MITSUBISHI MICROCOMPUTERS

M37906M4C-XXXFP, M37906M4C-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.

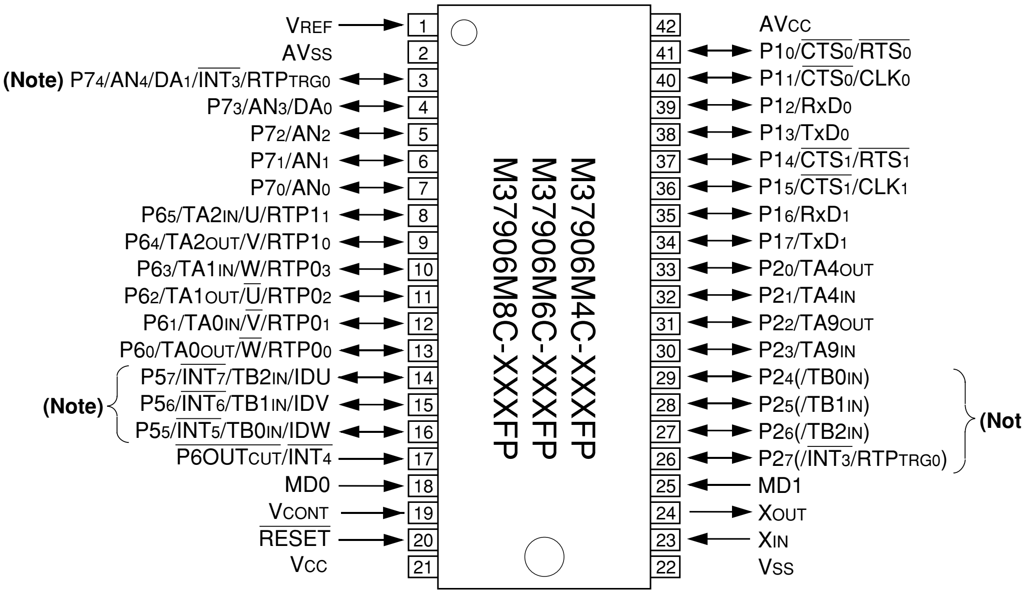
DISTINCTIVE FEATURES

- Number of basic machine instructions 203
- Memory
  - [M37906M4C-XXXFP, M37906M4C-XXXSP]
  - 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

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

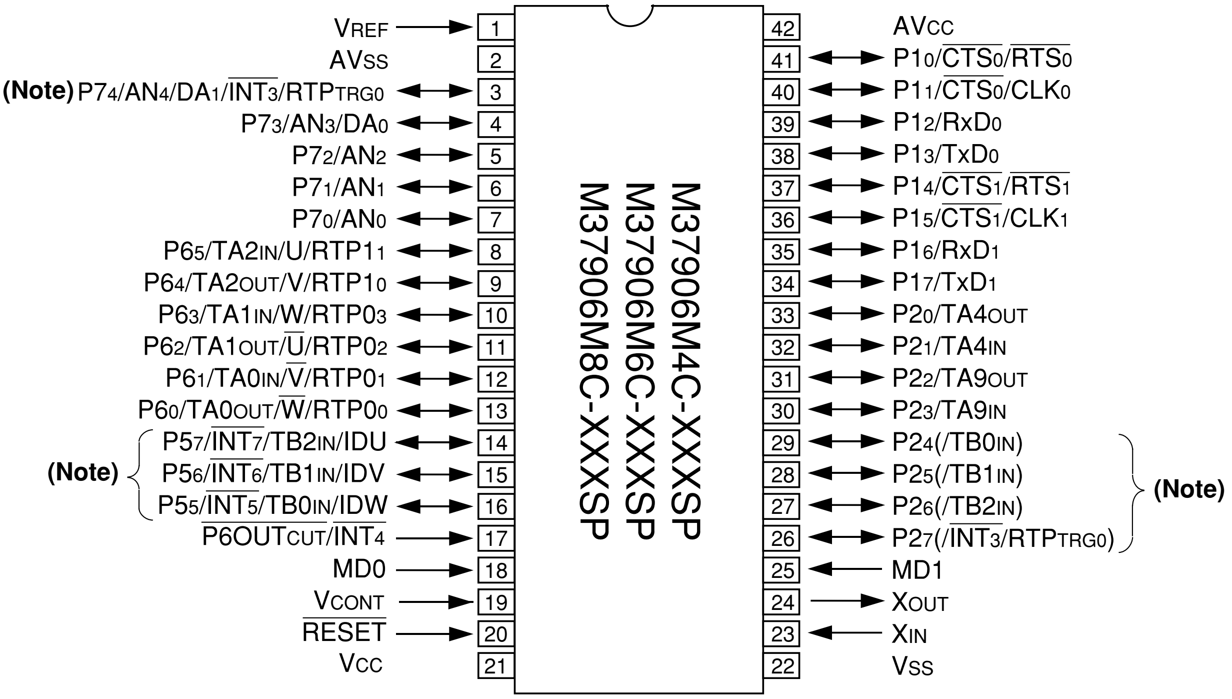
M37906MxC-XXXFP PIN CONFIGURATION (TOP VIEW)


VREF 1
AVSS 2
(Note) P74/AN4/DA1/INT3/RTPTRG0 3
P73/AN3/DA0 4
P72/AN2 5
P71/AN1 6
P70/AN0 7
P65/TA2IN/U/RTP11 8
P64/TA2OUT/V/RTP10 9
P63/TA1IN/W/RTP03 10
P62/TA1OUT/U/RTP02 11
P61/TA0IN/V/RTP01 12
P60/TA0OUT/W/RTP00 13
(Note) P57/INT7/TB2IN/IDU 14
P56/INT6/TB1IN/IDV 15
P55/INT5/TB0IN/IDW 16
P6OUTCUT/INT4 17
MD0 18
VCONT 19
RESET 20
VCC 21
42 AVCC
41 P10/CTS0/RTS0
40 P11/CTS0/CLK0
39 P12/RxD0
38 P13/TxD0
37 P14/CTS1/RTS1
36 P15/CTS1/CLK1
35 P16/RxD1
34 P17/TxD1
33 P20/TA4OUT
32 P21/TA4IN
31 P22/TA9OUT
30 P23/TA9IN
29 P24(/TB0IN)
28 P25(/TB1IN)
27 P26(/TB2IN)
26 P27(/INT3/RTPTRG0) (Note)
25 MD1
24 XOUT
23 XIN
22 VSS
M37906M4C-XXXFP
M37906M6C-XXXFP
M37906M8C-XXXFP

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

Outline 42P2R-E

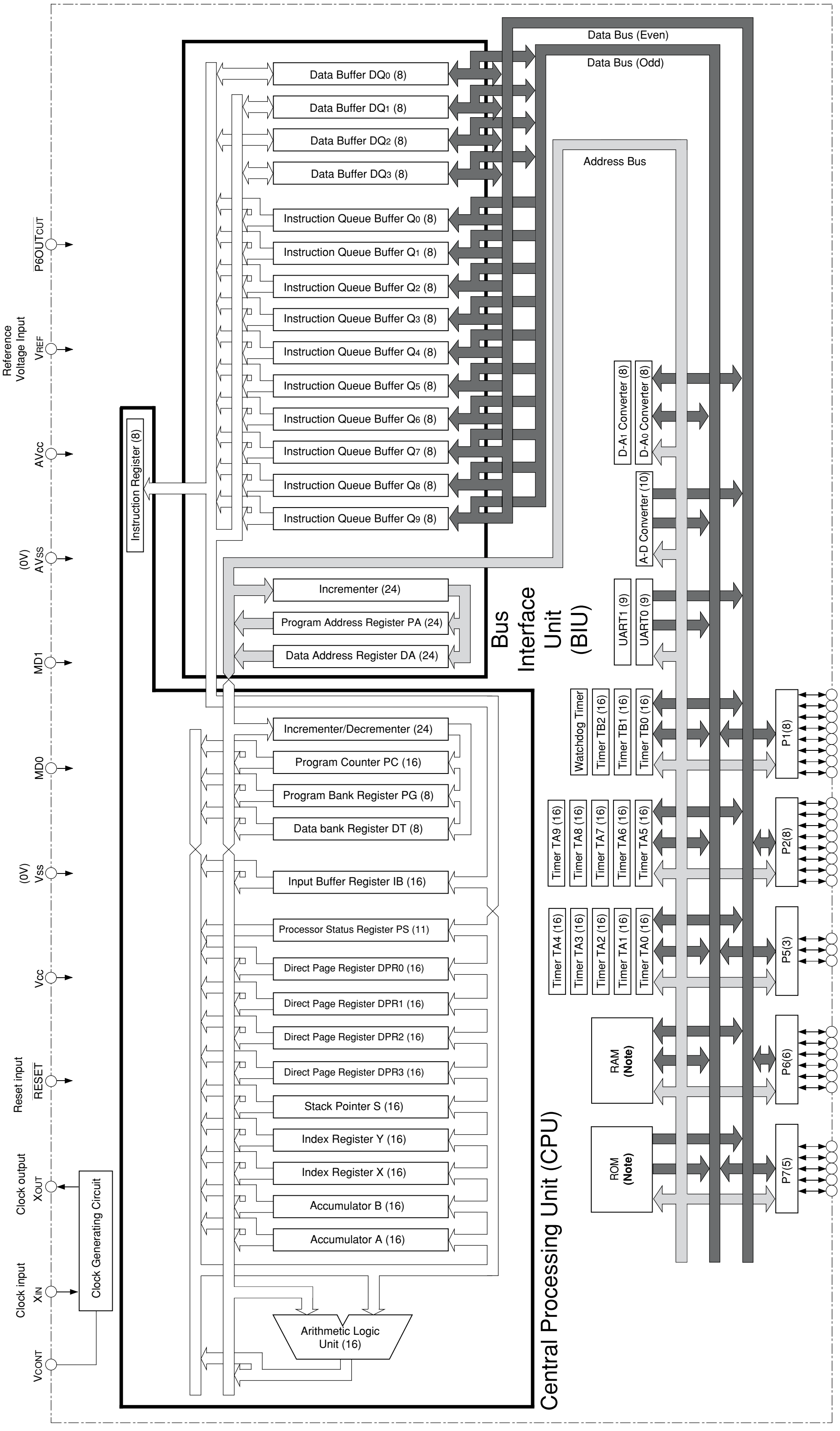
M37906MxC-XXXSP PIN CONFIGURATION (TOP VIEW)


VREF 1
AVSS 2
(Note) P74/AN4/DA1/INT3/RTPTRG0 3
P73/AN3/DA0 4
P72/AN2 5
P71/AN1 6
P70/AN0 7
P65/TA2IN/U/RTP11 8
P64/TA2OUT/V/RTP10 9
P63/TA1IN/W/RTP03 10
P62/TA1OUT/U/RTP02 11
P61/TA0IN/V/RTP01 12
P60/TA0OUT/W/RTP00 13
(Note)
P57/INT7/TB2IN/IDU 14
P56/INT6/TB1IN/IDV 15
P55/INT5/TB0IN/IDW 16
P6OUTCUT/INT4 17
MD0 18
VCONT 19
RESET 20
VCC 21
M37906M4C-XXXSP
M37906M6C-XXXSP
M37906M8C-XXXSP
42 AVCC
41 P10/CTS0/RTS0
40 P11/CTS0/CLK0
39 P12/RxD0
38 P13/TxD0
37 P14/CTS1/RTS1
36 P15/CTS1/CLK1
35 P16/RxD1
34 P17/TxD1
33 P20/TA4OUT
32 P21/TA4IN
31 P22/TA9OUT
30 P23/TA9IN
29 P24(/TB0IN)
28 P25(/TB1IN)
27 P26(/TB2IN)
26 P27(/INT3/RTPTRG0)
(Note)
25 MD1
24 XOUT
23 XIN
22 VSS

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

Outline 42P4B

BLOCK DIAGRAM


VCONT
Clock input
XIN
Clock output
XOUT
Clock Generating Circuit
Reset input
RESET
Vcc
(0V)
Vss
MD0
MD1
(0V)
AVSS
AVcc
Reference Voltage Input
VREF
P6OUTCUT
Central Processing Unit (CPU)
Arithmetic Logic Unit (16)
Accumulator A (16)
Accumulator B (16)
Index Register X (16)
Index Register Y (16)
Stack Pointer S (16)
Direct Page Register DPR3 (16)
Direct Page Register DPR2 (16)
Direct Page Register DPR1 (16)
Direct Page Register DPR0 (16)
Processor Status Register PS (11)
Input Buffer Register IB (16)
Data bank Register DT (8)
Program Bank Register PG (8)
Program Counter PC (16)
Incrementer/Decrementer (24)
Bus Interface Unit (BIU)
Instruction Register (8)
Data Address Register DA (24)
Program Address Register PA (24)
Incrementer (24)
Instruction Queue Buffer Q9 (8)
Instruction Queue Buffer Q8 (8)
Instruction Queue Buffer Q7 (8)
Instruction Queue Buffer Q6 (8)
Instruction Queue Buffer Q5 (8)
Instruction Queue Buffer Q4 (8)
Instruction Queue Buffer Q3 (8)
Instruction Queue Buffer Q2 (8)
Instruction Queue Buffer Q1 (8)
Instruction Queue Buffer Q0 (8)
Data Buffer DQ3 (8)
Data Buffer DQ2 (8)
Data Buffer DQ1 (8)
Data Buffer DQ0 (8)
Data Bus (Even)
Data Bus (Odd)
Address Bus
D-A1 Converter (8)
D-A0 Converter (8)
A-D Converter (10)
UART1 (9)
UART0 (9)
Watchdog Timer
Timer TB2 (16)
Timer TB1 (16)
Timer TB0 (16)
Timer TA9 (16)
Timer TA8 (16)
Timer TA7 (16)
Timer TA6 (16)
Timer TA5 (16)
Timer TA4 (16)
Timer TA3 (16)
Timer TA2 (16)
Timer TA1 (16)
Timer TA0 (16)
RAM (Note)
ROM (Note)
P1(8)
P2(8)
P5(3)
P6(6)
P7(5)
Input/Output port P1
Input/Output port P2
Input/Output port P5
Input/Output port P6
Input/Output port P7

Note:

	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

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(X_{IN})		20 MHz (Max.)
System clock input frequency f(f_{sys})		20 MHz (Max.)
Memory size	ROM	**(Note 1)**
	RAM	**(Note 1)**
Programmable input/output ports	P1, P2	8-bit × 2
	P5	3-bit × 1
	P6	6-bit × 1
	P7	5-bit × 1
Multi-functional timers	TA0–TA9	16-bit × 10
	TB0–TB2	16-bit × 3
Serial I/O	UART0 and UART1	(UART or Clock synchronous serial I/O) × 2
A-D converter		10-bit successive approximation method × 1 (5 channels)
D-A converter		8-bit × 2
Dead-time timer		8-bit × 3
Watchdog timer		12-bit × 1
Interrupts	Maskable interrups	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 interrups	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: × 2, × 3, × 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		**(Note 2)**

Notes 1:

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

2:

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

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.
$\overline{\mathrm{RESET}}$	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.
$P1_0$–$P1_7$	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.
$P2_0$–$P2_7$	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 $\overline{\mathrm{INT}}_3$, or as trigger input pins in the pulse output port mode.
$P5_5$–$P5_7$	I/O port P5	I/O	In addition to having the same functions as port P1, these pins function as input pins for $\overline{\mathrm{INT}}_5$ to $\overline{\mathrm{INT}}_7$. Also, they function as input pins for timers B0 to B2 and position data input pins in the three-phase waveform mode.
$P6_0$–$P6_5$	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.
$P7_0$–$P7_4$	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, $P7_3$ functions as an output pin for the D-A converter, $P7_4$ as an output pin for the D-A converter and an input pin for $\overline{\mathrm{INT}}_3$, and a trigger input pin in the pulse output port mode.
$\overline{\mathrm{P6OUT_{CUT}}}$	$\overline{\mathrm{P6OUT_{CUT}}}$ input	Input	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 $\overline{\mathrm{INT}}_4$, and as an input pin for the port-output-cutoff function in the motor drive waveform output mode.

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 0_{16} through $FFFF_{16}$. This address space is called “bank 0_{16}”.

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

Addresses $FFB4_{16}$ through $FFFF_{16}$ contain the RESET and the interrupt vector addresses, and the interrupt vectors are stored there. For details, refer to the section on interrupts.

Allocated to addresses 0_{16} through FF_{16} 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.

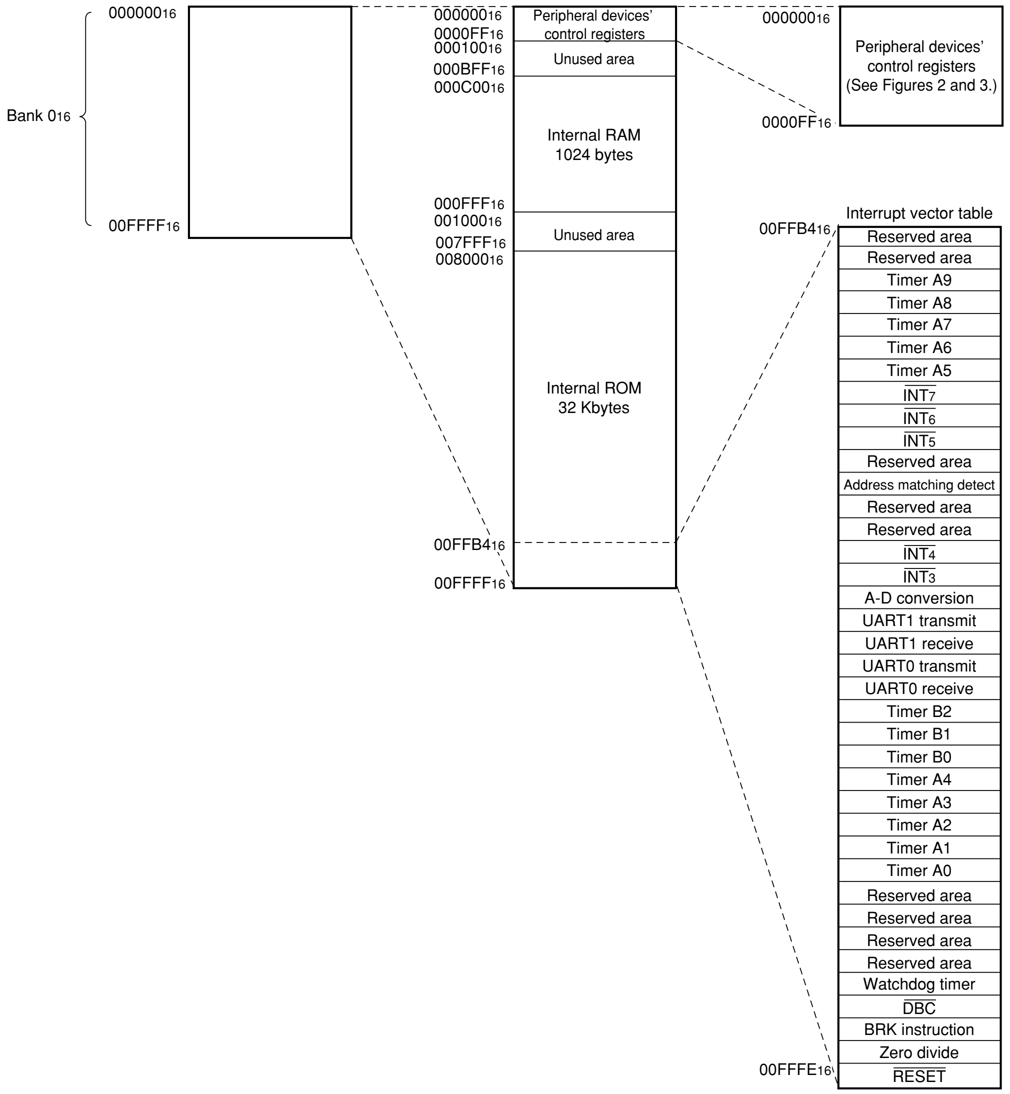

000000_{16}
Bank 0_{16}
$00FFFF_{16}$
000000_{16}
$0000FF_{16}$
000100_{16}
$000BFF_{16}$
$000C00_{16}$
$000FFF_{16}$
001000_{16}
$007FFF_{16}$
008000_{16}
$00FFB4_{16}$
$00FFFF_{16}$
Peripheral devices’ control registers
Unused area
Internal RAM 1024 bytes
Unused area
Internal ROM 32 Kbytes
000000_{16}
$0000FF_{16}$
Peripheral devices’ control registers (See Figures 2 and 3.)
Interrupt vector table
$00FFB4_{16}$
$00FFFE_{16}$
Reserved area
Reserved area
Timer A9
Timer A8
Timer A7
Timer A6
Timer A5
$\overline{\mathrm{INT}_7}$
$\overline{\mathrm{INT}_6}$
$\overline{\mathrm{INT}_5}$
Reserved area
Address matching detect
Reserved area
Reserved area
$\overline{\mathrm{INT}_4}$
$\overline{\mathrm{INT}_3}$
A-D conversion
UART1 transmit
UART1 receive
UART0 transmit
UART0 receive
Timer B2
Timer B1
Timer B0
Timer A4
Timer A3
Timer A2
Timer A1
Timer A0
Reserved area
Reserved area
Reserved area
Reserved area
Watchdog timer
$\overline{\mathrm{DBC}}$
BRK instruction
Zero divide
$\overline{\mathrm{RESET}}$

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

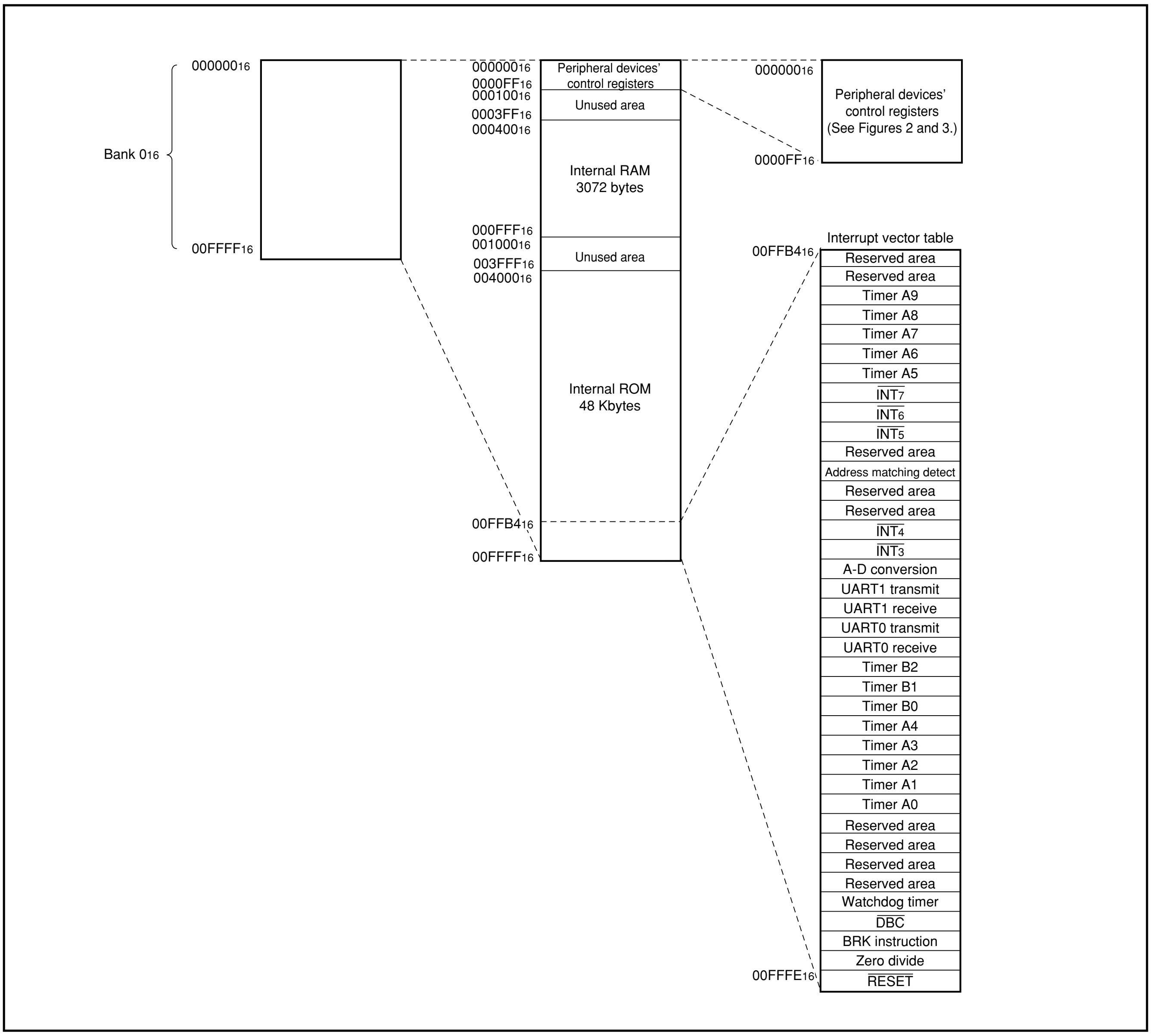

Bank 0_{16}
000000_{16}
$00FFFF_{16}$
000000_{16}
$0000FF_{16}$
000100_{16}
$0003FF_{16}$
000400_{16}
$000FFF_{16}$
001000_{16}
$003FFF_{16}$
004000_{16}
$00FFB4_{16}$
$00FFFF_{16}$
Peripheral devices' control registers
Unused area
Internal RAM 3072 bytes
Unused area
Internal ROM 48 Kbytes
000000_{16}
$0000FF_{16}$
Peripheral devices' control registers (See Figures 2 and 3.)
Interrupt vector table
$00FFB4_{16}$
$00FFFE_{16}$
Reserved area
Reserved area
Timer A9
Timer A8
Timer A7
Timer A6
Timer A5
$\overline{INT_7}$
$\overline{INT_6}$
$\overline{INT_5}$
Reserved area
Address matching detect
Reserved area
Reserved area
$\overline{INT_4}$
$\overline{INT_3}$
A-D conversion
UART1 transmit
UART1 receive
UART0 transmit
UART0 receive
Timer B2
Timer B1
Timer B0
Timer A4
Timer A3
Timer A2
Timer A1
Timer A0
Reserved area
Reserved area
Reserved area
Reserved area
Watchdog timer
$\overline{DBC}$
BRK instruction
Zero divide
$\overline{RESET}$

