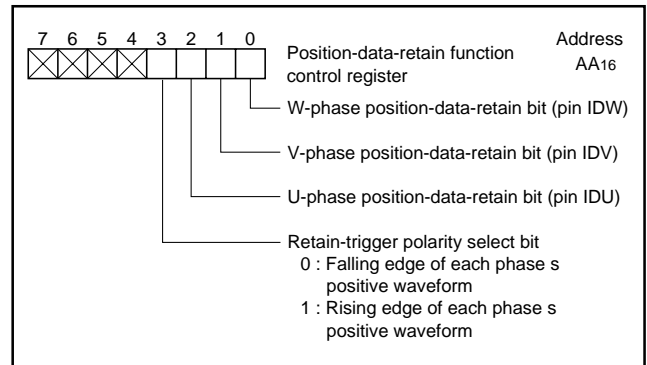


Fig. 47 Bit configuration of position-data-retain function control register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**PULSE OUTPUT PORT MODE**

Figure 48 shows the block diagram in the pulse output port mode. This mode has a 6-bit pulse output port. The waveform output select bits (bits 0 to 2) of waveform output mode register (address A616) select use of pulse output port. The 6-bit pulse output port can also be divided into 4 bits and 2 bits with the pulse output mode select bit (bit 3) of waveform output mode register (address A616); each of them can be individually controlled. Set timers A3 and A0 to the timer mode because they are used in the pulse output port mode. Figure 50 shows the bit configuration of timer A3 and A0 mode registers in the pulse output port mode. Timers A3 and A0 start count when setting the corresponding timer count start bit to "1", and they stop it when clearing that bit to "0".

Each bit using timer A0 as a trigger can also be controlled by an input trigger from pin RTPTRG0. This control is selected by the pulse output trigger select bits of the three-phase output data register 0 (bits 7 and 6 at address A816). Also, this externally-input trigger can be selected from the following three types: falling edges, rising edges, and falling and rising edges. The reversed content of the pulse output data bit can be output to each pulse output port by the pulse output polarity select bit of the three-phase output data register 1 (bit 3 at address A916). When the pulse output polarity select bit = "0", the content of the pulse output data bit is output as it is; when the pulse output polarity select bit = "1", the reversed content is output.

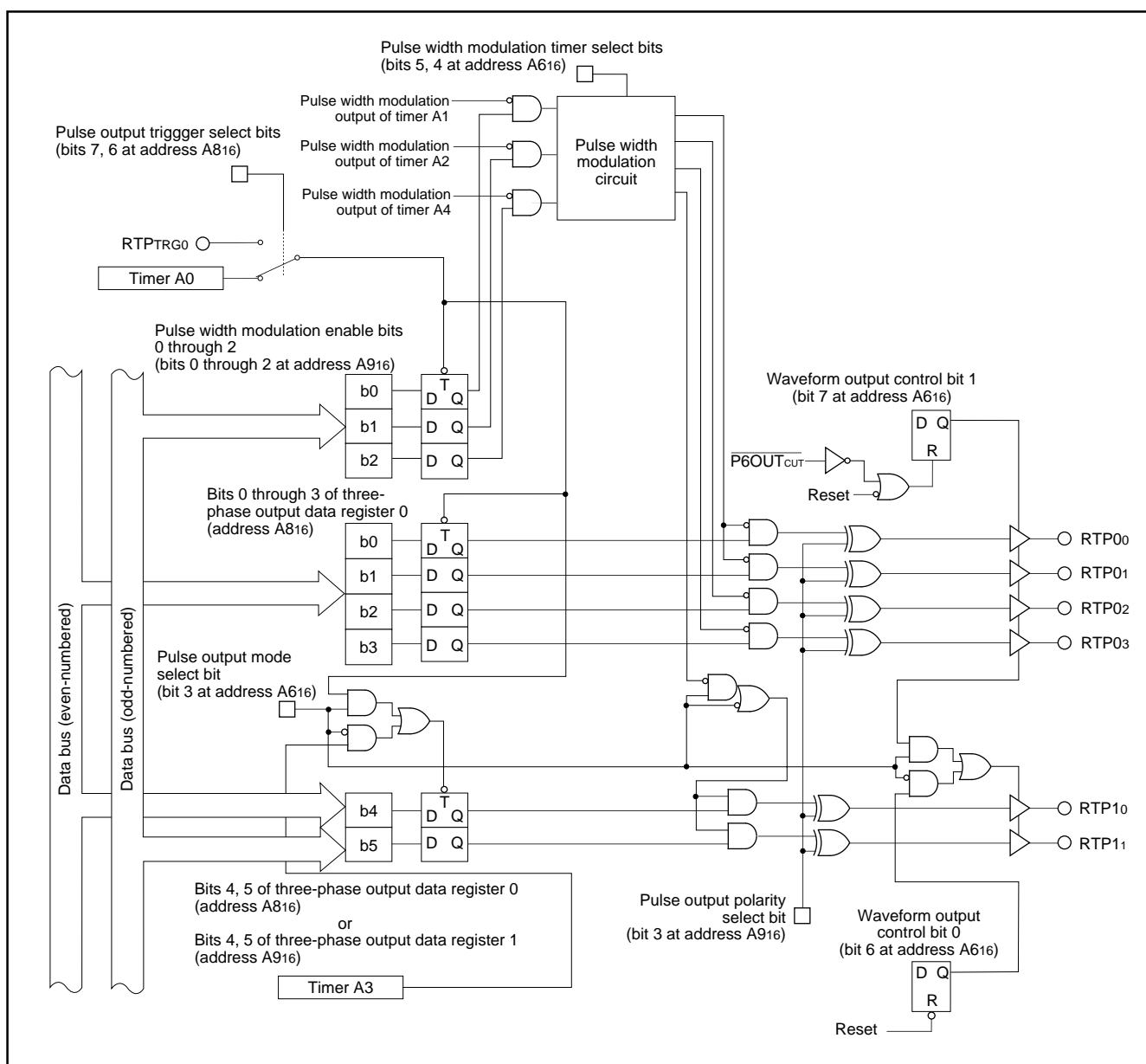


Fig. 48 Block diagram in pulse output port mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

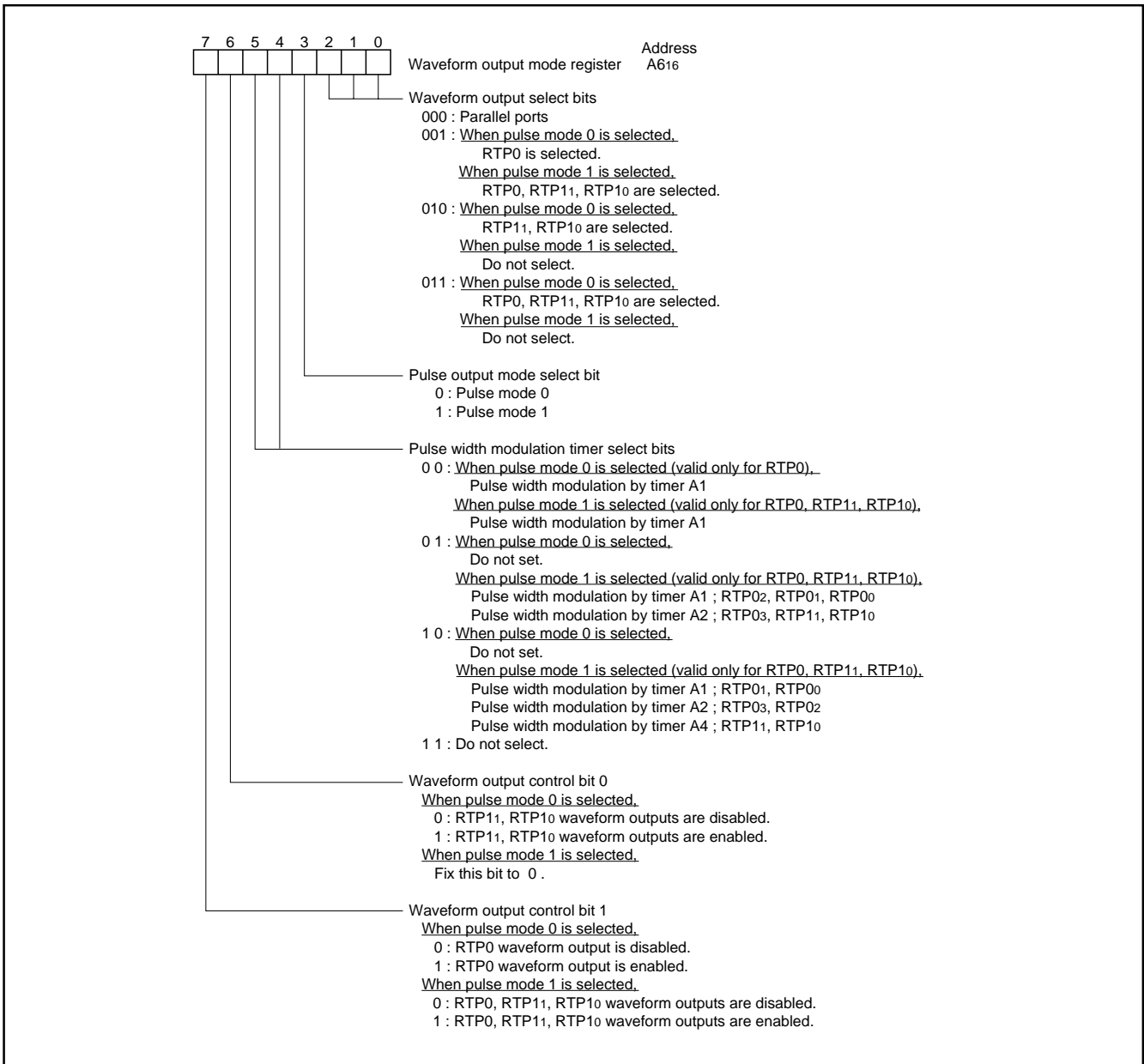


Fig. 49 Bit configuration of waveform output mode register in pulse output port mode

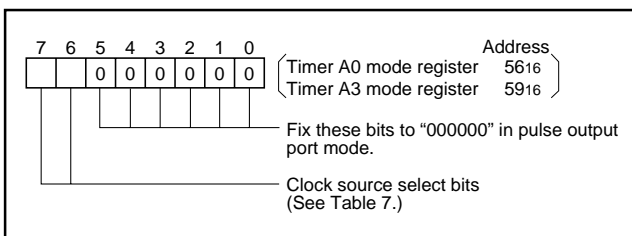


Fig. 50 Bit configuration of timer A3 and A0 mode registers in pulse output port mode

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

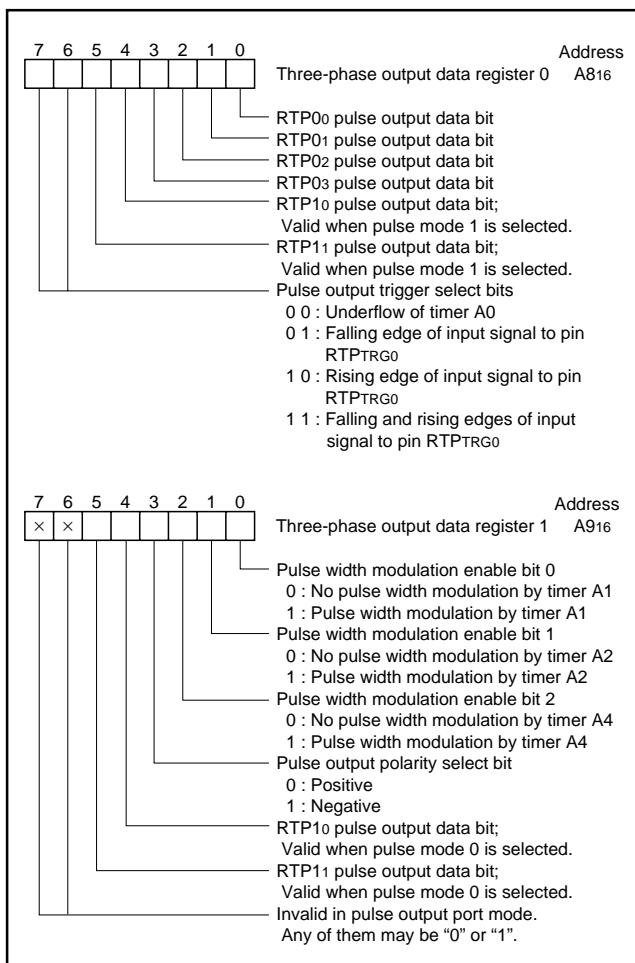


Fig. 51 Bit configuration of three-phase output data registers 1 and 0 in pulse output port mode

**Pulse mode 0**

This mode divides a pulse output port into 4 bits and 2 bits and individually controls them.

When setting the pulse output mode select bit to "0", and setting bits 2 and 1 to "0" and bit 0 to "1" of the waveform output select bits, four of RTP03, RTP02, RTP01, and RTP00 become the pulse output ports with RTP0 selected.

When setting the pulse output mode select bit to "0", and setting bits 2 and 0 to "0" and bit 1 to "1" of the waveform output select bits, two of RTP11, RTP10 become the pulse output ports.

When setting the pulse output mode select bit to "0", and setting bit 2 to "0" and bits 1 and 0 to "1" of the waveform output select bits, the following two groups become the pulse output ports:

- Two of RTP11, RTP10
- Four of RTP03, RTP02, RTP01, RTP00.

Each time the contents of timer A3 counter become 0000<sub>16</sub>, the contents of three-phase output data register 1 (address A916)'s bits 5 and 4, which corresponding to RTP11 and RTP10, are output from ports.

Each time the contents of timer A0 counter become 0000<sub>16</sub>, the contents of three-phase output data register 0 (address A816)'s low order 4 bits, which corresponding to RTP03, RTP02, RTP01, RTP00, are output from ports.

When writing "0" to the specified one of the pulse output data bits, "L" level is output from the pulse output port when the contents of the corresponding timer counter become 0000<sub>16</sub>; when writing "1" to it, "H" level is output from the pulse output port.

In the case that an input trigger of pin RTPTRG0 is selected, the data written to the pulse output data bit is output from the corresponding pulse output port by this selected trigger.

Additionally, pulse width modulation can be applied for RTP03, RTP02, RTP01, and RTP00. Because timer A1 is used for pulse width modulation, actuate timer A1 in the pulse width modulation mode. When any of pulse output data bits is "1", the pulse to which pulse width modulation has been applied is output from the pulse output port when the contents of timer A0 counter become 0000<sub>16</sub>. The pulse width modulation using timer A1 can be applied by setting the pulse width modulation enable bit 0 of the three-phase output data register 1 (bit 0 at address A916) to "1" and the pulse width modulation timer select bits of the waveform output mode register (bits 5 and 4 at address A616) to "00". Figure 52 shows example waveforms in pulse mode 0.

In ports selecting pulse mode 0, output of RTP11 and RTP10 is controlled by the waveform output control bit 0 (bit 6) of waveform output mode register; output of RTP03, RTP02, RTP01 and RTP00 is done by the waveform output control bit 1 (bit 7).

When setting the waveform output control bit to "1", waveform is output from the corresponding port. When clearing that bit to "0", waveform output from the corresponding port stops, and the port becomes floating. The waveform output control bits are cleared to "0" by reset or by executing instructions.

Also, the waveform output control bit 1 can be cleared to "0" by inputting a falling edge to pin P6OUTCUT.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

## Pulse mode 1

This mode controls a pulse output port as a 6-bit unit.

When setting the pulse output mode select bit to "1", and setting bits 2 and 1 to "0" and bit 0 to "1" of the waveform output select bits, the following become the pulse output ports:

•Six of RTP11, RTP10, RTP03, RTP02, RTP01, RTP00

Whether to control these pulse output ports by an underflow of timer A0 or an input edge to pin RTPTRG0 is selected by the pulse output trigger select bit of the three-phase output data register 0 (bits 7 and 6 at address A816).

Additionally, pulse width modulation can be applied to the pulse output ports. The pulse width modulation timer select bits of the waveform output mode register (bits 5 and 4 at address A616) select the type of pulse width modulation from the following:

- (1) When the pulse width modulation timer select bits = "00", the common modulation to six of RTP11, RTP10, RTP03, RTP02, RTP01, RTP00 is selected. For this modulation, since timer A1 is necessary, be sure to actuate this timer in the pulse width modulation mode.
- (2) When the pulse width modulation timer select bits = "01", the modulation to the following two groups is selected; one consists of RTP11, RTP10, RTP03, and the other consists of RTP02, RTP01, RTP00. For this modulation, since timers A1 and A2 are necessary, be sure to actuate these timers in the pulse width modulation mode.
- (3) When the pulse width modulation timer select bits = "10", the modulation to the following three groups is selected; one consists of RTP11, RTP10, and another consists of RTP03, RTP02, and the other consists of RTP01, RTP00. For this modulation, since timers A1, A2, and A4 are necessary, be sure to actuate these timers in the pulse width modulation mode.

Additionally, at this time, be sure to set the corresponding pulse width modulation enable bit of the three-phase output data register 1 (bits 2 through 0 at address A916) to "1" in order to enable the pulse width modulation.

The other operations are the same as that of pulse mode 0. Figure 53 shows example waveforms in pulse mode 1.

In ports selecting pulse mode 1, output can be controlled by the waveform output control bit 1 of the waveform output mode register (bit 7 at address A616).

When setting the waveform output control bit to "1", waveform is output from the corresponding port. When clearing that bit to "0", waveform output from the corresponding port stops and the port becomes floating. The waveform output control bit 1 is cleared to "0" by reset, by inputting a falling edge to pin P6OUTCUT, or by executing instructions.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

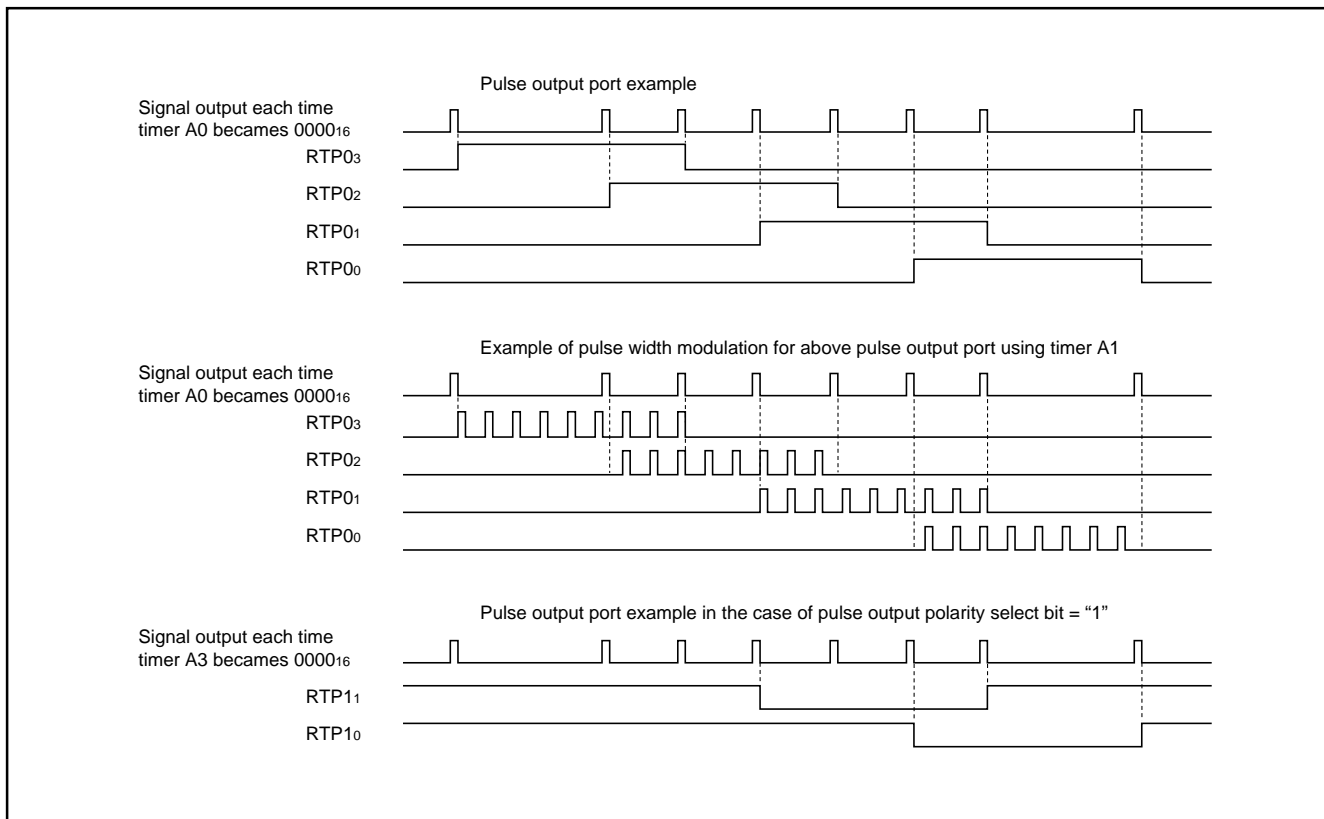


Fig. 52 Example waveforms in pulse mode 0

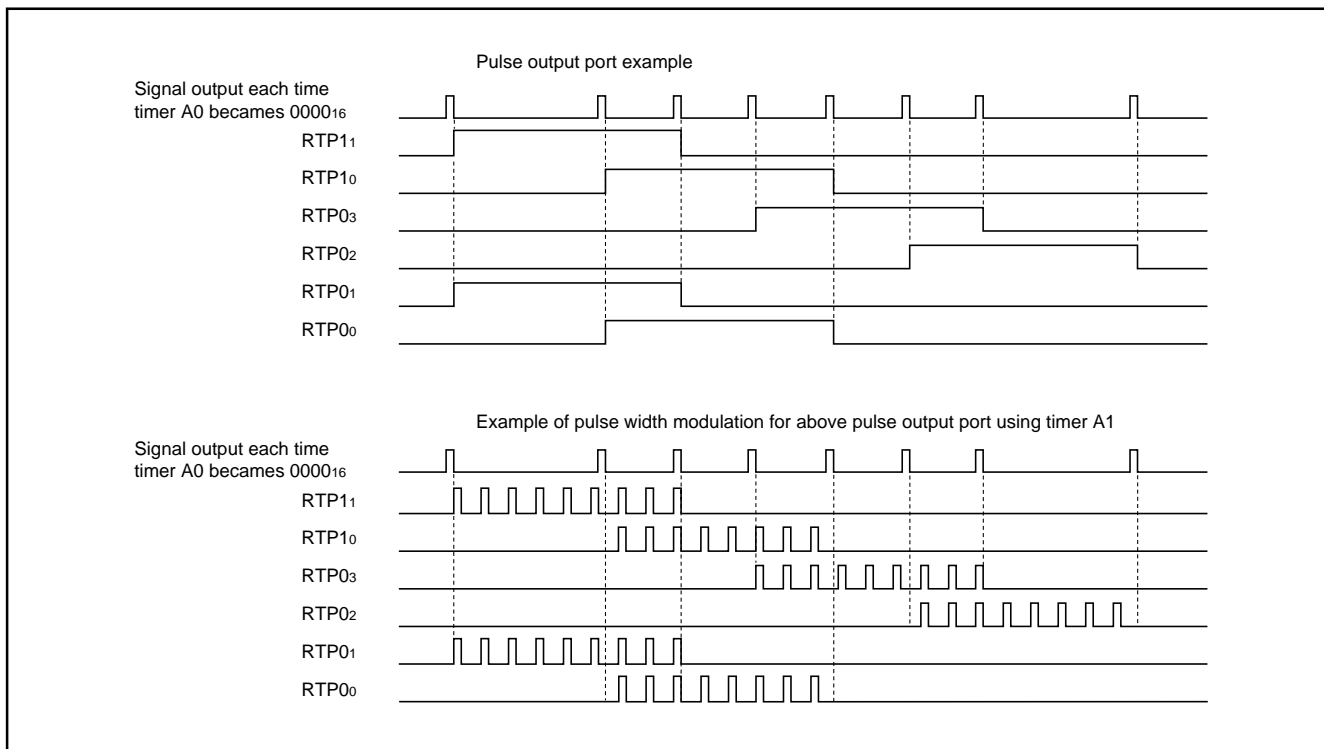


Fig. 53 Example waveforms in pulse mode 1

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**SERIAL I/O PORTS**

Two independent serial I/O ports are provided. Figure 54 shows a block diagram of the serial I/O ports.

Bits 0 through 2 of the UART<sub>i</sub>(i = 0, 1) transmit/receive mode register shown in Figure 55 are used to determine whether to use port P1 as a programmable I/O port, clock synchronous serial I/O port, or asyn-

chronous (UART) serial I/O port which uses start and stop bits. Figures 56 and 57 show the block diagrams of the receiver/transmitter.

Figure 58 shows the bit configuration of the UART<sub>i</sub> transmit/receive control register.

Each communication method is described below.

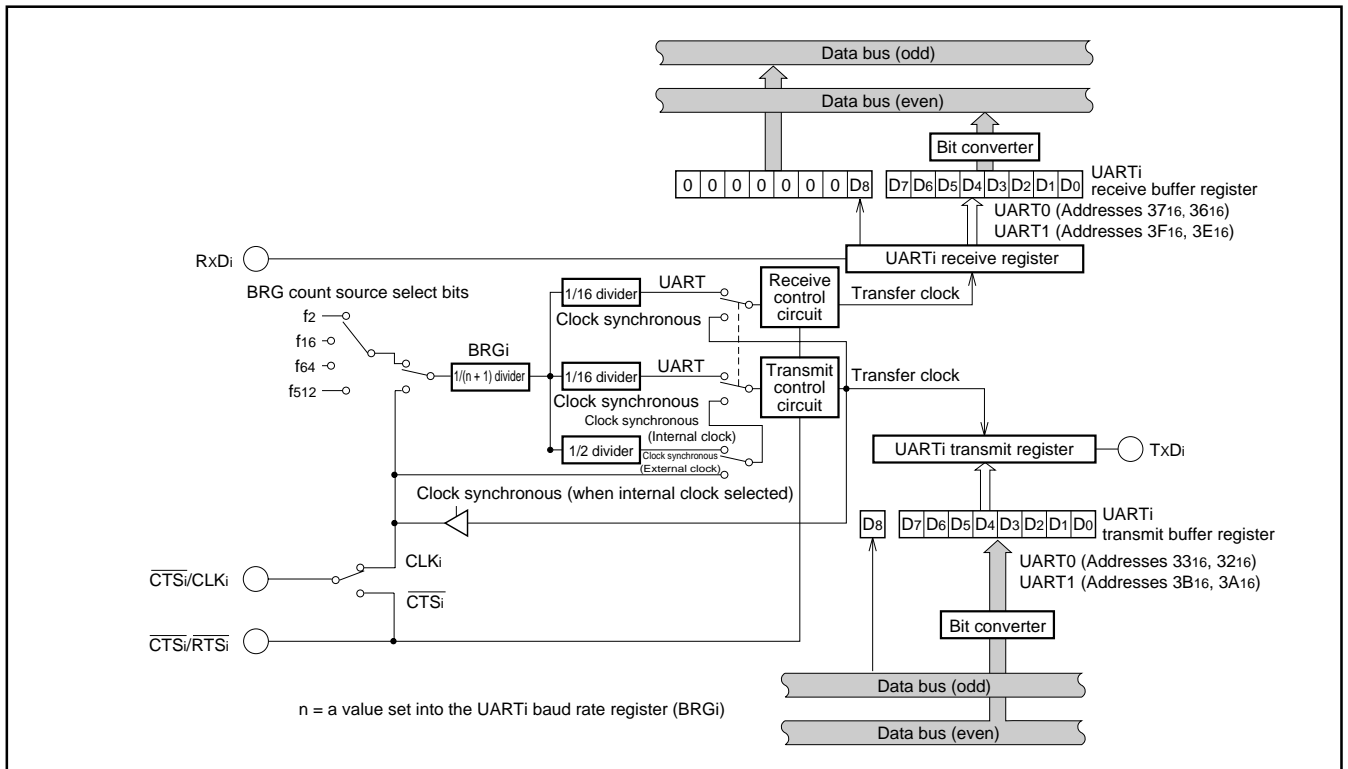


Fig. 54 Block diagram of serial I/O port

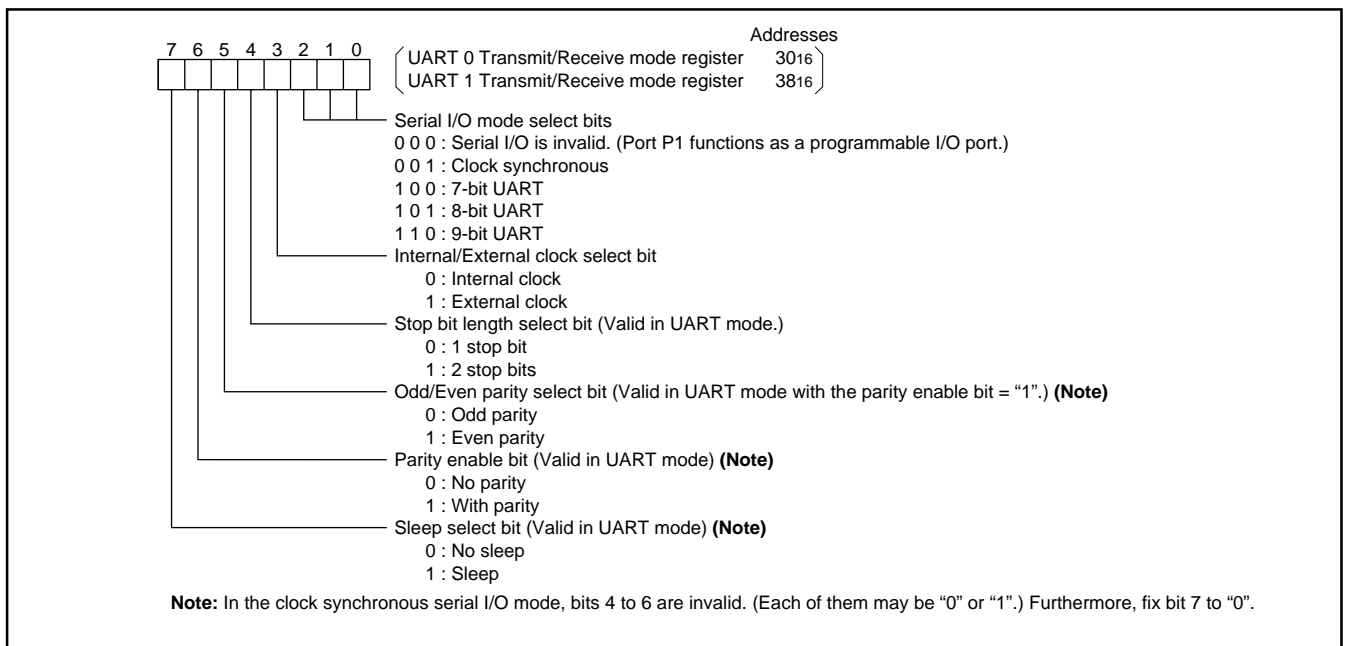


Fig. 55 Bit configuration of UART<sub>i</sub> transmit/receive mode register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

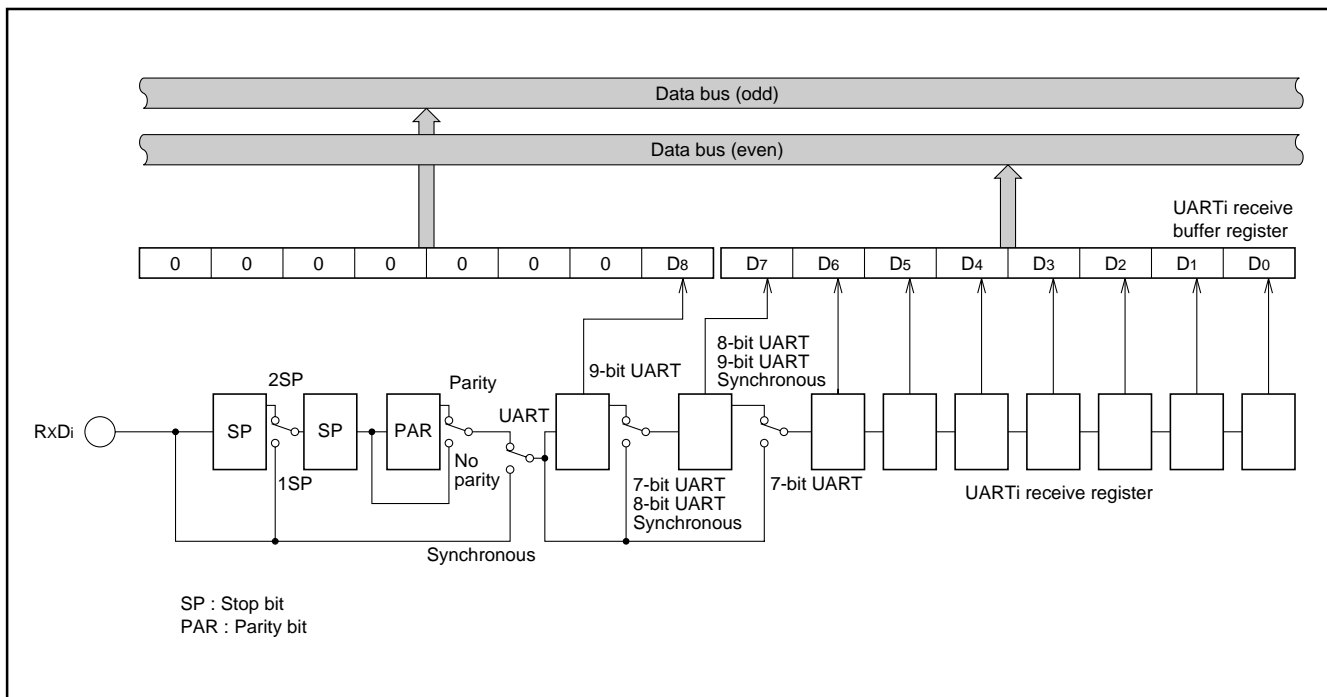


Fig. 56 Block diagram of receiver

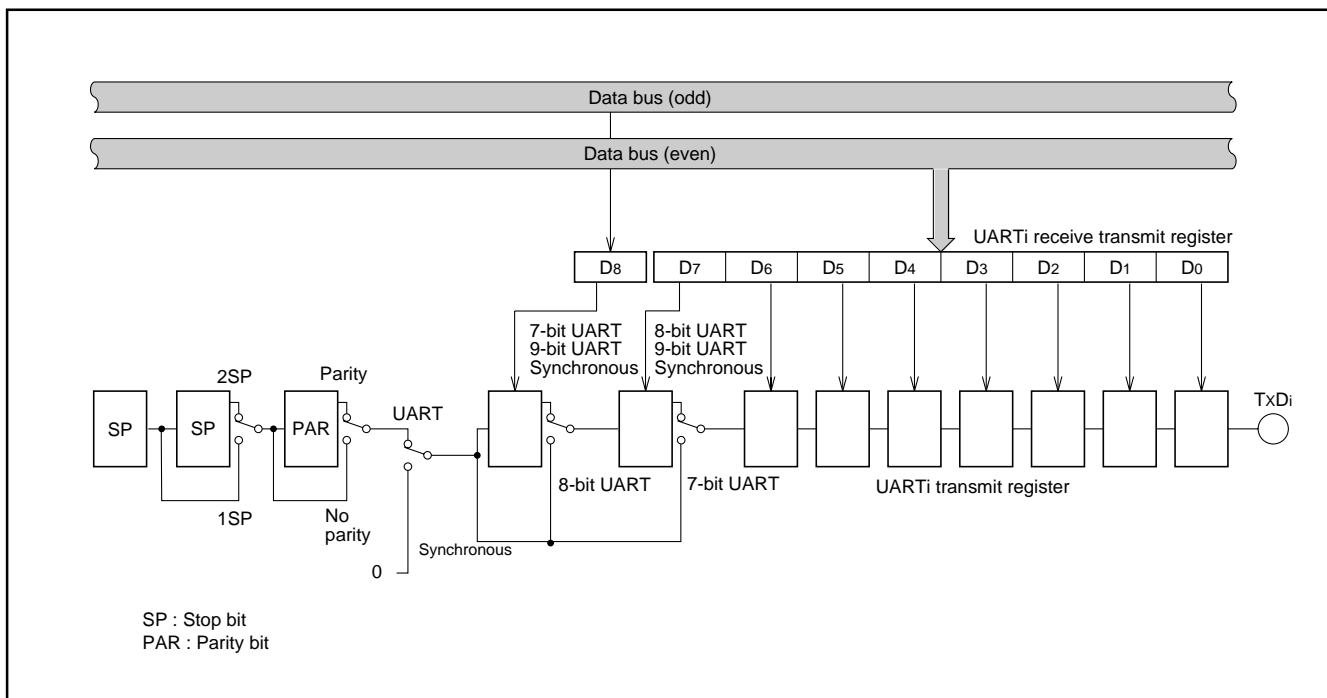


Fig. 57 Block diagram of transmitter

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

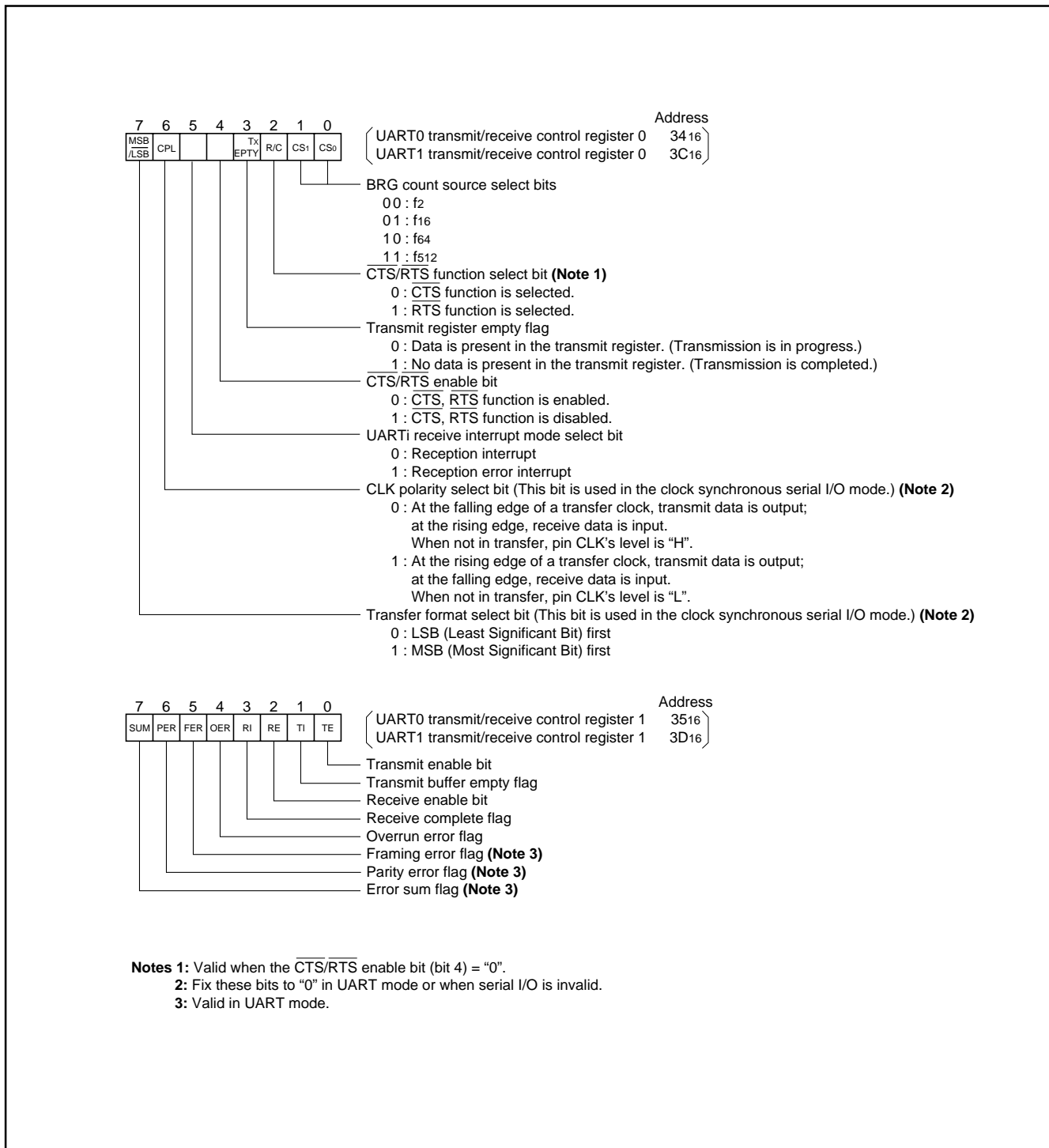


Fig. 58 Bit configuration of UARTi transmit/receive control register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**CLOCK SYNCHRONOUS SERIAL COMMUNICATION**

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 59 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 (CS0) and bit 1 (CS1) of the clock-sending-side UARTj transmit/receive control register 0. As shown in Figure 54, 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 CS0 and CS1 bits of the UARTk transmit/receive control register 0 are ignored because an external clock is selected.

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

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

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

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

The case using  $\overline{\text{CTS}}$  and  $\overline{\text{RTS}}$  signals are explained below. As shown in Figure 66, bits 2 and 3 of the serial I/O pin control register can determine whether port pins P13 and P17 are used as pins TxDi or as port pins. When bits 2 and 3 are "0", P13 and P17 function as pins TxDi; when bits 2 and 3 are "1", P13 and P17 function as port pins. Therefore, in the input-only system where pins TxDi are not used, pins TxDi can function as port pins.

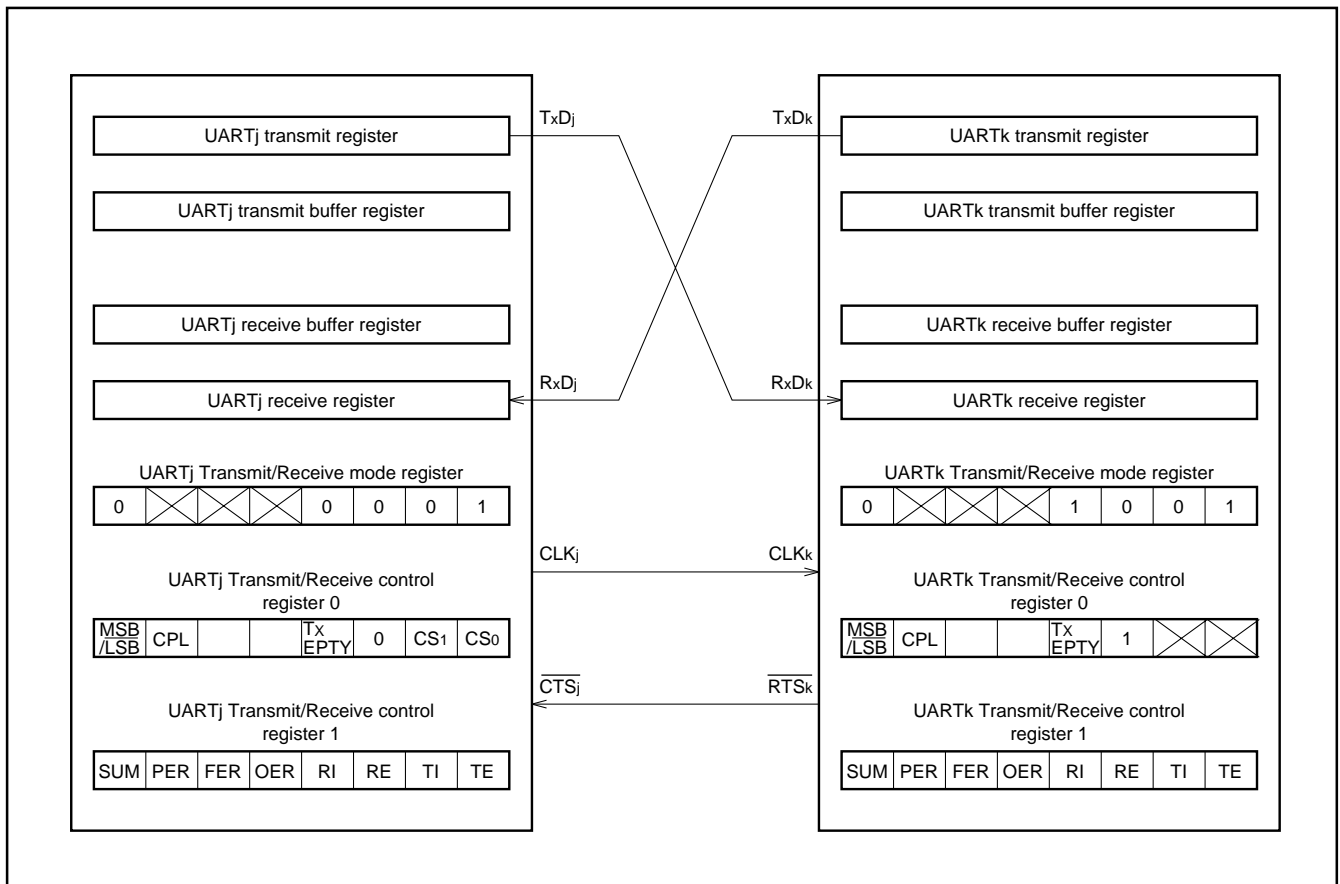


Fig. 59 Clock synchronous serial communication

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

## Transmission

Transmission is started when bit 0 (TEj flag: transmit enable bit) of UARTj transmit/receive control register 1 is "1", bit 1 (Tlj flag) of one is "0", and  $\overline{CTSj}$  input is "L". The Tlj 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 60, 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, Tlj flag, and  $\overline{CTSj}$  signals are ignored until data transmission completes. Therefore, transmission is not interrupted when  $\overline{CTSj}$  input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, Tlj flag, and  $\overline{CTSj}$  is checked while the TENDj 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 Tlj flag is cleared to "0" before the TENDj signal goes "H".

Bit 3 (TxEMPTYj 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 Tlj 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  $\overline{RTSk}$  output is "H" when the REk flag is "0". When the REk flag is set to "1", the  $\overline{RTSk}$  output becomes "L". This informs the transmitter side that reception becomes enabled. When the receive operation starts, the  $\overline{RTSk}$  output automatically becomes "H".

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

(Note that, in the clock synchronous serial communication, the polarity of a transfer clock can be inverted. For details, refer to the section on the polarity of the transfer clock.) When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and bit 3 (RIk flag) of UARTk transmit/receive control register 1 is set to "1". In other words, the setting "1" to the RIk flag indicates that the receive buffer register contains the received data. At this time, if the low-order byte of the UARTk receive buffer register is read out, the  $\overline{RTSk}$  output turns back to "L". This indicates that the

next data reception becomes enabled. Bit 4 (OERk flag) of UARTk transmit/receive control register 1 is set to "1" when the next data is transferred from the receive register to the receive buffer register while RIk flag is "1", and indicates that the next data was transferred to the receive register before the contents of the receive buffer register was read. (In other words, this indicates that an overrun error has occurred.) RIk flag is automatically cleared to "0" when the low-order byte of the receive buffer register is read or when the REk flag is cleared to "0". The OERk flag is cleared when the REk flag is cleared. Bit 5 (FERk flag), bit 6 (PERk flag), and bit 7 (SUMk flag) are ignored in clock synchronous mode.

As shown in Figure 54, 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.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

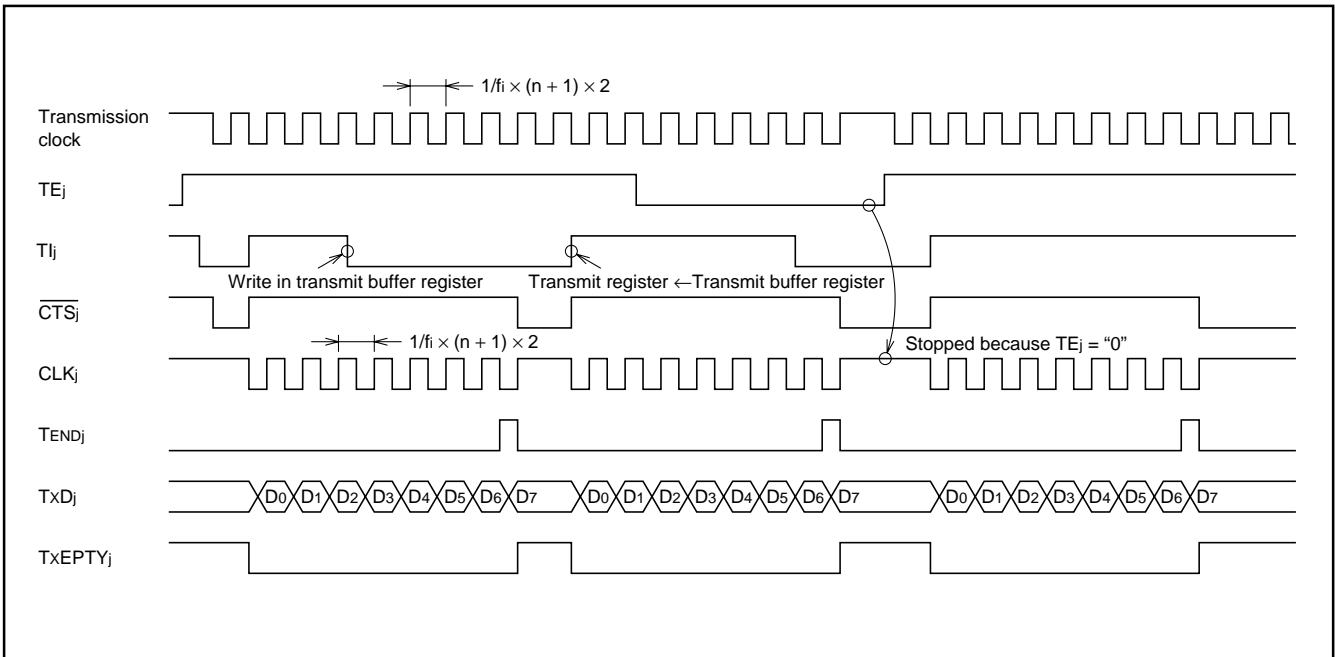


Fig. 60 Clock synchronous serial I/O timing

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

**Polarity of transfer clock**

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

As shown in Figure 61, when bit 6 = "0", the polarity is as follows:

- In transmission, transmit data is output at the falling edge of CLK<sub>j</sub>.
- In reception, receive data is input at the rising edge of CLK<sub>k</sub>.
- When not in transfer, CLK<sub>i</sub> is at "H" level.

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

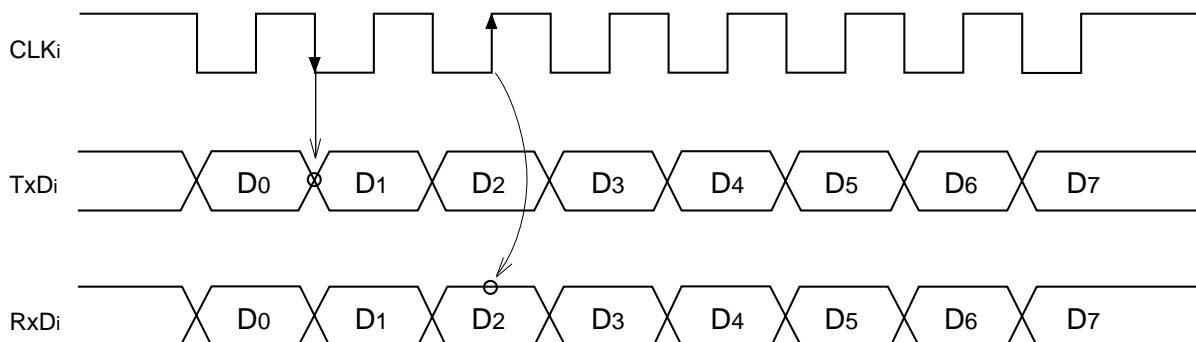
- In transmission, transmit data is output at the rising edge of CLK<sub>j</sub>.
- In reception, receive data is input at the rising edge of CLK<sub>k</sub>.
- When not in transfer, CLK<sub>i</sub> is at "L" level.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

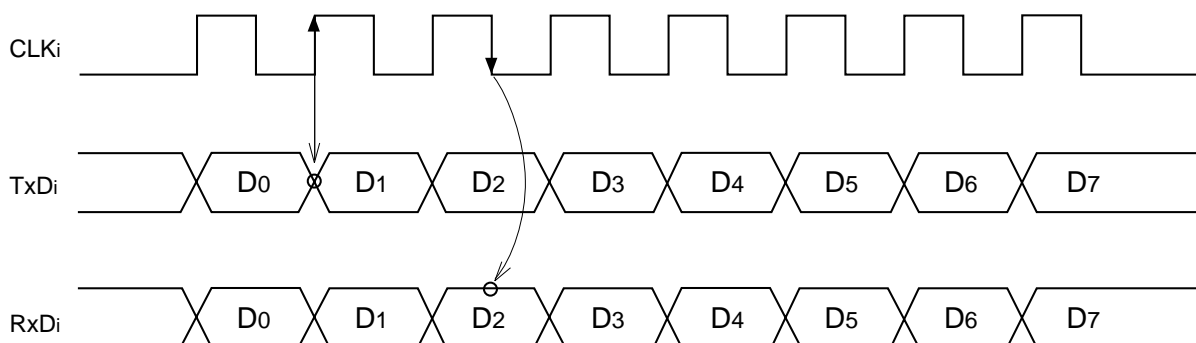
16-BIT CMOS MICROCOMPUTER

■ 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. 61 Polarity of transfer clock

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**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 62 shows the connection relation.