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

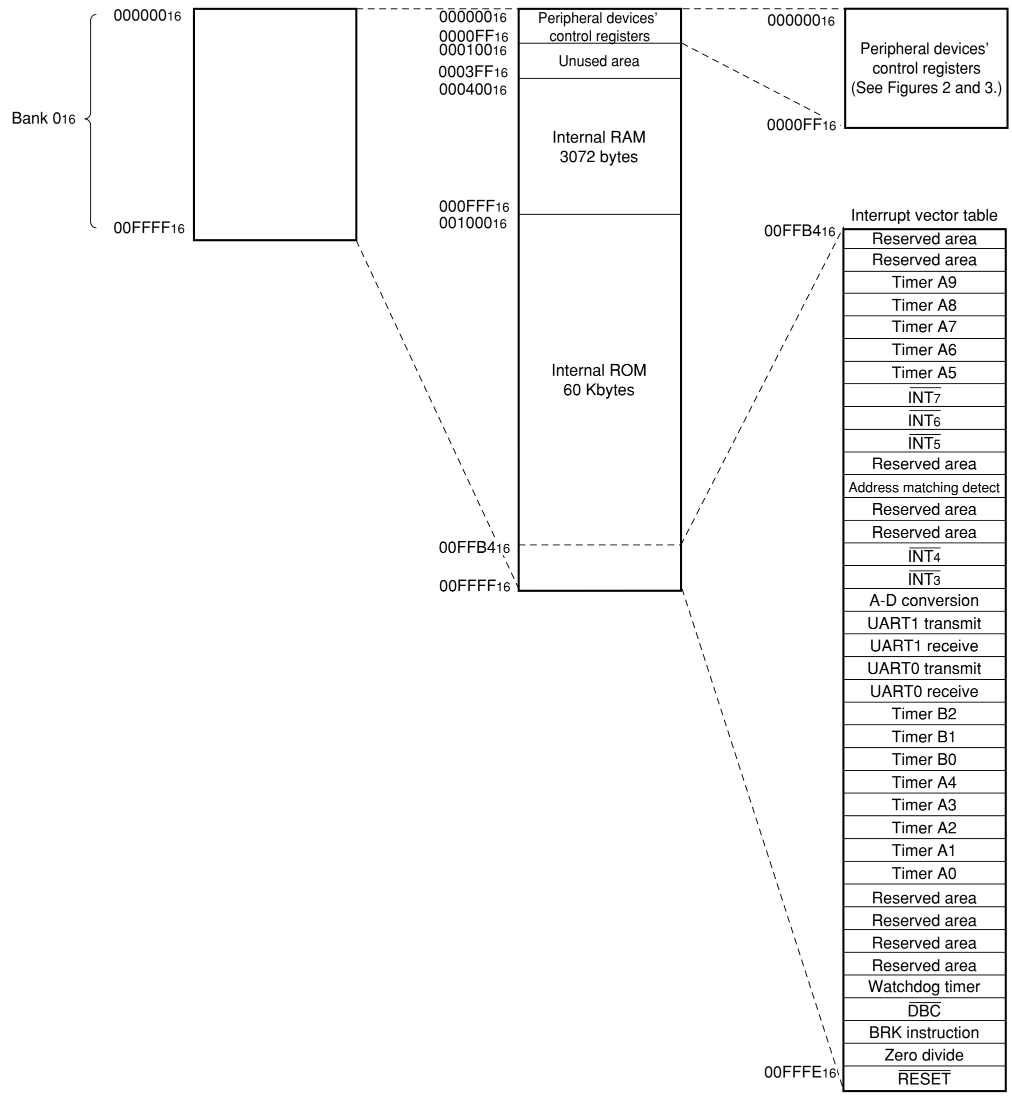

Bank 0₁₆
000000₁₆
00FFFF₁₆
000000₁₆
0000FF₁₆
000100₁₆
0003FF₁₆
000400₁₆
000FFF₁₆
001000₁₆
00FFB4₁₆
00FFFF₁₆
Peripheral devices' control registers
Unused area
Internal RAM 3072 bytes
Internal ROM 60 Kbytes
000000₁₆
0000FF₁₆
Peripheral devices' control registers (See Figures 2 and 3.)
Interrupt vector table
00FFB4₁₆
00FFFE₁₆
Reserved area
Reserved area
Timer A9
Timer A8
Timer A7
Timer A6
Timer A5
INT7
INT6
INT5
Reserved area
Address matching detect
Reserved area
Reserved area
INT4
INT3
A-D conversion
UART1 transmit
UART1 receive
UART0 transmit
UART0 receive
Timer B2
Timer B1
Timer B0
Timer A4
Timer A3
Timer A2
Timer A1
Timer A0
Reserved area
Reserved area
Reserved area
Reserved area
Watchdog timer
DBC
BRK instruction
Zero divide
RESET

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

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

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

Note: Do not write to this address.

Fig. 2 Location of SFRs (1)

Address (Hexadecimal notation)	
000080_{16}	Reserved area **(Note)**
000081_{16}	Reserved area **(Note)**
000082_{16}	Reserved area **(Note)**
000083_{16}	Reserved area **(Note)**
000084_{16}	Reserved area **(Note)**
000085_{16}	Reserved area **(Note)**
000086_{16}	Reserved area **(Note)**
000087_{16}	Reserved area **(Note)**
000088_{16}	
000089_{16}	
$00008A_{16}$	Reserved area **(Note)**
$00008B_{16}$	
$00008C_{16}$	Reserved area **(Note)**
$00008D_{16}$	
$00008E_{16}$	Reserved area **(Note)**
$00008F_{16}$	
000090_{16}	Reserved area **(Note)**
000091_{16}	
000092_{16}	Reserved area **(Note)**
000093_{16}	
000094_{16}	
000095_{16}	External interrupt input read-out register
000096_{16}	D-A control register
000097_{16}	
000098_{16}	D-A register 0
000099_{16}	D-A register 1
$00009A_{16}$	
$00009B_{16}$	
$00009C_{16}$	
$00009D_{16}$	
$00009E_{16}$	
$00009F_{16}$	
$0000A0_{16}$	Reserved area **(Note)**
$0000A1_{16}$	
$0000A2_{16}$	Reserved area **(Note)**
$0000A3_{16}$	
$0000A4_{16}$	Reserved area **(Note)**
$0000A5_{16}$	
$0000A6_{16}$	Waveform output mode register
$0000A7_{16}$	Dead-time timer
$0000A8_{16}$	Three-phase output data register 0
$0000A9_{16}$	Three-phase output data register 1
$0000AA_{16}$	Position-data-retain function control register
$0000AB_{16}$	
$0000AC_{16}$	Serial I/O pin control register
$0000AD_{16}$	
$0000AE_{16}$	Port P2 pin function control register
$0000AF_{16}$	
$0000B0_{16}$	Reserved area **(Note)**
$0000B1_{16}$	Reserved area **(Note)**
$0000B2_{16}$	Reserved area **(Note)**
$0000B3_{16}$	Reserved area **(Note)**
$0000B4_{16}$	Reserved area **(Note)**
$0000B5_{16}$	Reserved area **(Note)**
$0000B6_{16}$	Reserved area **(Note)**
$0000B7_{16}$	Reserved area **(Note)**
$0000B8_{16}$	Reserved area **(Note)**
$0000B9_{16}$	
$0000BA_{16}$	Reserved area **(Note)**
$0000BB_{16}$	Reserved area **(Note)**
$0000BC_{16}$	Clock control register 0
$0000BD_{16}$	Reserved area **(Note)**
$0000BE_{16}$	Reserved area **(Note)**
$0000BF_{16}$	Reserved area **(Note)**

Address (Hexadecimal notation)	
$0000C0_{16}$	
$0000C1_{16}$	
$0000C2_{16}$	
$0000C3_{16}$	
$0000C4_{16}$	Up-down register 1
$0000C5_{16}$	
$0000C6_{16}$	Timer A5 register
$0000C7_{16}$	
$0000C8_{16}$	Timer A6 register
$0000C9_{16}$	
$0000CA_{16}$	Timer A7 register
$0000CB_{16}$	
$0000CC_{16}$	Timer A8 register
$0000CD_{16}$	
$0000CE_{16}$	Timer A9 register
$0000CF_{16}$	
$0000D0_{16}$	Timer $A0_1$ register
$0000D1_{16}$	
$0000D2_{16}$	Timer $A1_1$ register
$0000D3_{16}$	
$0000D4_{16}$	Timer $A2_1$ register
$0000D5_{16}$	
$0000D6_{16}$	Timer A5 mode register
$0000D7_{16}$	Timer A6 mode register
$0000D8_{16}$	Timer A7 mode register
$0000D9_{16}$	Timer A8 mode register
$0000DA_{16}$	Timer A9 mode register
$0000DB_{16}$	Reserved area **(Note)**
$0000DC_{16}$	Comparator function select register 0
$0000DD_{16}$	Reserved area **(Note)**
$0000DE_{16}$	Comparator result register 0
$0000DF_{16}$	Reserved area **(Note)**
$0000E0_{16}$	Reserved area **(Note)**
$0000E1_{16}$	Reserved area **(Note)**
$0000E2_{16}$	Reserved area **(Note)**
$0000E3_{16}$	Reserved area **(Note)**
$0000E4_{16}$	Reserved area **(Note)**
$0000E5_{16}$	Reserved area **(Note)**
$0000E6_{16}$	Reserved area **(Note)**
$0000E7_{16}$	Reserved area **(Note)**
$0000E8_{16}$	Reserved area **(Note)**
$0000E9_{16}$	Reserved area **(Note)**
$0000EA_{16}$	Reserved area **(Note)**
$0000EB_{16}$	Reserved area **(Note)**
$0000EC_{16}$	Reserved area **(Note)**
$0000ED_{16}$	Reserved area **(Note)**
$0000EE_{16}$	Reserved area **(Note)**
$0000EF_{16}$	Reserved area **(Note)**
$0000F0_{16}$	
$0000F1_{16}$	Reserved area **(Note)**
$0000F2_{16}$	Reserved area **(Note)**
$0000F3_{16}$	
$0000F4_{16}$	
$0000F5_{16}$	Timer A5 interrupt control register
$0000F6_{16}$	Timer A6 interrupt control register
$0000F7_{16}$	Timer A7 interrupt control register
$0000F8_{16}$	Timer A8 interrupt control register
$0000F9_{16}$	Timer A9 interrupt control register
$0000FA_{16}$	
$0000FB_{16}$	
$0000FC_{16}$	
$0000FD_{16}$	$\overline{INT_5}$ interrupt control register
$0000FE_{16}$	$\overline{INT_6}$ interrupt control register
$0000FF_{16}$	$\overline{INT_7}$ interrupt control register

Note: Do not write to this address.

Fig. 3 Location of SFRs (2)

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).

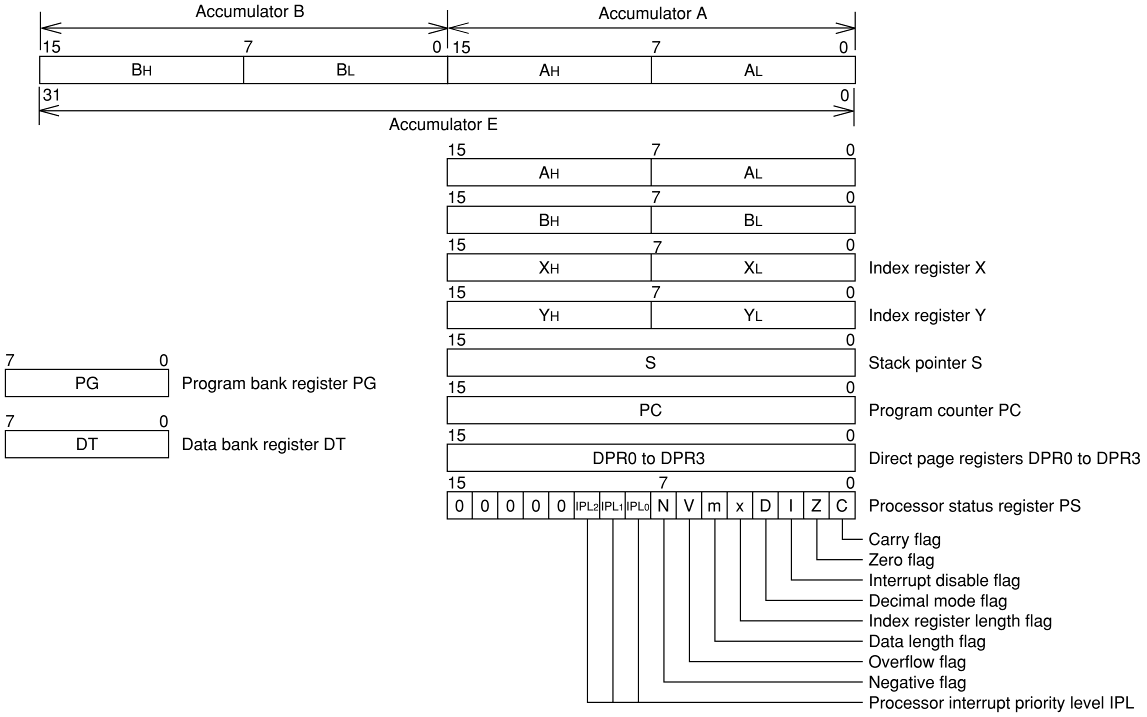

Accumulator B
Accumulator A
15
7
0
15
7
0
BH
BL
AH
AL
31
0
Accumulator E
AH
AL
BH
BL
XH
XL
Index register X
YH
YL
Index register Y
S
Stack pointer S
PG
Program bank register PG
PC
Program counter PC
DT
Data bank register DT
DPR0 to DPR3
Direct page registers DPR0 to DPR3
0 0 0 0 0 IPL2 IPL1 IPL0 N V m x D I Z C
Processor status register PS
Carry flag
Zero flag
Interrupt disable flag
Decimal mode flag
Index register length flag
Data length flag
Overflow flag
Negative flag
Processor interrupt priority level IPL

Fig. 4 Register structure

STACK POINTER (S)

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

PROGRAM COUNTER (PC)

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

PROGRAM BANK REGISTER (PG)

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

DATA BANK REGISTER (DT)

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

DIRECT PAGE REGISTERS 0 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.

5. Index register length flag (x)

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

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

6. Data length flag (m)

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

7. Overflow flag (V)

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

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

8. Negative flag (N)

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

9. Processor interrupt priority level (IPL)

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

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

BANK

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

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

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

Note: Some addressing modes directly specify a bank.

DIRECT PAGE

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

■ Conventional direct addressing mode

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

■ Expanded direct addressing mode

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

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

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

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

Instruction Set

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

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

BUS INTERFACE UNIT

Data transfer between the central processing unit (CPU) and internal memory, internal peripheral devices 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 peripheral devices uses the following internal common buses: 32-bit code bus, 16-bit data bus, 24-bit address bus, and control signals.

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

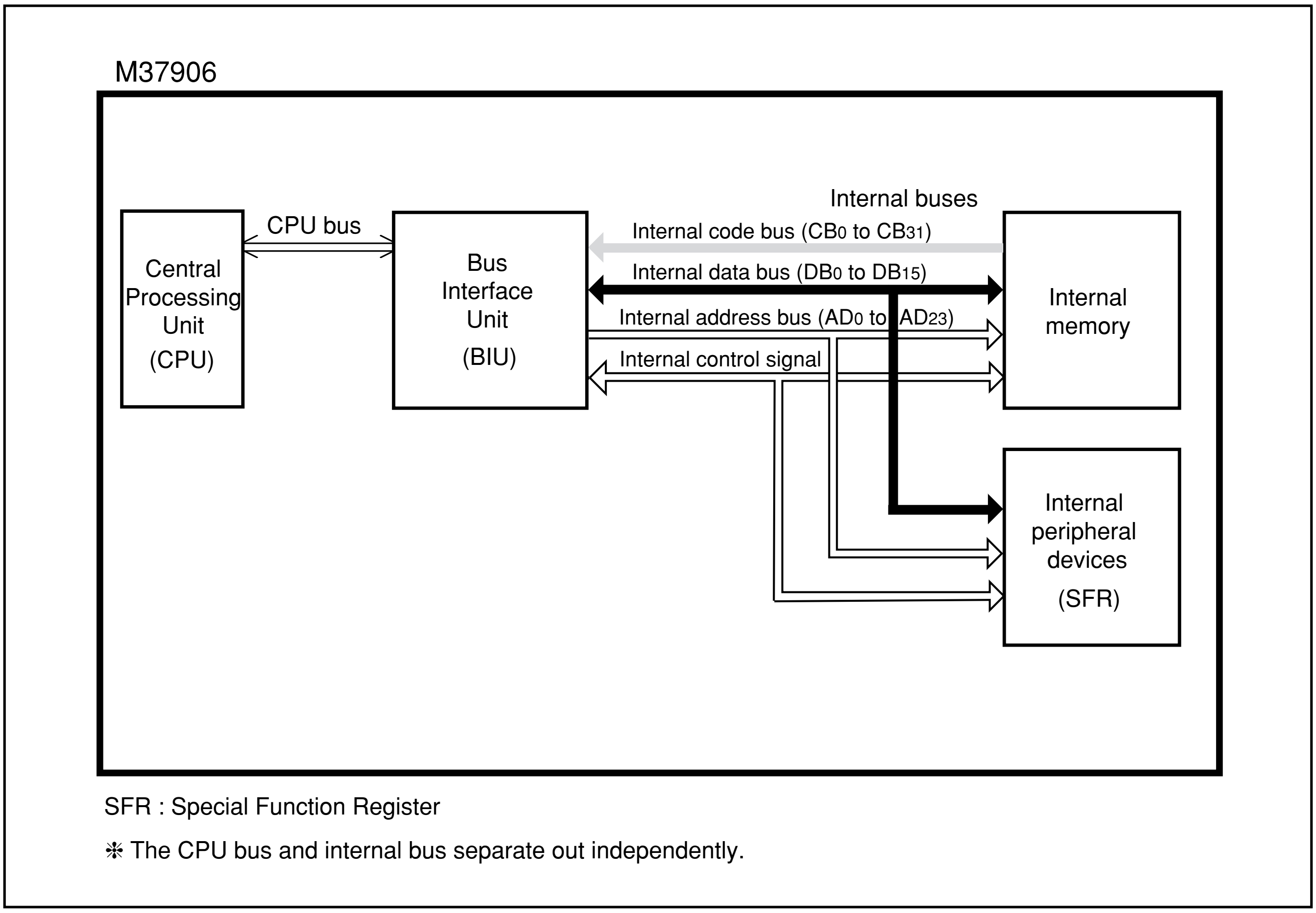

M37906
Central Processing Unit (CPU)
CPU bus
Bus Interface Unit (BIU)
Internal buses
Internal code bus (CB0 to CB31)
Internal data bus (DB0 to DB15)
Internal address bus (AD0 to AD23)
Internal control signal
Internal memory
Internal peripheral devices (SFR)
SFR : Special Function Register
✻ The CPU bus and internal bus separate out independently.

Fig. 5 BIU and bus structure

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.

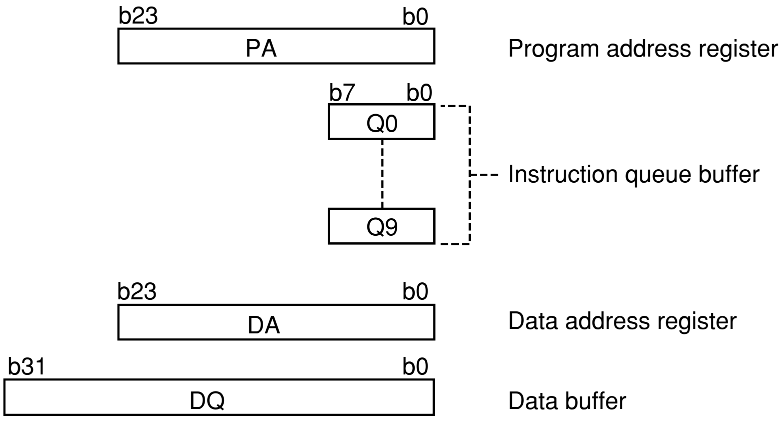

b23
b0
PA
Program address register
b7
b0
Q0
Q9
Instruction queue buffer
b23
b0
DA
Data address register
b31
b0
DQ
Data buffer

Fig. 6 Register structure of BIU

BIU Functions

(1) Instruction prefetch

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

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

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

(2) Data read operation

When executing an instruction for reading data from the internal memory 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 lengh 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		
	AD_2 (A_2)	AD_1 (A_1)	AD_0 (A_0)
Even address	X	X	0
4-byte boundary	X	0	0
8-byte boundary	0	0	0