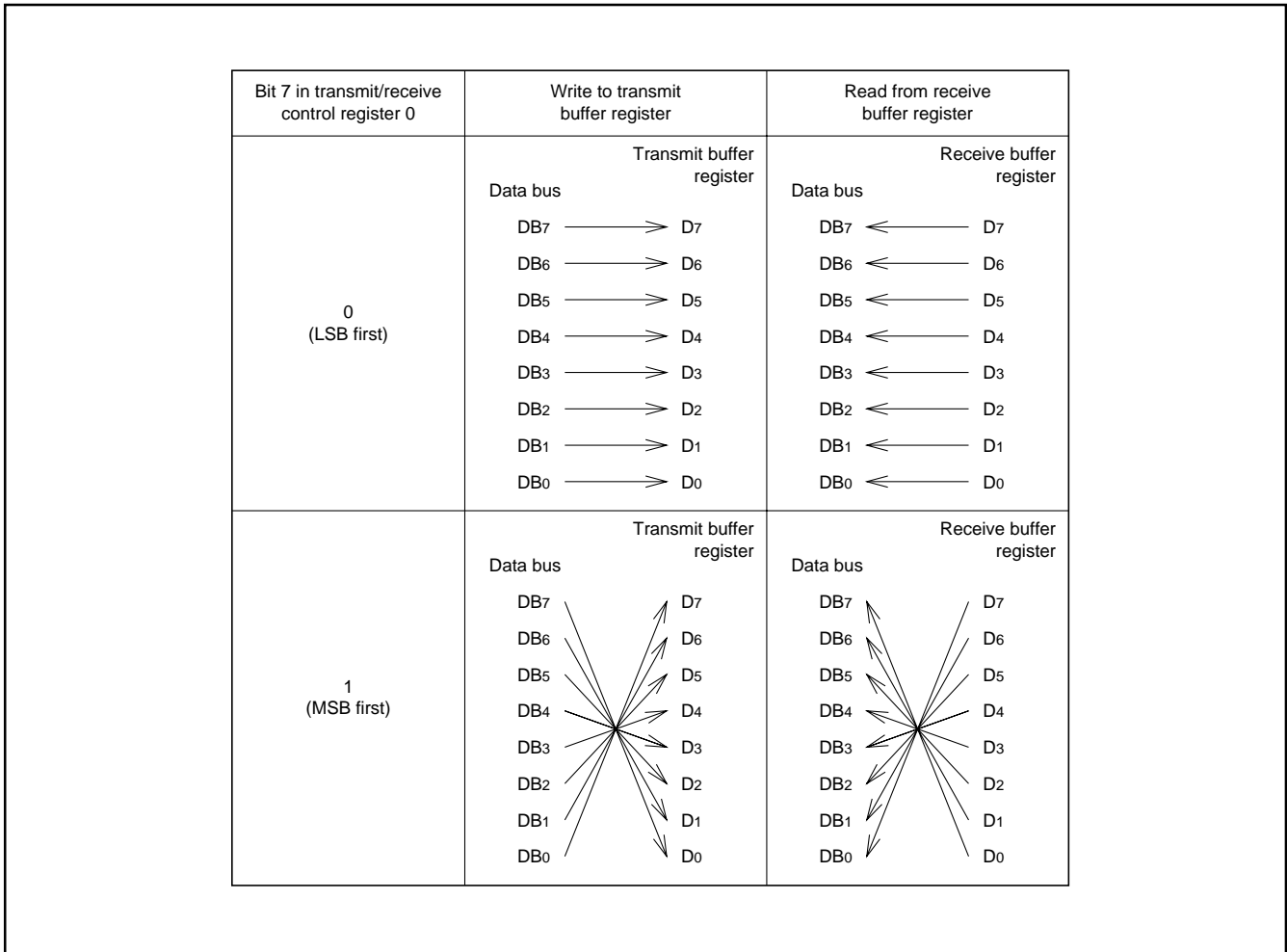


Fig. 62 Connection relation between transmit buffer register, receive buffer register, and data bus

**Precautions for clock synchronous serial communication**

In the clock synchronous serial communication, the separate function for CTS<sub>i</sub>/RTS<sub>i</sub> cannot be selected. Furthermore, when an internal clock is selected,  $\overline{\text{RTS}}$  output is undefined. Therefore, do not use the  $\overline{\text{RTS}}$  function.

Before transmit operation is performed, be sure to clear bits 2 and 3 of the serial I/O pin control register (address AC16) to "00".

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**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 (CS0) and bit 1 (CS1) 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 fEXT,

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

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.

Figure 66 shows the bit configuration of the serial I/O pin control register. By bits 0 and 1 of the serial I/O pin control register (CTS<sub>i</sub>/RTS<sub>i</sub> separate select bits), the function of the CTS/RTS pin can be separated into two functions, and each function can be assigned to two different pins. When bits 0 and 1 = "11", the above separation is performed. When bits 0 and 1 = "00", no separation is performed.

Table 8 lists the selection methods of the CTS/RTS function.

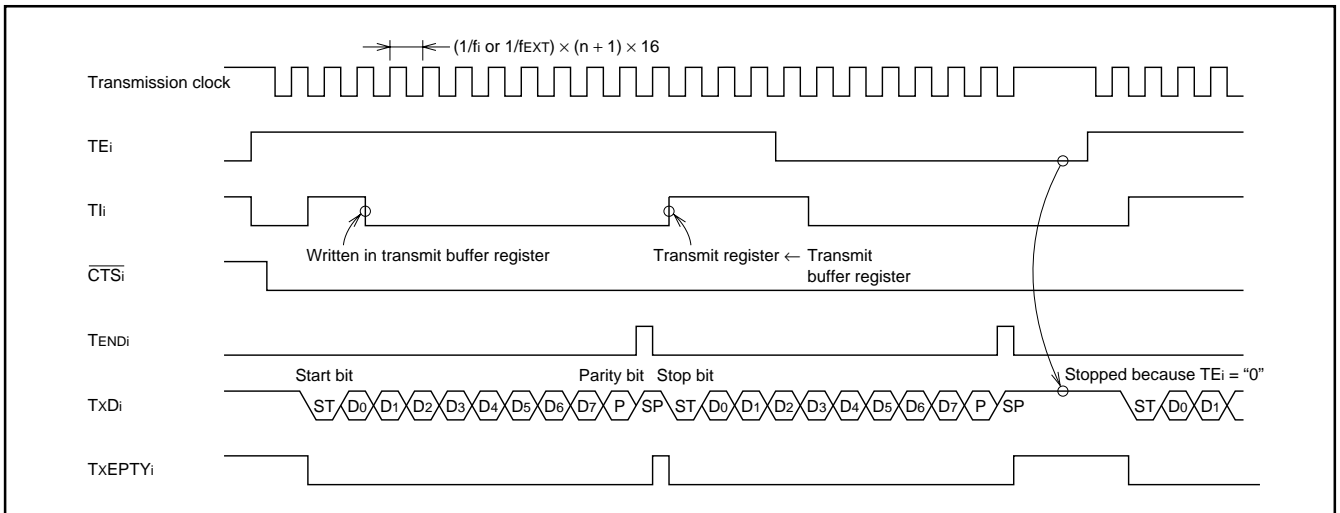


Fig. 63 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit selected

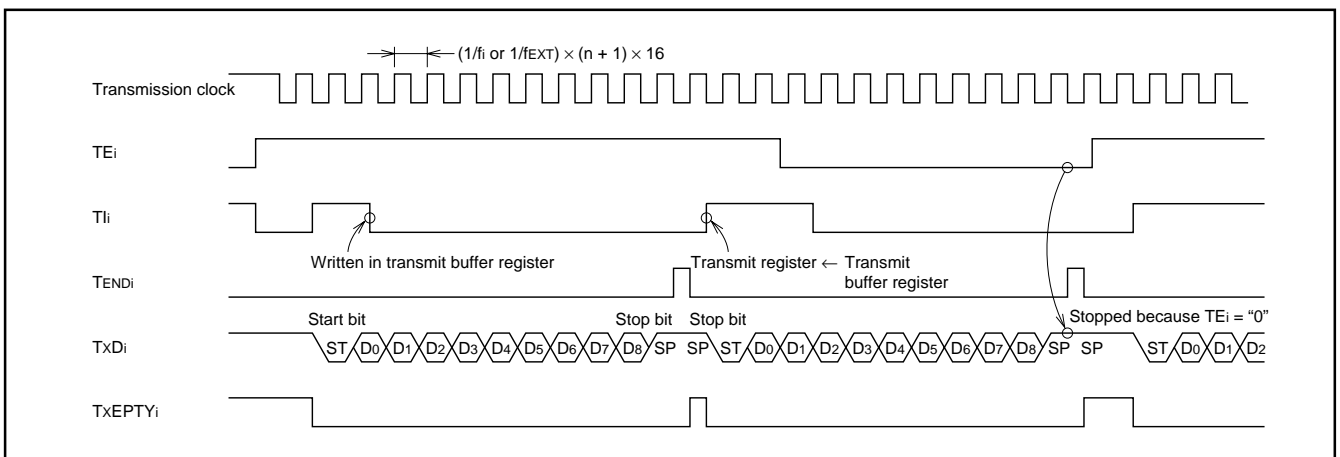


Fig. 64 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits selected

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**Transmission**

Transmission is started when bit 0 (TEi flag transmit enable flag) of UARTi transmit/receive control register 1 is "1", bit 1 (Tli flag) is "0", and  $\overline{\text{CTS}}_i$  input (in other words, transmit enable signal input from receiver) is "L." The Tli 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 63 and 64, data is output from the TxDi pin with the stop bit or parity bit specified by bits 4 through 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

successfully.

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

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

The transmission start condition indicated by TEi flag, Tli flag, and  $\overline{\text{CTS}}_i$  is checked while the TENDi signal shown in Figure 63 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmit buffer register and Tli flag is cleared to "0" before the TENDi signal goes "H".

Bit 3 (TxEMPTYi 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 Tli flag changes from "0" to "1", the interrupt request bit of the UARTi transmit interrupt control register is set to "1".

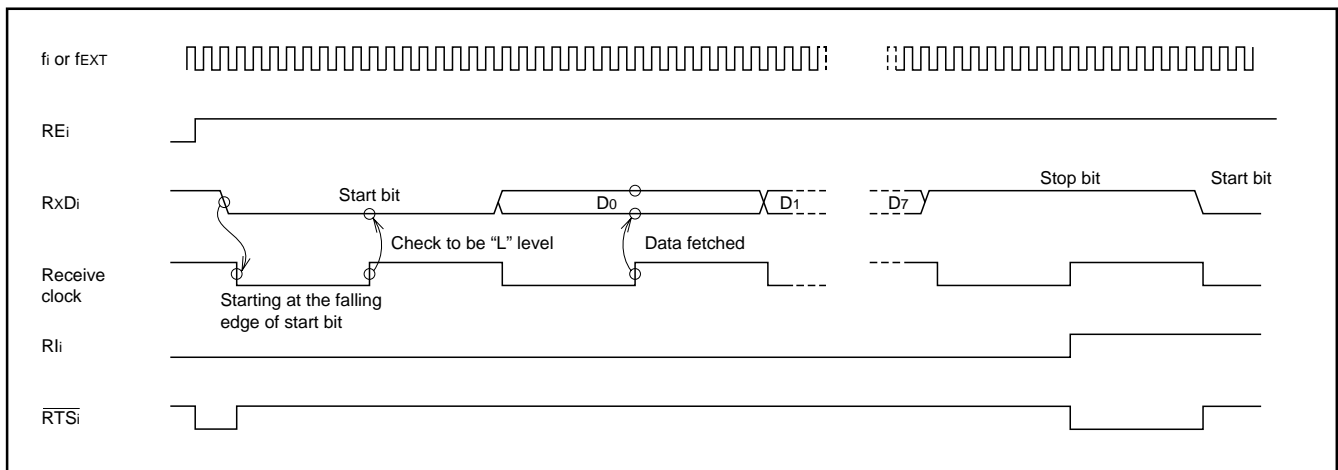


Fig. 65 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit selected

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

Table 8. Selection methods of  $\overline{\text{CTS}}/\overline{\text{RTS}}$  function

CTS/RTS enable bit	CTS/RTSi separate select bit	CTS/RTS function select bit	Functions			
			Pin P10/ $\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$	Pin P11/ $\overline{\text{CTS}}_0/\text{CLK}_0$	Pin P14/ $\overline{\text{CTS}}_1/\overline{\text{RTS}}_1$	Pin P15/ $\overline{\text{CTS}}_1/\text{CLK}_1$
0	0	0	$\overline{\text{CTS}}_0$	P11 or CLK0	$\overline{\text{CTS}}_1$	P15 or CLK1
		1	$\overline{\text{RTS}}_0$	P11 or CLK0	$\overline{\text{RTS}}_1$	P15 or CLK1
1	X	X	P10	P11 or CLK0	P14	P15 or CLK1

X: It may be "0" or "1".

**Notes 1:** When using the  $\overline{\text{CTS}}$  function, be sure to clear the corresponding bit of the port P1 direction register to "0".

**2:** When  $\overline{\text{CTS}}_i$  and  $\overline{\text{RTS}}_i$  has been separated, the CLK<sub>i</sub> pin cannot be used. Therefore, in the clock synchronous serial communication,  $\overline{\text{CTS}}_i$  and  $\overline{\text{RTS}}_i$  cannot be separated. Also, when  $\overline{\text{CTS}}_i$  and  $\overline{\text{RTS}}_i$  are separated in UART mode, be sure to select an internal clock.

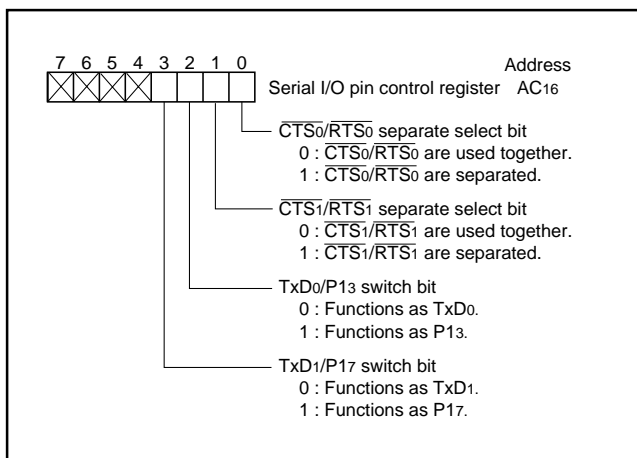


Fig. 66 Bit configuration of serial I/O pin control register

### Receive

Receive is enabled when bit 2 (REi flag) of UARTi transmit/receive control register 1 is set to "1." As shown in Figure 65, 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 56. At this point, the contents of the receive register is transferred to the receive buffer register and bit 3 (Rli flag) of UARTi transmit/receive control register 1 is set to "1." In other words, the Rli 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 Rli flag is "1", in other words, when an overrun error occurs. If the OERi flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive buffer register has been read.

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

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

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

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

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

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

The OERi flag is cleared to "0" when writing "0" to the REi flag.

### Interrupt request at completion of reception

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

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

- Each reception
- When an error occurs at reception

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

### Sleep mode

The sleep mode is used to communicate only between certain microcomputers when multiple microcomputers are connected through serial I/O.

The microcomputer enters the sleep mode when bit 7 of UARTi transmit/receive mode register is set to "1."

The operation of the sleep mode for an 8-bit asynchronous communication is described below.

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asynchronous communication) of the received data is "0". Also the RLi, 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 through 6 are its own address and sets the sleep bit to "1" if not. Next, the main microcomputer sends data with bit 7 cleared. Then the microcomputer which cleared the sleep bit will receive the data, but the microcomputers which set the sleep bit to "1" will not. In this way, the main microcomputer is able to communicate only with the designated microcomputer.

### Precautions for clock asynchronous (UART) serial communication

When  $\overline{\text{CTS}}_i$  and  $\overline{\text{RTS}}_i$  are separated, pin CLKi cannot be used. Therefore, when  $\overline{\text{CTS}}_i$  and  $\overline{\text{RTS}}_i$  are separated in UART mode, be sure to select an internal clock.

Before transmit operation is performed, be sure to clear bits 2 and 3 of the serial I/O pin control register (address AC16) to "00".

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**A-D CONVERTER**

The A-D converter is a 10-bit successive approximation converter. The use of A-D converter or the use of comparator can be selected for each A-D input pin. The contents of the comparator function select register specify it. Figure 67 shows a block diagram of the A-D converter.

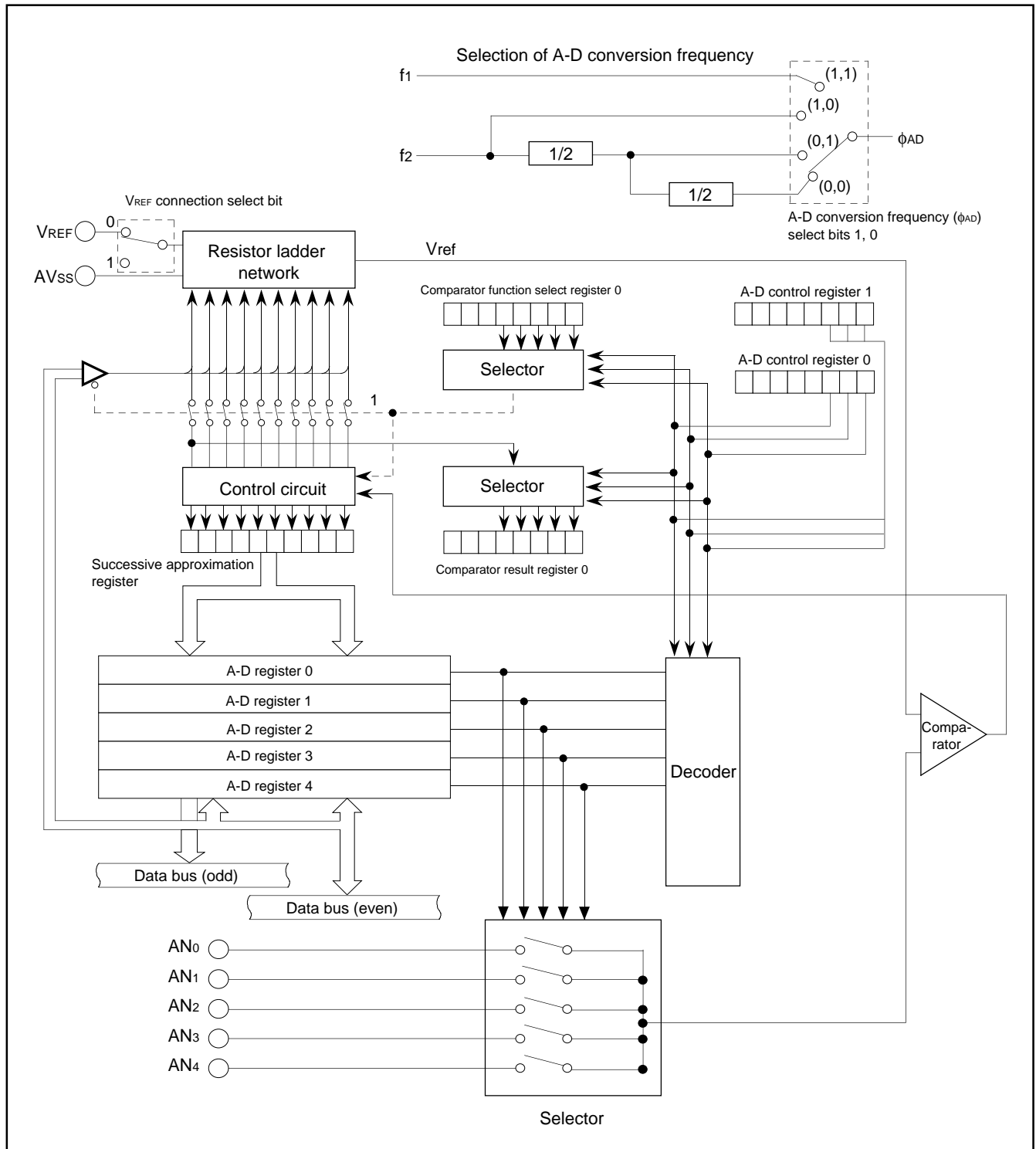


Fig. 67 Block diagram of A-D converter

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

Figure 68 shows the bit configuration of comparator function select register 0 (address DC16). Bits 7 to 0 correspond to channels 7 to 0 respectively. Each channel can be selected as either an A-D converter or a comparator. When the bit is "0", the channel corresponding to it functions as a 10-bit or an 8-bit A-D converter. When the bit is "1", the channel functions as a comparator.

When selecting an A-D converter, an input voltage to a selected analog input pin is A-D converted and the result is stored into the A-D register.

When selecting a comparator, D-A conversion is performed to the value of which high-order 8 bits are the value stored in an even address of the A-D converter and of which low-order 2 bits are "10<sub>2</sub>." Then, this D-A converted value is compared with the voltage supplied to an analog input pin. After the comparison, when the voltage supplied to an analog input pin is higher, "1" is stored into the comparator result register 0 (address DE16) shown in Figure 69. When it is lower, "0" is stored into that register.

Be sure to perform only read to the A-D register of which channel is selected as an A-D converter, and perform only write to the A-D register of which channel is selected as a comparator. Additionally, do not write to the comparator function select register 0 and the A-D register while an A-D converter or a comparator is operating.

Port direction register's bits corresponding to pins to be A-D converted must be "0" (input mode) because analog input ports are multiplexed with port P7.

Figure 70 shows the bit configuration of the A-D control register 0 (address 1E16), and Figure 71 shows that of the A-D control register 1 (address 1F16).

The operation clock of the A-D converter,  $\phi_{AD}$ , is selected by the following bits: 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 = "0",  $\phi_{AD}$  is selected as follows:

- if bit 7 of the A-D control register 0 = "0",  $\phi_{AD} = f_2/4$ .
- if bit 7 of the A-D control register 0 = "1",  $\phi_{AD} = f_2/2$ .

When bit 4 of the A-D control register 1 = "1",  $\phi_{AD}$  is selected as follows:

- if bit 7 of the A-D control register 0 = "0",  $\phi_{AD} = f_2$ .
- if bit 7 of the A-D control register 0 = "1",  $\phi_{AD} = f_1$ .

Note that the highest frequency,  $\phi_{AD} = f_1$ , can be selected only in the 8-bit resolution mode.

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

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

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

When the conversion result is used as 8-bit data, the high-order 8 bits of the 10-bit A-D conversion result are stored in even address of the corresponding A-D register. In this case, the value at the A-D register's odd address is "0016" when read.

Whether to connect the reference voltage input (VREF) with the lad-

der network or not depends on bit 5 of the A-D control register 1. The VREF pin is connected when bit 5 is "0" and is disconnected when bit 5 is "1" (High impedance state).

When A-D or D-A conversion is not performed, current from the VREF pin to the ladder network can be cut off by disconnecting ladder network from the VREF pin.

Before starting A-D or D-A conversion, wait for 1  $\mu$ s or more after clearing bit 5 to "0".

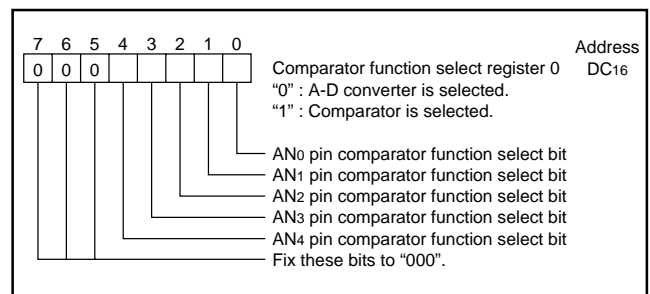


Fig. 68 Bit configuration of comparator function select register 0

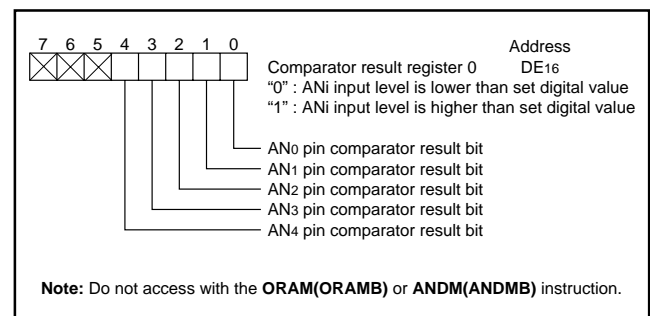


Fig. 69 Bit configuration of comparator result register 0

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**Operation mode**

The operation mode is selected by bits 3 and 4 of A-D control register 0 and bit 2 of A-D control register 1. The available operation modes are one-shot, repeat, single sweep, and repeat sweep 0. Either an A-D converter or a comparator can be selected respectively for every pin in the following 4 modes. The following description applies to the case where the bit of the comparator function select register 0 is "0" and an A-D converter is selected. It also applies to a comparator's operation except that an A-D conversion is changed to a comparator operation and the result of a comparison is stored into the comparator result register 0.

**(1) One-shot mode**

One-shot mode is selected when bits 3 and 4 of A-D control register 0 are "0". The A-D conversion pins are selected with bits 0 to 2 of A-D control register 0. A-D conversion or comparator operation is started when bit 6 of A-D control register 0 (A-D conversion start bit)

is set to "1".

When the AN<sub>i</sub> (i = 4 through 0) comparator function select bit (bits 4 through 0) of the comparator function select register 0 = "0" and bit 3 of the A-D control register 1 = "1", A-D conversion ends 59 φ<sub>AD</sub> cycles after, and the interrupt request bit of the A-D conversion interrupt control register is set to "1". At the same time, bit 6 of the A-D control register 0 (A-D conversion start bit) is cleared to "0" and this A-D conversion stops. The result of A-D conversion is stored into the A-D register corresponding to the selected pin.

When the AN<sub>i</sub> (i = 4 through 0) comparator function select bit (bits 4 through 0) of the comparator function select register 0 = "1", a comparator operation ends 14 φ<sub>AD</sub> cycles after, and the interrupt request bit of the A-D conversion interrupt control register is set to "1". At the same time, bit 6 of the A-D control register 0 (A-D conversion start bit) is cleared to "0" and the comparator operation stops. The result of the comparison is stored into the bits of the comparator result register corresponding to the selected pin.

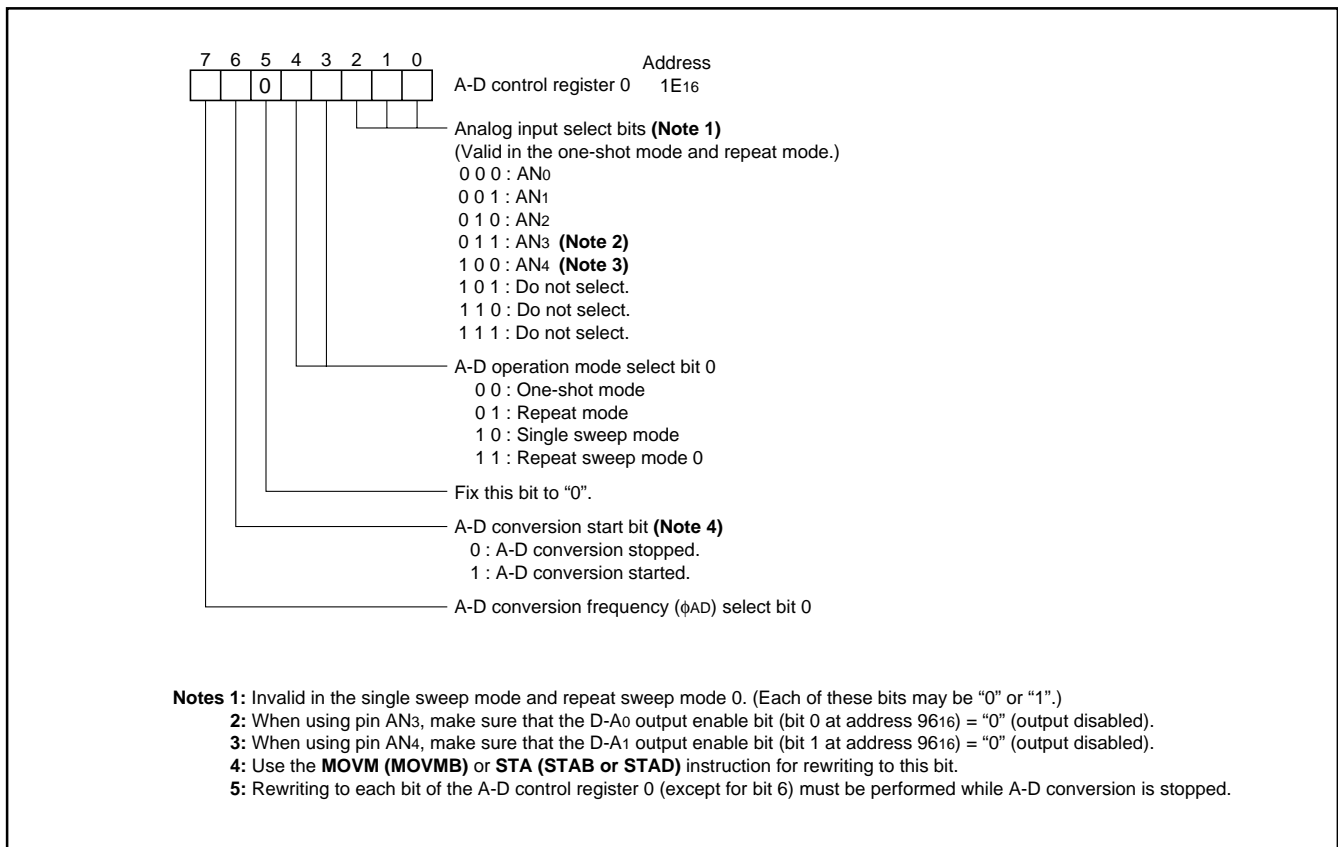


Fig. 70 Bit configuration of A-D control register 0

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

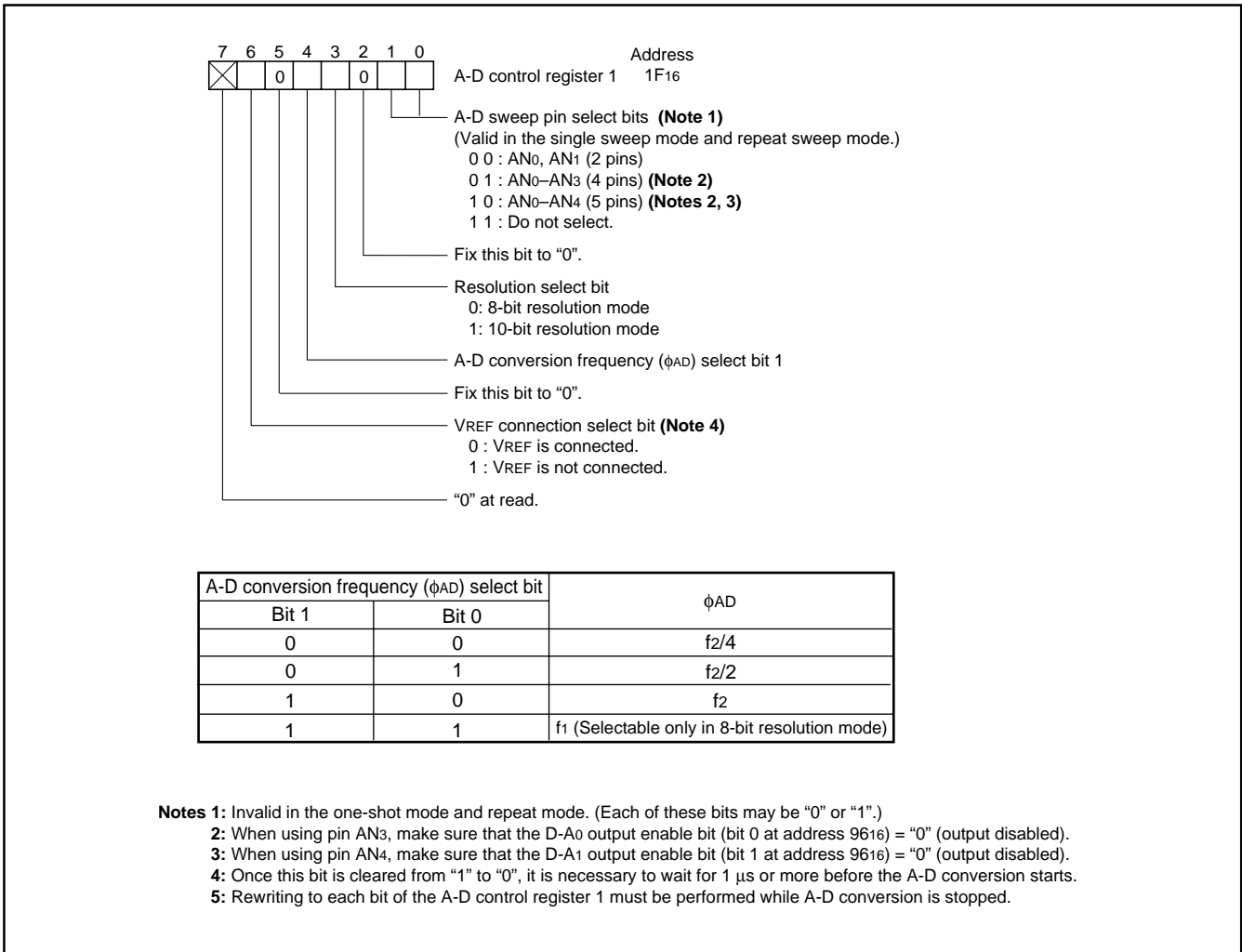


Fig. 71 Bit configuration of A-D control register 1

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

## (2) Repeat mode

Repeat mode is selected when bit 3 of the A-D control register 0 = "1" and bit 4 = "0".