X: 0 or 1

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

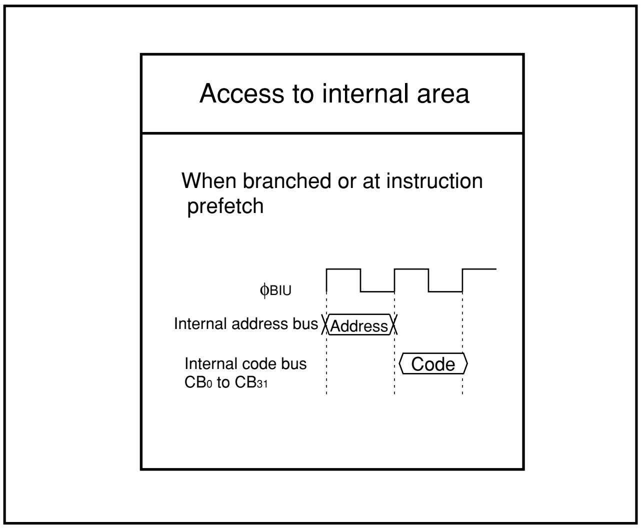


Fig. 7 Bus cycle waveform example for instruction prefetch

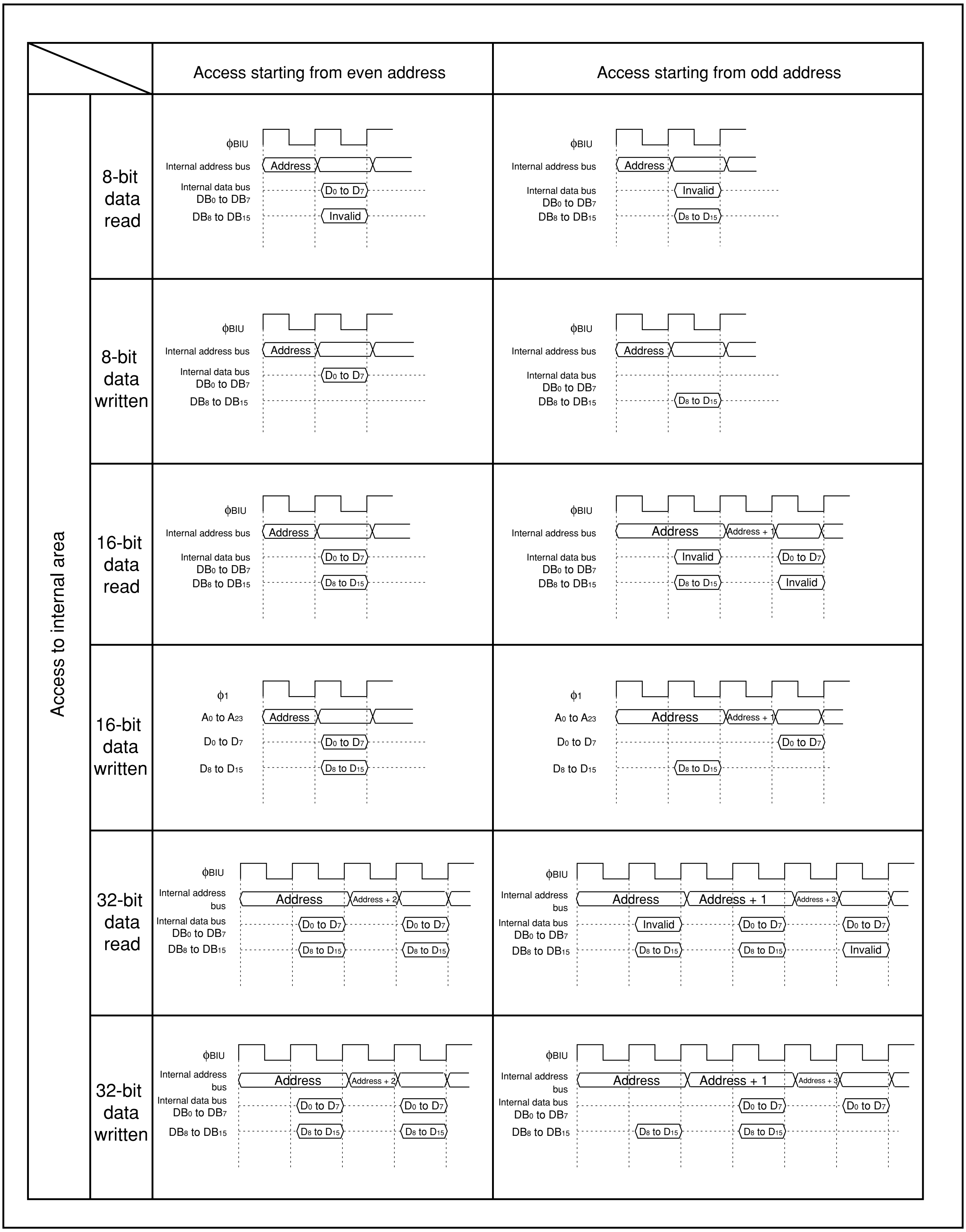
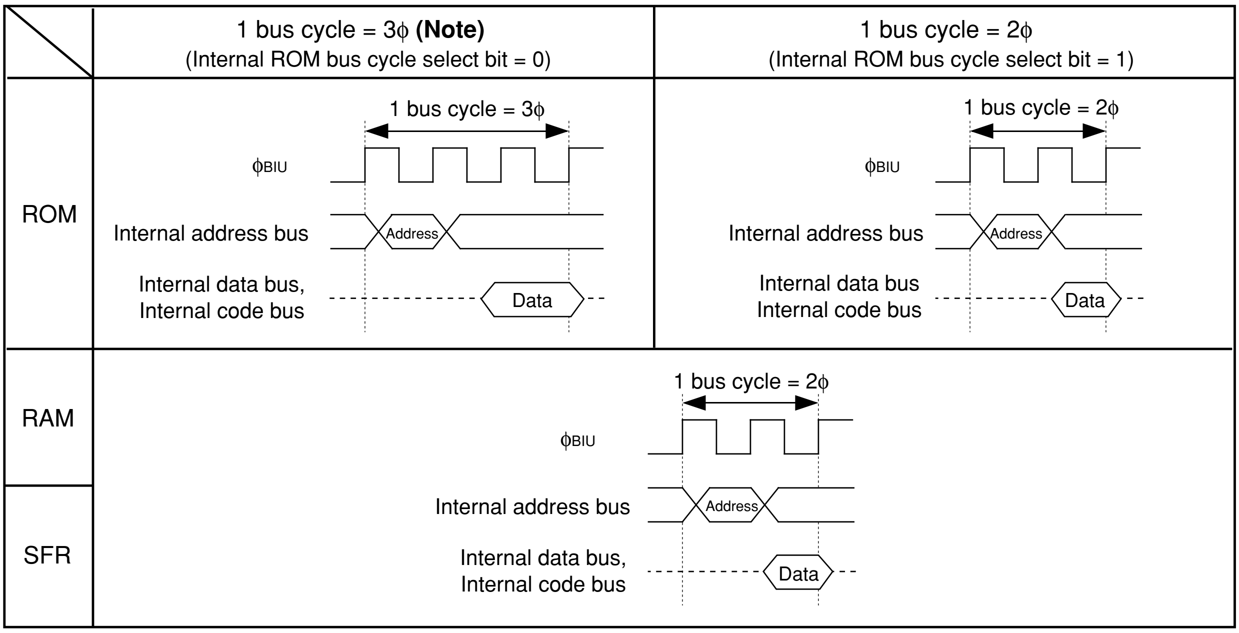

Access starting from even address
Access starting from odd address
Access to internal area
8-bit data read
8-bit data written
16-bit data read
16-bit data written
32-bit data read
32-bit data written
φBIU
φ1
Internal address bus
Internal data bus
DB0 to DB7
DB8 to DB15
A0 to A23
D0 to D7
D8 to D15
Address
Address + 1
Address + 2
Address + 3
D0 to D7
D8 to D15
Invalid

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

● 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 $5F_{16}$), 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.


1 bus cycle = 3ϕ (Note)
(Internal ROM bus cycle select bit = 0)
1 bus cycle = 2ϕ
(Internal ROM bus cycle select bit = 1)
ROM
1 bus cycle = 3ϕ
ϕBIU
Internal address bus
Address
Internal data bus,
Internal code bus
Data
1 bus cycle = 2ϕ
ϕBIU
Internal address bus
Address
Internal data bus
Internal code bus
Data
RAM
SFR
1 bus cycle = 2ϕ
ϕBIU
Internal address bus
Address
Internal data bus,
Internal code bus
Data

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

Fig. 9 Bus cycle waveform at access to internal area

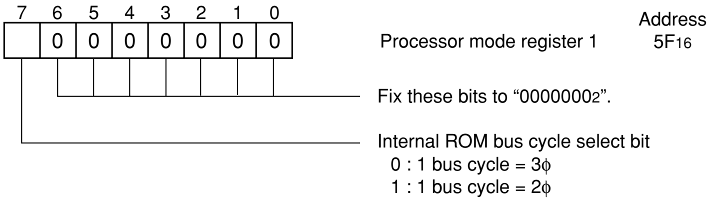

7 6 5 4 3 2 1 0
0 0 0 0 0 0 0
Processor mode register 1
Address
5F16
Fix these bits to "00000002".
Internal ROM bus cycle select bit
0 : 1 bus cycle = 3ϕ
1 : 1 bus cycle = 2ϕ

Fig. 10 Bit configuration of processor mode register 1

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 $5E_{16}$), which is shown in Figure 11, to "00_2".

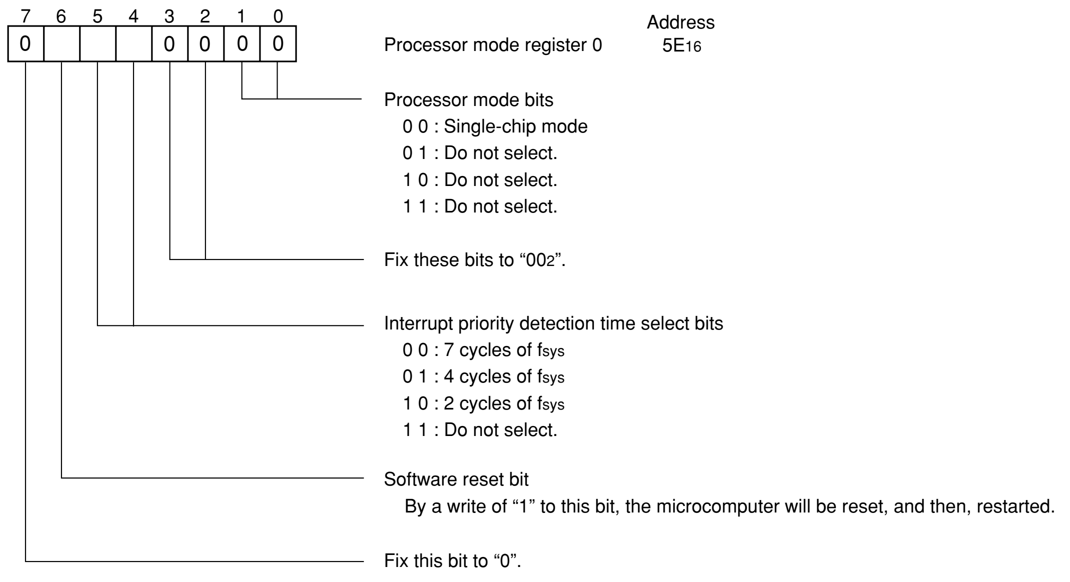

7 6 5 4 3 2 1 0
0 0 0 0 0
Address
Processor mode register 0 $5E_{16}$
Processor mode bits
0 0 : Single-chip mode
0 1 : Do not select.
1 0 : Do not select.
1 1 : Do not select.
Fix these bits to "00_2".
Interrupt priority detection time select bits
0 0 : 7 cycles of f_{sys}
0 1 : 4 cycles of f_{sys}
1 0 : 2 cycles of f_{sys}
1 1 : Do not select.
Software reset bit
By a write of "1" to this bit, the microcomputer will be reset, and then, restarted.
Fix this bit to "0".

Fig. 11 Bit configuration of processor mode register 0

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.

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

Interrupts other than reset, watchdog timer, zero divide, 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 $\overline{\text{INT}}_i$ (i = 3 to 7) interrupt request is generated by an external input.

$\overline{\text{INT}}_i$ 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 $\overline{\text{INT}}_3$/RTP$_{\text{TRG0}}$ is allocated can be selected by the pin $\overline{\text{INT}}_3$/RTP$_{\text{TRG0}}$ 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$_7$/$\overline{\text{INT}}_3$, P5$_5$/$\overline{\text{INT}}_5$, P5$_6$/$\overline{\text{INT}}_6$, and P5$_7$(P7$_4$)/$\overline{\text{INT}}_7$.

When the external interrupt input read register (address 95$_{16}$), which is shown in Figure 12, is read out, the status of pins $\overline{\text{INT}}_3$ through $\overline{\text{INT}}_7$ 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$_{16}$ 00FFB9$_{16}$
Timer A8	00FFBA$_{16}$ 00FFBB$_{16}$
Timer A7	00FFBC$_{16}$ 00FFBD$_{16}$
Timer A6	00FFBE$_{16}$ 00FFBF$_{16}$
Timer A5	00FFC0$_{16}$ 00FFC1$_{16}$
$\overline{\text{INT}}_7$ external interrupt	00FFC2$_{16}$ 00FFC3$_{16}$
$\overline{\text{INT}}_6$ external interrupt	00FFC4$_{16}$ 00FFC5$_{16}$
$\overline{\text{INT}}_5$ external interrupt	00FFC6$_{16}$ 00FFC7$_{16}$
Address matching detection interrupt	00FFCA$_{16}$ 00FFCB$_{16}$
$\overline{\text{INT}}_4$ external interrupt	00FFD0$_{16}$ 00FFD1$_{16}$
$\overline{\text{INT}}_3$ external interrupt	00FFD2$_{16}$ 00FFD3$_{16}$
A-D conversion	00FFD4$_{16}$ 00FFD5$_{16}$
UART1 transmit	00FFD6$_{16}$ 00FFD7$_{16}$
UART1 receive	00FFD8$_{16}$ 00FFD9$_{16}$
UART0 transmit	00FFDA$_{16}$ 00FFDB$_{16}$
UART0 receive	00FFDC$_{16}$ 00FFDD$_{16}$
Timer B2	00FFDE$_{16}$ 00FFDF$_{16}$
Timer B1	00FFE0$_{16}$ 00FFE1$_{16}$
Timer B0	00FFE2$_{16}$ 00FFE3$_{16}$
Timer A4	00FFE4$_{16}$ 00FFE5$_{16}$
Timer A3	00FFE6$_{16}$ 00FFE7$_{16}$
Timer A2	00FFE8$_{16}$ 00FFE9$_{16}$
Timer A1	00FFEA$_{16}$ 00FFEB$_{16}$
Timer A0	00FFEC$_{16}$ 00FFED$_{16}$
Watchdog timer	00FFF6$_{16}$ 00FFF7$_{16}$
$\overline{\text{DBC}}$ (Do not select.)	00FFF8$_{16}$ 00FFF9$_{16}$
Break instruction (Do not select.)	00FFFA$_{16}$ 00FFFB$_{16}$
Zero divide	00FFFC$_{16}$ 00FFFD$_{16}$
Reset	00FFFE$_{16}$ 00FFFF$_{16}$

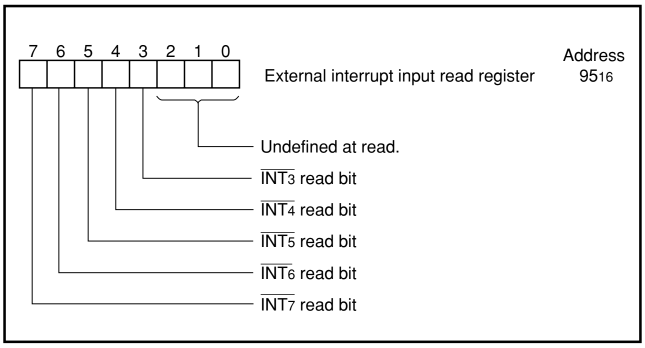
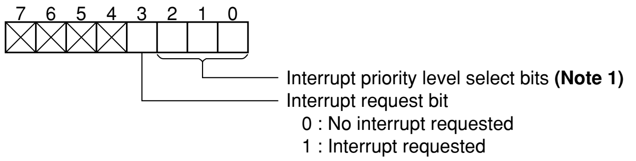
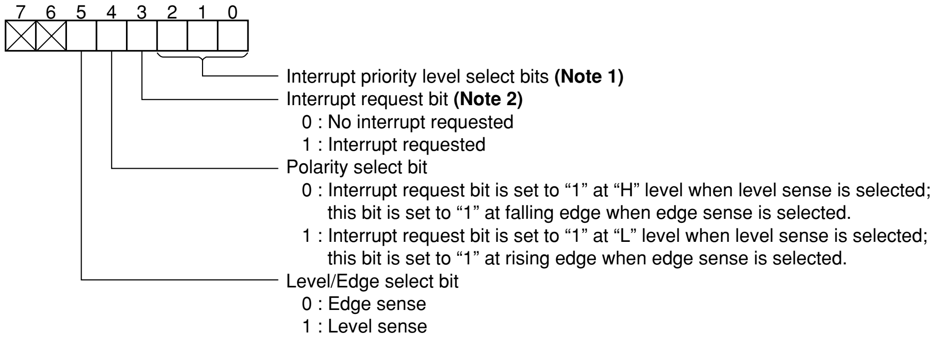

7 6 5 4 3 2 1 0
External interrupt input read register
Address
95$_{16}$
Undefined at read.
$\overline{\text{INT}}_3$ read bit
$\overline{\text{INT}}_4$ read bit
$\overline{\text{INT}}_5$ read bit
$\overline{\text{INT}}_6$ read bit
$\overline{\text{INT}}_7$ read bit

Fig. 12 Bit configuration of external interrupt input read register



Interrupt priority level select bits **(Note 1)**
Interrupt request bit
0 : No interrupt requested
1 : Interrupt requested

Interrupt control register bit configuration for A-D converter, UART0, UART1, timer A0 to timer A9, and timer B0 to timer B2.



Interrupt priority level select bits **(Note 1)**
Interrupt request bit **(Note 2)**
0 : No interrupt requested
1 : Interrupt requested
Polarity select bit
0 : Interrupt request bit is set to “1” at “H” level when level sense is selected; this bit is set to “1” at falling edge when edge sense is selected.
1 : Interrupt request bit is set to “1” at “L” level when level sense is selected; this bit is set to “1” at rising edge when edge sense is selected.
Level/Edge select bit
0 : Edge sense
1 : Level sense

Interrupt control register bit configuration for $\overline{\text{INT}}_3$– $\overline{\text{INT}}_7$

Notes 1: Use the **MOVM (MOVMB)** instruction or the **STA (STAB, STAD)** instruction for writing to this bit.
2: Interrupt request bits of $\overline{\text{INT}}_3$ to $\overline{\text{INT}}_7$ are invalid when the level sense is selected.

Fig. 13 Bit configuration of interrupt control register

Table 4. Addresses of interrupt control registers

Interrupt control registers	Addresses
$\overline{\mathrm{INT}_3}$ interrupt control register	00006E$_{16}$
$\overline{\mathrm{INT}_4}$ interrupt control register	00006F$_{16}$
A-D interrupt control register	000070$_{16}$
UART0 transmit interrupt control register	000071$_{16}$
UART0 receive interrupt control register	000072$_{16}$
UART1 transmit interrupt control register	000073$_{16}$
UART1 receive interrupt control register	000074$_{16}$
Timer A0 interrupt control register	000075$_{16}$
Timer A1 interrupt control register	000076$_{16}$
Timer A2 interrupt control register	000077$_{16}$
Timer A3 interrupt control register	000078$_{16}$
Timer A4 interrupt control register	000079$_{16}$
Timer B0 interrupt control register	00007A$_{16}$
Timer B1 interrupt control register	00007B$_{16}$
Timer B2 interrupt control register	00007C$_{16}$
Timer A5 interrupt control register	0000F5$_{16}$
Timer A6 interrupt control register	0000F6$_{16}$
Timer A7 interrupt control register	0000F7$_{16}$
Timer A8 interrupt control register	0000F8$_{16}$
Timer A9 interrupt control register	0000F9$_{16}$
$\overline{\mathrm{INT}_5}$ interrupt control register	0000FD$_{16}$
$\overline{\mathrm{INT}_6}$ interrupt control register	0000FE$_{16}$
$\overline{\mathrm{INT}_7}$ interrupt control register	0000FF$_{16}$

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.

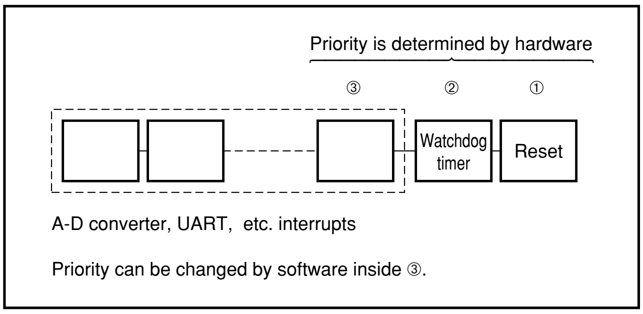

Priority is determined by hardware
③
②
①
Watchdog timer
Reset
A-D converter, UART, etc. interrupts
Priority can be changed by software inside ③.

Fig. 14 Interrupt priority

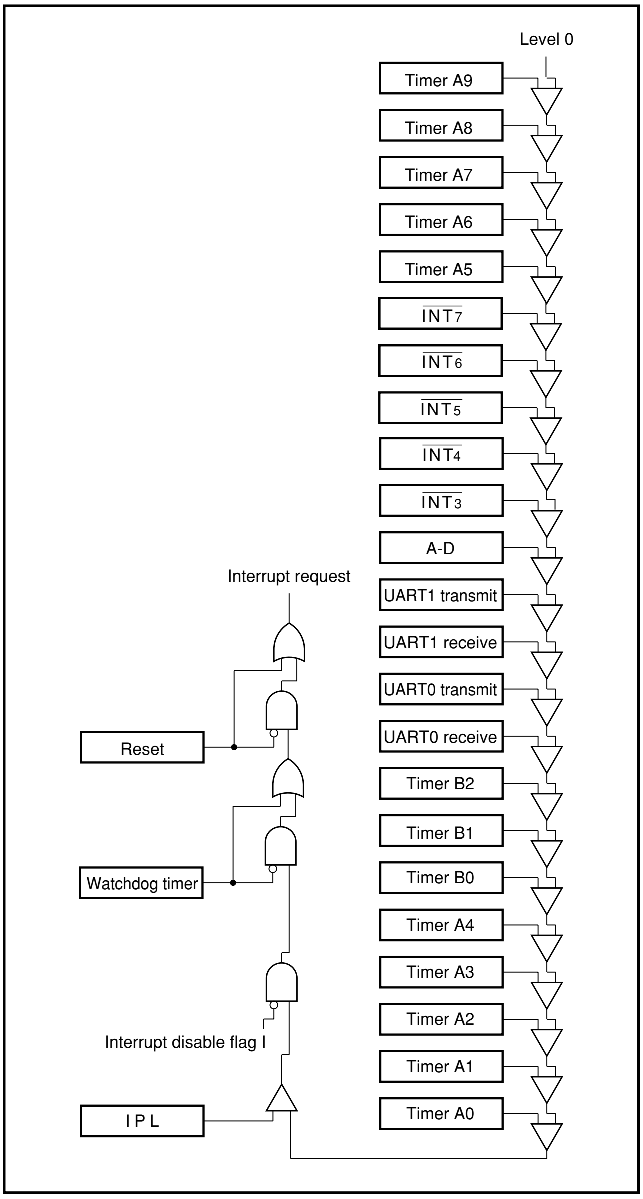

Level 0
Timer A9
Timer A8
Timer A7
Timer A6
Timer A5
INT7
INT6
INT5
INT4
INT3
A-D
UART1 transmit
UART1 receive
UART0 transmit
UART0 receive
Timer B2
Timer B1
Timer B0
Timer A4
Timer A3
Timer A2
Timer A1
Timer A0
Interrupt request
Reset
Watchdog timer
Interrupt disable flag I
I P L

Fig. 15 Interrupt priority detection

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 $5E_{16}$) 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 “00_{16}.” Therefore, the longest time is automatically set, however, the shortest time must be selected by software.

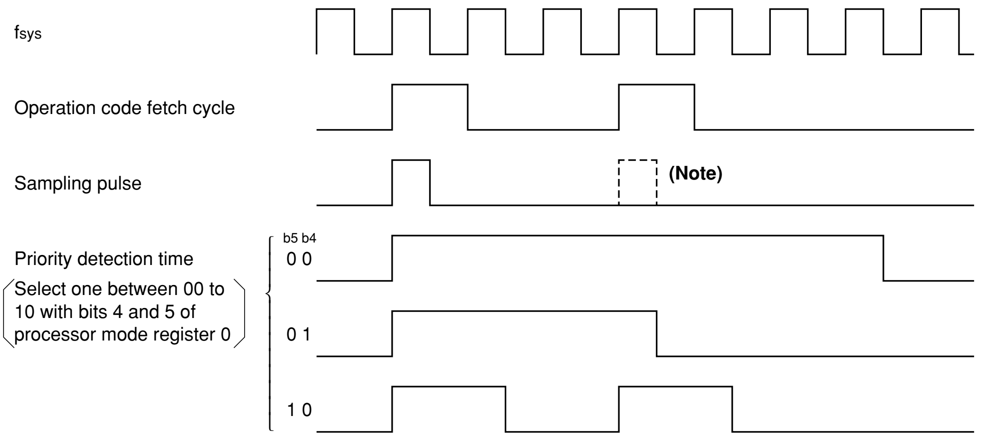
Table 5. Value loaded in processor interrupt level (IPL) during an interrupt

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

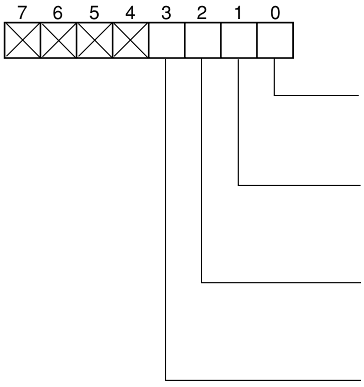
Priority detection time select bit		Number of cycles **(Note)**
Bit 5	Bit 4	
0	0	7 cycles of f_{sys}
0	1	4 cycles of f_{sys}
1	0	2 cycles of f_{sys}

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


fsys
Operation code fetch cycle
Sampling pulse
(Note)
Priority detection time
Select one between 00 to 10 with bits 4 and 5 of processor mode register 0
b5 b4
0 0
0 1
1 0

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

Fig. 16 Interrupt priority detection time



Port P2 pin function control register

Address
AE$_{16}$

Pin TB0$_{\mathrm{IN}}$ select bit
0: Allocate pin TB0$_{\mathrm{IN}}$ to P5$_5$.
1: Allocate pin TB0$_{\mathrm{IN}}$ to P2$_4$.

Pin TB1$_{\mathrm{IN}}$ select bit
0: Allocate pin TB1$_{\mathrm{IN}}$ to P5$_6$.
1: Allocate pin TB1$_{\mathrm{IN}}$ to P2$_5$.

Pin TB2$_{\mathrm{IN}}$ select bit
0: Allocate pin TB2$_{\mathrm{IN}}$ to P5$_7$.
1: Allocate pin TB2$_{\mathrm{IN}}$ to P2$_6$.

Pin $\overline{\mathrm{INT_3}}$/RTP$_{\mathrm{TRG0}}$ select bit
0: Allocate pin $\overline{\mathrm{INT_3}}$/RTP$_{\mathrm{TRG0}}$ to P7$_4$ **(Note)**.
1: Allocate pin $\overline{\mathrm{INT_3}}$/RTP$_{\mathrm{TRG0}}$ to P2$_7$.

Note: When pin $\overline{\mathrm{INT_3}}$/RTP$_{\mathrm{TRG0}}$ is allocated to P7$_4$, be sure that the D-A$_1$ output enable bit (bit 1 at address 96$_{16}$) = “0” (output disabled).

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

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.

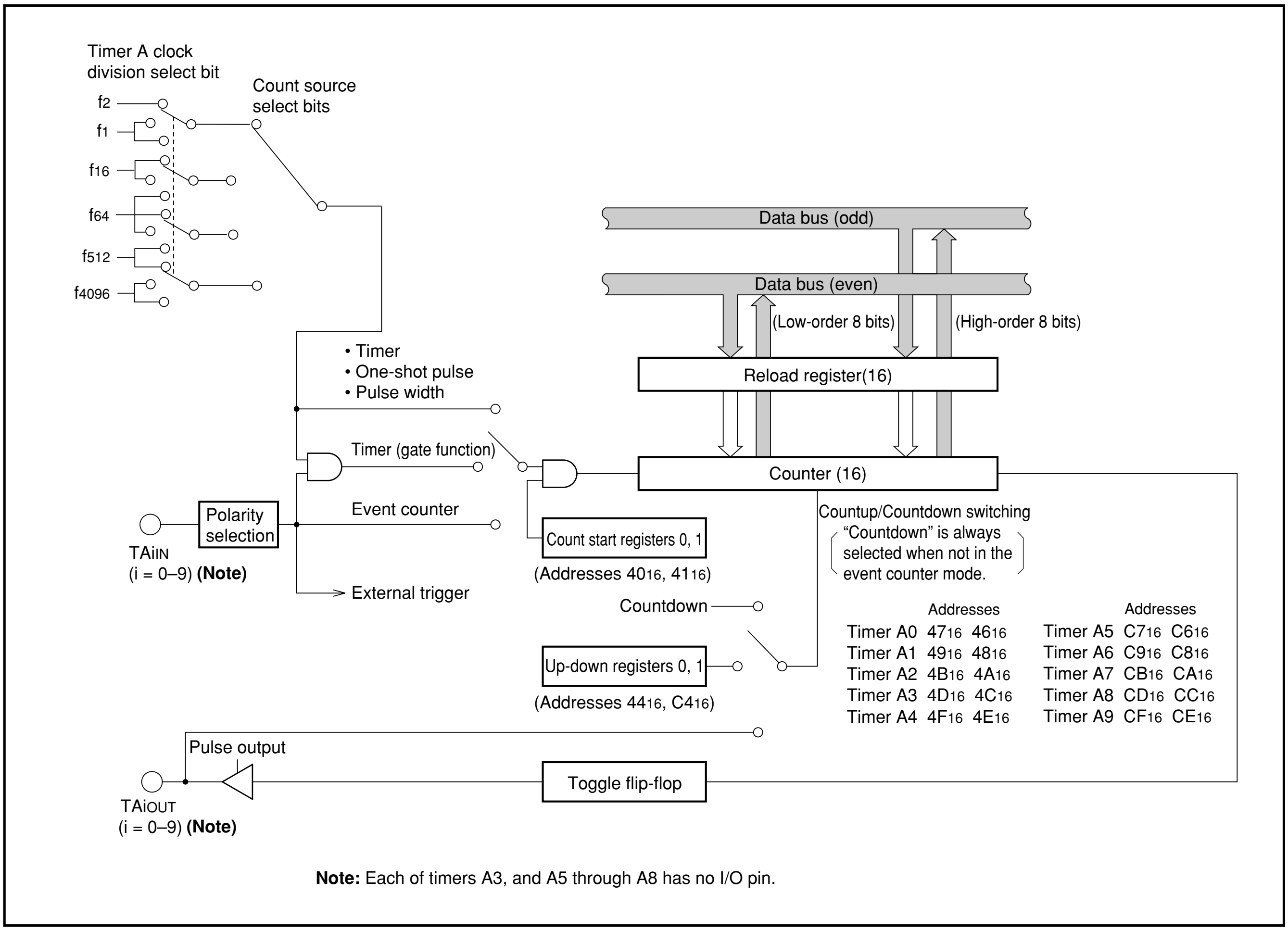
Each of timers A3, and A5 through A8 has no its own I/O pins. Therefore, these timers are available only in the timer mode.

(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_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.


Timer A clock division select bit
Count source select bits
f2
f1
f16
f64
f512
f4096
Data bus (odd)
Data bus (even)
(Low-order 8 bits)
(High-order 8 bits)
• Timer
• One-shot pulse
• Pulse width
Reload register(16)
Timer (gate function)
Counter (16)
Polarity selection
Event counter
TAiIN
(i = 0–9) (Note)
Count start registers 0, 1
(Addresses 40_{16}, 41_{16})
External trigger
Countup/Countdown switching
“Countdown” is always selected when not in the event counter mode.
Countdown
Up-down registers 0, 1
(Addresses 44_{16}, $C4_{16}$)
Addresses
Timer A0 47_{16} 46_{16}
Timer A1 49_{16} 48_{16}
Timer A2 $4B_{16}$ $4A_{16}$
Timer A3 $4D_{16}$ $4C_{16}$
Timer A4 $4F_{16}$ $4E_{16}$
Addresses
Timer A5 $C7_{16}$ $C6_{16}$
Timer A6 $C9_{16}$ $C8_{16}$
Timer A7 CB_{16} CA_{16}
Timer A8 CD_{16} CC_{16}
Timer A9 CF_{16} CE_{16}
Pulse output
Toggle flip-flop
TAiOUT
(i = 0–9) (Note)

Note: Each of timers A3, and A5 through A8 has no I/O pin.

Fig. 18 Block diagram of timer A

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

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

When bit 4 is “1”, counting is performed only while the input signal from the TAiIN pin is “H” or “L” as shown in Figure 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).

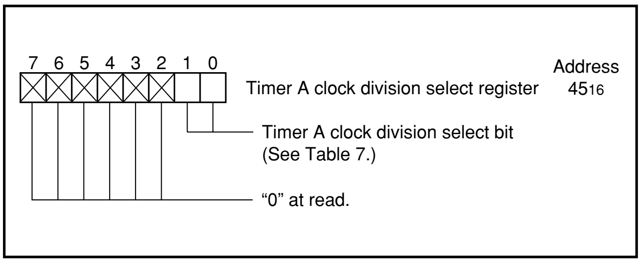
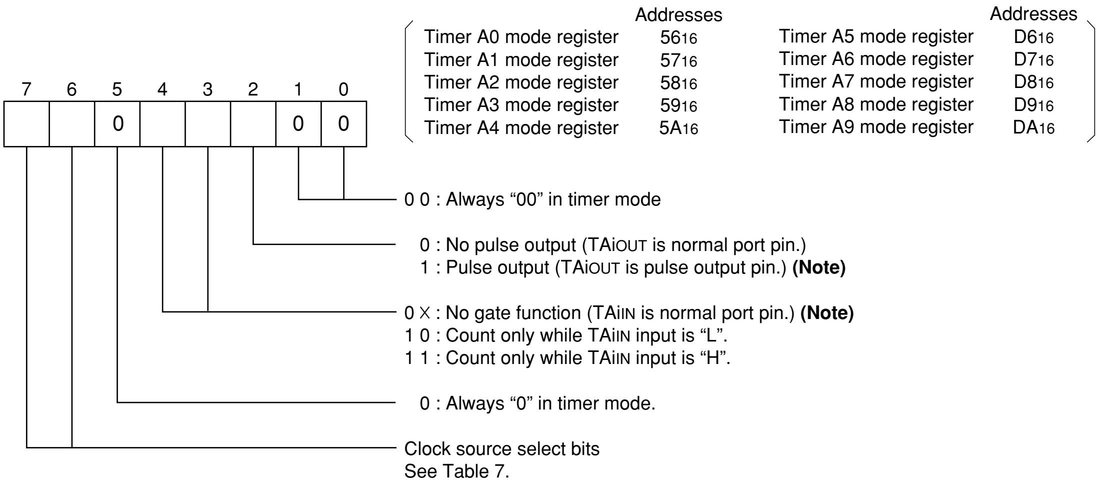

7 6 5 4 3 2 1 0
Address
Timer A clock division select register
45_{16}
Timer A clock division select bit
(See Table 7.)
“0” at read.

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_{16} to $5A_{16}$) (bits 7 and 6 at addresses $D6_{16}$ to DA_{16})	Timer A clock division select bits (bits 1 and 0 at address 45_{16})			
	00	01	10	11
0 0	f_2	f_1	f_1	Do not select.
0 1	f_{16}	f_{16}	f_{64}	
1 0	f_{64}	f_{64}	f_{512}	
1 1	f_{512}	f_{4096}	f_{4096}	

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


7 6 5 4 3 2 1 0
0
0 0
Addresses
Timer A0 mode register 56_{16}
Timer A1 mode register 57_{16}
Timer A2 mode register 58_{16}
Timer A3 mode register 59_{16}
Timer A4 mode register $5A_{16}$
Addresses
Timer A5 mode register $D6_{16}$
Timer A6 mode register $D7_{16}$
Timer A7 mode register $D8_{16}$
Timer A8 mode register $D9_{16}$
Timer A9 mode register DA_{16}
0 0 : Always “00” in timer mode
0 : No pulse output (TAiOUT is normal port pin.)
1 : Pulse output (TAiOUT is pulse output pin.) (Note)
0 X : No gate function (TAiIN is normal port pin.) (Note)
1 0 : Count only while TAiIN input is “L”.
1 1 : Count only while TAiIN input is “H”.
0 : Always “0” in timer mode.
Clock source select bits
See Table 7.

Note: Each of timers A3, and A5 through A8 has no I/O pin. Therfore, fix each of these bits to “0”.

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

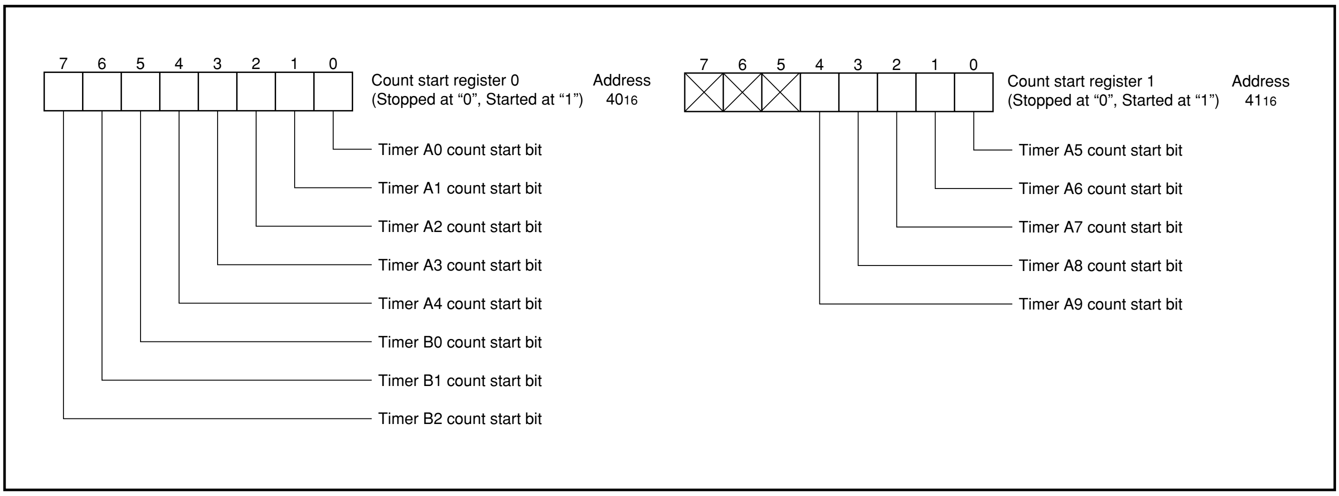

7 6 5 4 3 2 1 0
Count start register 0
(Stopped at "0", Started at "1")
Address
40_{16}
Timer A0 count start bit
Timer A1 count start bit
Timer A2 count start bit
Timer A3 count start bit
Timer A4 count start bit
Timer B0 count start bit
Timer B1 count start bit
Timer B2 count start bit
7 6 5 4 3 2 1 0
Count start register 1
(Stopped at "0", Started at "1")
Address
41_{16}
Timer A5 count start bit
Timer A6 count start bit
Timer A7 count start bit
Timer A8 count start bit
Timer A9 count start bit

Fig. 21 Bit configuration of count start register

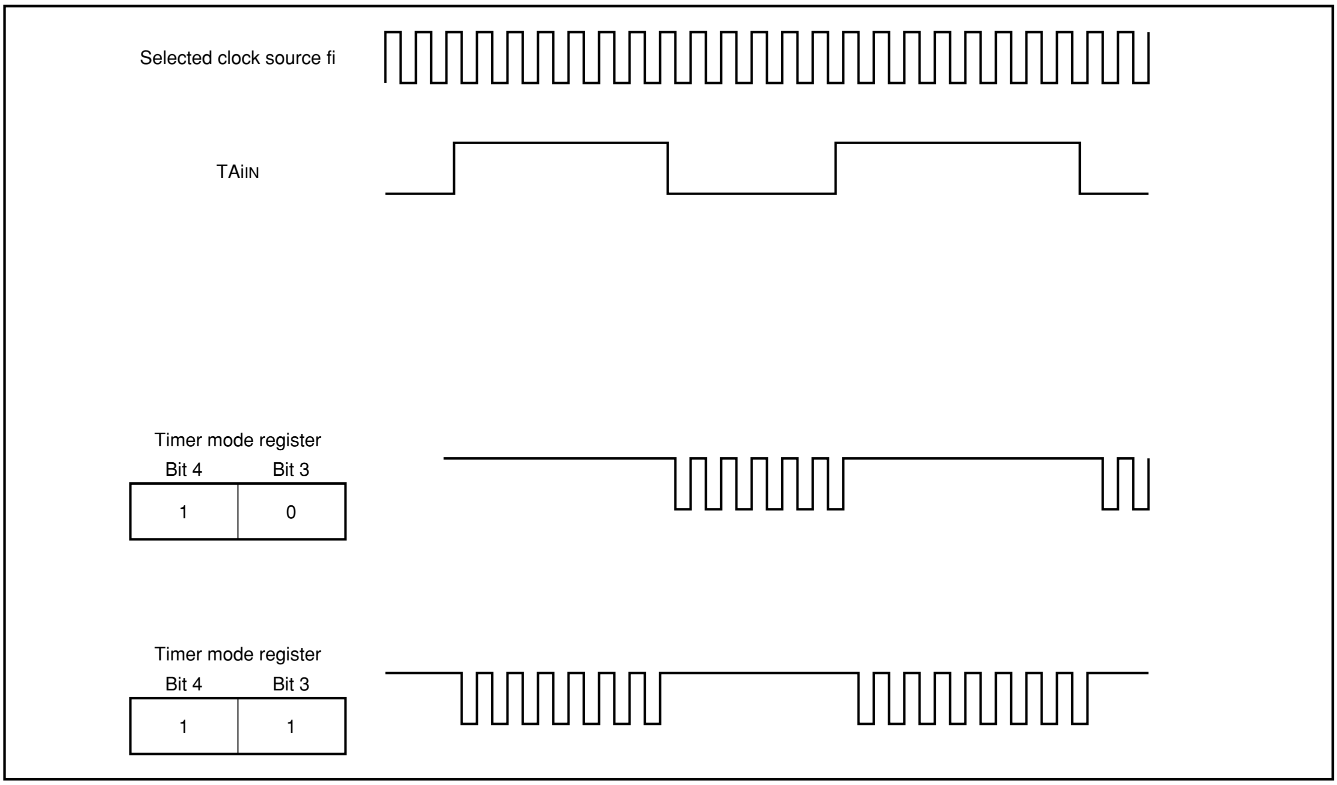

Selected clock source fi
TAiIN
Timer mode register
Bit 4
Bit 3
1
0
Timer mode register
Bit 4
Bit 3
1
1

Fig. 22 Count waveform when gate function is available

(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_{16} (decrement count) or $FFFF_{16}$ (increment count). At the same time, the contents of the reload register is transferred to the counter and the count is continued.

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

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

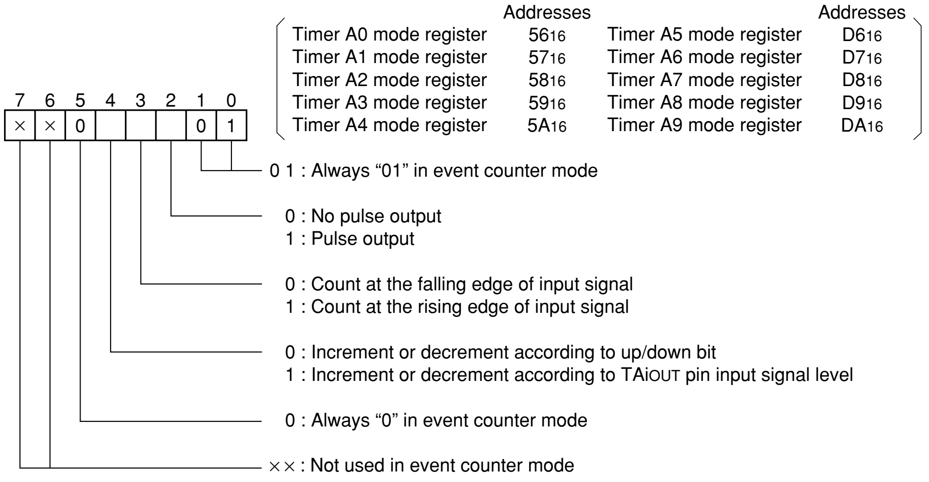
However, if bit 4 is “1” and the TAiOUT pin is used as an output pin, the output from the pin changes the count direction. Therefore, bit 4 must be “0” unless the output from the TAiOUT pin is to be used to select the count direction.

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.


Addresses
Timer A0 mode register 56₁₆
Timer A1 mode register 57₁₆
Timer A2 mode register 58₁₆
Timer A3 mode register 59₁₆
Timer A4 mode register 5A₁₆
Addresses
Timer A5 mode register D6₁₆
Timer A6 mode register D7₁₆
Timer A7 mode register D8₁₆
Timer A8 mode register D9₁₆
Timer A9 mode register DA₁₆
7 6 5 4 3 2 1 0
× × 0 0 1
0 1 : Always “01” in event counter mode
0 : No pulse output
1 : Pulse output
0 : Count at the falling edge of input signal
1 : Count at the rising edge of input signal
0 : Increment or decrement according to up/down bit
1 : Increment or decrement according to TAiOUT pin input signal level
0 : Always “0” in event counter mode
× × : Not used in event counter mode

Note: Each of timers A3, and A5 through A8 has no I/O pin. Therefore, this mode cannot be selected for these timers.

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

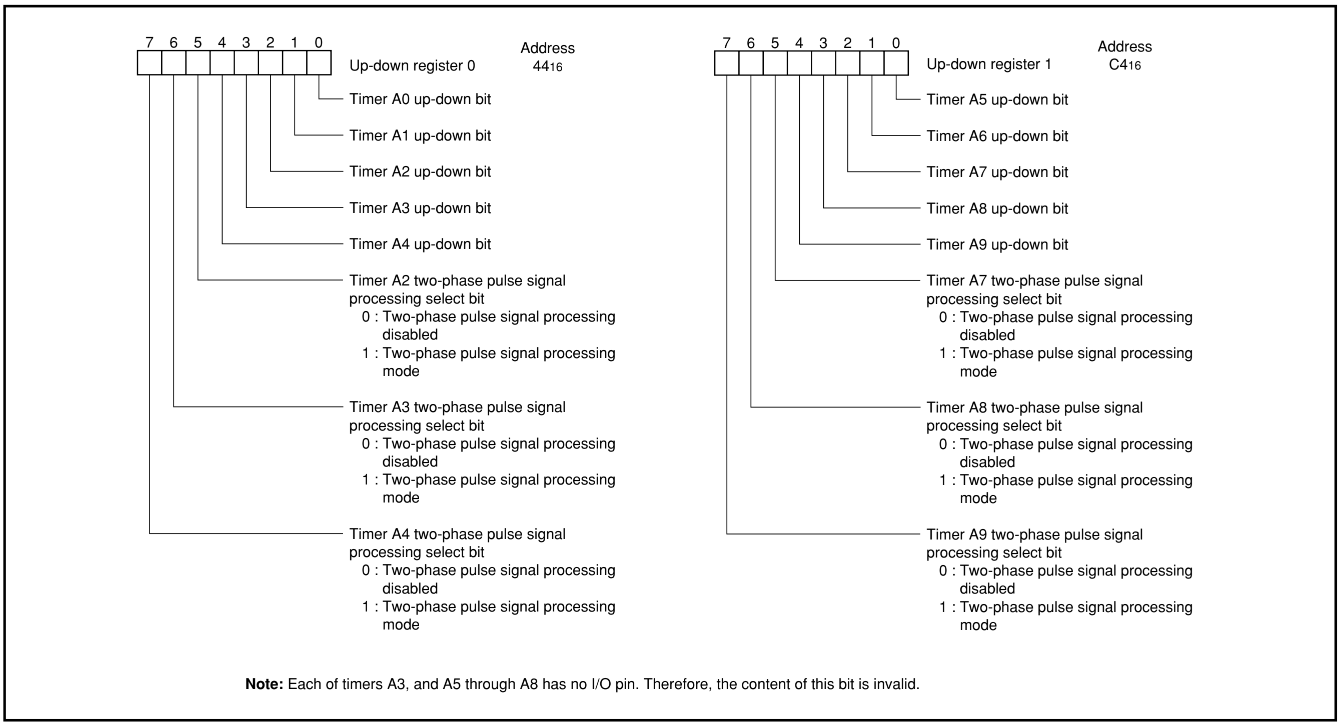

7 6 5 4 3 2 1 0
Up-down register 0
Address
44_{16}
Timer A0 up-down bit
Timer A1 up-down bit
Timer A2 up-down bit
Timer A3 up-down bit
Timer A4 up-down bit
Timer A2 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Timer A3 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Timer A4 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
7 6 5 4 3 2 1 0
Up-down register 1
Address
$C4_{16}$
Timer A5 up-down bit
Timer A6 up-down bit
Timer A7 up-down bit
Timer A8 up-down bit
Timer A9 up-down bit
Timer A7 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Timer A8 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Timer A9 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
1 : Two-phase pulse signal processing mode
Note: Each of timers A3, and A5 through A8 has no I/O pin. Therefore, the content of this bit is invalid.

Fig. 24 Bit configuration of up-down register

When a phase-related pulse with a falling edge input to the TAkOUT pin is input after the level of TAkIN pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the TAkIN pin and TAkOUT pin. When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be "0". Bits 6 and 7 are ignored. (See Figure 27.) Note that bits 5 and 7 of the up-down register 0 (address 44_{16}) 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 $C4_{16}$) 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.

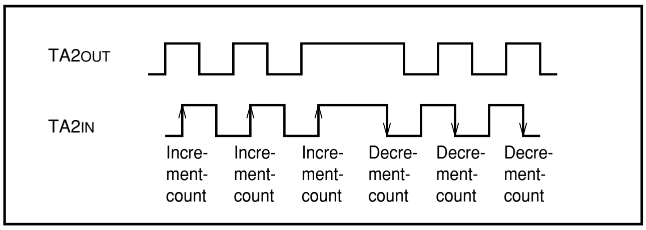

TA2OUT
TA2IN
Increment-count
Increment-count
Increment-count
Decrement-count
Decrement-count
Decrement-count

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

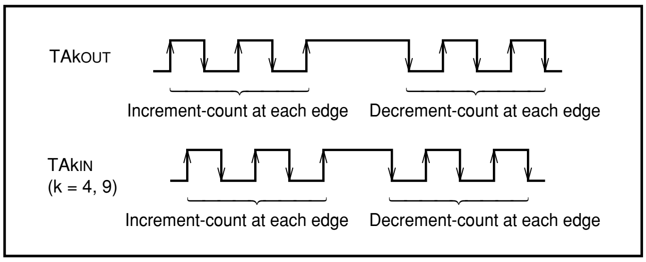

TAkOUT
Increment-count at each edge
Decrement-count at each edge
TAkIN
(k = 4, 9)
Increment-count at each edge
Decrement-count at each edge

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

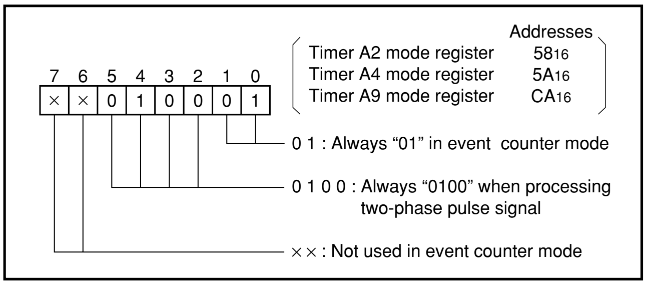

7 6 5 4 3 2 1 0
× × 0 1 0 0 0 1
Addresses
Timer A2 mode register 58_{16}
Timer A4 mode register $5A_{16}$
Timer A9 mode register CA_{16}
0 1 : Always "01" in event counter mode
0 1 0 0 : Always "0100" when processing two-phase pulse signal
× × : Not used in event counter mode

Fig. 27 Bit configuration of timer Aj mode register when performing two-phase pulse signal processing in event counter mode

(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_{16}, the TAiOUT pin goes “H” when a trigger signal is received. The count direction is decrement. When the counter reaches 0001_{16}, the TAiOUT pin goes “L” and count is stopped. The contents of the reload register is transferred to the counter. At the same time, an interrupt request signal is generated and the interrupt request bit in the timer Ai interrupt control register is set. This is repeated each time a trigger signal is received.