The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion for 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, the A-D conversion start bit is not cleared.

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

Be sure not to write to the A-D register corresponding to the pins selected for a comparator during operation.

## (3) Single sweep mode

Single sweep mode is selected when bit 3 of the A-D control register 0 = "0" and bit 4 = "1".

In the single sweep mode, the number of analog input pins to be swept can be selected. Analog input pins are selected by bits 1 and 0 of the A-D control register 1 (address 1F16). Two pins, four pins, or five 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 AN0 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 is started when bit 6 of the A-D control register 0 (A-D conversion start bit) is set to "1". When A-D conversion for 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.

## (4) Repeat sweep mode 0

Repeat sweep mode 0 is selected when bit 3 of the A-D control register 0 = "1" and bit 4 = "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 AN0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, the A-D conversion start bit is not cleared. The A-D register can be read at any time.

Be sure not to write to the A-D register corresponding to the pins selected for a comparator during operation.

## Precaution for A-D conversion interrupts

Clear the interrupt request bit of the A-D conversion interrupt control register (bit 3 at address 7016) before using an A-D conversion interrupt. It is because this interrupt request bit is undefined just after reset.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**D-A CONVERTER**

Two independent D-A converters are included in this microcomputer, and each D-A converter adopts an 8-bit R-2R method. Figure 72 shows the block diagram of the D-A converter, and Figure 73 shows the bit configuration of the D-A control register (address 96<sub>16</sub>).

D-A conversion is performed by writing a value to the corresponding D-A register *i*. Whether to output the analog voltage or not is determined by bits 0 and 1 of the D-A control register. When any of bits 0 and 1 = "1", the corresponding pin (D-A<sub>0</sub> or D-A<sub>1</sub>) outputs the analog voltage.

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

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

*V<sub>REF</sub>* : Reference voltage

The contents of the corresponding D-A output enable bit and D-A register are cleared to "0" at reset.

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

Pin D-A<sub>*i*</sub> (*i* = 0, 1) is multiplexed with I/O port pins, analog input pins, and external interrupt input pins. When a D-A<sub>*i*</sub> output enable bit = "1" (in other words, output is enabled.), however, the corresponding pin cannot function as another I/O pin, which is multiplexed with pin D-A<sub>*i*</sub>.

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

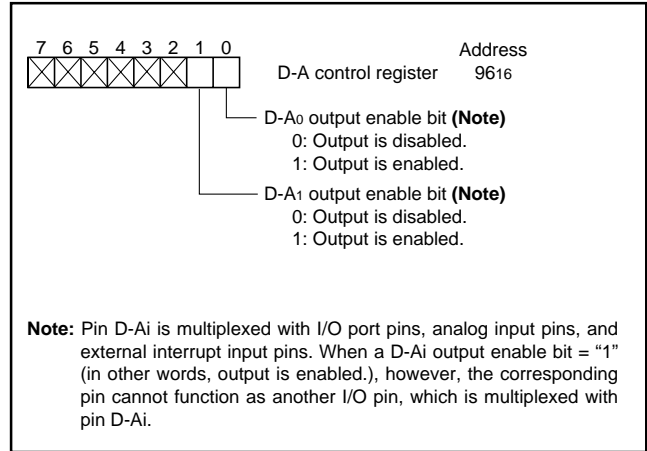


Fig. 73 Bit configuration of D-A control register

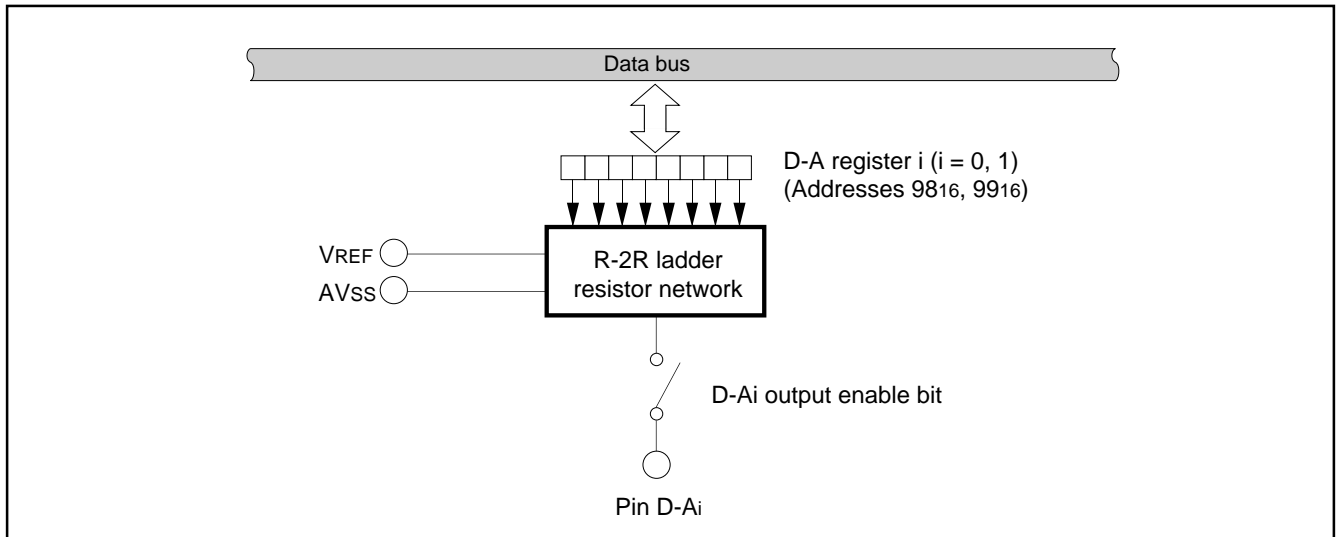


Fig. 72 Block diagram of D-A converter

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**WATCHDOG TIMER**

The watchdog timer is used to detect unexpected execution sequence caused by software runaway and others. Figure 74 shows the block diagram of the watchdog timer.

The watchdog timer consists of a 12-bit binary counter. The watchdog timer counts clock Wf32, which is obtained by dividing the peripheral devices' clock f2 by 16; or clock Wf512, which is obtained by doing it by 256. Bit 0 of the watchdog timer frequency select register (watchdog timer frequency select bit) shown in Figure 75 selects which clock is to be counted.

Wf512 is selected when this bit 0 is "0", and Wf32 is selected when bit 0 is "1". Bit 0 is cleared to "0" after reset.

FFF16 is set in the watchdog timer when "L" level voltage is applied to pin RESET, STP instruction is executed, data is written to the watchdog timer register (address 6016), or the most significant bit of the watchdog timer becomes "0".

After FFF16 is set in the watchdog timer, when the watchdog timer counts Wf32 or Wf512 by 2048 counts, the most significant bit of watchdog timer becomes "0", the watchdog timer interrupt request bit is set to "1", and FFF16 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 be-

comes "0" and an interrupt is generated.

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

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

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

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

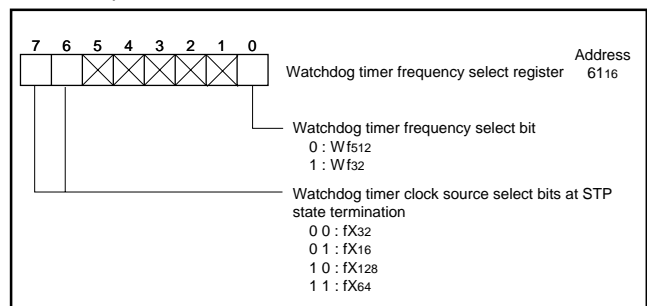


Fig. 75 Bit configuration of watchdog timer frequency select register

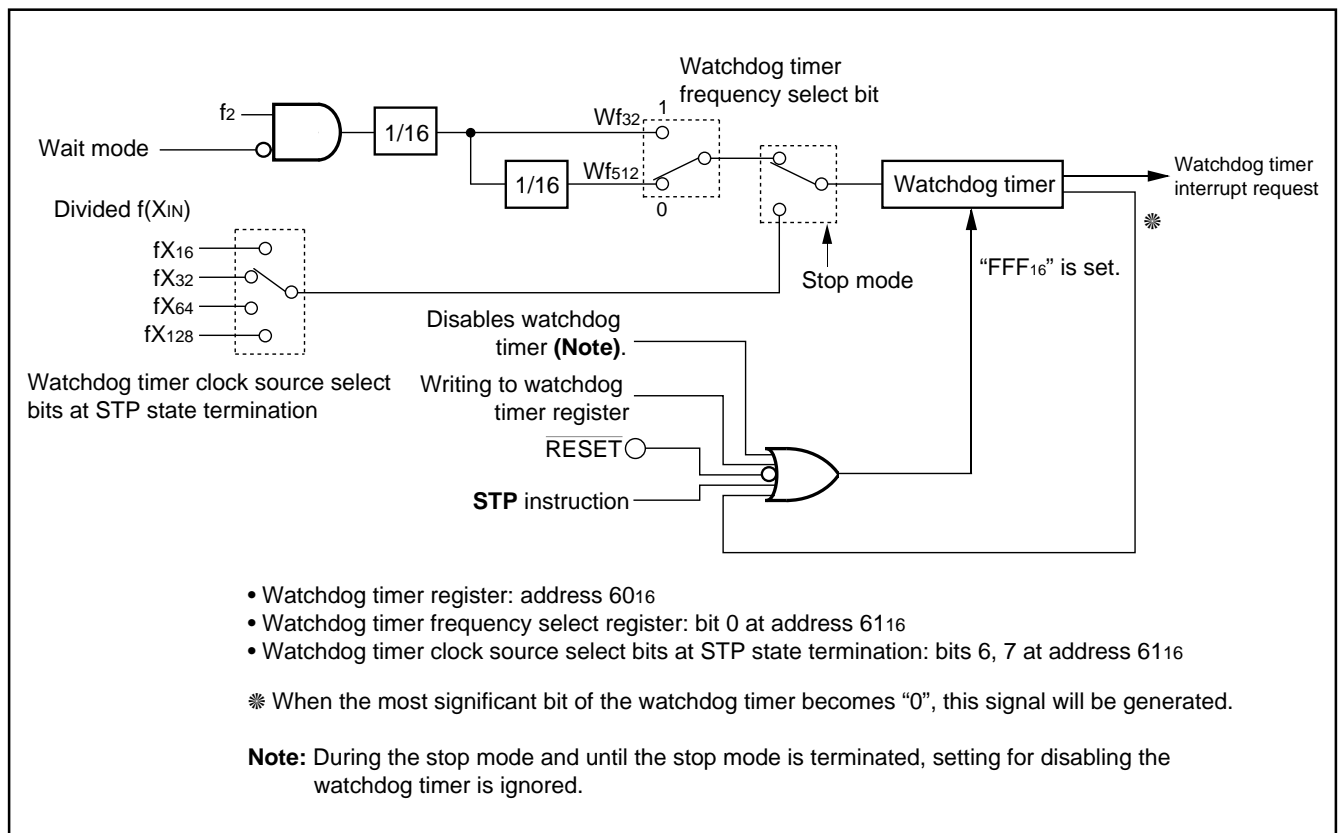


Fig. 74 Block diagram of watchdog timer

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

---

### How to disable watchdog timer

When not using the watchdog timer, it can be disabled. When the watchdog timer is disabled, its operation stops and no watchdog timer interrupt has been generated.

Setting for disabling the watchdog timer is possible by writing "7916" and "5016" to the particular function select register 2 (address 6416) sequentially with the following instructions:

- **MOVMB/STAB** instruction, or
- **MOVm/STAm** instruction (m = 1)

If any method other than above has been adopted in order to access (in other words, read/write) the particular function select register 2, the watchdog timer will not be disabled until reset operation is performed. (Also, reset is the only one method to remove the setting for disabling the watchdog timer.)

Moreover, this setting for disabling the watchdog timer is ignored at return from the STP mode, and the watchdog timer operates. (For details, refer to the section on the standby function.)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**INPUT/OUTPUT PINS**

Ports P1, P2, and P5 through P7 all have the direction register, and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding bit of direction register is "1", and an input pin when it is "0".

Also, each bit of the port P6 direction register can be cleared to "0" by inputting a falling edge to pin  $\overline{P6OUTCUT}$  or by executing instructions.

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

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

Each of Figures 76 and 77 shows the block diagram for each port pin.

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

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

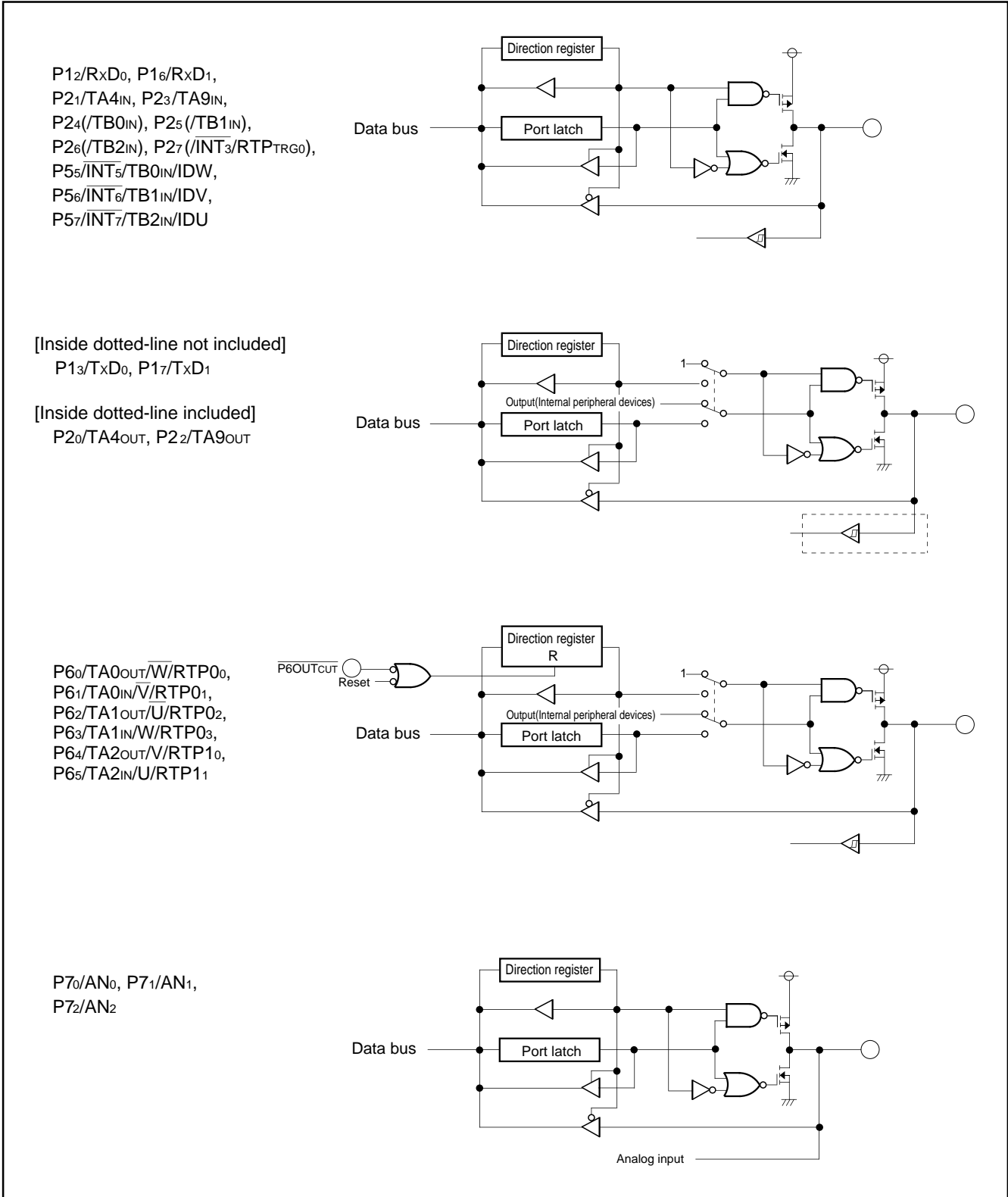


Fig. 76 Block diagram for each port pin (1)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

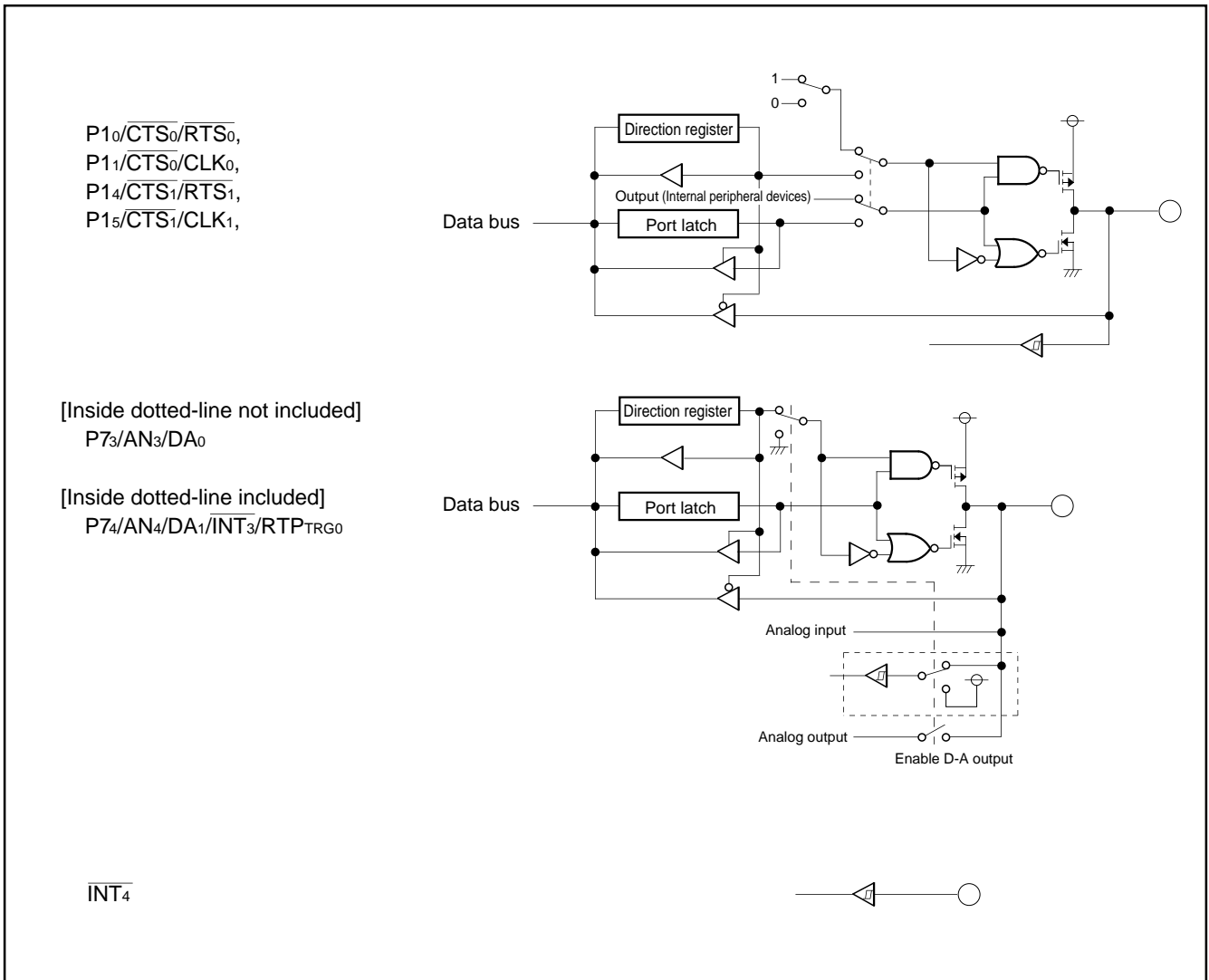


Fig. 77 Block diagram for each port pin (2)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**RESET CIRCUIT**

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

- A23 to A16 = 0016
- A15 to A8 = Contents at address FFFF16
- A7 to A0 = Contents at address FFFE16

Figures 78 and 79 show the microcomputer internal register's status at reset, and Figure 80 shows an operation example of the reset circuit. Apply "L" level voltage to pin  $\overline{\text{RESET}}$  for a period (10  $\mu\text{s}$  or more) under the following conditions:

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

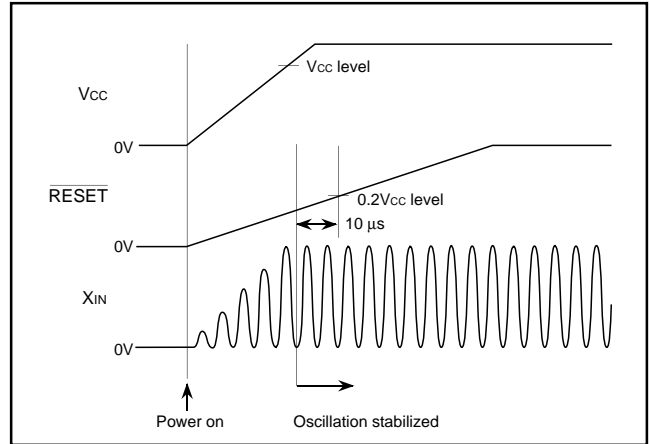


Fig. 80 Operation example of reset circuit (Note that proper evaluation is necessary in the system development stage.)

	Address			Address																	
Waveform output mode register	(A6 <sub>16</sub> )...	00 <sub>16</sub>		Timer A5 interrupt control register	(F5 <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0								
×	×	×	×	0	0	0	0														
Three-phase output data register 0	(A8 <sub>16</sub> )...	00 <sub>16</sub>		Timer A6 interrupt control register	(F6 <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0								
×	×	×	×	0	0	0	0														
Three-phase output data register 1	(A9 <sub>16</sub> )...	00 <sub>16</sub>		Timer A7 interrupt control register	(F7 <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0								
×	×	×	×	0	0	0	0														
Position-data-retain function control register	(AA <sub>16</sub> )...	<table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0		Timer A8 interrupt control register	(F8 <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0
×	×	×	×	0	0	0	0														
×	×	×	×	0	0	0	0														
Serial I/O pin control register	(AC <sub>16</sub> )...	<table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0		Timer A9 interrupt control register	(F9 <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0
×	×	×	×	0	0	0	0														
×	×	×	×	0	0	0	0														
Port P2 pin function control register	(AE <sub>16</sub> )...	<table border="1"><tr><td>×</td><td>×</td><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	×	×	0	0	0	0		$\overline{\text{INT}}_5$ interrupt control register	(FD <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	0	0	0	0	0	0
×	×	×	×	0	0	0	0														
×	×	0	0	0	0	0	0														
Clock control register 0	(BC <sub>16</sub> )...	0 0 0 1 0 1 1 1		$\overline{\text{INT}}_6$ interrupt control register	(FE <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	0	0	0	0	0	0								
×	×	0	0	0	0	0	0														
Up-down register 1	(C4 <sub>16</sub> )...	00 <sub>16</sub>		$\overline{\text{INT}}_7$ interrupt control register	(FF <sub>16</sub> )... <table border="1"><tr><td>×</td><td>×</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	×	×	0	0	0	0	0	0								
×	×	0	0	0	0	0	0														
Timer A5 mode register	(D6 <sub>16</sub> )...	00 <sub>16</sub>		Processor status register PS	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>?</td><td>?</td><td>0</td><td>0</td><td>0</td><td>1</td><td>?</td><td>?</td></tr></table>	0	0	0	?	?	0	0	0	1	?	?					
0	0	0	?	?	0	0	0	1	?	?											
Timer A6 mode register	(D7 <sub>16</sub> )...	00 <sub>16</sub>		Program bank register PG	00 <sub>16</sub>																
Timer A7 mode register	(D8 <sub>16</sub> )...	00 <sub>16</sub>		Program counter PCH	Contents at address FFFF <sub>16</sub>																
Timer A8 mode register	(D9 <sub>16</sub> )...	00 <sub>16</sub>		Program counter PCL	Contents at address FFFE <sub>16</sub>																
Timer A9 mode register	(DA <sub>16</sub> )...	00 <sub>16</sub>		Direct page registers DPR0 to DPR3	0000 <sub>16</sub>																
Comparator function select register 0	(DC <sub>16</sub> )...	00 <sub>16</sub>		Data bank register DT	00 <sub>16</sub>																
Comparator result register 0	(DE <sub>16</sub> )...	00 <sub>16</sub>		Stack pointer	FFF <sub>16</sub>																

**Note:** The contents of the other registers and RAM are undefined at reset and must be initialized by software.

Fig. 79 Microcomputer internal register's status at reset (2)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

	Address		Address	
Port P1 direction register	(05 <sub>16</sub> )...	00 <sub>16</sub>	Processor mode register 0	(5E <sub>16</sub> )... 0 0 0 0 1 0 0 0
Port P2 direction register	(08 <sub>16</sub> )...	00 <sub>16</sub>	Processor mode register 1	(5F <sub>16</sub> )... 0 0 0 0 0 0 0 1
Port P5 direction register	(0D <sub>16</sub> )...	0 0 0 $\times$ $\times$ $\times$ $\times$ $\times$ $\times$	Watchdog timer	(60 <sub>16</sub> )... FFF <sub>16</sub>
Port P6 direction register	(10 <sub>16</sub> )...	$\times$ $\times$ 0 0 0 0 0 0	Watchdog timer frequency select register	(61 <sub>16</sub> )... 0 0 $\times$ $\times$ $\times$ $\times$ $\times$ $\times$ 0
Port P7 direction register	(11 <sub>16</sub> )...	$\times$ $\times$ $\times$ 0 0 0 0 0 0	Particular function select register 0	(62 <sub>16</sub> )... 0 $\times$ 0 $\times$ $\times$ $\times$ $\times$ 0 0
A-D control register 0	(1E <sub>16</sub> )...	0 0 0 0 0 ? ? ?	Particular function select register 1	(63 <sub>16</sub> )... $\times$ 0 $\times$ 0 0 0 (Note 2)
A-D control register 1	(1F <sub>16</sub> )...	$\times$ 0 0 0 0 0 ? ?	Debug control register 0	(66 <sub>16</sub> )... 1 (Note 2)
UART 0 transmit/receive mode register	(30 <sub>16</sub> )...	00 <sub>16</sub>	Debug control register 1	(67 <sub>16</sub> )... 0 0 0 $\times$ 0 0 0 (Note 2)
UART 1 transmit/receive mode register	(38 <sub>16</sub> )...	00 <sub>16</sub>	$\overline{\text{INT}}_3$ interrupt control register	(6E <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
UART 0 transmit/receive control register 0	(34 <sub>16</sub> )...	0 0 0 0 1 0 0 0	$\overline{\text{INT}}_4$ interrupt control register	(6F <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
UART 1 transmit/receive control register 0	(3C <sub>16</sub> )...	0 0 0 0 1 0 0 0	A-D conversion interrupt control register	(70 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ ? 0 0 0
UART 0 transmit/receive control register 1	(35 <sub>16</sub> )...	0 0 0 0 0 0 1 0	UART 0 transmit interrupt control register	(71 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
UART 1 transmit/receive control register 1	(3D <sub>16</sub> )...	0 0 0 0 0 0 1 0	UART 0 receive interrupt control register	(72 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Count start register 0	(40 <sub>16</sub> )...	00 <sub>16</sub>	UART 1 transmit interrupt control register	(73 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Count start register 1	(41 <sub>16</sub> )...	$\times$ $\times$ $\times$ 0 0 0 0 0 0	UART 1 receive interrupt control register	(74 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
One-shot start register 0	(42 <sub>16</sub> )...	0 $\times$ $\times$ 0 0 0 0 0 0	Timer A0 interrupt control register	(75 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
One-shot start register 1	(43 <sub>16</sub> )...	0 $\times$ $\times$ 0 0 0 0 0 0	Timer A1 interrupt control register	(76 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Up-down register 0	(44 <sub>16</sub> )...	00 <sub>16</sub>	Timer A2 interrupt control register	(77 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Timer A clock frequency select register	(45 <sub>16</sub> )...	$\times$ $\times$ $\times$ $\times$ $\times$ $\times$ 0 0	Timer A3 interrupt control register	(78 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Timer A0 mode register	(56 <sub>16</sub> )...	00 <sub>16</sub>	Timer A4 interrupt control register	(79 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Timer A1 mode register	(57 <sub>16</sub> )...	00 <sub>16</sub>	Timer B0 interrupt control register	(7A <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Timer A2 mode register	(58 <sub>16</sub> )...	00 <sub>16</sub>	Timer B1 interrupt control register	(7B <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Timer A3 mode register	(59 <sub>16</sub> )...	00 <sub>16</sub>	Timer B2 interrupt control register	(7C <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ 0 0 0 0
Timer A4 mode register	(5A <sub>16</sub> )...	00 <sub>16</sub>	D-A control register	(96 <sub>16</sub> )... $\times$ $\times$ $\times$ $\times$ $\times$ $\times$ 0 0
Timer B0 mode register	(5B <sub>16</sub> )...	0 0 ? 0 0 0 0 0	D-A register 0	(98 <sub>16</sub> )... 00 <sub>16</sub>
Timer B1 mode register	(5C <sub>16</sub> )...	0 0 ? 0 0 0 0 0	D-A register 1	(99 <sub>16</sub> )... 00 <sub>16</sub>
Timer B2 mode register	(5D <sub>16</sub> )...	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:** At power-on reset, these bits are clear to "0". At hardware or software reset, on the other hand, these bits retain the value just before reset.

Fig. 78 Microcomputer internal register's status at reset (1)

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**OSCILLATION CIRCUIT**

An oscillation circuit locates between pins XIN and XOUT, and Figure 81 shows a circuit example with an external ceramic resonator or quartz-crystal oscillator. The constants such as capacitance etc. depend on an oscillator. Therefore, for these constants, adopt the resonator/oscillator manufacturer's recommended values.

Figure 82 shows a circuit example with an external clock source. When an external clock is input, be sure to leave pin XOUT open. Also, in this case, when the external clock input select bit (bit 1 of the particular function select register 0; See Figure 86.) is set to "1", the oscillation circuit stops its operation and resumes the current dissipation. Moreover, this bit has another function, which selects the return condition from the stop mode. For details, refer to the section on the standby function.

On the other hand, the PLL (Phase Locked Loop) frequency multiplier (hereafter, referred as PLL circuit.) is included, also. This PLL circuit uses a clock input from pin XIN and generates a multiplied clock. When using the PLL circuit, be sure to connect pin VCONT with an external filter circuit. (See Figure 83.) When not using the PLL circuit, be sure to leave pin VCONT open.

When not using the PLL circuit, be sure to clear the PLL circuit operation enable bit (bit 1 of the clock control register 0; See Figure 85.), so that the PLL circuit will stop its operation.

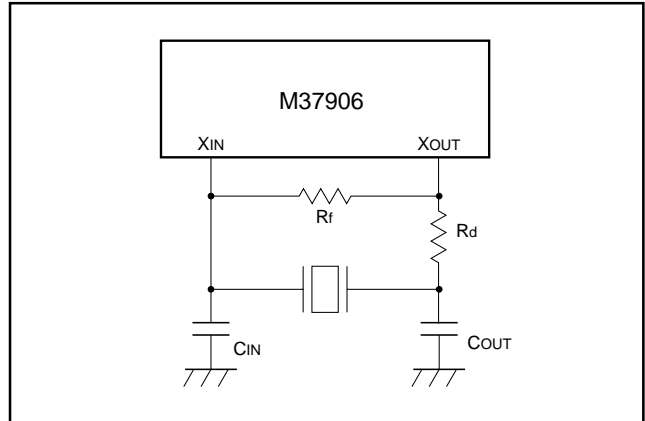


Fig. 81 Circuit example with external ceramic resonator or quartz-crystal oscillator

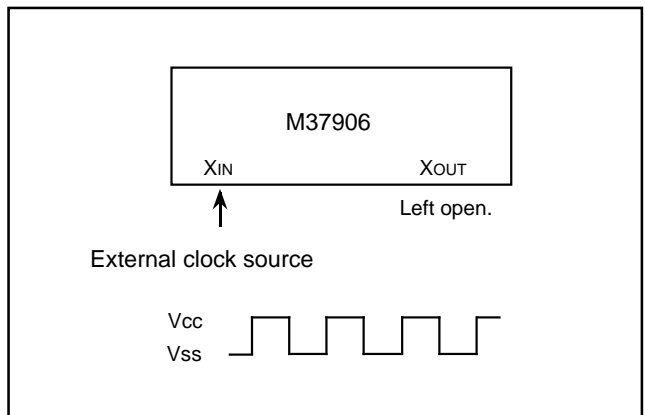


Fig. 82 Circuit example with external clock source

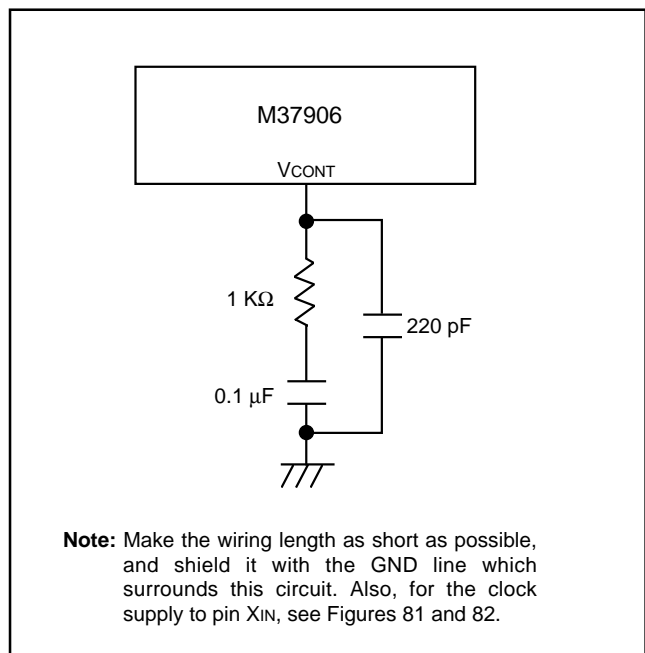


Fig. 83 Circuit example of connection with pin VCONT when PLL circuit used

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**CLOCK GENERATING CIRCUIT**

Figure 84 shows the block diagram of the clock generating circuit. The clock generating circuit consists of the clock oscillation circuit, PLL frequency multiplier (PLL circuit), system clock switch circuit, peripheral devices' clock switch circuit, clock divider, standby control circuit, etc. As control registers for the clock generating circuit, also, the clock control register 0 (address BC16), particular function select register 0 (address 6216) are provided. (See Figures 85 and 86.)

As shown in Figure 84, clocks used in the CPU, BIU, peripheral devices, watchdog timer (in other words, clocks  $\phi_{CPU}$ ,  $\phi_{BIU}$ ,  $f_1$  to  $f_{4096}$ ,  $Wf_{32}$ ,  $Wf_{512}$ ) are made from system clock  $f_{sys}$ . System clock  $f_{sys}$  can be selected between  $f_{XIN}$  (in other words, a clock input from pin  $XIN$ ) and  $f_{PLL}$  (in other words, an output clock generated by the PLL circuit).

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

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

Bits 2 and 3 of the clock control register 0 (the PLL multiplication ratio select bits) select the ratio of  $f_{PLL}/f_{XIN}$ . The PLL multiplication ratio

must be set so that the frequency of  $f_{PLL}$  must be in the range from 10 MHz to 20 MHz. At reset, the PLL multiplication ratio select bits become "0,1" (X 2). The change of the multiplication ratio must be performed while input clock  $f_{XIN}$  is set as system clock. (In this case, bit 5 of the clock control register 0 = "0".) After that, be sure to wait that the operation-stabilizing time of the PLL circuit has passed, and switch the system clock to  $f_{PLL}$ . (In other words, set bit 5 to "1".) Note that, after reset, the PLL multiplication ratio select bits are allowed to be changed only once.

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

Table 9 lists the  $f_{sys}$  selection.

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

Table 10 lists the internal peripheral devices' operation clock frequency. At reset, these bits become "0, 0".

Table 9.  $f_{sys}$  selection

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

**Note:** The PLL multiplication ratio must be set so that the frequency of  $f_{PLL}$  must be in the range from 10 MHz to 20 MHz.  $f(XIN)$  means the frequency of the input clock from pin  $XIN$  ( $f_{XIN}$ ). After reset, the PLL multiplication ratio select bits are allowed to be changed only once.

Table 10. Internal peripheral devices' operation clock frequency

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

**Note:** When selecting the peripheral devices' clock select bits 1, 0 = "012", be sure that system clock  $f_{sys}$  does not exceed 10 MHz.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

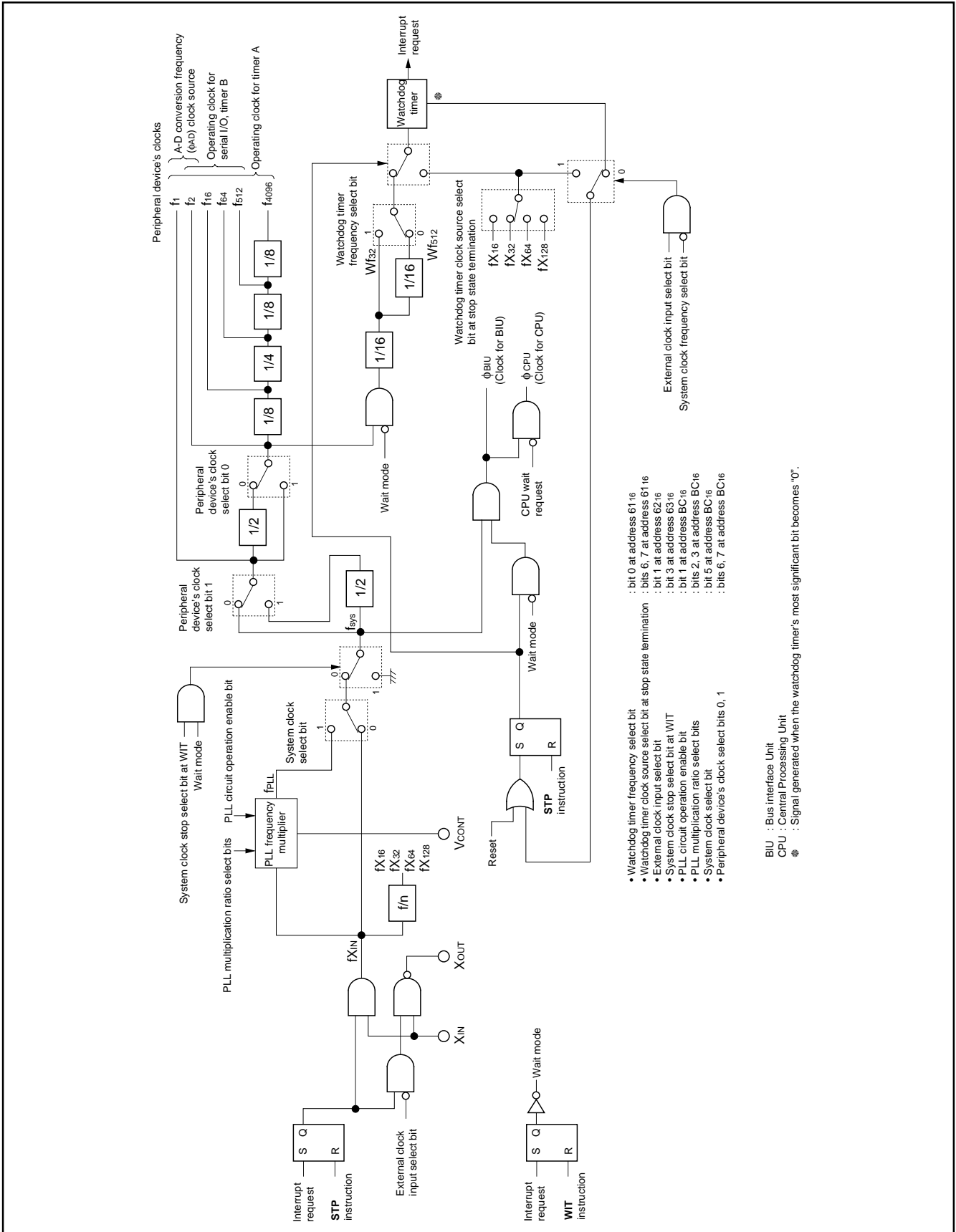


Fig. 84 Block diagram of clock generating circuit

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

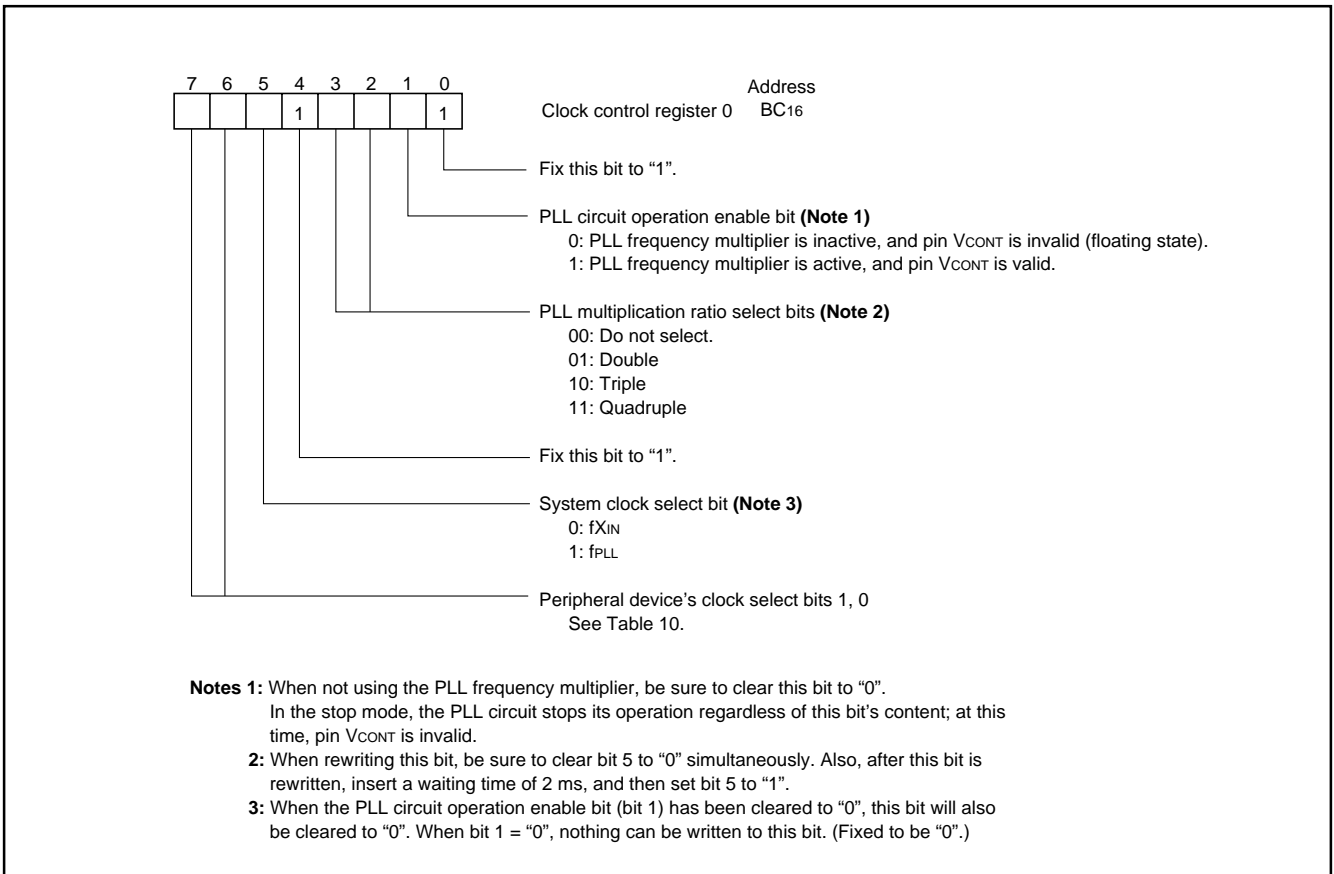


Fig. 85 Bit configuration of clock control register 0

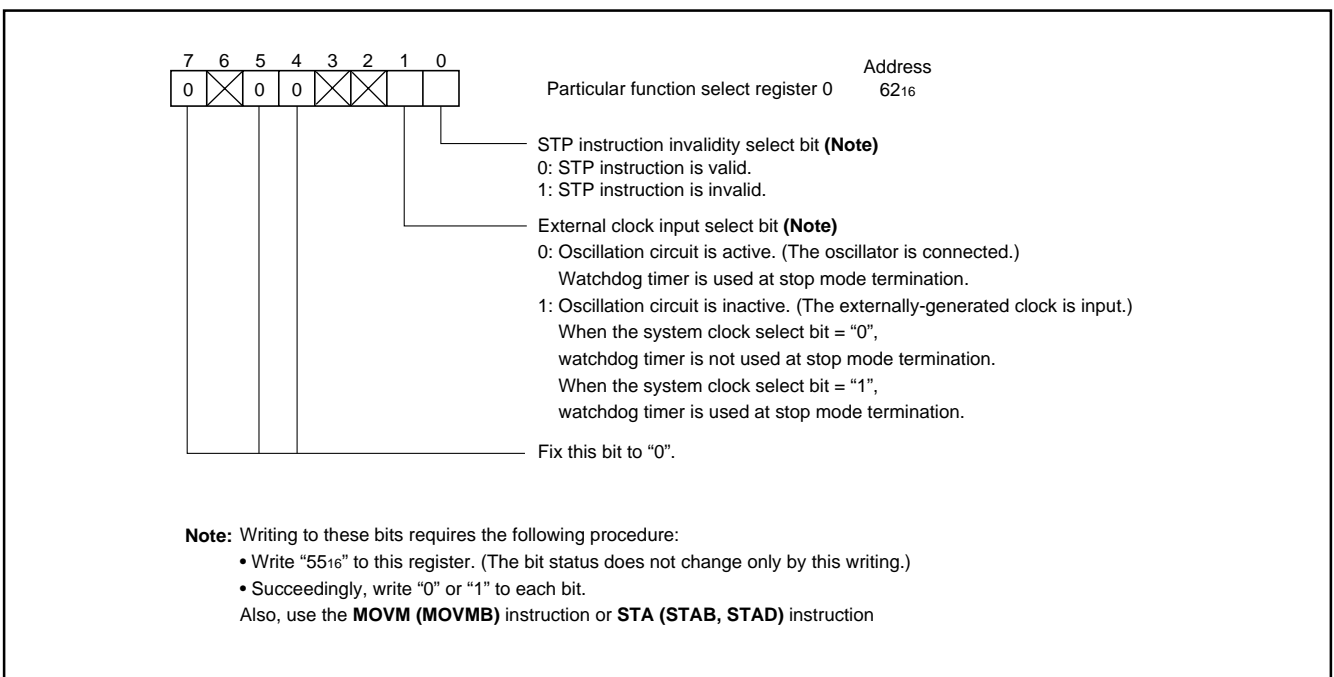


Fig. 86 Bit configuration of particular function select register 0

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**STANDBY FUNCTION**

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

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 needs to be higher than the processor interrupt priority level (IPL) of the routine where the STP or WIT instruction will be executed.

Figures 86 through 88 show the bit configurations of the particular function select registers 0, 1, and watchdog timer frequency select register respectively. Setting the STP instruction invalidity select bit (bit 0 of the particular function select register 0) to "1" invalidates the STP instruction, and the STP instruction will be ignored. Since the above bit is cleared to "0" after reset is removed, however, the STP instruction is valid.

The STP- or the WIT-instruction-execution status bit (bit 0 or 1 of the particular function select register 1) is set to "1" by the execution of the STP or the WIT instruction, and so, after the STP or WIT mode has been terminated, each bit will indicate that the STP or WIT instruction has been executed. Accordingly, each of these bits must be cleared to "0" by software at termination of the STP or the WIT mode. Table 11 explains the microcomputer's operation in the STP and WIT modes.

**STP mode**

The execution of the STP instruction makes the oscillation circuit and PLL circuit inactive. It also makes the following inactive: input clock fXIN, system clock fsys, φBIU, φCPU, and peripheral devices' clocks f1 to f4096, Wf32 and Wf512 with the "L" state, and divide clocks fX16 to fX128 with the "H" state. In the watchdog timer, "FFF16" is automatically set. As shown in Figure 84, any one of divide clocks fX16 to fX128, which is selected by the watchdog timer clock source select bits at STP termination (bits 6 and 7 of the watchdog timer frequency select register), becomes the watchdog timer's clock source.

In the STP mode, the A-D converter and watchdog timer, which uses peripheral devices' clocks f1 to f4096, Wf32 and Wf512, are stopped. At this time, timers A and B can be active only in the event counter mode, and serial I/O communication is active while an external clock is selected.

The STP mode is terminated by acceptance of an interrupt request or reset, and the oscillation circuit and PLL circuit restart their operations. Input clock fXIN, system clock fsys, and peripheral devices' clocks f1 to f4096, Wf32 and Wf512 are also supplied.

When the STP mode is terminated by reset, supply of φBIU and φCPU starts immediately after the oscillation circuit and PLL circuit restart their operations. Therefore, the reset input must be raised "H" after the operation-stabilizing time for these circuits has passed.

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

- (1) The watchdog timer is used in order to measure the period from the operation restart of the oscillation circuit and PLL circuit until the supply start of φBIU and φCPU.
- (2) The supply of φBIU and φCPU is started immediately after the operation restart of the oscillation circuit and PLL circuit.

Table 11. Microcomputer's operation in STP and WIT modes

Mode	System clock stop select bit at WIT	Operations of function while WIT, STP modes					
		Oscillation circuit	PLL circuit	fsys, φ1, f1 to f4096	Wf32, Wf512	φBIU, φCPU	Peripheral devices using f1 to f4096, Wf32, Wf512
STP	—	Inactive	Inactive	Inactive ("L")	Inactive ("L")	Inactive ("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: Inactive. (Watchdog timer: Inactive.)
WIT	"0"	Active (Note 1)	Active (Note 2)	Active	Inactive ("L")	Inactive ("L")	Timers A, B, Serial I/O, A-D converter: Operation is enabled. (Watchdog timer: Inactive.)
	"1"	Active (Note 1)	Active (Note 2)	Inactive ("L")	Inactive ("L")	Inactive ("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: Inactive. (Watchdog timer: Inactive.)

**Notes 1:** When the external clock input select bit = "1", the oscillation circuit is inactive. Also, clock input from pin XIN is allowed.  
**2:** When the PLL operation enable bit = "0", the PLL circuit is inactive.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

When the external clock input select bit (bit 1 of the particular function select register 0) = "0" or the system clock select bit (bit 5 of the clock control register 0) = "1", the watchdog timer will start counting down with one of the above divide clocks, fX16 to fX128, after the oscillation circuit and PLL circuit have been restarted their operations owing to an interrupt. The most significant bit of the watchdog timer reaching "0", supply of  $\phi_{BIU}$  and  $\phi_{CPU}$  restarts.

On the other hand, when the external clock input select bit = "1" and the system clock select bit = "0", supply of  $\phi_{BIU}$  and  $\phi_{CPU}$  will restart immediately after the oscillation circuit and PLL circuit have been restarted their operations owing to an interrupt. (In actual fact, after the selected one of the above divide clocks, fX16 to fX128, has been changed from "H" to "L", this supply will restart.)