The output pulse width is

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

If the count start flag is “0”, TAiOUT goes “L”. Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start bit.

As shown in Figure 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.

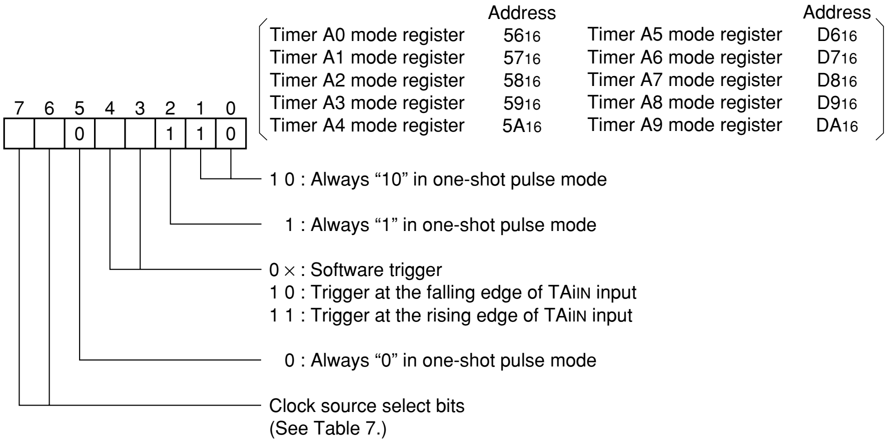
When retriggering, there must be at least one timer count source cycle before a new trigger can be issued.

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.


7 6 5 4 3 2 1 0
0
1 1 0
Address
Timer A0 mode register 56₁₆
Timer A1 mode register 57₁₆
Timer A2 mode register 58₁₆
Timer A3 mode register 59₁₆
Timer A4 mode register 5A₁₆
Address
Timer A5 mode register D6₁₆
Timer A6 mode register D7₁₆
Timer A7 mode register D8₁₆
Timer A8 mode register D9₁₆
Timer A9 mode register DA₁₆
1 0 : Always “10” in one-shot pulse mode
1 : Always “1” in one-shot pulse mode
0 × : Software trigger
1 0 : Trigger at the falling edge of TAiIN input
1 1 : Trigger at the rising edge of TAiIN input
0 : Always “0” in one-shot pulse mode
Clock source select bits
(See Table 7.)

Note: Each of timers A3, and A5 through A8 has no I/O pin. Therefore, this mode cannot be selected for these timers.

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

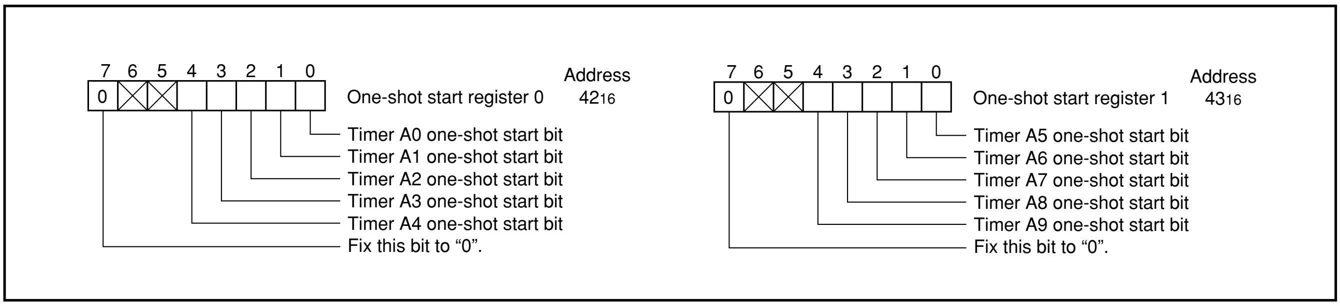

7 6 5 4 3 2 1 0
0
One-shot start register 0
Address
42_{16}
Timer A0 one-shot start bit
Timer A1 one-shot start bit
Timer A2 one-shot start bit
Timer A3 one-shot start bit
Timer A4 one-shot start bit
Fix this bit to “0”.
7 6 5 4 3 2 1 0
0
One-shot start register 1
Address
43_{16}
Timer A5 one-shot start bit
Timer A6 one-shot start bit
Timer A7 one-shot start bit
Timer A8 one-shot start bit
Timer A9 one-shot start bit
Fix this bit to “0”.

Fig. 29 Bit configuration of one-shot start register

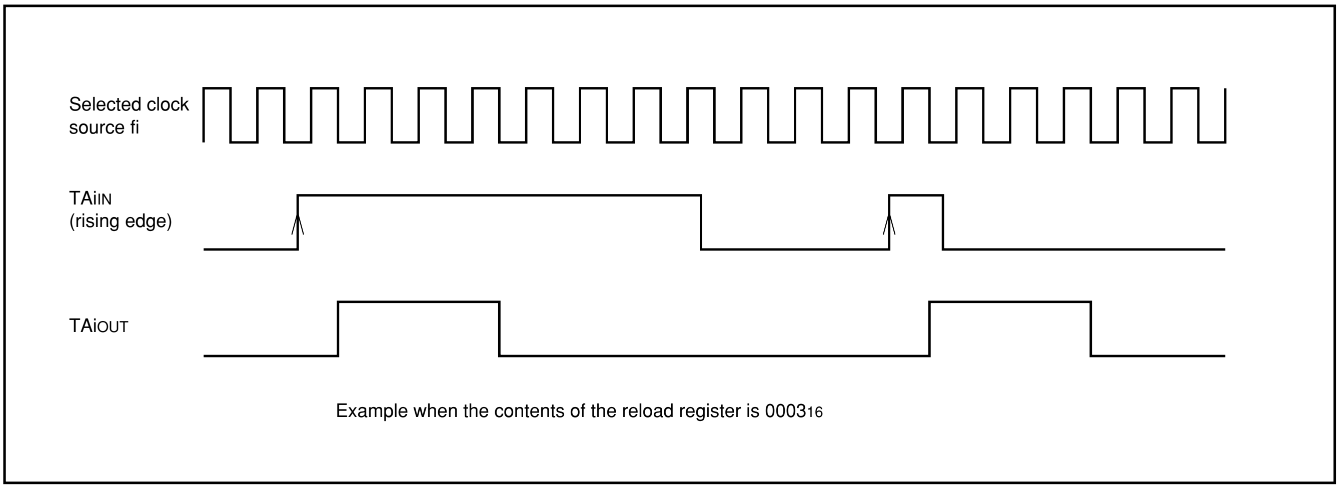

Selected clock source fi
TAiIN (rising edge)
TAiOUT
Example when the contents of the reload register is 0003_{16}

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

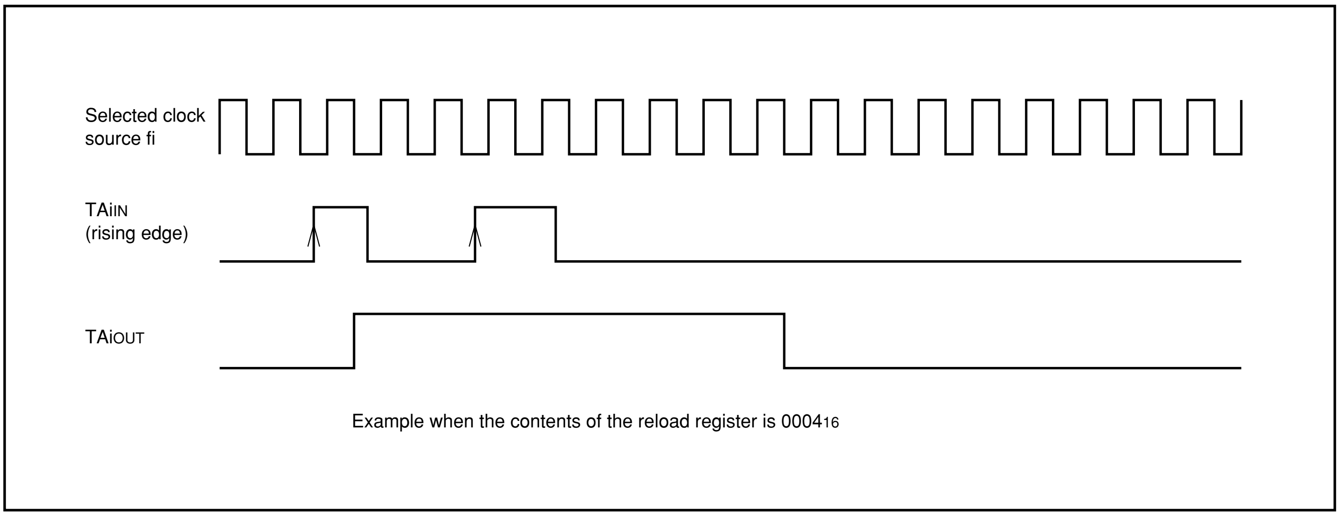

Selected clock source fi
TAiIN (rising edge)
TAiOUT
Example when the contents of the reload register is 0004_{16}

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

(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 000_{16} 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

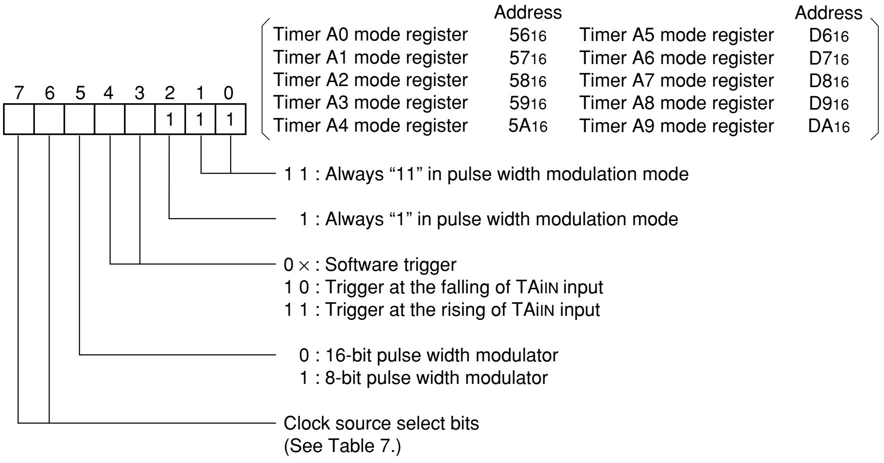
$$\frac{1}{\text{selected clock frequency}} \times (n + 1).$$

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

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



	Address		Address
Timer A0 mode register	56_{16}	Timer A5 mode register	$D6_{16}$
Timer A1 mode register	57_{16}	Timer A6 mode register	$D7_{16}$
Timer A2 mode register	58_{16}	Timer A7 mode register	$D8_{16}$
Timer A3 mode register	59_{16}	Timer A8 mode register	$D9_{16}$
Timer A4 mode register	$5A_{16}$	Timer A9 mode register	DA_{16}

1 1 : Always “11” in pulse width modulation mode

1 : Always “1” in pulse width modulation mode

0 × : Software trigger
1 0 : Trigger at the falling of TAiIN input
1 1 : Trigger at the rising of TAiIN input

0 : 16-bit pulse width modulator
1 : 8-bit pulse width modulator

Clock source select bits
(See Table 7.)

Note: Each of timers A3, and A5 through A8 has no I/O pin. Therefore, this mode cannot be selected for these timers.

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

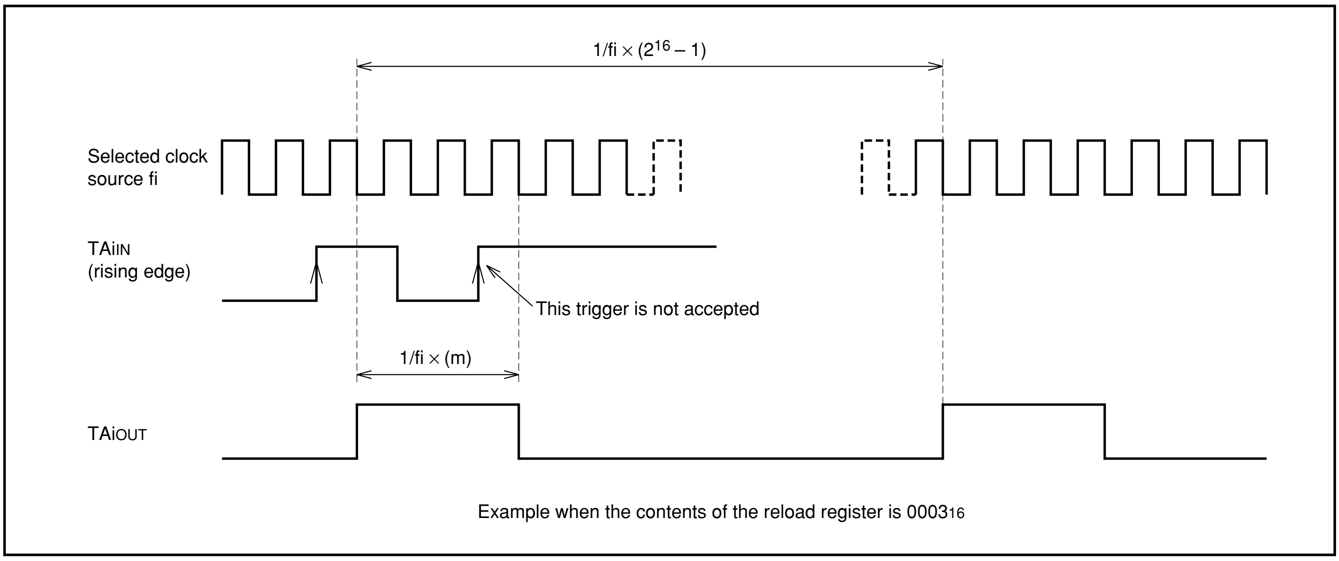

1/fi × (2^16 – 1)
Selected clock source fi
TAiIN (rising edge)
This trigger is not accepted
1/fi × (m)
TAiOUT
Example when the contents of the reload register is 0003₁₆

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

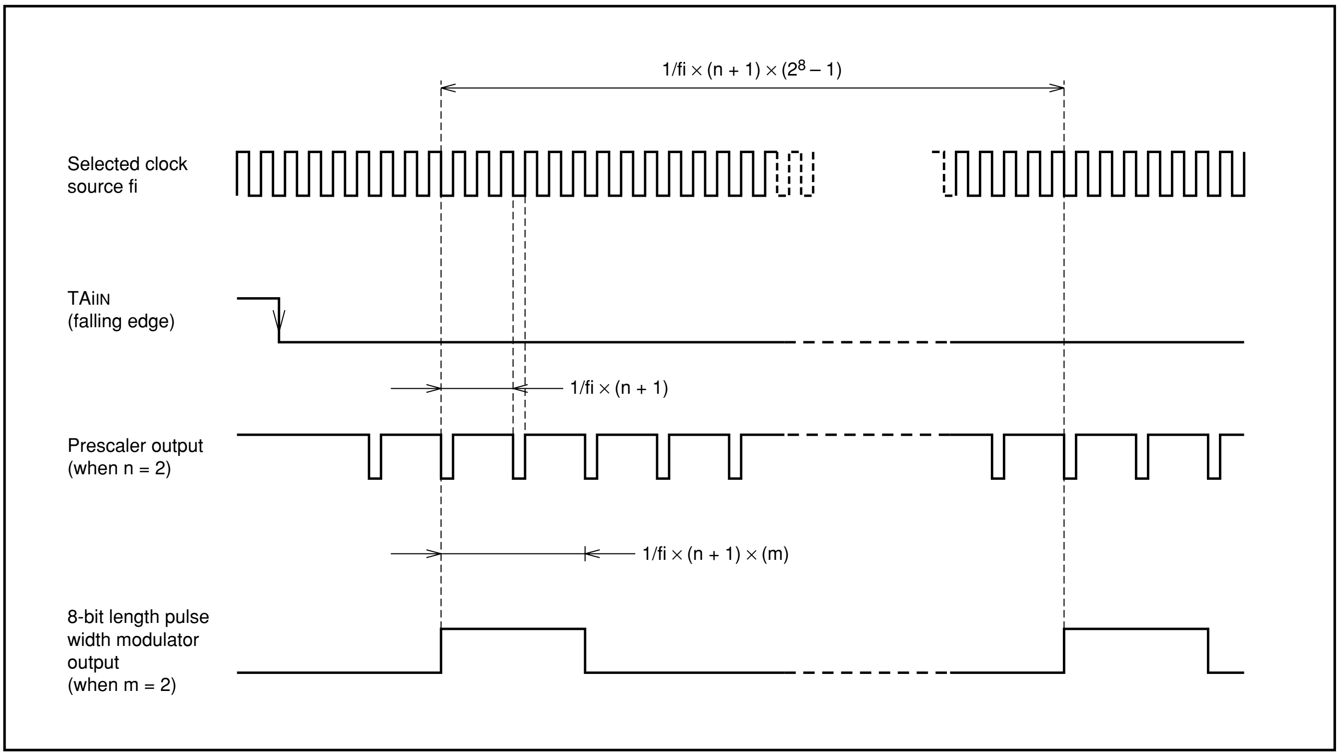

1/fi × (n + 1) × (2^8 – 1)
Selected clock source fi
TAiIN (falling edge)
1/fi × (n + 1)
Prescaler output (when n = 2)
1/fi × (n + 1) × (m)
8-bit length pulse width modulator output (when m = 2)

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

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.

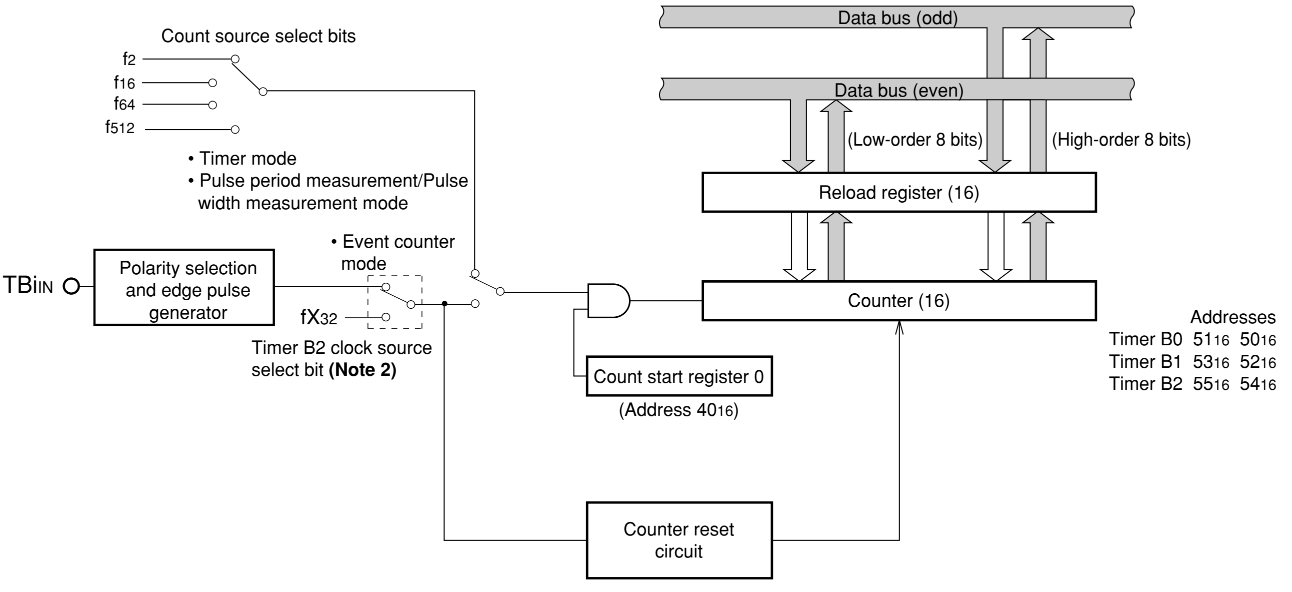
Bits 6 and 7 are used to select the clock source. The counting of the selected clock starts when the count start bit is “1” and stops when “0”.

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_{16}. At the same time, the contents of the reload register is stored in the counter and count is continued.

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

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

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


Count source select bits
f_2
f_{16}
f_{64}
f_{512}
• Timer mode
• Pulse period measurement/Pulse width measurement mode
• Event counter mode
TBi_{IN}
Polarity selection and edge pulse generator
fX_{32}
Timer B2 clock source select bit (Note 2)
Data bus (odd)
Data bus (even)
(Low-order 8 bits)
(High-order 8 bits)
Reload register (16)
Counter (16)
Count start register 0
(Address 40_{16})
Counter reset circuit
Addresses
Timer B0 51_{16} 50_{16}
Timer B1 53_{16} 52_{16}
Timer B2 55_{16} 54_{16}

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

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

Fig. 35 Block diagram of timer B

(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 TBi_{IN} pin is counted when the count start bit is “1” and counting is stopped when it is “0”.

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

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

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

Only for timer B2, when the timer B2 clock source select bit of the particular function select register 1 (bit 6 at address 63_{16}) = “1” in the event counter mode, fX_{32} **(Note)** can be selected. (When this bit is “0”, an input signal from pin $TB2_{IN}$ becomes the count source as described above.) For the bit configuration of the particular function select register 1, refer to the section on the standby function.

Also, the pin position where pin TBi_{IN} is to be allocated can be selected by the pin TBi_{IN} select bit (bit 0, 1, or 2 at address AE_{16}; the port P2 pin function control register).

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

(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 TBi_{IN} pin to the next fall or at the rise of the input signal to the next rise; the result is stored in the reload register. In this case, the reload register acts as a buffer register.

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

In the case of counting from the fall of the input signal to the next fall, counting is performed as follows. As shown in Figure 39, when the fall of the input signal from TBi_{IN} pin is detected, the contents of the counter is transferred to the reload register. Next, the counter is cleared and count is started from the next clock. When the fall of the next input signal is detected, the contents of the counter is transferred to the reload register once more, the counter is cleared, and the count is started. The period from the fall of the input signal to the next fall is measured in this way.

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

When bit 3 is “1”, the pulse width measurement mode is selected. Pulse width measurement mode is the same as the pulse period measurement mode except that the clock is counted from the fall of the TBi_{IN} pin input signal to the next rise or from the rise of the input signal to the next fall as shown in Figure 40.