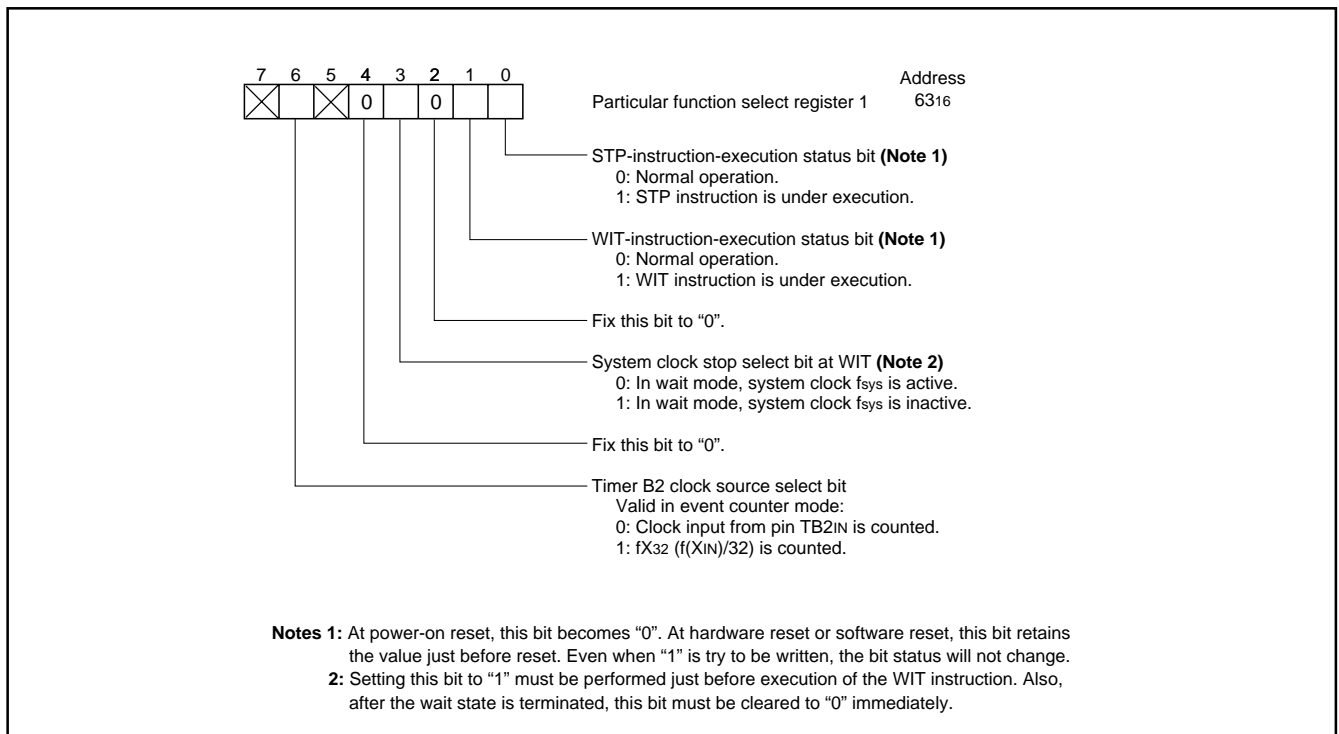


Fig. 87 Bit configuration of particular function select register 1

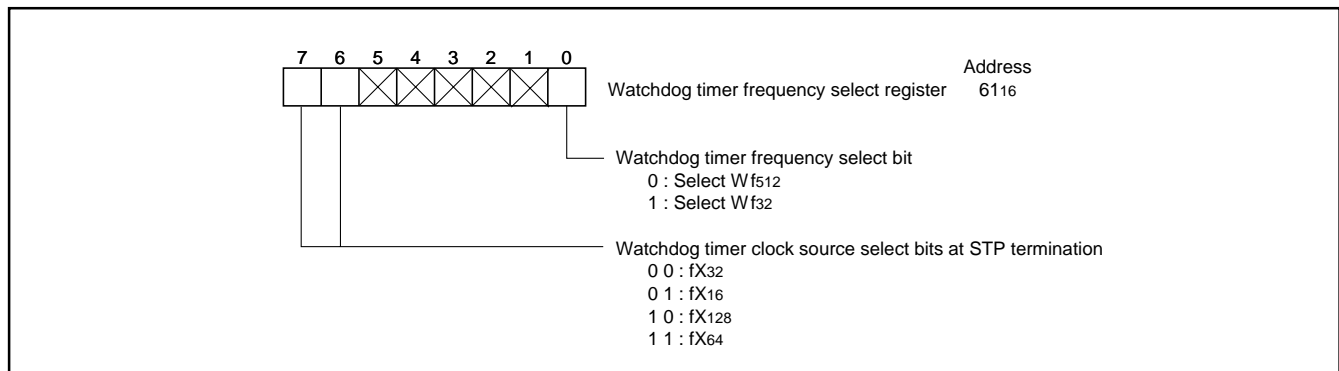


Fig. 88 Bit configuration of watchdog timer frequency select register

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

---

### WIT mode

When the WIT instruction is executed with the system clock stop select bit at WIT (bit 3 of the particular function select register 1 in Figure 87) being "0",  $\phi_{BIU}$ ,  $\phi_{CPU}$ , and divide clocks  $Wf_{32}$  and  $Wf_{512}$  are inactive with the "L" state. However, the oscillation circuit, PLL circuit, input clock  $f_{XIN}$ , system clock  $f_{sys}$ ,  $\phi_1$ , and peripheral devices' clocks  $f_1$  to  $f_{4096}$  remain active. Therefore, BIU and CPU are inactive, whereas timers A and B, serial I/O, and the A-D converter, which use the peripheral devices' clocks  $f_1$  to  $f_{4096}$ , are still active. Note that the watchdog timer is inactive.

On the other hand, when the WIT instruction is executed with the system clock stop select bit at WIT being "1", the oscillation circuit, PLL circuit, and input clock  $f_{XIN}$  are active, while system clock  $f_{sys}$ ,  $\phi_{BIU}$ ,  $\phi_{CPU}$ , and peripheral devices' clocks are inactive. As a result, the A-D converter and watchdog timer, which use peripheral devices' clocks  $f_1$  to  $f_{4096}$ ,  $Wf_{32}$  and  $Wf_{512}$ , become inactive. At this time, timers A and B are active only in the event counter mode, and serial I/O communication is active only while an external clock is selected. If the internal peripheral devices are not used in the WIT mode, the latter is better because the current dissipation is more saved. Note that the system clock stop select bit at WIT needs 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  $\phi_{BIU}$  and  $\phi_{CPU}$  will restart. Since the oscillation circuit, PLL circuit, and clock input  $f_{XIN}$  are active in the WIT mode, an interrupt processing can be executed just after the WIT mode termination.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

## POWER SAVING FUNCTION

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

### (1) Inactive system clock in wait mode

In the wait mode, if the internal peripheral devices need not to operate, when the system clock stop select bit at WIT (bit 3 of the particular function select register 1) = "1", both of system clock  $f_{sys}$  and peripheral devices' clock are inactive, and the power dissipation can be saved.

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

### (2) Inactive oscillation circuit

When an externally-generated stable clock is input to pin XIN, 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 XIN and XOUT is inactive. At this time, the output level at pin XOUT is fixed to "H". When not using  $f_{PLL}$ , also, the supply of  $\phi_{BIU}$  and  $\phi_{CPU}$  restarts just after the microcomputer returns from the stop mode, owing to an interrupt request occurrence. Therefore, an instruction can be executed just after the termination of the stop mode. For details, refer to the section on the clock generating circuit and standby function.

### (3) Disconnection from pin VREF

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

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**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 89 shows the block diagram of the debug function. Figures 90 and 91 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.

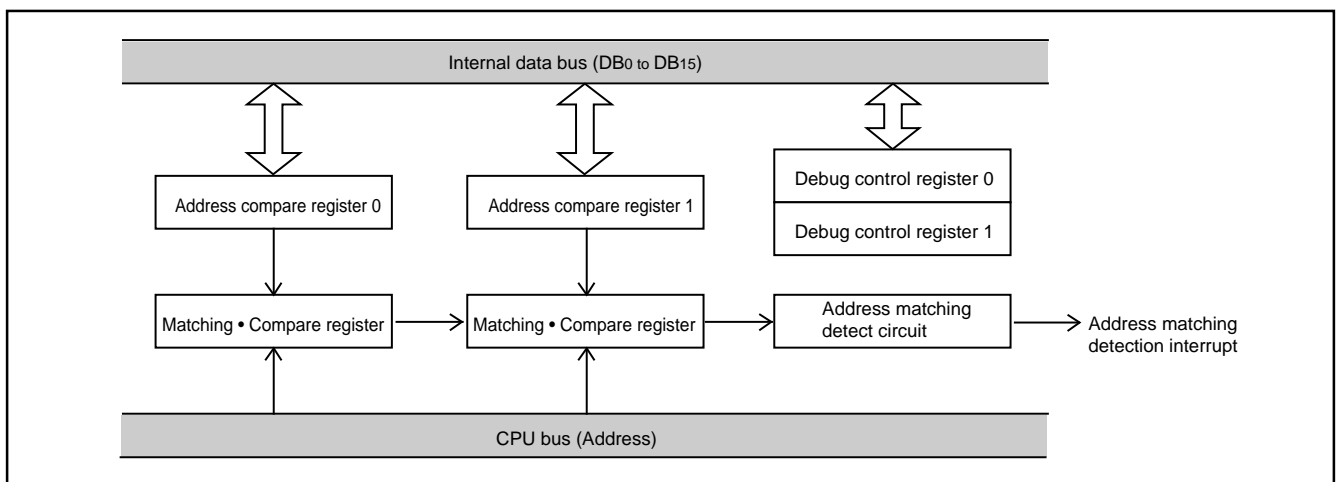


Fig. 89 Block diagram of debug function

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

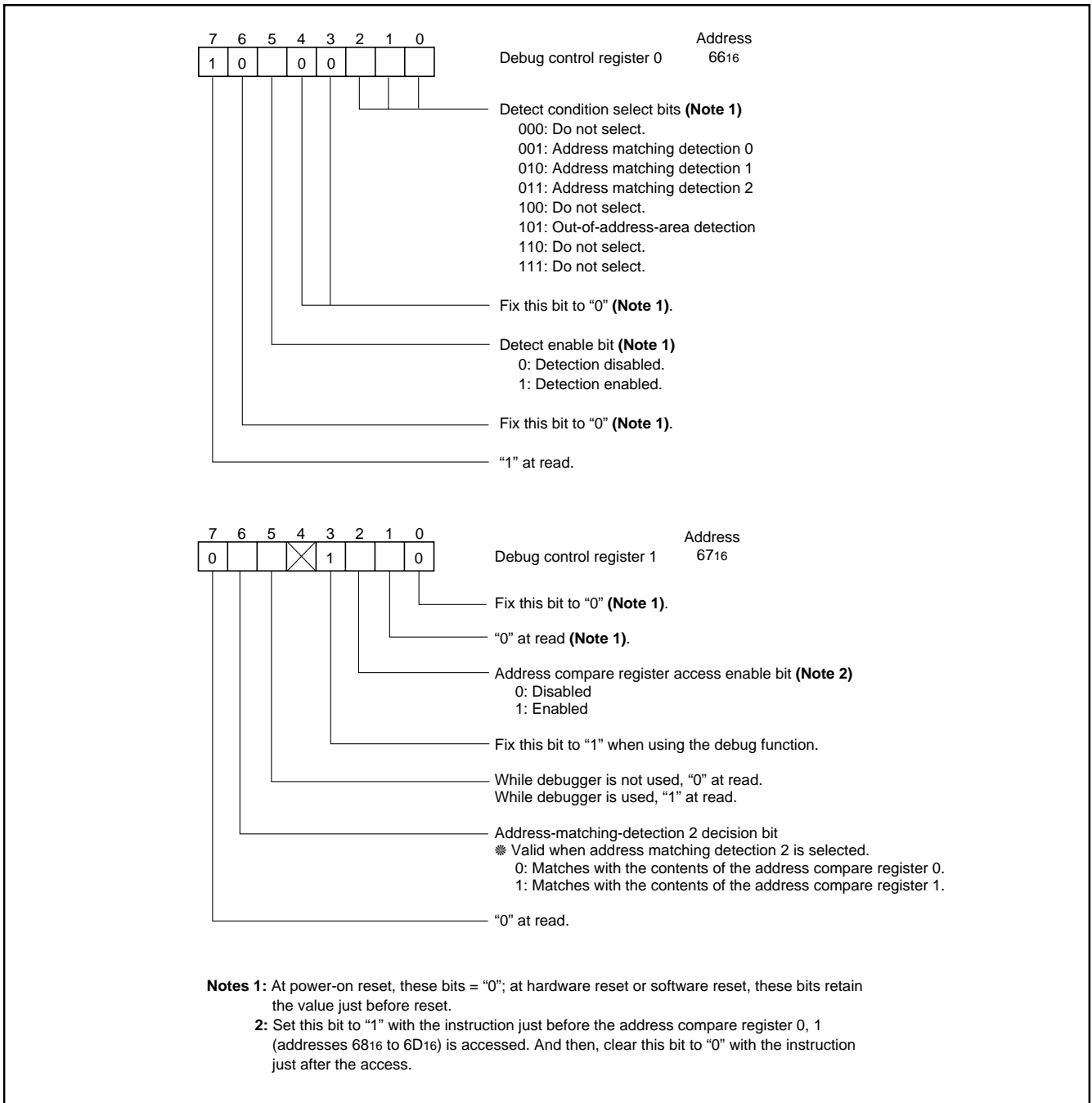


Fig. 90 Bit configuration of debug control register 0, 1

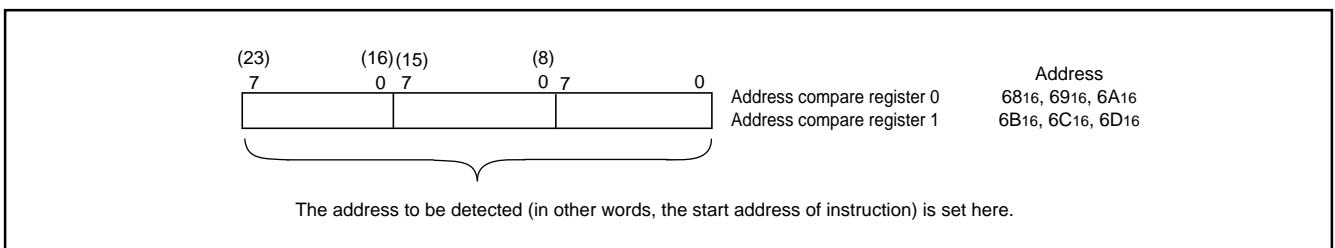


Fig. 91 Bit configuration of address compare register 0, 1

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Ratings	Unit
V <sub>CC</sub>	Power source voltage	-0.3 to 6.5	V
AV <sub>CC</sub>	Analog power source voltage	-0.3 to 6.5	V
V <sub>I</sub>	Input voltage P10-P17, P20-P27, P50-P57, P60-P65, P70-P74, P6OUTCUT, VCONT, VREF, XIN, RESET, BYTE, MD0, MD1	-0.3 to V <sub>CC</sub> +0.3	V
V <sub>O</sub>	Output voltage P10-P17, P20-P27, P50-P57, P60-P65, P70-P74, XOUT	-0.3 to V <sub>CC</sub> +0.3	V
P <sub>d</sub>	Power dissipation	300	mW
T <sub>opr</sub>	Operating ambient temperature	-20 to 85	°C
T <sub>stg</sub>	Storage temperature	-40 to 150	°C

**RECOMMENDED OPERATING CONDITIONS** (V<sub>CC</sub> = 5 V, T<sub>a</sub> = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V <sub>CC</sub>	Power source voltage	4.5	5.0	5.5	V
AV <sub>CC</sub>	Analog power source voltage		V <sub>CC</sub>		V
V <sub>SS</sub>	Power source voltage		0		V
AV <sub>SS</sub>	Analog power source voltage		0		V
V <sub>IH</sub>	High-level input voltage P10-P17, P20-P27, P50-P57, P60-P65, P70-P74, P6OUTCUT, XIN, RESET, MD0, MD1	0.8 V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	Low-level input voltage P10-P17, P20-P27, P50-P57, P60-P65, P70-P74, P6OUTCUT, XIN, RESET, MD0, MD1	0		0.2 V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	High-level peak output current P10-P17, P20-P27, P50-P57, P60-P65, P70-P74			-10	mA
I <sub>OH(avg)</sub>	High-level average output current P10-P17, P20-P27, P50-P57, P60-P65, P70-P74			-5	mA
I <sub>OL(peak)</sub>	Low-level peak output current P10-P17, P20-P27, P50-P57, P70-P74			10	mA
I <sub>OL(peak)</sub>	Low-level peak output current P60-P65			20	mA
I <sub>OL(avg)</sub>	Low-level average output current P10-P17, P20-P27, P50-P57, P70-P74			5	mA
I <sub>OL(avg)</sub>	Low-level average output current P60-P65			15	mA
f(XIN)	External clock input frequency ( <b>Note 1</b> )			20	MHz
f(f <sub>sys</sub> )	System clock frequency			20	MHz

- Notes 1:** When using the PLL frequency multiplier, be sure that f(f<sub>sys</sub>) = 20 MHz or less.  
**2:** The average output current is the average value of an interval of 100 ms.  
**3:** The sum of I<sub>OL(peak)</sub> must be 110 mA or less, the sum of I<sub>OH(peak)</sub> must be 80 mA or less.

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**DC ELECTRICAL CHARACTERISTICS** ( $V_{CC} = 5\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $T_a = -20\text{ to }85\text{ }^\circ\text{C}$ ,  $f(f_{sys}) = 20\text{ MHz}$ , unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
VOH	High-level output voltage P10–P17, P20–P27, P55–P57, P60–P65, P70–P74	I <sub>OH</sub> = –10 mA	3			V
VOL	Low-level output voltage P10–P17, P20–P27, P55–P57, P60–P65, P70–P74	I <sub>OL</sub> = 10 mA			2	V
VT+ –VT–	Hysteresis TA0IN–TA2IN, TA4IN, TA9IN, TA0OUT–TA2OUT, TA4OUT, TA9OUT, TB0IN–TB2IN, INT3–INT7, CTS0, CTS1, CLK0, CLK1, RxD0, RxD1, RTPTRG0, P6OUTCUT		0.4		1	V
VT+ –VT–	Hysteresis RESET		0.5		1.5	V
VT+ –VT–	Hysteresis XIN		0.1		0.3	V
I <sub>IH</sub>	High-level input current P10–P17, P20–P27, P50–P57, P60–P65, P70–P74, P6OUTCUT, XIN, RESET, MD0, MD1	V <sub>I</sub> = 5.0 V			5	μA
I <sub>IL</sub>	Low-level input current P10–P17, P20–P27, P50–P57, P60–P65, P70–P74, P6OUTCUT, XIN, RESET, MD0, MD1	V <sub>I</sub> = 0 V			–5	μA
VRAM	RAM hold voltage	When clock is inactive.	2			V
I <sub>CC</sub>	Power source current	Output-only pins are open, and the other pins are connected to V <sub>SS</sub> or V <sub>CC</sub> . An external square-waveform clock is input. (Pin X <sub>OUT</sub> is open.) The PLL frequency multiplier is inactive.	f(f <sub>sys</sub> ) = 20 MHz. CPU is active.	25	50	mA
			T <sub>a</sub> = 25 °C when clock is inactive.		1	μA
			T <sub>a</sub> = 85 °C when clock is inactive.		20	

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**A-D CONVERTER CHARACTERISTICS**

(VCC = AVCC = 5 V ± 0.5 V, VSS = AVSS = 0 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	VREF = VCC	A-D converter		10	Bits
			Comparator		$\frac{1}{256} V_{REF}$	V
—	Absolute accuracy	VREF = VCC	10-bit resolution mode		± 3	LSB
			8-bit resolution mode		± 2	LSB
			Comparator		± 40	mV
RLADDER	Ladder resistance	VREF = VCC	5			kΩ
tCONV	Conversion time	f(fs <sub>ys</sub> ) ≤ 20 MHz	10-bit resolution mode	5.9		μs
			8-bit resolution mode	2.45 (Note)		
			Comparator	0.7 (Note)		
VREF	Reference voltage		2.7		VCC	V
VIA	Analog input voltage		0		VREF	V

Note: This is applied when A-D conversion frequency (φ<sub>AD</sub>) = f<sub>1</sub> (φ).

**D-A CONVERTER CHARACTERISTICS**

(VCC = 5 V, VSS = AVSS = 0 V, VREF = 5 V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute accuracy				± 1.0	%
t <sub>su</sub>	Set time				3	μs
RO	Output resistance		2	3.5	4.5	kΩ
IVREF	Reference power source input current	(Note)			3.2	mA

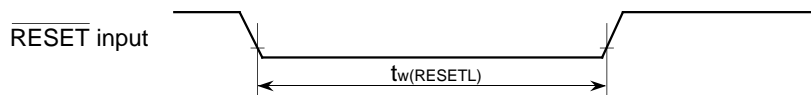
Note: The test conditions are as follows:

- One D-A converter is used.
- The D-A register value of the unused D-A converter is "0016."
- The reference power source input current for the ladder resistance of the A-D converter is excluded.

**RESET INPUT**

**Reset input timing requirements** (VCC = 5 V ± 0.5 V, VSS = 0V, Ta = -20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t <sub>w</sub> (RESETL)	RESETL input low-level pulse width	10			μs



**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**PERIPHERAL DEVICE INPUT/OUTPUT TIMING**

(VCC = 5 V±0.5 V, VSS = 0 V, Ta = -20 to 85 °C, f(fsyst) = 20 MHz unless otherwise noted)

\* For limits depending on f(fsyst), their calculation formulas are shown below. Also, the values at f(fsyst) = 20 MHz are shown in ( ).

**Timer A input** (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	80		ns
tw(TAH)	TAiIN input high-level pulse width	40		ns
tw(TAL)	TAiIN input low-level pulse width	40		ns

**Timer A input** (Gating input in timer mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
tc(TA)	TAiIN input cycle time	f(fsyst) ≤ 20 MHz	$\frac{16 \times 10^9}{f(fsyst)}$ (800)		ns
tw(TAH)	TAiIN input high-level pulse width	f(fsyst) ≤ 20 MHz	$\frac{8 \times 10^9}{f(fsyst)}$ (400)		ns
tw(TAL)	TAiIN input low-level pulse width	f(fsyst) ≤ 20 MHz	$\frac{8 \times 10^9}{f(fsyst)}$ (400)		ns

**Note** : The TAiIN input cycle time requires 4 or more cycles of a count source. The TAiIN input high-level pulse width and the TAiIN 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 = f2 at f(fsyst) ≤ 20 MHz.

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

Symbol	Parameter		Limits		Unit
			Min.	Max.	
tc(TA)	TAiIN input cycle time	f(fsyst) ≤ 20 MHz	$\frac{8 \times 10^9}{f(fsyst)}$ (400)		ns
tw(TAH)	TAiIN input high-level pulse width		80		ns
tw(TAL)	TAiIN input low-level pulse width		80		ns

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

Symbol	Parameter	Limits		Unit
		Min.	Max.	
tw(TAH)	TAiIN input high-level pulse width	80		ns
tw(TAL)	TAiIN 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.	
tc(UP)	TAiOUT input cycle time	2000		ns
tw(UPH)	TAiOUT input high-level pulse width	1000		ns
tw(UPL)	TAiOUT input low-level pulse width	1000		ns
tsu(UP-TiN)	TAiOUT input setup time	400		ns
th(TiN-UP)	TAiOUT input hold time	400		ns

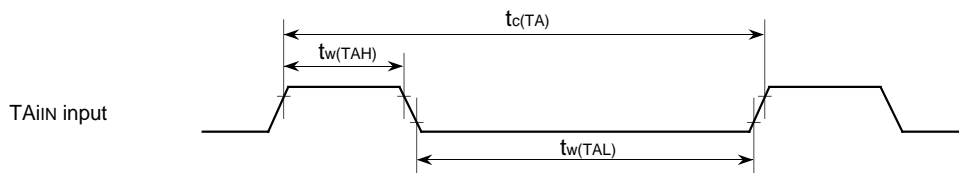
**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

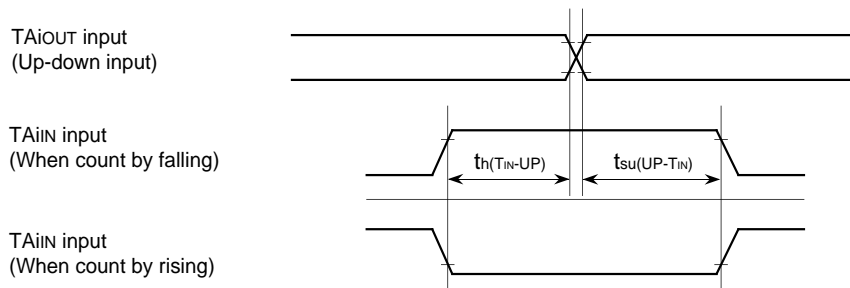
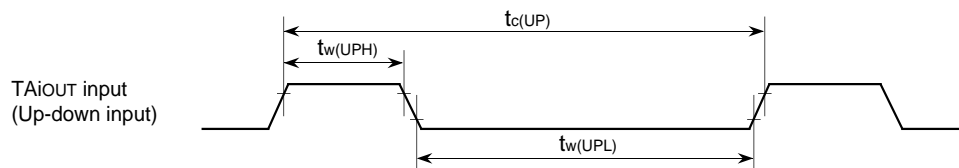
**Timer A input** (Two-phase pulse input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_c(TA)$	TAJIN input cycle time	800		ns
$t_{su}(TA_{JIN}-TA_{JOUT})$	TAJIN input setup time	200		ns
$t_{su}(TA_{JOUT}-TA_{JIN})$	TAJOUT input setup time	200		ns

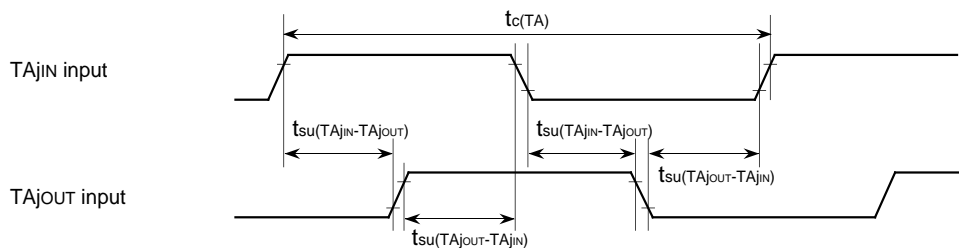
- Gating input in timer mode
- Count input in event counter mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode



- Up-down and Count input in event counter mode



- Two-phase pulse input in event counter mode



**Test conditions**

- $V_{CC} = 5 V \pm 0.5 V$ ,  $T_a = -20$  to  $85^\circ C$
- Input timing voltage :  $V_{IL} = 1.0 V$ ,  $V_{IH} = 4.0 V$

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**Timer B input** (Count input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t <sub>c</sub> (TB)	TBiIn input cycle time (one edge count)	80		ns
t <sub>w</sub> (TBH)	TBiIn input high-level pulse width (one edge count)	40		ns
t <sub>w</sub> (TBL)	TBiIn input low-level pulse width (one edge count)	40		ns
t <sub>c</sub> (TB)	TBiIn input cycle time (both edge count)	160		ns
t <sub>w</sub> (TBH)	TBiIn input high-level pulse width (both edge count)	80		ns
t <sub>w</sub> (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 <sub>c</sub> (TB)	TBiIn input cycle time	f(f <sub>sys</sub> ) ≤ 20 MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (800)		ns
t <sub>w</sub> (TBH)	TBiIn input high-level pulse width	f(f <sub>sys</sub> ) ≤ 20 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns
t <sub>w</sub> (TBL)	TBiIn input low-level pulse width	f(f <sub>sys</sub> ) ≤ 20 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (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<sub>2</sub> at f(f<sub>sys</sub>) ≤ 20 MHz.

**Timer B input** (Pulse width measurement mode)

Symbol	Parameter		Limits		Unit
			Min.	Max.	
t <sub>c</sub> (TB)	TBiIn input cycle time	f(f <sub>sys</sub> ) ≤ 20 MHz	$\frac{16 \times 10^9}{f(f_{sys})}$ (800)		ns
t <sub>w</sub> (TBH)	TBiIn input high-level pulse width	f(f <sub>sys</sub> ) ≤ 20 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (400)		ns
t <sub>w</sub> (TBL)	TBiIn input low-level pulse width	f(f <sub>sys</sub> ) ≤ 20 MHz	$\frac{8 \times 10^9}{f(f_{sys})}$ (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<sub>2</sub> at f(f<sub>sys</sub>) ≤ 20 MHz.