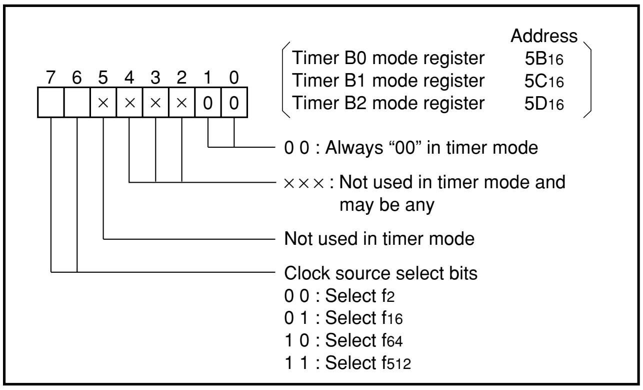

7 6 5 4 3 2 1 0
× × × × 0 0
Address
Timer B0 mode register 5B16
Timer B1 mode register 5C16
Timer B2 mode register 5D16
0 0 : Always “00” in timer mode
× × × : Not used in timer mode and may be any
Not used in timer mode
Clock source select bits
0 0 : Select f2
0 1 : Select f16
1 0 : Select f64
1 1 :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

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

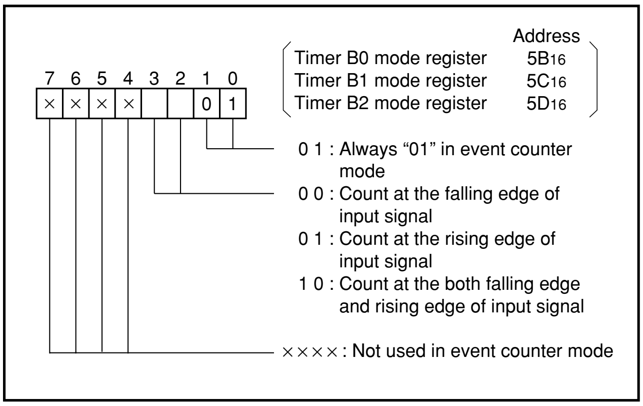

7 6 5 4 3 2 1 0
× × × × 0 1
Address
Timer B0 mode register 5B16
Timer B1 mode register 5C16
Timer B2 mode register 5D16
0 1 : Always “01” in event counter mode
0 0 : Count at the falling edge of input signal
0 1 : Count at the rising edge of input signal
1 0 : Count at the both falling edge and rising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

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

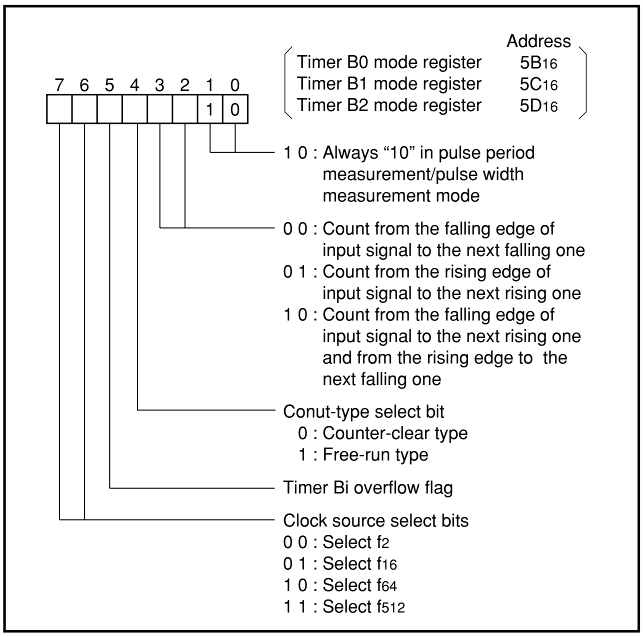

7 6 5 4 3 2 1 0
1 0
Address
Timer B0 mode register 5B16
Timer B1 mode register 5C16
Timer B2 mode register 5D16
1 0 : Always “10” in pulse period measurement/pulse width measurement mode
0 0 : Count from the falling edge of input signal to the next falling one
0 1 : Count from the rising edge of input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

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

When timer Bi is read, the contents of the reload register is read. Note that in this mode, the interval between the fall of the TBiIN pin input signal to the next rise or from the rise to the next fall must be at least two cycles of the timer count source.
Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}, which indicates that a pulse width or pulse period is longer than that which can be measured by a 16-bit length.
This flag is cleared by writing data to the corresponding timer Bi mode register. This flag is set to "1"at reset.
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_{16}", and counting will continue. However, when a valid edge is input, an interrupt-requesting signal will be generated.

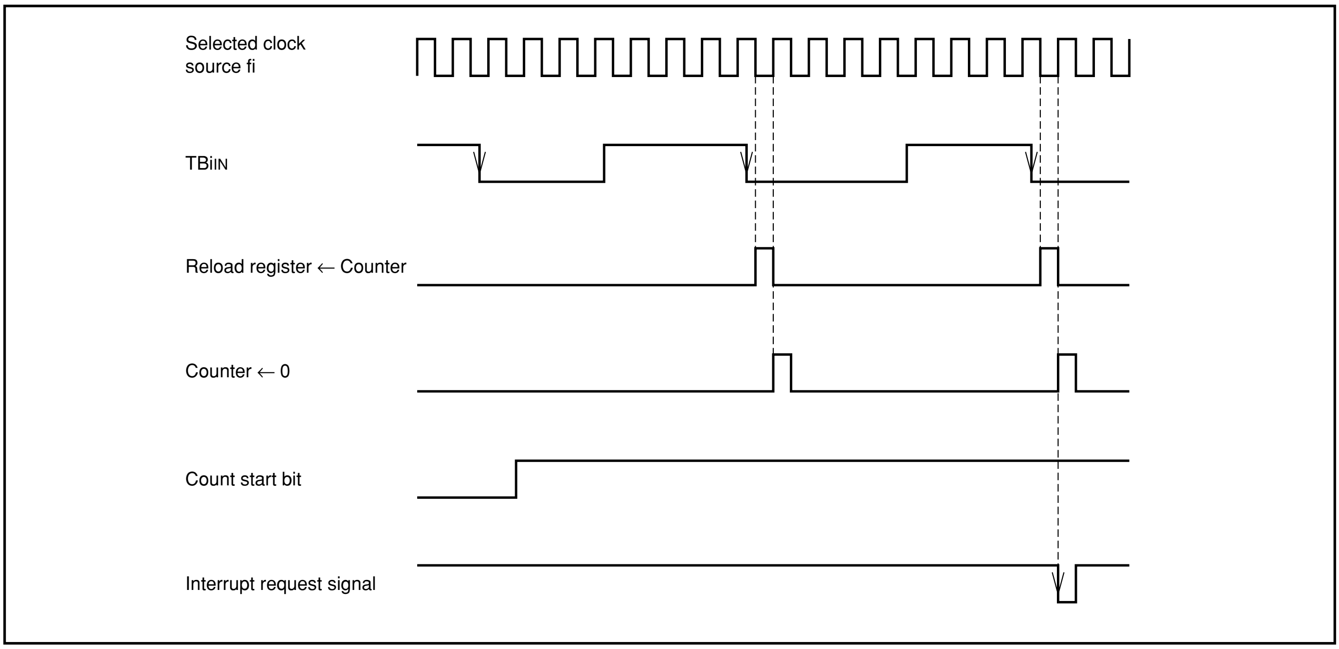

Selected clock source fi
TBiIN
Reload register ← Counter
Counter ← 0
Count start bit
Interrupt request signal

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

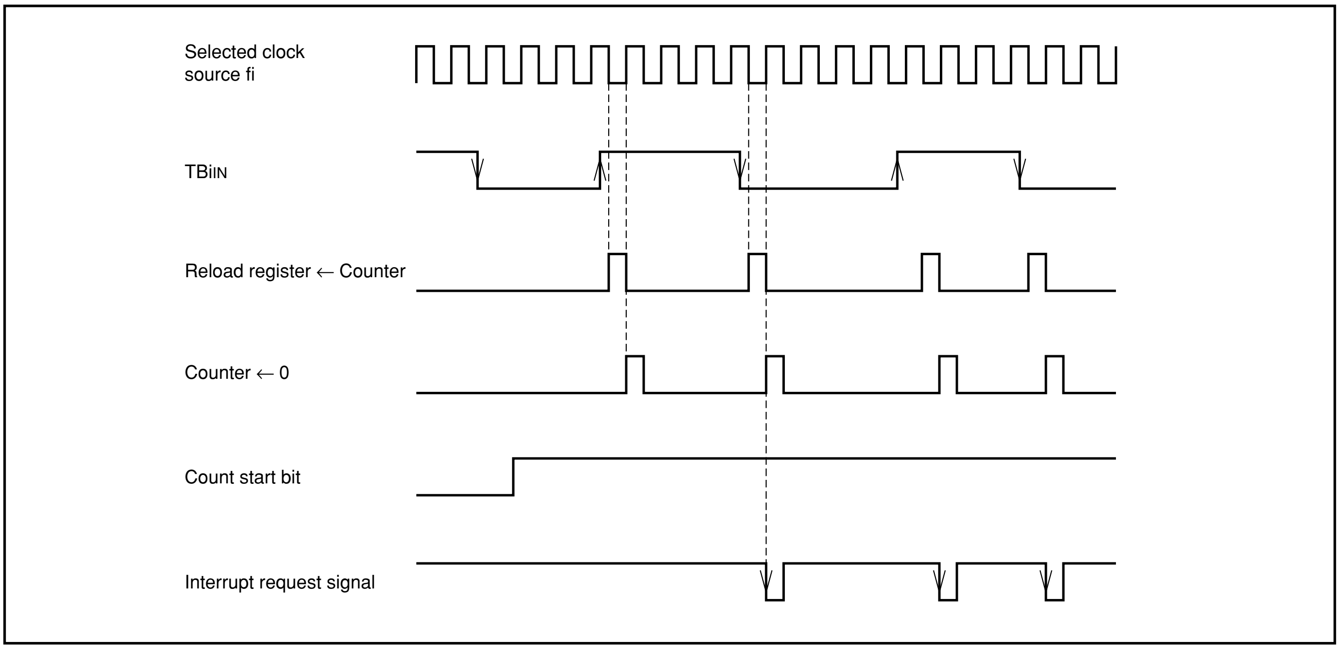

Selected clock source fi
TBiIN
Reload register ← Counter
Counter ← 0
Count start bit
Interrupt request signal

Fig. 40 Pulse width measurement mode operation

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 bellow.

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 A6$_{16}$, Figure 41) to "100$_2$".

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 ($\overline{U}$, $\overline{V}$, $\overline{W}$ phases), from the respective ports with "L" level active.

Timer A2 controls U and $\overline{U}$ phases; timer A1 does V and $\overline{V}$ phases and timer A0 does W and $\overline{W}$ 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 ($\overline{U}$, $\overline{V}$, $\overline{W}$ 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 A6$_{16}$) 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.

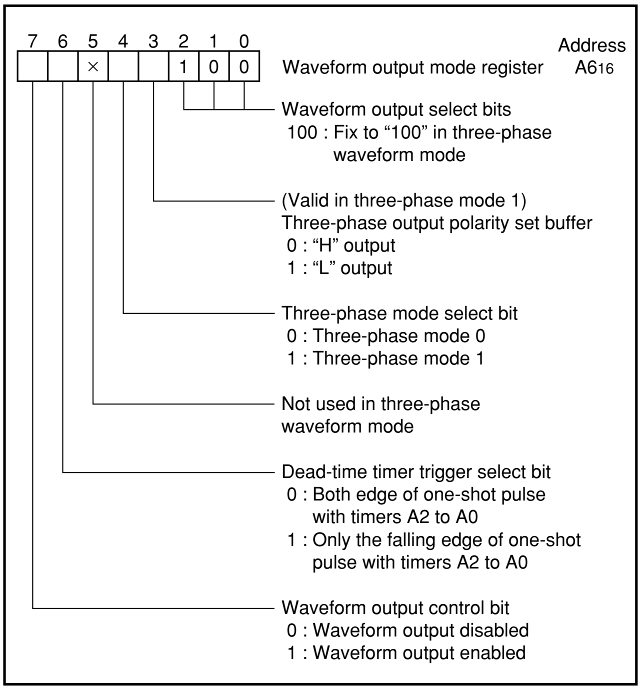

7 6 5 4 3 2 1 0
×
1 0 0
Address
Waveform output mode register A6₁₆
Waveform output select bits
100 : Fix to "100" in three-phase waveform mode
(Valid in three-phase mode 1)
Three-phase output polarity set buffer
0 : "H" output
1 : "L" output
Three-phase mode select bit
0 : Three-phase mode 0
1 : Three-phase mode 1
Not used in three-phase waveform mode
Dead-time timer trigger select bit
0 : Both edge of one-shot pulse with timers A2 to A0
1 : Only the falling edge of one-shot pulse with timers A2 to A0
Waveform output control bit
0 : Waveform output disabled
1 : Waveform output enabled

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

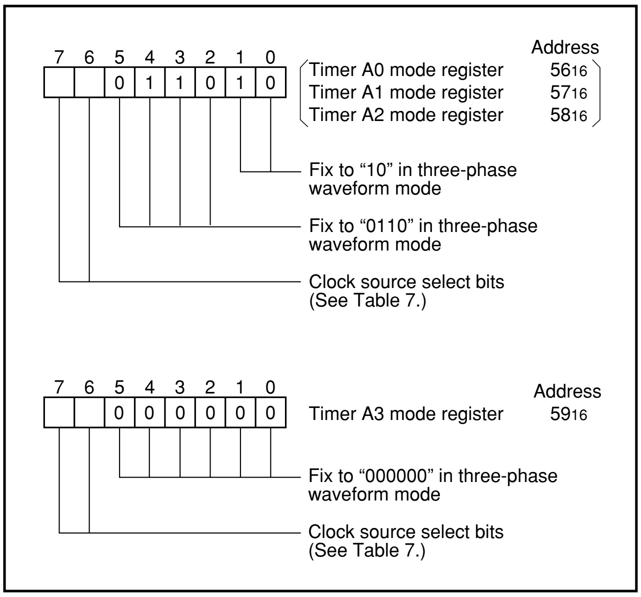

7 6 5 4 3 2 1 0
0 1 1 0 1 0
Address
Timer A0 mode register 56₁₆
Timer A1 mode register 57₁₆
Timer A2 mode register 58₁₆
Fix to "10" in three-phase waveform mode
Fix to "0110" in three-phase waveform mode
Clock source select bits (See Table 7.)
7 6 5 4 3 2 1 0
0 0 0 0 0 0
Address
Timer A3 mode register 59₁₆
Fix to "000000" in three-phase waveform mode
Clock source select bits (See Table 7.)

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

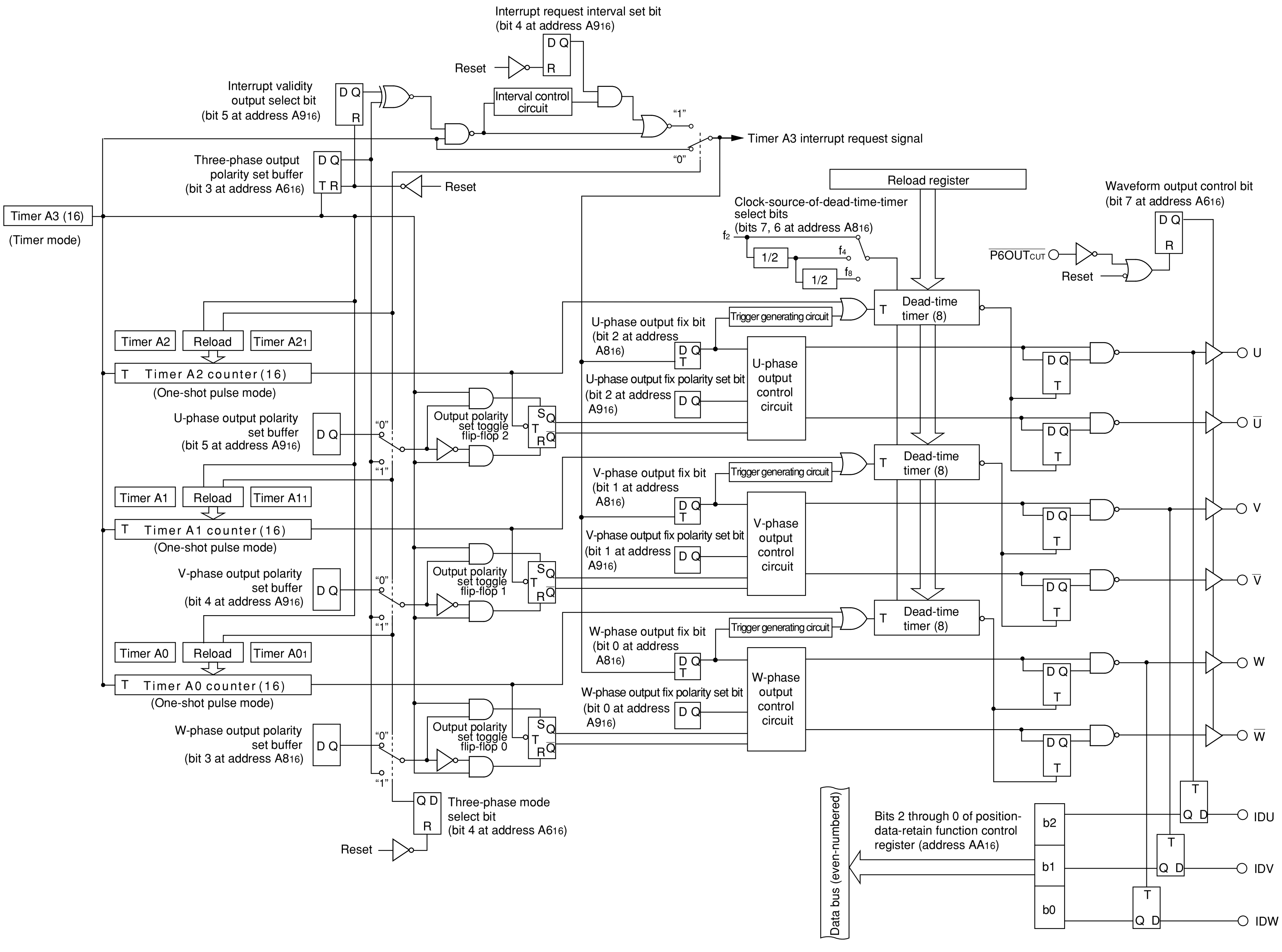

Interrupt request interval set bit
(bit 4 at address A9₁₆)
Reset
D Q
R
Interrupt validity
output select bit
(bit 5 at address A916)
Interval control circuit
"1"
"0"
Timer A3 interrupt request signal
Three-phase output
polarity set buffer
(bit 3 at address A616)
T R
Reset
Timer A3 (16)
(Timer mode)
Reload register
Clock-source-of-dead-time-timer
select bits
(bits 7, 6 at address A816)
f2
1/2
f4
f8
Waveform output control bit
(bit 7 at address A616)
P6OUTCUT
Reset
Dead-time timer (8)
Trigger generating circuit
U-phase output fix bit
(bit 2 at address A816)
U-phase output fix polarity set bit
(bit 2 at address A916)
U-phase output control circuit
Timer A2
Reload
Timer A21
T Timer A2 counter (16)
(One-shot pulse mode)
U-phase output polarity
set buffer
(bit 5 at address A916)
Output polarity set toggle flip-flop 2
S Q
T
R Q
V-phase output fix bit
(bit 1 at address A816)
V-phase output fix polarity set bit
(bit 1 at address A916)
V-phase output control circuit
Timer A1
Timer A11
T Timer A1 counter (16)
V-phase output polarity
set buffer
(bit 4 at address A916)
Output polarity set toggle flip-flop 1
W-phase output fix bit
(bit 0 at address A816)
W-phase output fix polarity set bit
(bit 0 at address A916)
W-phase output control circuit
Timer A0
Timer A01
T Timer A0 counter (16)
W-phase output polarity
set buffer
(bit 3 at address A816)
Output polarity set toggle flip-flop 0
Q D
Three-phase mode
select bit
(bit 4 at address A616)
U
U̅
V
V̅
W
W̅
IDU
IDV
IDW
b2
b1
b0
Bits 2 through 0 of position-
data-retain function control
register (address AA16)
Data bus (even-numbered)

Fig. 43 Block diagram in three-phase waveform mode

When writing data to the dead-time timer (address $A7_{16}$), 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 $A8_{16}$). 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 “00_{16}”, and waits next trigger.

In the three-phase waveform mode, setting bit 7 of the waveform output mode register (address $A6_{16}$) to “1” makes positive waveforms (U, V, W phases) and their negative waveforms ($\overline{U}$, $\overline{V}$, $\overline{W}$ 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 $\overline{P6OUT_{CUT}}$, by reset, or by executing instructions.

Additionally, setting bits 2 through 0 of the three-phase output data register 0 (address $A8_{16}$) 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 $A9_{16}$). Clearing these bits to “0” makes the corresponding waveform outputs fixed to “H”; setting these bits to “1” makes the outputs fixed to “L”.

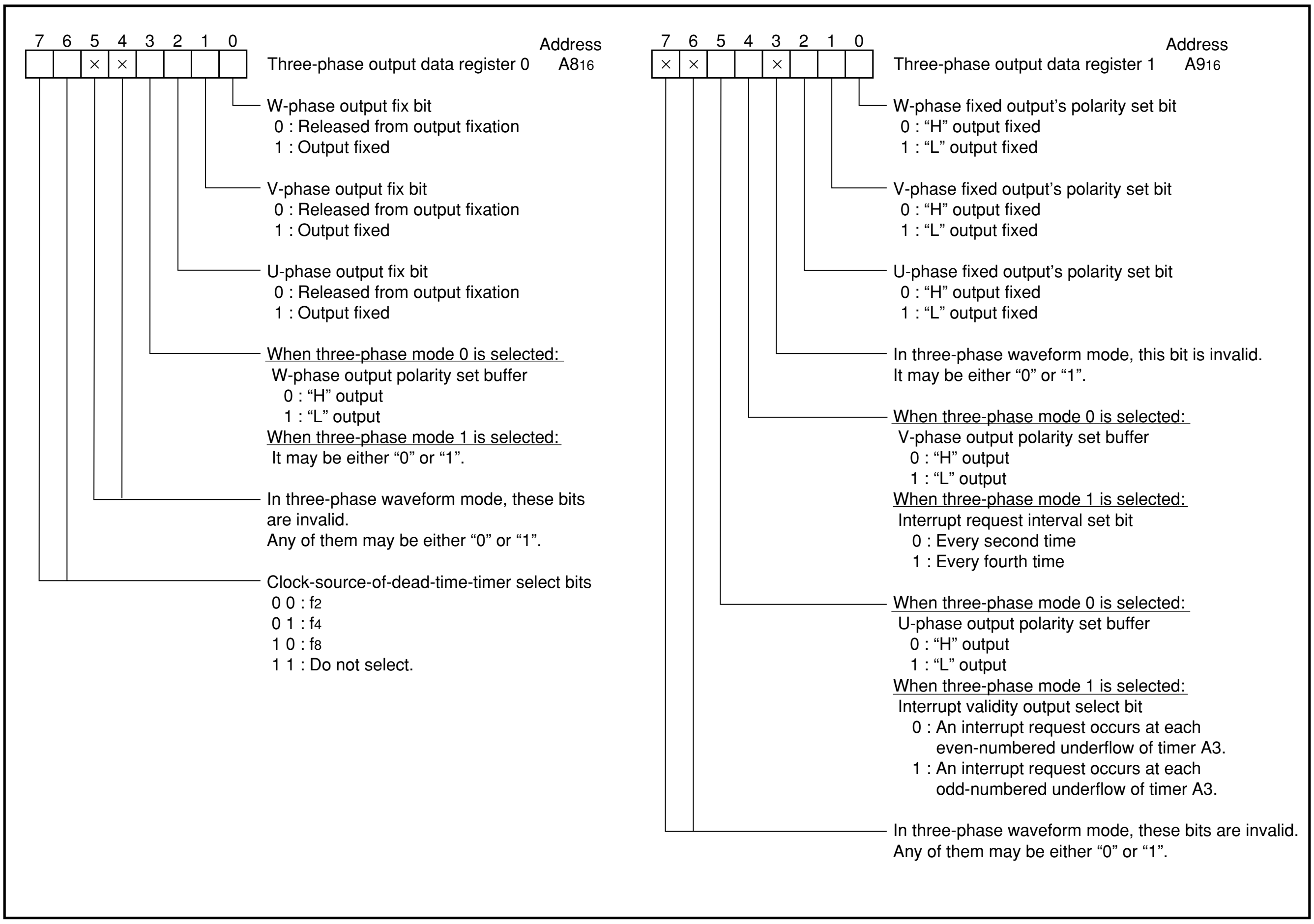

7 6 5 4 3 2 1 0
×
×
Three-phase output data register 0
Address
$A8_{16}$
W-phase output fix bit
0 : Released from output fixation
1 : Output fixed
V-phase output fix bit
0 : Released from output fixation
1 : Output fixed
U-phase output fix bit
0 : Released from output fixation
1 : Output fixed
When three-phase mode 0 is selected:
W-phase output polarity set buffer
0 : “H” output
1 : “L” output
When three-phase mode 1 is selected:
It may be either “0” or “1”.
In three-phase waveform mode, these bits are invalid.
Any of them may be either “0” or “1”.
Clock-source-of-dead-time-timer select bits
0 0 : f_2
0 1 : f_4
1 0 : f_8
1 1 : Do not select.
7 6 5 4 3 2 1 0
×
×
×
Three-phase output data register 1
Address
$A9_{16}$
W-phase fixed output’s polarity set bit
0 : “H” output fixed
1 : “L” output fixed
V-phase fixed output’s polarity set bit
0 : “H” output fixed
1 : “L” output fixed
U-phase fixed output’s polarity set bit
0 : “H” output fixed
1 : “L” output fixed
In three-phase waveform mode, this bit is invalid.
It may be either “0” or “1”.
When three-phase mode 0 is selected:
V-phase output polarity set buffer
0 : “H” output
1 : “L” output
When three-phase mode 1 is selected:
Interrupt request interval set bit
0 : Every second time
1 : Every fourth time
When three-phase mode 0 is selected:
U-phase output polarity set buffer
0 : “H” output
1 : “L” output
When three-phase mode 1 is selected:
Interrupt validity output select bit
0 : An interrupt request occurs at each even-numbered underflow of timer A3.
1 : An interrupt request occurs at each odd-numbered underflow of timer A3.
In three-phase waveform mode, these bits are invalid.
Any of them may be either “0” or “1”.

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

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 A6$_{16}$) 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 0000$_{16}$, 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 A9$_{16}$) and actuating the timer A3. When the counter of timer A3 becomes 0000$_{16}$, 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 $\overline{\text{U}}$ 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 A9$_{16}$) before the counter of timer A3 becomes 0000$_{16}$.

After that, when the counter of timer A3 becomes 0000$_{16}$, 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 $\overline{\text{U}}$-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 $\overline{\text{U}}$-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 $\overline{\text{V}}$-, $\overline{\text{W}}$-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.

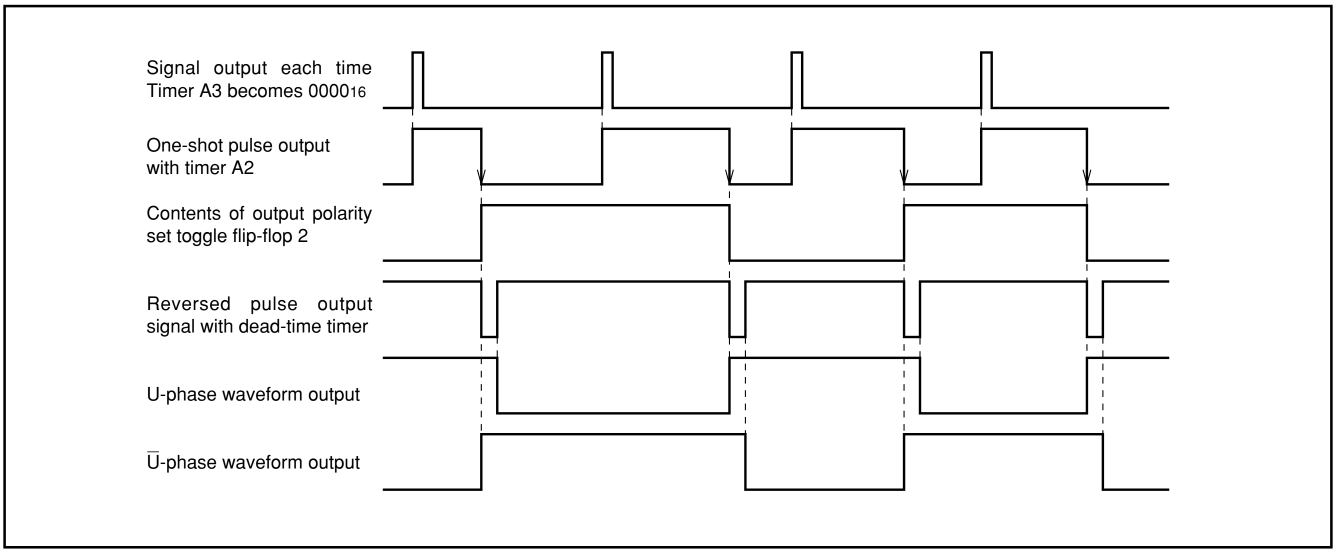

Signal output each time Timer A3 becomes 0000$_{16}$
One-shot pulse output with timer A2
Contents of output polarity set toggle flip-flop 2
Reversed pulse output signal with dead-time timer
U-phase waveform output
$\overline{\text{U}}$-phase waveform output

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

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 $A6_{16}$) 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_{16}.

The interrupt request normally occurs when the counter of timer A3 becomes 0000_{16}. 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 $A9_{16}$) 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 $A9_{16}$) selects that.

When the timer A3’s counter contents become 0000_{16}, 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 $A6_{16}$). Clear the interrupt request interval set bit (bit 4 at address $A9_{16}$) 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 $A9_{16}$) 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_{16}, 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 $\overline{\text{U}}$-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_{16}, 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, because 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 $\overline{\text{U}}$-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 $\overline{\text{U}}$-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, $A2_1$, or A3.

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

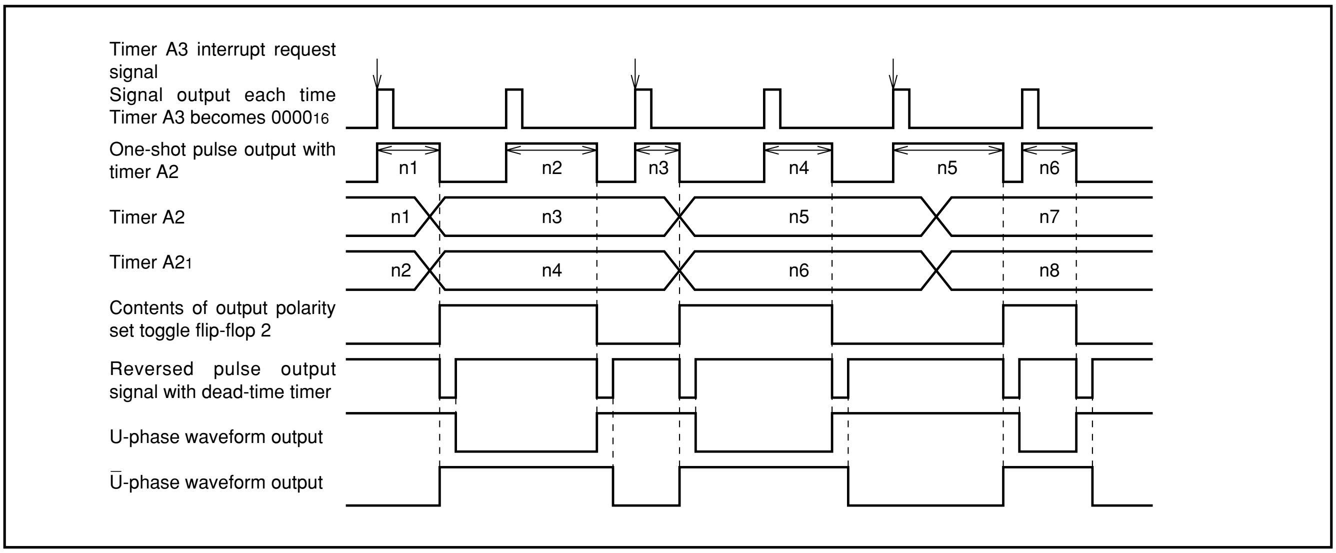

Timer A3 interrupt request signal
Signal output each time Timer A3 becomes 0000₁₆
One-shot pulse output with timer A2
n1
n2
n3
n4
n5
n6
Timer A2
n1
n3
n5
n7
Timer A2₁
n2
n4
n6
n8
Contents of output polarity set toggle flip-flop 2
Reversed pulse output signal with dead-time timer
U-phase waveform output
Ū-phase waveform output

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 AA_{16}).

Retain data can be read out by bits 2 through 0 of the position-data-retain function control register (address AA_{16}).

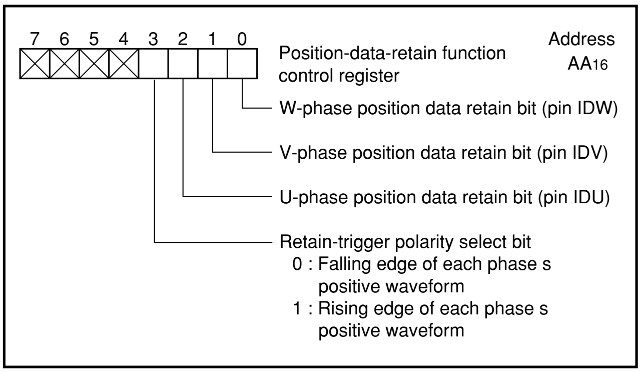

7 6 5 4 3 2 1 0
Position-data-retain function control register
Address AA₁₆
W-phase position data retain bit (pin IDW)
V-phase position data retain bit (pin IDV)
U-phase position data retain bit (pin IDU)
Retain-trigger polarity select bit
0 : Falling edge of each phase s positive waveform
1 : Rising edge of each phase s positive waveform

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

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 $A6_{16}$) 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 $A6_{16}$); 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 RTP_{TRG0}. 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 $A8_{16}$). 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 $A9_{16}$). 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.

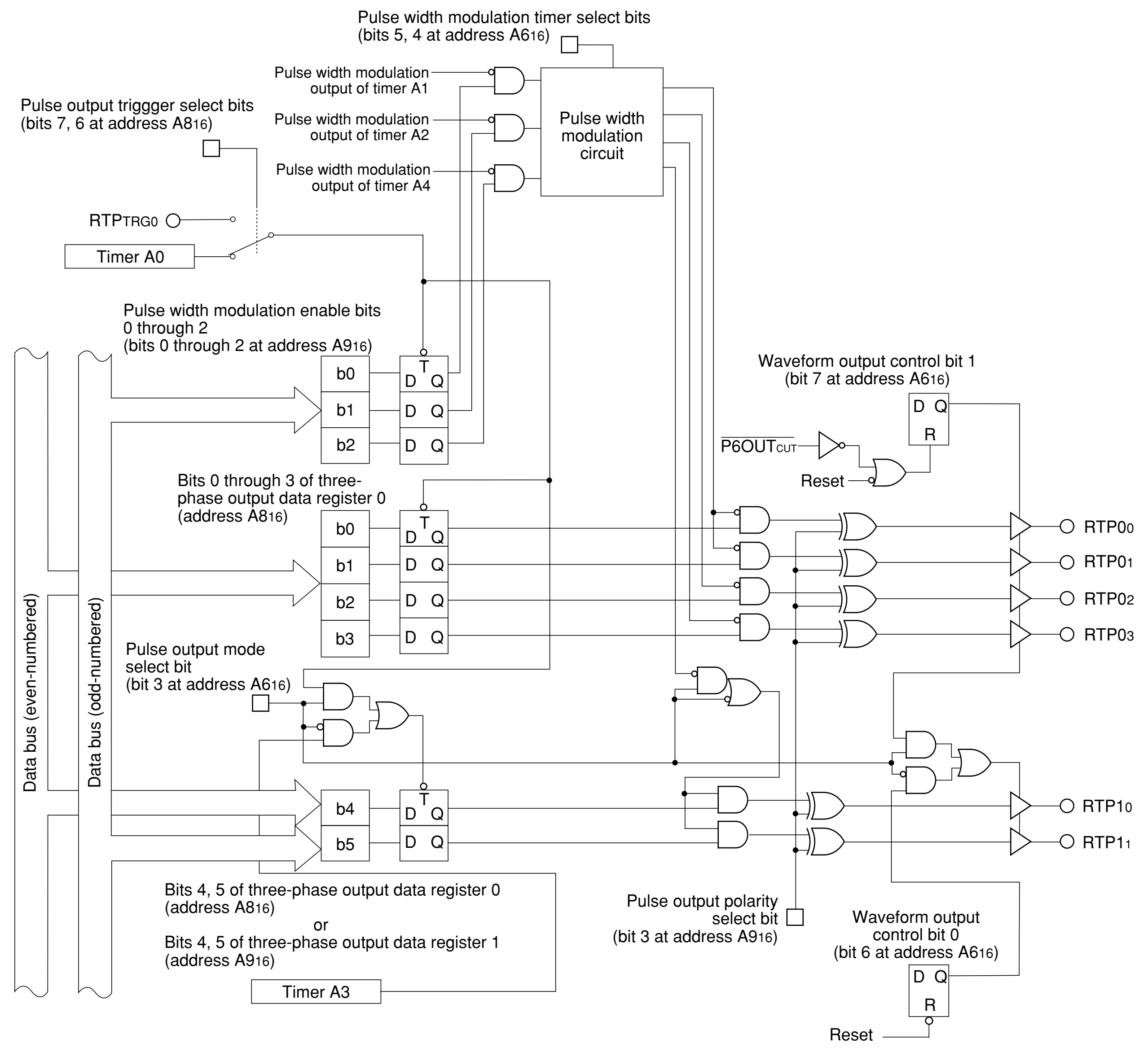
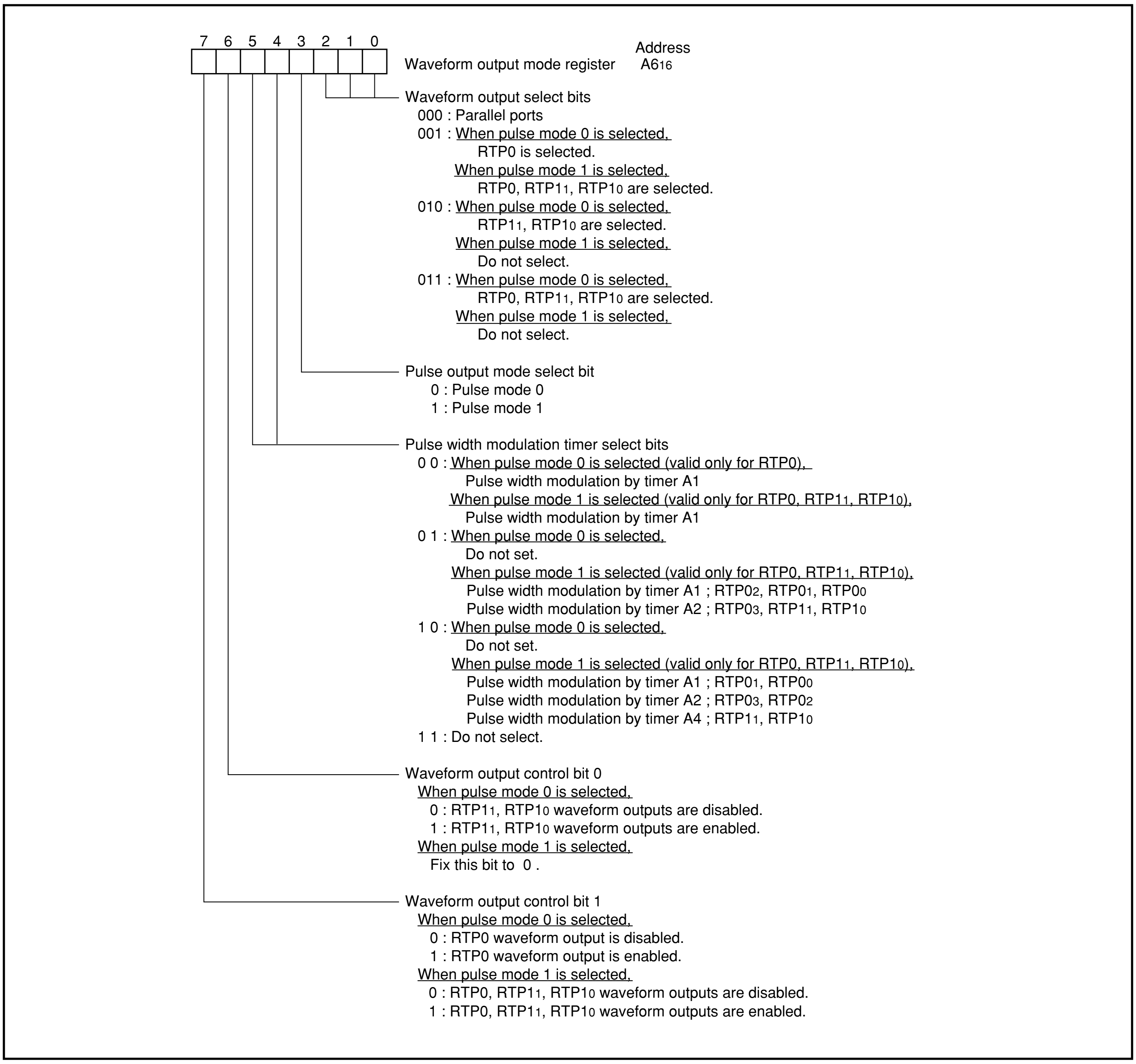

Pulse width modulation timer select bits
(bits 5, 4 at address A616)
Pulse width modulation output of timer A1
Pulse width modulation output of timer A2
Pulse width modulation output of timer A4
Pulse width modulation circuit
Pulse output triggger select bits
(bits 7, 6 at address A816)
RTPTRG0
Timer A0
Pulse width modulation enable bits
0 through 2
(bits 0 through 2 at address A916)
b0
b1
b2
T
D Q
D Q
D Q
Waveform output control bit 1
(bit 7 at address A616)
D Q
R
P6OUTCUT
Reset
Bits 0 through 3 of three-phase output data register 0
(address A816)
b0
b1
b2
b3
RTP00
RTP01
RTP02
RTP03
Data bus (even-numbered)
Data bus (odd-numbered)
Pulse output mode select bit
(bit 3 at address A616)
b4
b5
RTP10
RTP11
Bits 4, 5 of three-phase output data register 0
(address A816)
or
Bits 4, 5 of three-phase output data register 1
(address A916)
Timer A3
Pulse output polarity select bit
(bit 3 at address A916)
Waveform output control bit 0
(bit 6 at address A616)
D Q
R
Reset

Fig. 48 Block diagram in pulse output port mode



Waveform output mode register $A6_{16}$

Waveform output select bits
000 : Parallel ports
001 : When pulse mode 0 is selected,
RTP0 is selected.
When pulse mode 1 is selected,
RTP0, $RTP1_1$, $RTP1_0$ are selected.
010 : When pulse mode 0 is selected,
$RTP1_1$, $RTP1_0$ are selected.
When pulse mode 1 is selected,
Do not select.
011 : When pulse mode 0 is selected,
RTP0, $RTP1_1$, $RTP1_0$ are selected.
When pulse mode 1 is selected,
Do not select.

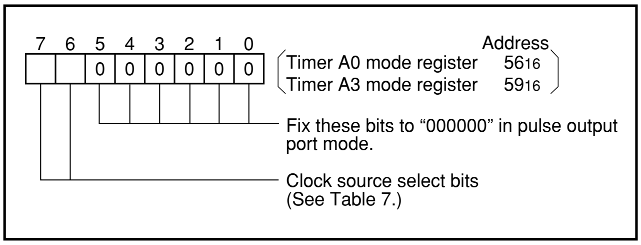
Pulse output mode select bit
0 : Pulse mode 0
1 : Pulse mode 1

Pulse width modulation timer select bits
0 0 : When pulse mode 0 is selected (valid only for RTP0),
Pulse width modulation by timer A1
When pulse mode 1 is selected (valid only for RTP0, $RTP1_1$, $RTP1_0$),
Pulse width modulation by timer A1
0 1 : When pulse mode 0 is selected,
Do not set.
When pulse mode 1 is selected (valid only for RTP0, $RTP1_1$, $RTP1_0$),
Pulse width modulation by timer A1 ; $RTP0_2$, $RTP0_1$, $RTP0_0$
Pulse width modulation by timer A2 ; $RTP0_3$, $RTP1_1$, $RTP1_0$
1 0 : When pulse mode 0 is selected,
Do not set.
When pulse mode 1 is selected (valid only for RTP0, $RTP1_1$, $RTP1_0$),
Pulse width modulation by timer A1 ; $RTP0_1$, $RTP0_0$
Pulse width modulation by timer A2 ; $RTP0_3$, $RTP0_2$
Pulse width modulation by timer A4 ; $RTP1_1$, $RTP1_0$
1 1 : Do not select.

Waveform output control bit 0
When pulse mode 0 is selected,
0 : $RTP1_1$, $RTP1_0$ waveform outputs are disabled.
1 : $RTP1_1$, $RTP1_0$ waveform outputs are enabled.
When pulse mode 1 is selected,
Fix this bit to 0 .

Waveform output control bit 1
When pulse mode 0 is selected,
0 : RTP0 waveform output is disabled.
1 : RTP0 waveform output is enabled.
When pulse mode 1 is selected,
0 : RTP0, $RTP1_1$, $RTP1_0$ waveform outputs are disabled.
1 : RTP0, $RTP1_1$, $RTP1_0$ waveform outputs are enabled.

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



Timer A0 mode register 56_{16}
Timer A3 mode register 59_{16}

Fix these bits to “000000” in pulse output port mode.

Clock source select bits
(See Table 7.)

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

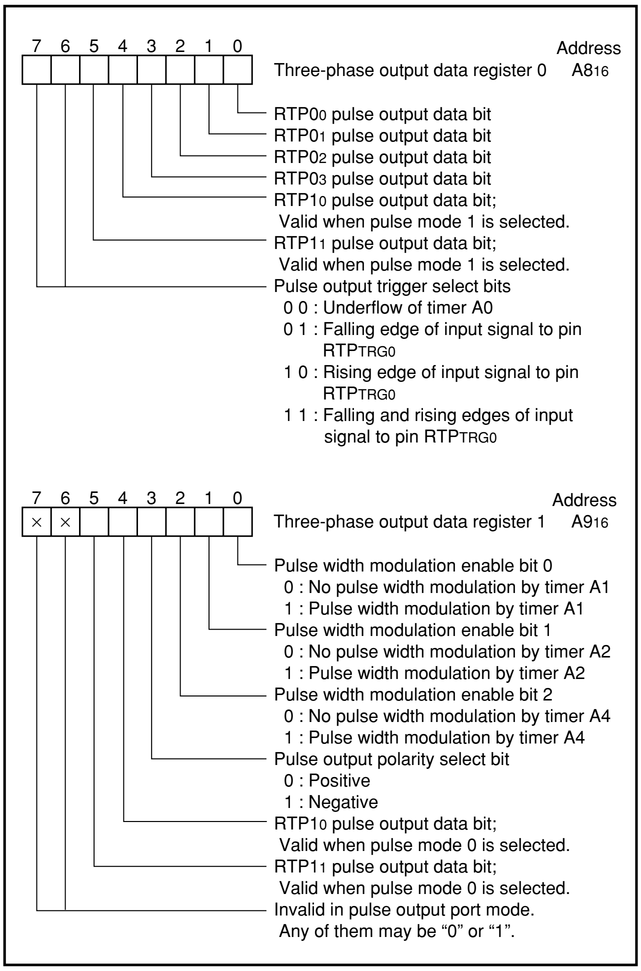

7 6 5 4 3 2 1 0
Address
Three-phase output data register 0 A8$_{16}$
RTP0$_0$ pulse output data bit
RTP0$_1$ pulse output data bit
RTP0$_2$ pulse output data bit
RTP0$_3$ pulse output data bit
RTP1$_0$ pulse output data bit; Valid when pulse mode 1 is selected.
RTP1$_1$ pulse output data bit; Valid when pulse mode 1 is selected.
Pulse output trigger select bits
0 0 : Underflow of timer A0
0 1 : Falling edge of input signal to pin RTP$_{\text{TRG0}}$
1 0 : Rising edge of input signal to pin RTP$_{\text{TRG0}}$
1 1 : Falling and rising edges of input signal to pin RTP$_{\text{TRG0}}$
7 6 5 4 3 2 1 0
× ×
Address
Three-phase output data register 1 A9$_{16}$
Pulse width modulation enable bit 0
0 : No pulse width modulation by timer A1
1 : Pulse width modulation by timer A1
Pulse width modulation enable bit 1
0 : No pulse width modulation by timer A2
1 : Pulse width modulation by timer A2
Pulse width modulation enable bit 2
0 : No pulse width modulation by timer A4
1 : Pulse width modulation by timer A4
Pulse output polarity select bit
0 : Positive
1 : Negative
RTP1$_0$ pulse output data bit; Valid when pulse mode 0 is selected.
RTP1$_1$ pulse output data bit; Valid when pulse mode 0 is selected.
Invalid in pulse output port mode. Any of them may be “0” or “1”.

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 RTP0$_3$, RTP0$_2$, RTP0$_1$, and RTP0$_0$ 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 RTP1$_1$, RTP1$_0$ 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 RTP1$_1$, RTP1$_0$

•Four of RTP0$_3$, RTP0$_2$, RTP0$_1$, RTP0$_0$.

Each time the contents of timer A3 counter become 0000$_{16}$, the contents of three-phase output data register 1 (address A9$_{16}$)’s bits 5 and 4, which corresponding to RTP1$_1$ and RTP1$_0$, are output from ports.

Each time the contents of timer A0 counter become 0000$_{16}$, the contents of three-phase output data register 0 (address A8$_{16}$)’s low order 4 bits, which corresponding to RTP0$_3$, RTP0$_2$, RTP0$_1$, RTP0$_0$, 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$_{16}$; when writing “1” to it, “H” level is output from the pulse output port.

In the case that an input trigger of pin RTP$_{\text{TRG0}}$ 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 RTP0$_3$, RTP0$_2$, RTP0$_1$, and RTP0$_0$. 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$_{16}$.

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 A9$_{16}$) to “1” and the pulse width modulation timer select bits of the waveform output mode register (bits 5 and 4 at address A6$_{16}$) to “00”. Figure 52 shows example waveforms in pulse mode 0.

In ports selecting pulse mode 0, output of RTP1$_1$ and RTP1$_0$ is controlled by the waveform output control bit 0 (bit 6) of waveform output mode register; output of RTP0$_3$, RTP0$_2$, RTP0$_1$ and RTP0$_0$ 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 $\overline{\text{P6OUT}_{\text{CUT}}}$.