**Serial I/O**

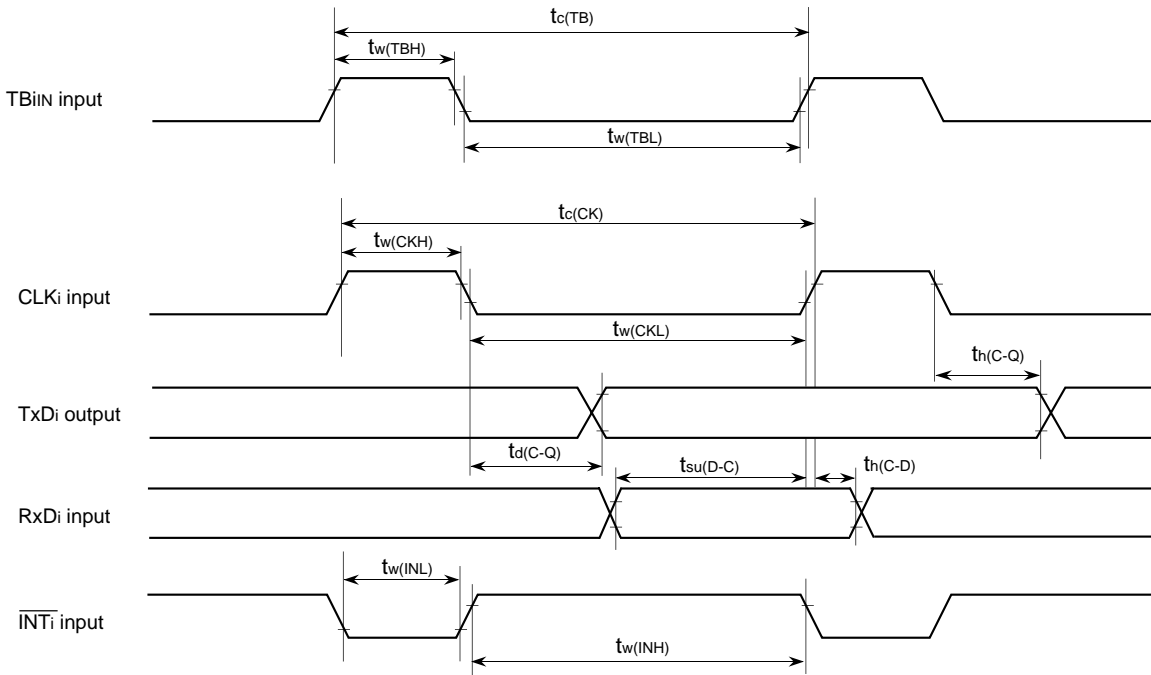
Symbol	Parameter	Limits		Unit
		Min.	Max.	
t <sub>c</sub> (CK)	CLKi input cycle time	200		ns
t <sub>w</sub> (CKH)	CLKi input high-level pulse width	100		ns
t <sub>w</sub> (CKL)	CLKi input low-level pulse width	100		ns
t <sub>d</sub> (C-Q)	TxDi output delay time		80	ns
t <sub>h</sub> (C-Q)	TxDi hold time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

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**External interrupt ( $\overline{\text{INTi}}$ ) input**

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_w(\text{INH})$	$\overline{\text{INTi}}$ input high-level pulse width	250		ns
$t_w(\text{INL})$	$\overline{\text{INTi}}$ input low-level pulse width	250		ns



**Test conditions**

- $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$ ,  $T_a = -20 \text{ to } 85 \text{ }^\circ\text{C}$
- Input timing voltage :  $V_{IL} = 1.0 \text{ V}$ ,  $V_{IH} = 4.0 \text{ V}$
- Output timing voltage :  $V_{OL} = 0.8 \text{ V}$ ,  $V_{OH} = 2.0 \text{ V}$ ,  $C_L = 50 \text{ pF}$

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

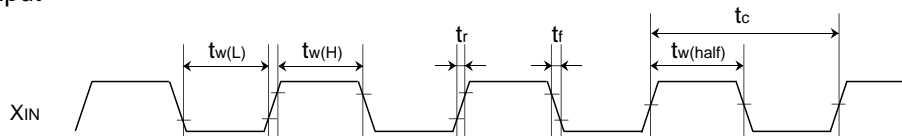
M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

**External clock input**

**Timing Requirements** ( $V_{CC} = 5 V \pm 0.5 V$ ,  $V_{SS} = 0 V$ ,  $T_a = -20$  to  $85\text{ }^\circ\text{C}$ ,  $f(X_{IN}) = 20\text{ MHz}$ , unless otherwise noted)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	50		ns
$t_{w(half)}$	External clock input pulse width with half input-voltage	$0.45 t_c$	$0.55 t_c$	ns
$t_{w(H)}$	External clock input high-level pulse width	$0.5 t_c - 8$		ns
$t_{w(L)}$	External clock input low-level pulse width	$0.5 t_c - 8$		ns
$t_r$	External clock input rise time		8	ns
$t_f$	External clock input fall time		8	ns

External clock input



Test conditions

- $V_{CC} = 5 V \pm 0.5 V$ ,  $T_a = -20$  to  $85\text{ }^\circ\text{C}$
- 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 :  $2.5 V$  ( $t_c$ ,  $t_{w(half)}$ )

**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

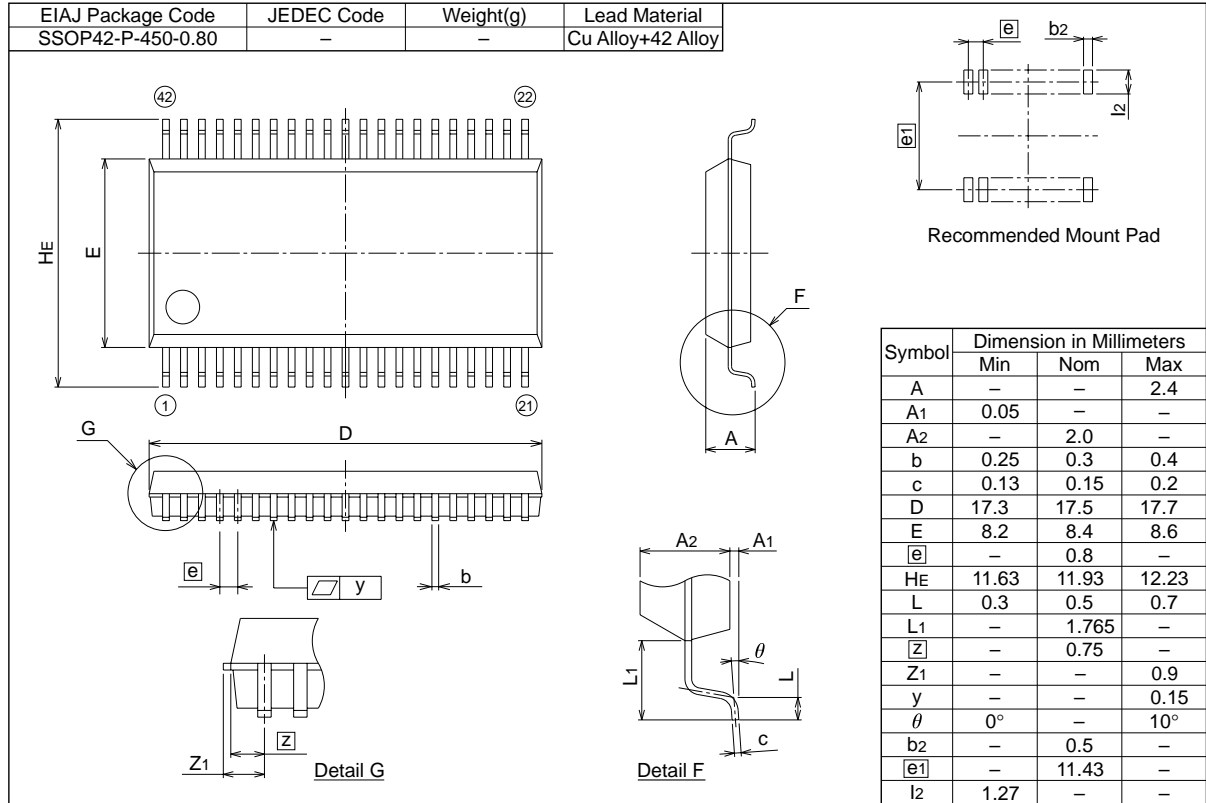
M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP

16-BIT CMOS MICROCOMPUTER

**PACKAGE OUTLINE**

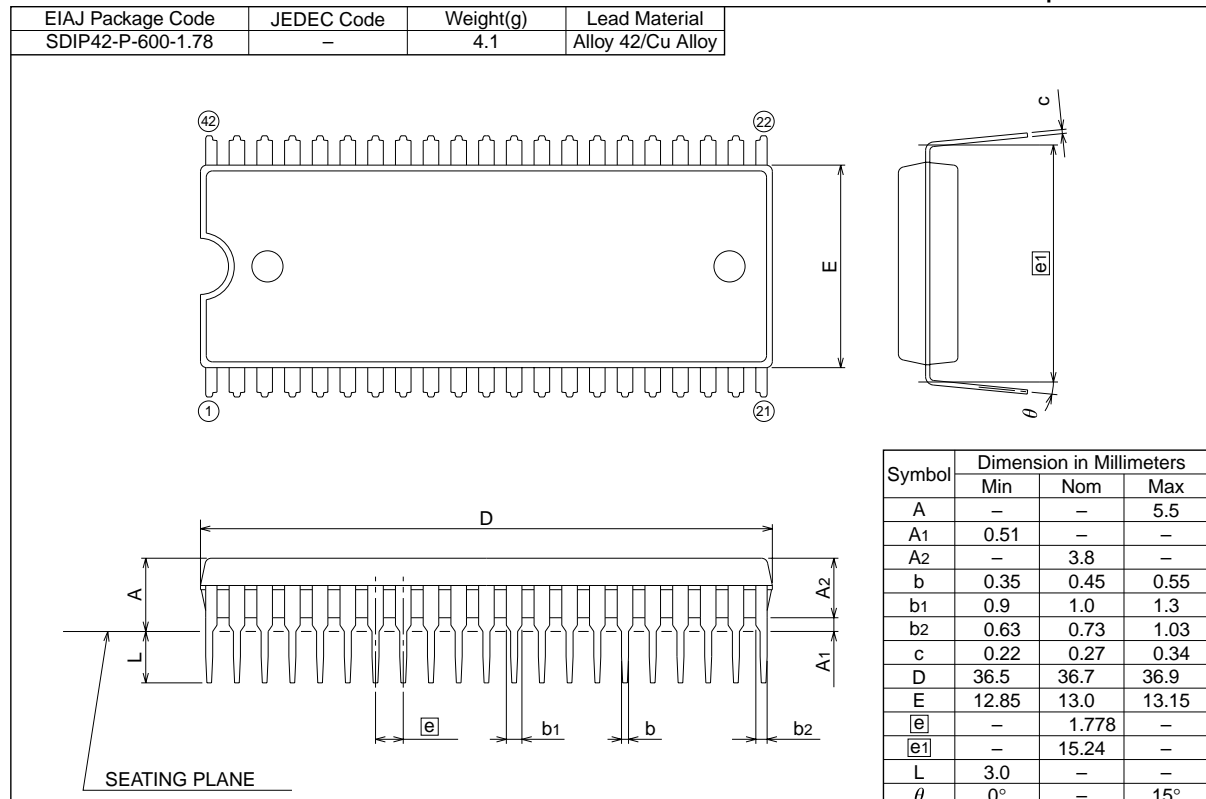
**42P2R-E**

Plastic 42pin 450mil SSOP



**42P4B**

Plastic 42pin 600mil SDIP



**PRELIMINARY**  
 Notice: This is not a final specification.  
 Some parametric limits are subject to change.

M37906M4C-XXXFP, M37906M4C-XXXSP, M37906M4H-XXXFP  
 M37906M4H-XXXSP, M37906M6C-XXXFP, M37906M6C-XXXSP  
 M37906M8C-XXXFP, M37906M8C-XXXSP  
 16-BIT CMOS MICROCOMPUTER

# Renesas Technology Corp.

Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

## Keep safety first in your circuit designs!

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

## Notes regarding these materials

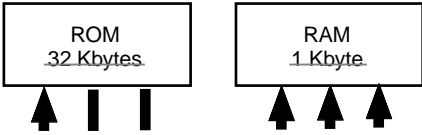
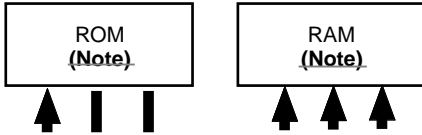
- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation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

## Revision History

## M37906MxX-XXXFP/SP Datasheet

Rev. No.	Revision Description	Rev. date
1.0	First Edition. The following are released. <ul style="list-style-type: none"> <li>• DESCRIPTION</li> <li>• Timer function for motor control</li> <li>• Pulse output port mode</li> </ul>	000331
2.0	(1) The following are revised: <ul style="list-style-type: none"> <li>• Figure 42 in page 36; the description for bits 6 and 7 of the timer Ax (x = 0 to 3) mode register &lt;Error&gt; (See <a href="#">the table below.</a>) &lt;Correction&gt; (See <a href="#">Table 7.</a>)</li> <li>• Figure 50 in page 43; the description for bits 6 and 7 of the timer Ax (x = 0, 3) mode register &lt;Error&gt; (See <a href="#">Fig. 42.</a>) &lt;Correction&gt; (See <a href="#">Table 7.</a>)</li> </ul> (2) The following are added: BASIC FUNCTION BLOCKS, MEMORY, CENTRAL PROCESSING UNIT (CPU), BUS INTERFACE UNIT, PROCESSOR MODES, INTERRUPTS, TIMER A, TIMER B, SERIAL I/O PORTS, A-D CONVERTER, D-A CONVERTER, WATCHDOG TIMER, INPUT/OUTPUT PINS, RESET CIRCUIT, OSCILLATION CIRCUIT, CLOCK GENERATING CIRCUIT, STANDBY FUNCTION, POWER SAVING FUNCTION, DEBUG FUNCTION, ELECTRICAL CHARACTERISTICS, PACKAGE OUTLINE, MASK ROM CONFIRMATION FORM, MARK SPECIFICATION FORM	000515
3.0	The new information added in Rev.3.0 are described in <b>Corrections and Supplementary Explanation for “M37906MxC Datasheet (Rev.A)”</b> .	000807
4.0	Refer to <b>“Corrections and Supplementary Explanation for M37906MxC-XXXFP/SP Datasheet (Rev.B)”</b> . <b>Note</b> : ★ represents the new information added in Rev.4.0.	010706
5.0	Refer to <b>“Corrections and Supplementary Explanation for M37906MxX-XXXFP/SP Datasheet (Rev.C)”</b> . <b>Note</b> : ★ represents the new information added in Rev.5.0.	011113

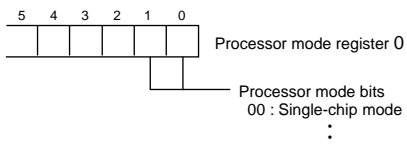
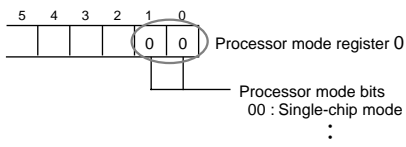
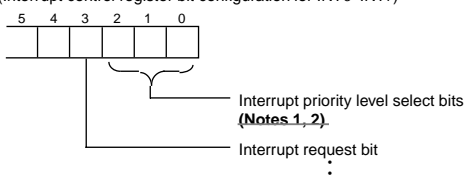
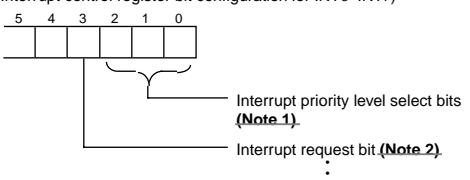
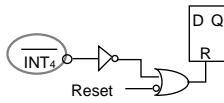
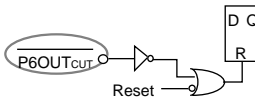
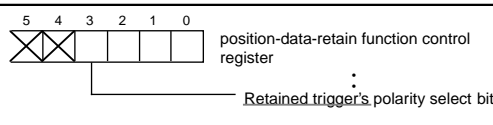
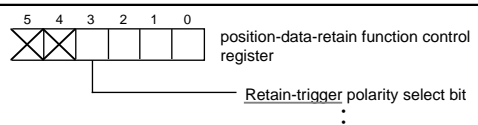
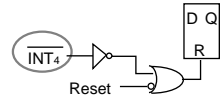
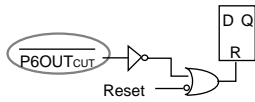


# Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.1

Page	Error	Correction																									
★ All pages, Header	M37906M4C-XXXFP, M37906M4C-XXXSP M37906M6C-XXXFP, M37906M6C-XXXSP M37906M8C-XXXFP, M37906M8C-XXXSP	M37906M4C-XXXFP, M37906M4C-XXXSP M37906M4H-XXXFP, M37906M4H-XXXSP M37906M6C-XXXFP, M37906M6C-XXXSP M37906M8C-XXXFP, M37906M8C-XXXSP																									
★ Page 1, DISTINCTIVE FEATURES; Memory	[M37906M4C-XXXFP, M37906M4C-XXXSP] ROM.....32 Kbytes RAM.....1 Kbyte	[M37906M4C-XXXFP, M37906M4C-XXXSP] [M37906M4H-XXXFP, M37906M4H-XXXSP] <b>(Note)</b> ROM.....32 Kbytes RAM.....1 Kbyte																									
	_____	[M37906M6C-XXXFP, M37906M6C-XXXSP] ROM.....48 Kbytes RAM.....3 Kbytes																									
	_____	[M37906M8C-XXXFP, M37906M8C-XXXSP] ROM.....60 Kbytes RAM.....3 Kbytes																									
	• 8-bit A-D converter.....2-channel outputs	• 8-bit D-A converter.....2-channel outputs																									
★	_____	<b>Note:</b> The M37906M4C-XXXFP/SP and the M37906M4H-XXXFP/SP have the same functions, distinctive features, and electrical characteristics.																									
★ Page 1, PIN CONFIGURATION of outline 42P2R-E	M37906MxC-XXXFP PIN CONFIGURATION (TOP VIEW)  (Type) M37906M4C-XXXFP M37906M6C-XXXFP M37906M8C-XXXFP  $\overline{INT}_4$	M37906MxX-XXXFP PIN CONFIGURATION (TOP VIEW)  (Type) M37906M4C-XXXFP M37906M4H-XXXFP M37906M6C-XXXFP M37906M8C-XXXFP  $\overline{P6OUT}_{CUT}/\overline{INT}_4$																									
★ Page 2, PIN CONFIGURATION of outline 42P4B	M37906MxC-XXXSP PIN CONFIGURATION (TOP VIEW)  (Type) M37906M4C-XXXSP M37906M6C-XXXSP M37906M8C-XXXSP  $\overline{INT}_4$	M37906MxX-XXXSP PIN CONFIGURATION (TOP VIEW)  (Type) M37906M4C-XXXSP M37906M4H-XXXSP M37906M6C-XXXSP M37906M8C-XXXSP  $\overline{P6OUT}_{CUT}/\overline{INT}_4$																									
★ Pages 1, 2 PIN CONFIGURATION of outlines 42P2R-E and 42P4B	P74/AN4/DA1/ $\overline{INT}_3$ /RTP <sub>TRG0</sub>  P57/ $\overline{INT}_7$ /TB2IN/IDU P56/ $\overline{INT}_6$ /TB1IN/IDV P55/ $\overline{INT}_5$ /TB0IN/IDW  P24/(TB0IN) P25/(TB1IN) P26/(TB2IN) P26/( $\overline{INT}_3$ /RTP <sub>TRG0</sub> )  ( ) : Shift function	<b>(Note)</b> P74/AN4/DA1/ $\overline{INT}_3$ /RTP <sub>TRG0</sub>  <b>(Note)</b> { P57/ $\overline{INT}_7$ /TB2IN/IDU P56/ $\overline{INT}_6$ /TB1IN/IDV P55/ $\overline{INT}_5$ /TB0IN/IDW  P24/(TB0IN) P25/(TB1IN) P26/(TB2IN) P26/( $\overline{INT}_3$ /RTP <sub>TRG0</sub> ) } <b>(Note)</b>  <b>Note:</b> Allocation of these pins can be switched by software.																									
★ Page 3, BLOCK DIAGRAM	$\overline{INT}_4$  	$\overline{P6OUT}_{CUT}$  																									
★	<b>Note:</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>ROM</th> <th>RAM</th> </tr> </thead> <tbody> <tr> <td>M37906M4C-XXXFP, M37906M4C-XXXSP</td> <td>32 Kbytes</td> <td>1 Kbyte</td> </tr> <tr> <td>M37906M6C-XXXFP, M37906M6C-XXXSP</td> <td>48 Kbytes</td> <td>3 Kbytes</td> </tr> <tr> <td>M37906M8C-XXXFP, M37906M8C-XXXSP</td> <td>60 Kbytes</td> <td>3 Kbytes</td> </tr> </tbody> </table>		ROM	RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes	1 Kbyte	M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes	3 Kbytes	M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes	3 Kbytes	<b>Note:</b> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>ROM</th> <th>RAM</th> </tr> </thead> <tbody> <tr> <td>M37906M4C-XXXFP, M37906M4C-XXXSP</td> <td rowspan="2">32 Kbytes</td> <td rowspan="2">1 Kbyte</td> </tr> <tr> <td>M37906M4H-XXXFP, M37906M4H-XXXSP</td> </tr> <tr> <td>M37906M6C-XXXFP, M37906M6C-XXXSP</td> <td>48 Kbytes</td> <td>3 Kbytes</td> </tr> <tr> <td>M37906M8C-XXXFP, M37906M8C-XXXSP</td> <td>60 Kbytes</td> <td>3 Kbytes</td> </tr> </tbody> </table>		ROM	RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes	1 Kbyte	M37906M4H-XXXFP, M37906M4H-XXXSP	M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes	3 Kbytes	M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes	3 Kbytes
	ROM	RAM																									
M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes	1 Kbyte																									
M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes	3 Kbytes																									
M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes	3 Kbytes																									
	ROM	RAM																									
M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes	1 Kbyte																									
M37906M4H-XXXFP, M37906M4H-XXXSP																											
M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes	3 Kbytes																									
M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes	3 Kbytes																									
★ Page 4, FUNCTIONS	Memory size <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>RAM</td> <td>1 Kbyte</td> </tr> <tr> <td>ROM</td> <td>32 Kbytes.</td> </tr> </table>	RAM	1 Kbyte	ROM	32 Kbytes.	Memory size <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>RAM</td> <td><b>(Note 1)</b></td> </tr> <tr> <td>ROM</td> <td><b>(Note 1)</b></td> </tr> </table>	RAM	<b>(Note 1)</b>	ROM	<b>(Note 1)</b>																	
RAM	1 Kbyte																										
ROM	32 Kbytes.																										
RAM	<b>(Note 1)</b>																										
ROM	<b>(Note 1)</b>																										

# Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.2

Page	Error	Correction																																				
Page 4, FUNCTIONS	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Interrupts</td> <td style="width: 15%;">Maskable interrupts</td> <td style="width: 70%;">5 external types, 18 internal types. *****</td> </tr> <tr> <td></td> <td>Non-maskable interrupts</td> <td>3 internal types.</td> </tr> </table>	Interrupts	Maskable interrupts	5 external types, 18 internal types. *****		Non-maskable interrupts	3 internal types.	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Interrupts</td> <td style="width: 15%;">Maskable interrupts</td> <td style="width: 70%;">5 external sources, 18 internal sources. *****</td> </tr> <tr> <td></td> <td>Non-maskable interrupts</td> <td>3 internal sources.</td> </tr> </table>	Interrupts	Maskable interrupts	5 external sources, 18 internal sources. *****		Non-maskable interrupts	3 internal sources.																								
	Interrupts	Maskable interrupts	5 external types, 18 internal types. *****																																			
		Non-maskable interrupts	3 internal types.																																			
	Interrupts	Maskable interrupts	5 external sources, 18 internal sources. *****																																			
		Non-maskable interrupts	3 internal sources.																																			
	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Clock generating circuit</td> <td>Built-in (externally connected to a ceramic resonator or quartz crystal resonator).</td> </tr> </table>	Clock generating circuit	Built-in (externally connected to a ceramic resonator or quartz crystal resonator).	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Clock generating circuit</td> <td>Built-in (externally connected to a ceramic resonator or quartz crystal oscillator).</td> </tr> </table>	Clock generating circuit	Built-in (externally connected to a ceramic resonator or quartz crystal oscillator).																																
Clock generating circuit	Built-in (externally connected to a ceramic resonator or quartz crystal resonator).																																					
Clock generating circuit	Built-in (externally connected to a ceramic resonator or quartz crystal oscillator).																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">PLL frequency multiplier</td> <td>The following multiplication methods are available : <u>double</u>, <u>triple</u>, and <u>quadruple</u>.</td> </tr> </table>	PLL frequency multiplier	The following multiplication methods are available : <u>double</u> , <u>triple</u> , and <u>quadruple</u> .	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">PLL frequency multiplier</td> <td>The following multiplication ratios are available : <u>X 2</u>, <u>X 3</u>, <u>X 4</u>.</td> </tr> </table>	PLL frequency multiplier	The following multiplication ratios are available : <u>X 2</u> , <u>X 3</u> , <u>X 4</u> .																																	
PLL frequency multiplier	The following multiplication methods are available : <u>double</u> , <u>triple</u> , and <u>quadruple</u> .																																					
PLL frequency multiplier	The following multiplication ratios are available : <u>X 2</u> , <u>X 3</u> , <u>X 4</u> .																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Power dissipation</td> <td>125 mW (at f(X<sub>IN</sub>) = 20 MHz, typ., PLL frequency multiplier stopped.)</td> </tr> </table>	Power dissipation	125 mW (at f(X <sub>IN</sub> ) = 20 MHz, typ., PLL frequency multiplier stopped.)	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Power dissipation</td> <td>125 mW (at f(f<sub>sys</sub>) = 20 MHz, typ., the PLL frequency multiplier is inactive.)</td> </tr> </table>	Power dissipation	125 mW (at f(f <sub>sys</sub> ) = 20 MHz, typ., the PLL frequency multiplier is inactive.)																																	
Power dissipation	125 mW (at f(X <sub>IN</sub> ) = 20 MHz, typ., PLL frequency multiplier stopped.)																																					
Power dissipation	125 mW (at f(f <sub>sys</sub> ) = 20 MHz, typ., the PLL frequency multiplier is inactive.)																																					
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Package</td> <td style="text-align: center;"><b>(Note)</b></td> </tr> </table>	Package	<b>(Note)</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">Package</td> <td style="text-align: center;"><b>(Note 2)</b></td> </tr> </table>	Package	<b>(Note 2)</b>																																	
Package	<b>(Note)</b>																																					
Package	<b>(Note 2)</b>																																					
★	<p><b>Note 1:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">ROM</td> <td style="width: 60%;">M37906M4C-XXXFP, M37906M4C-XXXSP</td> <td style="width: 30%;">32 Kbytes</td> </tr> <tr> <td></td> <td>M37906M6C-XXXFP, M37906M6C-XXXSP</td> <td>48 Kbytes</td> </tr> <tr> <td></td> <td>M37906M8C-XXXFP, M37906M8C-XXXSP</td> <td>60 Kbytes</td> </tr> <tr> <td rowspan="3">RAM</td> <td>M37906M4C-XXXFP, M37906M4C-XXXSP</td> <td>1 Kbyte</td> </tr> <tr> <td>M37906M6C-XXXFP, M37906M6C-XXXSP</td> <td>3 Kbytes</td> </tr> <tr> <td>M37906M8C-XXXFP, M37906M8C-XXXSP</td> <td>3 Kbytes</td> </tr> </table>	ROM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes		M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes		M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes	RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	1 Kbyte	M37906M6C-XXXFP, M37906M6C-XXXSP	3 Kbytes	M37906M8C-XXXFP, M37906M8C-XXXSP	3 Kbytes	<p><b>Note 1:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">ROM</td> <td style="width: 60%;">M37906M4C-XXXFP, M37906M4C-XXXSP</td> <td style="width: 30%;">32 Kbytes</td> </tr> <tr> <td></td> <td>M37906M4H-XXXFP, M37906M4H-XXXSP</td> <td></td> </tr> <tr> <td></td> <td>M37906M6C-XXXFP, M37906M6C-XXXSP</td> <td>48 Kbytes</td> </tr> <tr> <td></td> <td>M37906M8C-XXXFP, M37906M8C-XXXSP</td> <td>60 Kbytes</td> </tr> <tr> <td rowspan="4">RAM</td> <td>M37906M4C-XXXFP, M37906M4C-XXXSP</td> <td rowspan="2">1 Kbyte</td> </tr> <tr> <td>M37906M4H-XXXFP, M37906M4H-XXXSP</td> </tr> <tr> <td>M37906M6C-XXXFP, M37906M6C-XXXSP</td> <td>3 Kbytes</td> </tr> <tr> <td>M37906M8C-XXXFP, M37906M8C-XXXSP</td> <td>3 Kbytes</td> </tr> </table>	ROM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes		M37906M4H-XXXFP, M37906M4H-XXXSP			M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes		M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes	RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	1 Kbyte	M37906M4H-XXXFP, M37906M4H-XXXSP	M37906M6C-XXXFP, M37906M6C-XXXSP	3 Kbytes	M37906M8C-XXXFP, M37906M8C-XXXSP	3 Kbytes
ROM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes																																				
	M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes																																				
	M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes																																				
RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	1 Kbyte																																				
	M37906M6C-XXXFP, M37906M6C-XXXSP	3 Kbytes																																				
	M37906M8C-XXXFP, M37906M8C-XXXSP	3 Kbytes																																				
ROM	M37906M4C-XXXFP, M37906M4C-XXXSP	32 Kbytes																																				
	M37906M4H-XXXFP, M37906M4H-XXXSP																																					
	M37906M6C-XXXFP, M37906M6C-XXXSP	48 Kbytes																																				
	M37906M8C-XXXFP, M37906M8C-XXXSP	60 Kbytes																																				
RAM	M37906M4C-XXXFP, M37906M4C-XXXSP	1 Kbyte																																				
	M37906M4H-XXXFP, M37906M4H-XXXSP																																					
	M37906M6C-XXXFP, M37906M6C-XXXSP	3 Kbytes																																				
	M37906M8C-XXXFP, M37906M8C-XXXSP	3 Kbytes																																				
★	<p><b>2:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Packages</td> <td style="width: 35%;">M37906M4C-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP</td> <td style="width: 50%;">42-pin plastic *****</td> </tr> <tr> <td></td> <td>M37906M4C-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP</td> <td>42-pin shrink *****</td> </tr> </table>	Packages	M37906M4C-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP	42-pin plastic *****		M37906M4C-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP	42-pin shrink *****	<p><b>2:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Packages</td> <td style="width: 35%;">M37906M4C-XXXFP, M37906M4H-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP</td> <td style="width: 50%;">42-pin plastic *****</td> </tr> <tr> <td></td> <td>M37906M4C-XXXSP, M37906M4H-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP</td> <td>42-pin shrink *****</td> </tr> </table>	Packages	M37906M4C-XXXFP, M37906M4H-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP	42-pin plastic *****		M37906M4C-XXXSP, M37906M4H-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP	42-pin shrink *****																								
Packages	M37906M4C-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP	42-pin plastic *****																																				
	M37906M4C-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP	42-pin shrink *****																																				
Packages	M37906M4C-XXXFP, M37906M4H-XXXFP, M37906M6C-XXXFP, M37906M8C-XXXFP	42-pin plastic *****																																				
	M37906M4C-XXXSP, M37906M4H-XXXSP, M37906M6C-XXXSP, M37906M8C-XXXSP	42-pin shrink *****																																				
Page 5	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><u>P6OUT<sub>CUT</sub></u></td> <td style="width: 25%;"><u>INT<sub>4</sub></u></td> <td style="width: 25%;"><u>INT<sub>4</sub></u> input</td> <td style="width: 25%;">input</td> <td style="width: 20%;">*****</td> </tr> </table>	<u>P6OUT<sub>CUT</sub></u>	<u>INT<sub>4</sub></u>	<u>INT<sub>4</sub></u> input	input	*****	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><u>P6OUT<sub>CUT</sub></u></td> <td style="width: 25%;"><u>P6OUT<sub>CUT</sub></u> input</td> <td style="width: 25%;">input</td> <td style="width: 25%;">*****</td> </tr> </table>	<u>P6OUT<sub>CUT</sub></u>	<u>P6OUT<sub>CUT</sub></u> input	input	*****																											
<u>P6OUT<sub>CUT</sub></u>	<u>INT<sub>4</sub></u>	<u>INT<sub>4</sub></u> input	input	*****																																		
<u>P6OUT<sub>CUT</sub></u>	<u>P6OUT<sub>CUT</sub></u> input	input	*****																																			
[Functions]	This pin functions as an input pin for <u>INT<sub>4</sub></u> . Also, this pin functions as an input pin for a forcibly-cut-off signal in the motor drive waveform output mode.	This pin has the function to forcibly place port P6 pins in the input mode (port-output-cutoff function). Also, this pin functions as an input pin for <u>INT<sub>4</sub></u> , and as an input pin for the port-output-cutoff function in the motor drive waveform output mode.																																				
Page 6, Memory	<p>(Line 5)</p> <p>The internal ROM is allocated to addresses 8000<sub>16</sub> through FFFF<sub>16</sub>, and its address space is 32 Kbytes.</p>	<p>(Line 5)</p> <p>The internal ROM and RAM are allocated as shown in Figures 1 (1) through (3).</p>																																				
	<p>The internal RAM is allocated to addresses C00<sub>16</sub> through FFF<sub>16</sub>, and its address space is 1 Kbyte. In addition to storing data, the RAM is used as a stack during a subroutine call, or interrupts.</p>	(Deleted)																																				
Page 6, Right column Line 8	Allocated to addresses 0 <sub>16</sub> through FF <sub>16</sub> are ***** A-D converter, D-A converter, UART, timers, interrupt control registers, etc.	Allocated to addresses 0 <sub>16</sub> through FF <sub>16</sub> are ***** A-D converter, D-A converter, serial I/O, timers, interrupt control registers, etc.																																				
★ Page 6	Figure 1 (1). Memory map of M37906M4C-XXXFP/SP (Single-chip mode)	Figure 1 (1). Memory map of M37906M4C-XXXFP/SP and M37906M4H-XXXFP/SP (Single-chip mode)																																				

# Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.3

Page	Error	Correction
Page 7	_____	Memory map of M37906M6C-XXXFP/SP (Single-chip mode)
Page 8	_____	Memory map of M37906M8C-XXXFP/SP (Single-chip mode)
Page 12, 3. Interrupt disable flag (I)	(Line 3) ••• all interrupts except watchdog timer, $\overline{\text{NMI}}$ , and software interrupt •••	(Line 3) ••• all interrupts except watchdog timer and software interrupts •••
Page 20, Fig. 11		
Page 21, Left column Line 3	••• Reset is also handled as a type of interrupt in this section, too.	••• Reset is also handled as an interrupt source in this section, too.
Page 21, Left column Line 21	The position to which pin $\overline{\text{INT}}_3$ is to be allocated can be selected •••	The position where pin $\overline{\text{INT}}_3/\text{RTPTRG0}$ is allocated can be selected •••
Page 22, Fig. 13	(Interrupt control register bit configuration for $\overline{\text{INT}}_3\text{--}\overline{\text{INT}}_7$ ) 	(Interrupt control register bit configuration for $\overline{\text{INT}}_3\text{--}\overline{\text{INT}}_7$ ) 
Page 39, Fig. 43	Waveform output control bit (bit 7 at address A6 <sub>16</sub> ) 	Waveform output control bit (bit 7 at address A6 <sub>16</sub> ) 
Page 40, Right column Lines 5, 6	••• This bit is cleared to "0" by inputting falling edge to the $\overline{\text{INT}}_4$ pin, by reset, or by clearing with instructions.	••• This bit is cleared to "0" by inputting a falling edge to pin P6OUTcut, by reset, or by executing instructions.
Page 43, Left column Line 7	Retained data can be read out •••	Retain data can be read out •••
Page 43, Fig. 47		
Page 44, Left column Last line	••• count start flag to "1", and they stop it when clearing that flag to "0".	••• count start bit to "1", and they stop it when clearing that bit to "0".
Page 44, Fig. 48	Waveform output control bit 1 (bit 7 at address A6 <sub>16</sub> ) 	Waveform output control bit 1 (bit 7 at address A6 <sub>16</sub> ) 
Page 45, Fig. 49	 Waveform output control register	 Waveform output mode register
Page 46, Right column Line 35	••• the pulse to which pulse width modulation is applied output port •••	••• the pulse to which pulse width modulation has been applied output port •••
Page 46, Right column Line 51	••• by reset or clearing with instructions.	••• by reset or by executing instructions.

## Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.4

Page	Error	Correction
Page 46, Right column Last line	••• by inputting a falling edge to an external interrupt pin, <u>INT4</u> .	••• by inputting a falling edge to pin <u>P6OUTcut</u> .
Page 47, Left column Line 45	••• by inputting a falling edge to an external interrupt pin, <u>INT4</u> , or clearing with instructions.	••• by inputting a falling edge to pin <u>P6OUTcut</u> , or by executing instructions.
Page 49, Left column Line 5	••• shown in Figure 55 are used to determine whether to use port <u>P8</u> as a •••	••• shown in Figure 55 are used to determine whether to use port <u>P1</u> as a •••
Page 49, Fig. 55		
Page 59, Table 8		
Page 63, Operation mode Line 8	••• the comparator function select register is "0" and •••	••• the comparator function select register <u>Ω</u> is "0" and •••
Page 63, Operation mode Last line	••• the comparator result register.	••• the comparator result register <u>Ω</u> .
Page 69, Lines 6, 7	••• an input pin when it is "0". When a pin is programmed for output, the data •••	••• an input pin when it is "0". Also, each bit of the port P6 direction register can be cleared to "0" by inputting a falling edge to pin <u>P6OUTcut</u> or by executing instructions. When a pin is programmed for output, the data •••
Page 70, Fig. 76	[Inside dotted-line included] P20/TA4 <sub>OUT</sub> , P22/TA9 <sub>OUT</sub> , P60/TA0 <sub>OUT</sub> /W/RTP0 <sub>0</sub> , P61/TA0 <sub>IN</sub> /V/RTP0 <sub>1</sub> , P62/TA1 <sub>OUT</sub> /U/RTP0 <sub>2</sub> , P63/TA1 <sub>IN</sub> /W/RTP0 <sub>3</sub> , P64/TA2 <sub>OUT</sub> /V/RTP1 <sub>0</sub> , P65/TA2 <sub>IN</sub> /U/RTP1 <sub>1</sub>	[Inside dotted-line included] P20/TA4 <sub>OUT</sub> , P22/TA9 <sub>OUT</sub>
	—————	The block diagram for port P6 pins (the 3rd block diagram) is added.
Page 72, Fig. 79	<u>Comparison</u> result register 0 (DE <sub>16</sub> )••• <span style="border: 1px solid black; padding: 2px;">00<sub>16</sub></span>	<u>Comparator</u> result register 0 (DE <sub>16</sub> )••• <span style="border: 1px solid black; padding: 2px;">00<sub>16</sub></span>
Page 73, Fig. 78	Debug control register 1 (81 <sub>16</sub> )••• <span style="border: 1px solid black; padding: 2px;">0 0 0 <span style="border: 1px solid black; border-radius: 50%; padding: 1px;">0</span> 0 0 0 <span style="border: 1px solid black; border-radius: 50%; padding: 1px;">0</span> <span style="font-size: small;">Note 9</span></span>	Debug control register 1 (81 <sub>16</sub> )••• <span style="border: 1px solid black; padding: 2px;">0 0 0 <span style="border: 1px solid black; border-radius: 50%; padding: 1px;">X</span> 0 0 0 <span style="border: 1px solid black; border-radius: 50%; padding: 1px;">0</span> <span style="font-size: small;">Note 9</span></span>
Page 74, Left column Lines 2 to 6	••• and Figure 81 shows a circuit example with an external ceramic resonator or quartz crystal oscillator. The constants such as capacitance etc. depend on a resonator/oscillator.	••• and Figure 81 shows a circuit example with an external ceramic resonator or quartz-crystal oscillator. The constants such as capacitance etc. depend on an oscillator.
Page 74, Left column Line 22	••• enable bit (bit 1 of the clock control register; See Figure 85.), •••	••• enable bit (bit 1 of the clock control register <u>Ω</u> ; See Figure 85.), •••
Page 74, Fig. 81	Fig. 81 Circuit example with external ceramic resonator or quartz crystal oscillator	Fig. 81 Circuit example with external ceramic resonator or quartz-crystal oscillator
Page 74, Fig. 83	Fig. 83 Circuit example with pin <u>VCONT</u> and <u>PLL</u> circuit	Fig. 83 Circuit example of connection with pin <u>VCONT</u> when <u>PLL</u> circuit used
Page 75, Left column Lines 7, 27 Right column Lines 15, 10, 21	••• the clock control register •••	••• the clock control register <u>Ω</u> •••

## Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.5

Page	Error	Correction
Page 75, Left column Line 20	••• selects the PLL circuit's operation ( <u>stopped</u> /active).	••• selects the PLL circuit's operation ( <u>inactive</u> /active).
Page 75, Left column Line 23	••• (In this case, the PLL circuit <u>operates</u> .)	••• (In this case, the PLL circuit <u>is active</u> .)
Page 75, Left column Line 25	At the STP instruction execution, the PLL circuit <u>stops its</u> operation, and •••	At the STP instruction execution, the PLL circuit <u>is active</u> , and •••
Page 75, Right column Lines 1, 7, 12	••• the frequency of the <u>PLL output clock (f<sub>PLL</sub>)</u> •••	••• the frequency of <u>f<sub>PLL</sub></u> •••
Page 75, Right column Line 11	•••, and f <sub>XIN</sub> is select as the system clock when bit 5 = "0".	•••, and <u>input clock</u> f <sub>XIN</sub> is select as the system clock when bit 5 = "0".
Page 75, Right column Line 22	•••, and these bits select the <u>multiplication</u> ratio of (f <sub>1</sub> to f <sub>4096</sub> )/(f <sub>sys</sub> ).	•••, and these bits select the <u>division</u> ratio of (f <sub>1</sub> to f <sub>4096</sub> )/(f <sub>sys</sub> ).
Page 75, Table 9 Note	<b>Note:</b> The PLL multiplication must be set so that the frequency of the <u>PLL output clock (f<sub>PLL</sub>)</u> must be •••	<b>Note:</b> The PLL multiplication must be set so that the frequency of <u>f<sub>PLL</sub></u> must be •••
Page 76, Fig. 84		
Page 77, Fig. 86		
Page 78, Left column Lines 4, 5	These modes are used to save the power dissipation of the system, by <u>stopping</u> oscillation or system clock in the case that the CPU needs not be <u>operating</u> .	These modes are used to save the power dissipation of the system, by <u>making</u> oscillation or system clock <u>inactive</u> in the case that the CPU needs not be <u>active</u> .
Page 78, Left column Line 12	The interrupt priority level of this interrupt <u>is required</u> to be higher than the •••	The interrupt priority level of this interrupt <u>needs</u> to be higher than the •••
Page 78, Left column After line 29	Table 11 explains the microcomputer's operation in the STP and WIT modes. The external bus fixation function can also be provided. This function enables the user to specify the states of the external bus and the bus control signals in the memory expansion and the microprocessor mode in the STP or WIT mode. For more information, refer to the section on the power saving function.	Table 11 explains the microcomputer's operation in the STP and WIT modes.
Page 78, Right column Lines 2, 3	The execution of the STP instruction <u>stops</u> the oscillation circuit and PLL circuit. It also <u>stops</u> input clock f <sub>XIN</sub> , system clock f <sub>sys</sub> , •••	The execution of the STP instruction <u>makes</u> the oscillation circuit and PLL circuit <u>inactive</u> . It also <u>makes the following</u> <u>inactive</u> : input clock f <sub>XIN</sub> , system clock f <sub>sys</sub> , •••
Page 78, Right column Lines 5, 6	•••, Wf <sub>32</sub> , and Wf <sub>512</sub> <u>in</u> the "L" state, and divide clocks f <sub>X16</sub> to f <sub>X128</sub> <u>in</u> the "H" state.	•••, Wf <sub>32</sub> , and Wf <sub>512</sub> <u>with</u> the "L" state, and divide clocks f <sub>X16</sub> to f <sub>X128</sub> <u>with</u> the "H" state.
Page 78, Right column Line 13	At this time, timers A and B <u>operate</u> only in the event counter mode, •••	At this time, timers A and B <u>can be active</u> only in the event counter mode, •••

Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.6

Page	Error	Correction
Page 78, Table 11	Stopped	Inactive
	<p><b>Notes 1:</b> •••, the oscillation circuit <u>stops</u>. Also, clock input from pin X<sub>IN</sub> <u>available</u>.</p> <p><b>2:</b> •••, the PLL circuit <u>stops</u>.</p>	<p><b>Notes 1:</b> •••, the oscillation circuit is <u>inactive</u>. Also, clock input from pin X<sub>IN</sub> is <u>allowed</u>.</p> <p><b>2:</b> •••, the PLL circuit is <u>inactive</u>.</p>
Page 79, Left column Line 3	••• (bit 5 of the clock control register) = "1", •••	••• (bit 5 of the clock control register <u>0</u> ) = "1", •••
Page 79, Left column Line 10	••• after the oscillation circuit <u>has</u> been restarted •••	••• after the oscillation circuit <u>and PLL circuit have</u> been restarted •••
Page 79, Fig. 87	<p>Particular function register 1</p> <p>STP-instruction-execution status bit (<b>Note 1</b>) 0: Normal operation. 1: STP instruction <u>has been executed</u>.</p> <p>WIT-instruction-execution status bit (<b>Note 1</b>) 0: Normal operation. 1: WIT instruction <u>has been executed</u>.</p> <p>System clock stop select bit at WIT (<b>Note 2</b>) 0: In wait mode, system clock f<sub>sys</sub> is active. 1: In wait mode, system clock f<sub>sys</sub> is <u>stopped</u>.</p>	<p>Particular function register 1</p> <p>STP-instruction-execution status bit (<b>Note 1</b>) 0: Normal operation. 1: STP instruction <u>is under execution</u>.</p> <p>WIT-instruction-execution status bit (<b>Note 1</b>) 0: Normal operation. 1: WIT instruction <u>is under execution</u>.</p> <p>System clock stop select bit at WIT (<b>Note 2</b>) 0: In wait mode, system clock f<sub>sys</sub> is active. 1: In wait mode, system clock f<sub>sys</sub> is <u>inactive</u>.</p>
	<b>Notes 1:</b> ••• Even when "1" is <u>written</u> , the bit status will not change.	<b>Notes 1:</b> ••• Even when "1" is <u>try to be written</u> , the bit status will not change.
Page 80, Left column Line 5	•••, and divide clocks Wf <sub>32</sub> and Wf <sub>512</sub> are <u>stopped</u> in the "L" state.	•••, and divide clocks Wf <sub>32</sub> and Wf <sub>512</sub> are <u>inactive</u> with the "L" state.
Page 80, Left column Lines 6, 7	•••, and peripheral devices' clock f <sub>1</sub> to f <sub>4096</sub> remain <u>operating</u> . Therefore, BIU and CPU are <u>stopped</u> , •••	•••, and peripheral devices' <u>clocks</u> f <sub>1</sub> to f <sub>4096</sub> remain <u>active</u> . Therefore, BIU and CPU are <u>inactive</u> , •••
Page 80, Left column Lines 9, 10	•••, which use the peripheral devices' clocks f <sub>1</sub> to f <sub>4096</sub> , are still <u>operating</u> . Note that the watchdog timer is <u>stopped</u> .	•••, which use the peripheral devices' clocks f <sub>1</sub> to f <sub>4096</sub> , are still <u>active</u> . Note that the watchdog timer is <u>inactive</u> .
Page 80, Left column Lines 13, 14	•••, and input clock f <sub>XIN</sub> are <u>operating</u> , while system clock f <sub>sys</sub> , φ <sub>BIU</sub> , φ <sub>CPU</sub> , and peripheral devices' clocks <u>stop operating</u> .	•••, and input clock f <sub>XIN</sub> are <u>active</u> , while system clock f <sub>sys</sub> , φ <sub>BIU</sub> , φ <sub>CPU</sub> , and peripheral devices' clocks <u>are inactive</u> .
Page 80, Left column Lines 16, 17	•••, Wf <sub>32</sub> and Wf <sub>512</sub> , are <u>stopped</u> . At this time, timers A and B <u>operate</u> only in the event counter mode, •••	•••, Wf <sub>32</sub> and Wf <sub>512</sub> , <u>become inactive</u> . At this time, timers A and B <u>are active</u> only in the event counter mode, •••
Page 80, Left column Line 21	••• the system clocks stop select bit at WIT <u>is to be set</u> to "1" •••	••• the system clocks stop select bit at WIT <u>needs to be set</u> to "1" •••
Page 80, Left column Line 26	•••, and clock input f <sub>XIN</sub> are <u>operating</u> in the WIT mode, •••	•••, and clock input f <sub>XIN</sub> are <u>active</u> in the WIT mode, •••
Page 81, Left column Line 4	<b>(1) <u>Stop of system clock in wait mode</u></b>	<b>(1) <u>Inactive system clock in wait mode</u></b>
Page 81, Left column Line 6	•••, if the internal peripheral devices need not to operate, the system clock stop select bit •••	•••, if the internal peripheral devices need not to operate, <u>when</u> the system clock stop select bit •••
Page 81, Left column Line 8	••• and peripheral devices' clock <u>stop their operations</u> , and •••	••• and peripheral devices' clock <u>are inactive</u> , and •••
Page 81, Left column Line 11	<b>(2) <u>Stop of oscillation circuit</u></b>	<b>(2) <u>Inactive oscillation circuit</u></b>
Page 81, Left column Line 12	When an externally-generated <u>stable</u> clock is input to pin X <sub>IN</sub> , •••	When an externally-generated stable clock is input to pin X <sub>IN</sub> , •••

## Corrections and Supplementary Explanation for M37906MxX Datasheet (Rev.C) No.7

Page	Error	Correction																														
Page 81, Left column Line 17	• the oscillation driver between pins X <sub>IN</sub> and X <sub>OUT</sub> stops its operation.	• the oscillation driver circuit between pins X <sub>IN</sub> and X <sub>OUT</sub> is inactive.																														
Page 81, Left column Line 18	When not using a PLL output clocks, also, the supply of φ <sub>BIU</sub> and φ <sub>CPU</sub> restarts their operations just after the microcomputers returns from the stop mode, ****	When not using f <sub>PLL</sub> , also, the supply of φ <sub>BIU</sub> and φ <sub>CPU</sub> restarts just after the microcomputer returns from the stop mode, ****																														
Page 81, Left column Line 31	****, be sure that a waiting time of 1 μs more has passed ****	****, be sure that a waiting time of 1 μs more has passed ****																														
Page 84, ABSOLUTE MAXIMUM RATINGS	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Symbol</td> <td style="width: 35%;">Parameter</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>VRAM</td> <td>Storage temperature</td> <td></td> <td></td> <td></td> </tr> </table>	Symbol	Parameter				VRAM	Storage temperature				<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Symbol</td> <td style="width: 35%;">Parameter</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>VRAM</td> <td>Storage temperature</td> <td></td> <td></td> <td></td> </tr> </table>	Symbol	Parameter				VRAM	Storage temperature													
Symbol	Parameter																															
VRAM	Storage temperature																															
Symbol	Parameter																															
VRAM	Storage temperature																															
Page 84, RECOMMENDED OPERATING CONDITIONS	2: Average output current is the average value of an interval of 100 ms.	2: The average output current is the average value of an interval of 100 ms.																														
Page 85, DC ELECTRICAL CHARACTERISTICS																																
Item 1	(V <sub>CC</sub> = 5V, ****, f <sub>(sys)</sub> = 20 MHz)	(V <sub>CC</sub> = 5V, ****, f <sub>(sys)</sub> = 20 MHz, unless otherwise noted)																														
Item 2	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Symbol</td> <td style="width: 35%;">Test conditions</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>VRAM</td> <td>When clock is stopped.</td> <td></td> <td></td> <td></td> </tr> <tr> <td>ICC</td> <td>Output-only ..... The PLL frequency multiplier stops its operation.</td> <td>f<sub>(sys)</sub> = 20 MHz CPU operates. Ta = 25°C when clock is stopped. Ta = 85°C when clock is stopped.</td> <td></td> <td></td> </tr> </table>	Symbol	Test conditions				VRAM	When clock is stopped.				ICC	Output-only ..... The PLL frequency multiplier stops its operation.	f <sub>(sys)</sub> = 20 MHz CPU operates. Ta = 25°C when clock is stopped. Ta = 85°C when clock is stopped.			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Symbol</td> <td style="width: 35%;">Test conditions</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>VRAM</td> <td>When clock is inactive.</td> <td></td> <td></td> <td></td> </tr> <tr> <td>ICC</td> <td>Output-only ..... The PLL frequency multiplier is inactive.</td> <td>f<sub>(sys)</sub> = 20 MHz CPU is active. Ta = 25°C when clock is inactive. Ta = 85°C when clock is inactive.</td> <td></td> <td></td> </tr> </table>	Symbol	Test conditions				VRAM	When clock is inactive.				ICC	Output-only ..... The PLL frequency multiplier is inactive.	f <sub>(sys)</sub> = 20 MHz CPU is active. Ta = 25°C when clock is inactive. Ta = 85°C when clock is inactive.		
Symbol	Test conditions																															
VRAM	When clock is stopped.																															
ICC	Output-only ..... The PLL frequency multiplier stops its operation.	f <sub>(sys)</sub> = 20 MHz CPU operates. Ta = 25°C when clock is stopped. Ta = 85°C when clock is stopped.																														
Symbol	Test conditions																															
VRAM	When clock is inactive.																															
ICC	Output-only ..... The PLL frequency multiplier is inactive.	f <sub>(sys)</sub> = 20 MHz CPU is active. Ta = 25°C when clock is inactive. Ta = 85°C when clock is inactive.																														
Page 91, External clock input; Timing Requirements	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Symbol</td> <td style="width: 35%;">Parameter</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>tw(half)</td> <td>External clock input pulse width with half input-voltage.</td> <td></td> <td></td> <td></td> </tr> </table>	Symbol	Parameter				tw(half)	External clock input pulse width with half input-voltage.				<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">Symbol</td> <td style="width: 35%;">Parameter</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>tw(half)</td> <td>External clock input pulse width with half input-voltage</td> <td></td> <td></td> <td></td> </tr> </table>	Symbol	Parameter				tw(half)	External clock input pulse width with half input-voltage													
Symbol	Parameter																															
tw(half)	External clock input pulse width with half input-voltage.																															
Symbol	Parameter																															
tw(half)	External clock input pulse width with half input-voltage																															
Pages 91 to 98	MASK ROM ORDER CONFIRMATION FORM • M37906M4C-XXXFP/SP • M37906M6C-XXXFP/SP • M37906M8C-XXXFP/SP	(Deleted)																														
Pages 99, 100	MARK SPECIFICATION FORM • 42P2R-E • 42P4B	(Deleted)																														
Last page	Cotents of Notes (Before revision)	Cotents of Notes (After revision)																														