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 RTP1$_1$, RTP1$_0$, RTP0$_3$, RTP0$_2$, RTP0$_1$, RTP0$_0$
Whether to control these pulse output ports by an underflow of timer A0 or an input edge to pin RTP$_{TRG0}$ is selected by the pulse output trigger select bit of the three-phase output data register 0 (bits 7 and 6 at address A8$_{16}$).
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 A6$_{16}$) 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 RTP1$_1$, RTP1$_0$, RTP0$_3$, RTP0$_2$, RTP0$_1$, RTP0$_0$ 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 RTP1$_1$, RTP1$_0$, RTP0$_3$, and the other consists of RTP0$_2$, RTP0$_1$, RTP0$_0$. 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 RTP1$_1$, RTP1$_0$, and another consists of RTP0$_3$, RTP0$_2$, and the other consists of RTP0$_1$, RTP0$_0$. 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 A9$_{16}$) 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 A6$_{16}$).
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 $\overline{\mathrm{P6OUT_{CUT}}}$, or by executing instructions.

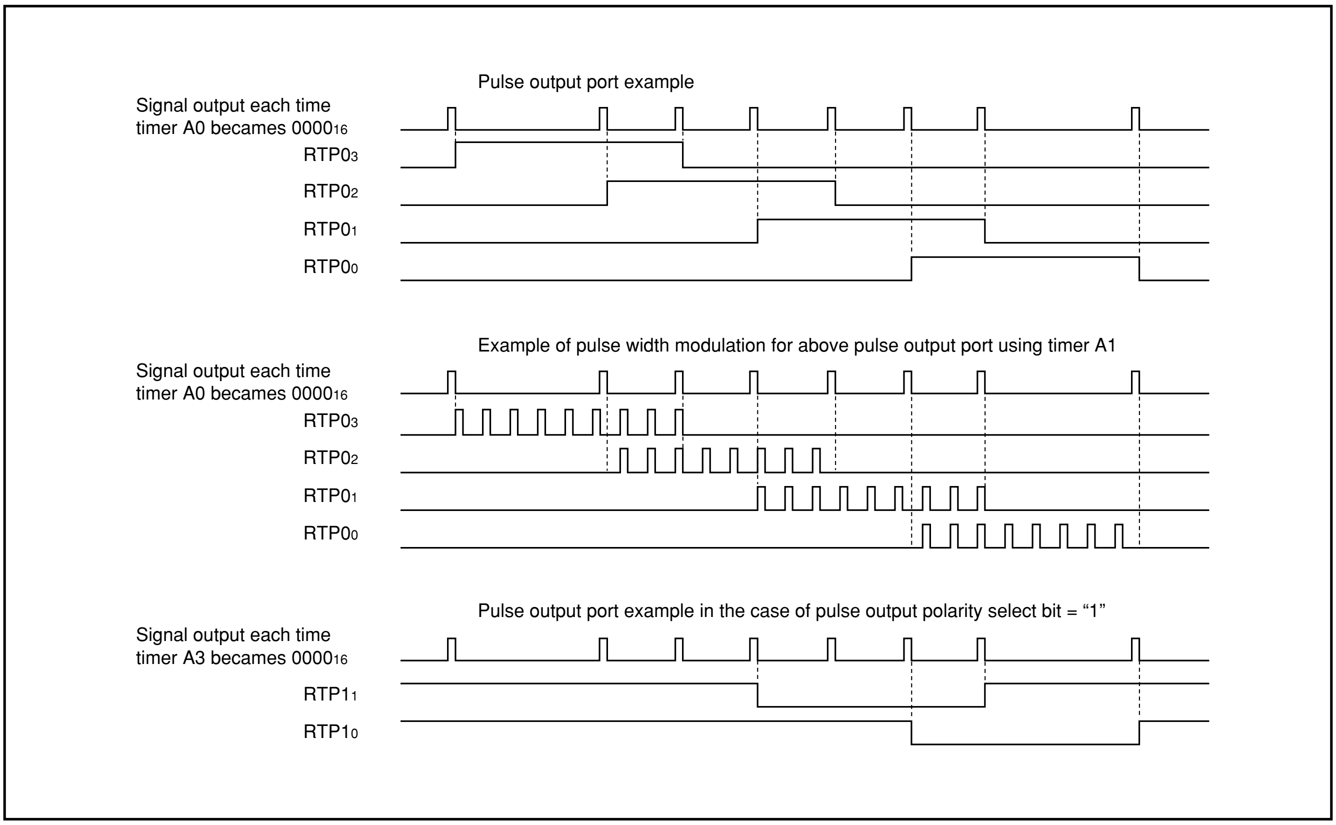

Pulse output port example
Signal output each time timer A0 becames 0000$_{16}$
RTP0$_3$
RTP0$_2$
RTP0$_1$
RTP0$_0$
Example of pulse width modulation for above pulse output port using timer A1
Signal output each time timer A0 becames 0000$_{16}$
RTP0$_3$
RTP0$_2$
RTP0$_1$
RTP0$_0$
Pulse output port example in the case of pulse output polarity select bit = “1”
Signal output each time timer A3 becames 0000$_{16}$
RTP1$_1$
RTP1$_0$

Fig. 52 Example waveforms in pulse mode 0

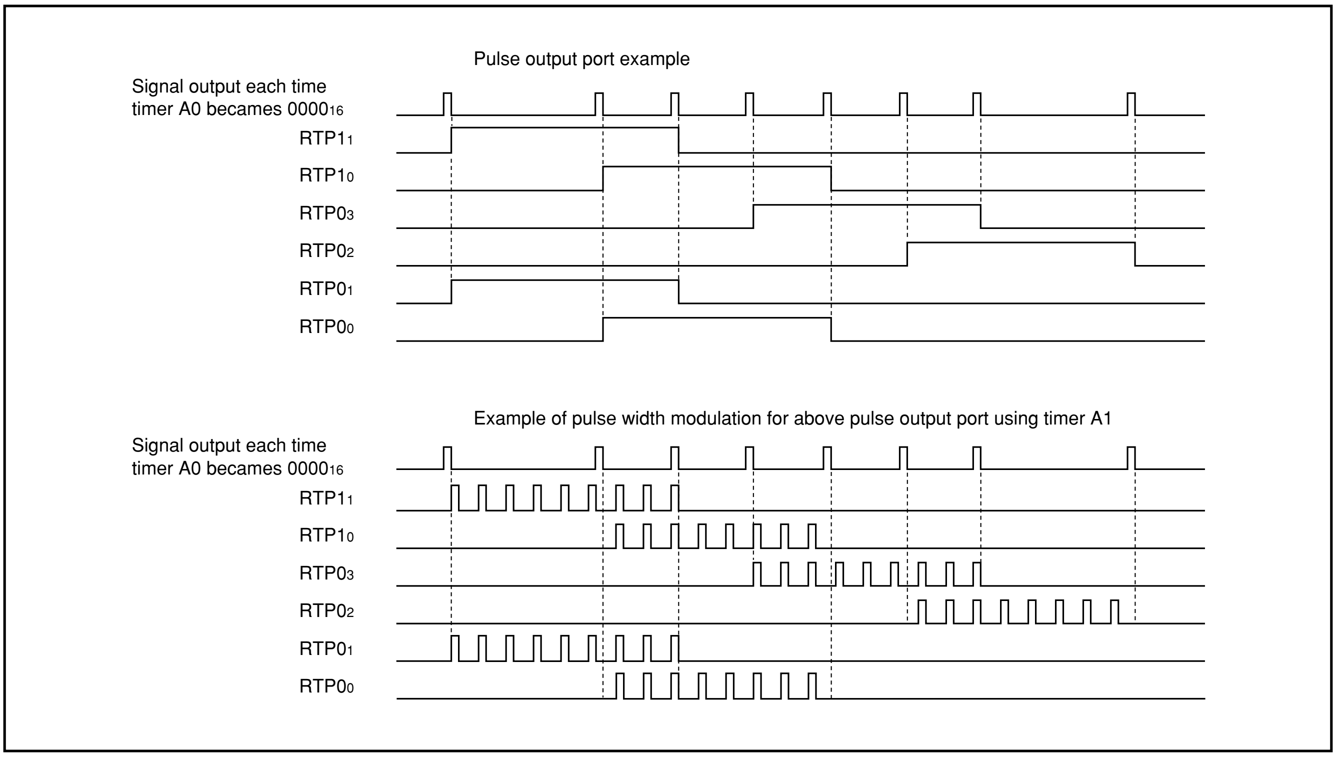
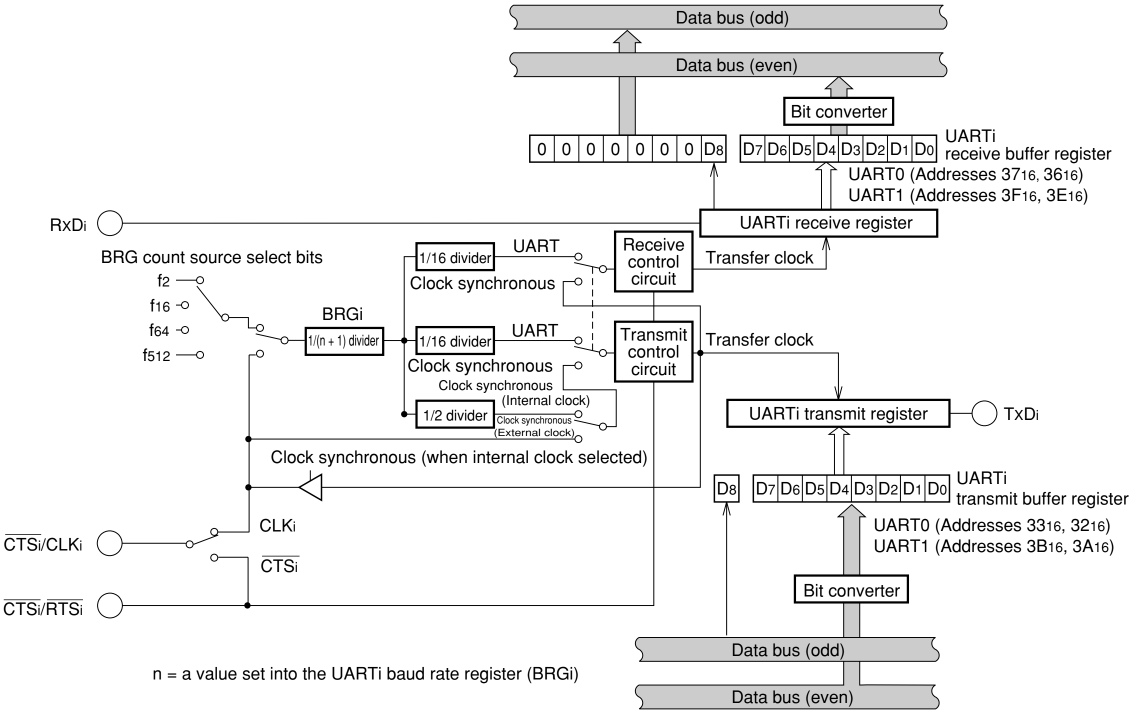

Pulse output port example
Signal output each time timer A0 becames 0000$_{16}$
RTP1$_1$
RTP1$_0$
RTP0$_3$
RTP0$_2$
RTP0$_1$
RTP0$_0$
Example of pulse width modulation for above pulse output port using timer A1
Signal output each time timer A0 becames 0000$_{16}$
RTP1$_1$
RTP1$_0$
RTP0$_3$
RTP0$_2$
RTP0$_1$
RTP0$_0$

Fig. 53 Example waveforms in pulse mode 1

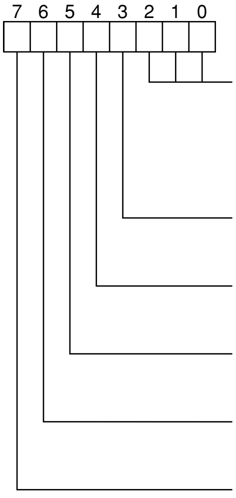
SERIAL I/O PORTS

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 UARTi(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 asynchronous (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 UARTi transmit/receive control register.
Each communication method is described below.


Data bus (odd)
Data bus (even)
Bit converter
0 0 0 0 0 0 0 D8
D7 D6 D5 D4 D3 D2 D1 D0
UARTi receive buffer register
UART0 (Addresses 37_{16}, 36_{16})
UART1 (Addresses $3F_{16}$, $3E_{16}$)
RxDi
UARTi receive register
BRG count source select bits
f2
f16
f64
f512
BRGi
1/(n + 1) divider
1/16 divider
UART
Clock synchronous
Receive control circuit
Transfer clock
1/16 divider
UART
Clock synchronous
Transmit control circuit
Transfer clock
Clock synchronous (Internal clock)
1/2 divider
Clock synchronous (External clock)
UARTi transmit register
TxDi
Clock synchronous (when internal clock selected)
D8
D7 D6 D5 D4 D3 D2 D1 D0
UARTi transmit buffer register
UART0 (Addresses 33_{16}, 32_{16})
UART1 (Addresses $3B_{16}$, $3A_{16}$)
CTSi/CLKi
CLKi
CTSi
CTSi/RTSi
Bit converter
Data bus (odd)
Data bus (even)

n = a value set into the UARTi baud rate register (BRGi)

Fig. 54 Block diagram of serial I/O port



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

Serial I/O mode select bits
0 0 0 : Serial I/O is invalid. (Port P1 functions as a programmable I/O port.)
0 0 1 : Clock synchronous
1 0 0 : 7-bit UART
1 0 1 : 8-bit UART
1 1 0 : 9-bit UART

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

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

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

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

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

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

Fig. 55 Bit configuration of UARTi transmit/receive mode register

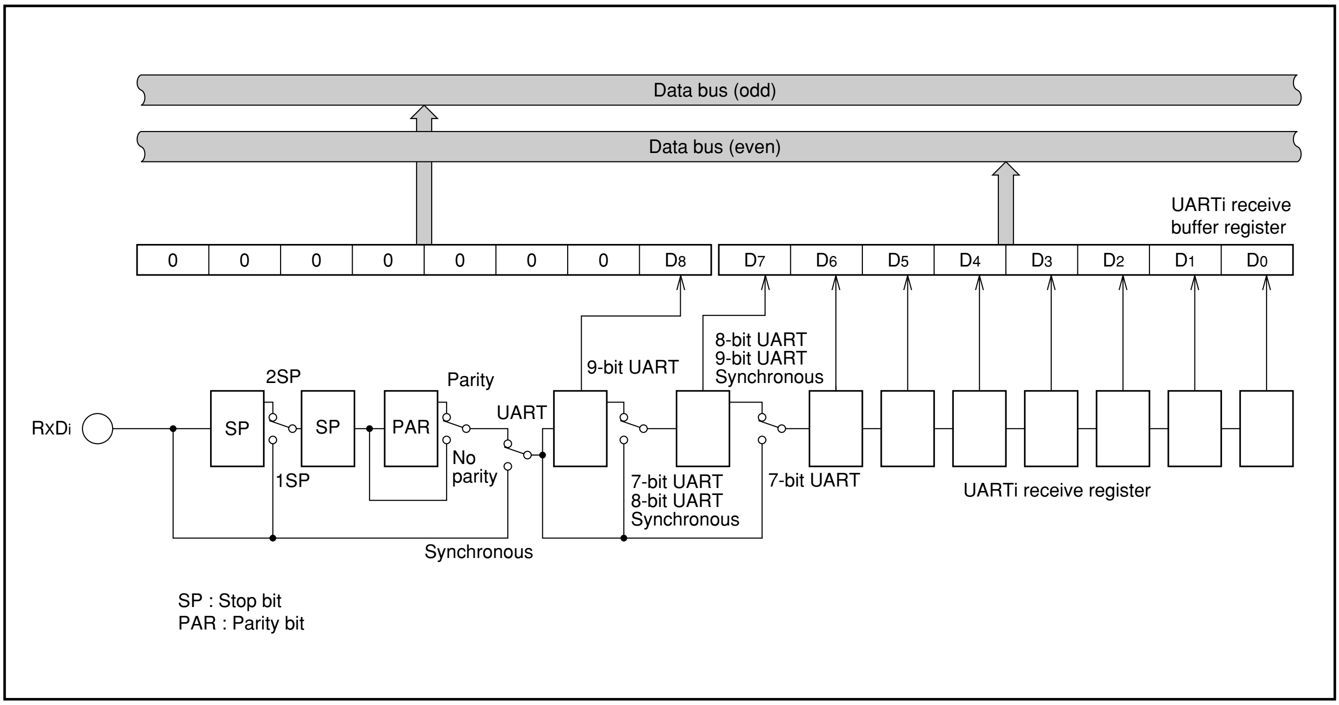

Data bus (odd)
Data bus (even)
UARTi receive buffer register
0
0
0
0
0
0
0
D8
D7
D6
D5
D4
D3
D2
D1
D0
9-bit UART
8-bit UART
9-bit UART
Synchronous
2SP
Parity
RxDi
SP
SP
PAR
UART
No parity
1SP
7-bit UART
8-bit UART
Synchronous
7-bit UART
UARTi receive register
Synchronous
SP : Stop bit
PAR : Parity bit

Fig. 56 Block diagram of receiver

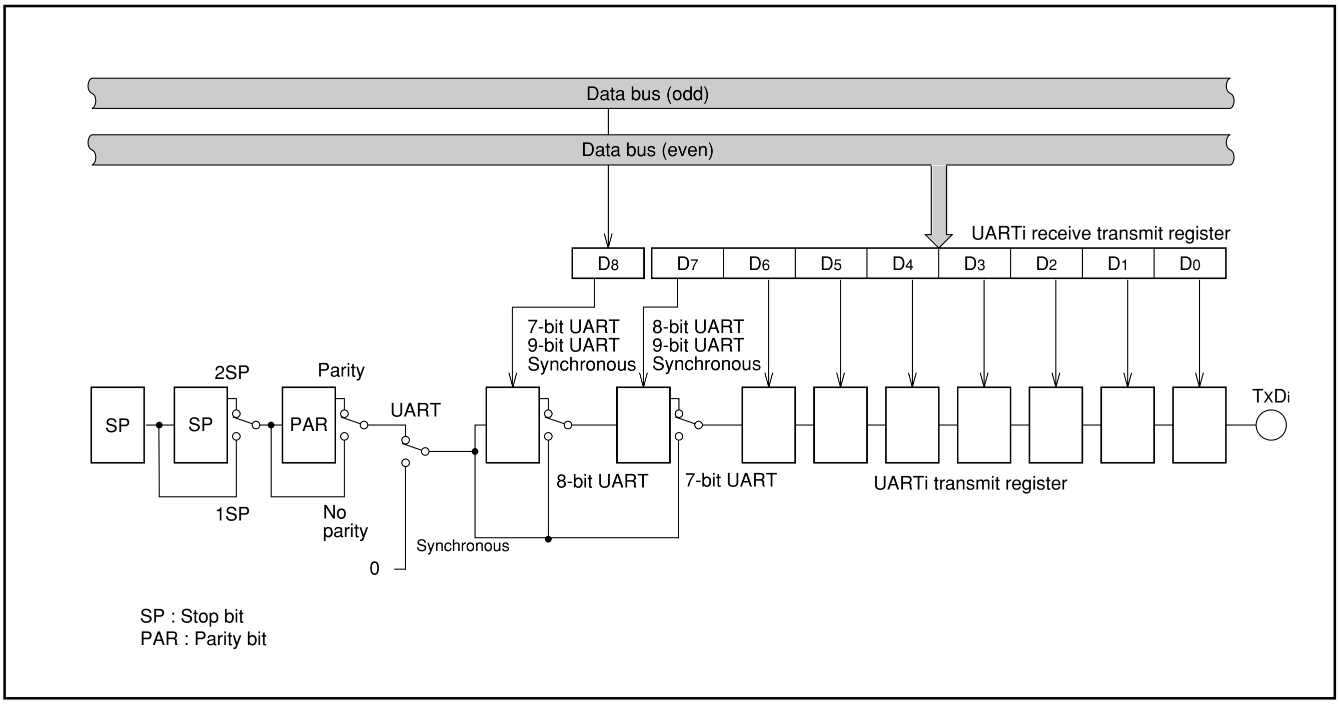
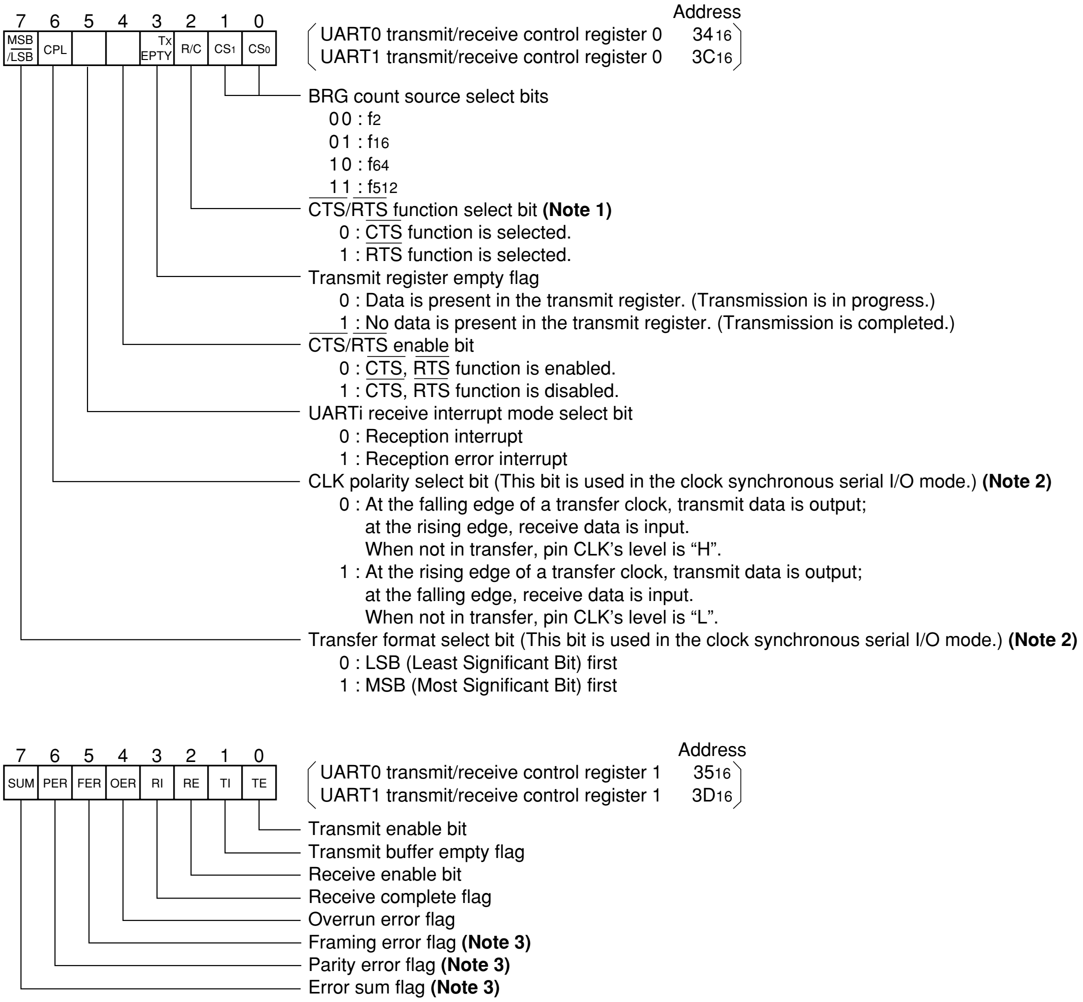

Data bus (odd)
Data bus (even)
UARTi receive transmit register
D8
D7
D6
D5
D4
D3
D2
D1
D0
7-bit UART
9-bit UART
Synchronous
8-bit UART
9-bit UART
Synchronous
2SP
Parity
SP
SP
PAR
UART
TxDi
8-bit UART
7-bit UART
UARTi transmit register
1SP
No parity
Synchronous
0
SP : Stop bit
PAR : Parity bit

Fig. 57 Block diagram of transmitter



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

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

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

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

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

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

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

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

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

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

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

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

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 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 (CS$_0$) and bit 1 (CS$_1$) 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 CLK$_j$. Therefore, when the selected clock is fi,

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

On the clock receiving side, the CS$_0$ and CS$_1$ bits of the UARTk transmit/receive control register 0 are ignored because an external clock is selected.

Both of UART0 and UART1 can use $\overline{\text{CTS}}$ 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 P1$_3$ and P1$_7$ are used as pins T$_X$D$_i$ or as port pins. When bits 2 and 3 are “0”, P1$_3$ and P1$_7$ function as pins T$_X$D$_i$; when bits 2 and 3 are “1”, P1$_3$ and P1$_7$ function as port pins. Therefore, in the input-only system where pins T$_X$D$_i$ are not used, pins T$_X$D$_i$ can function as port pins.

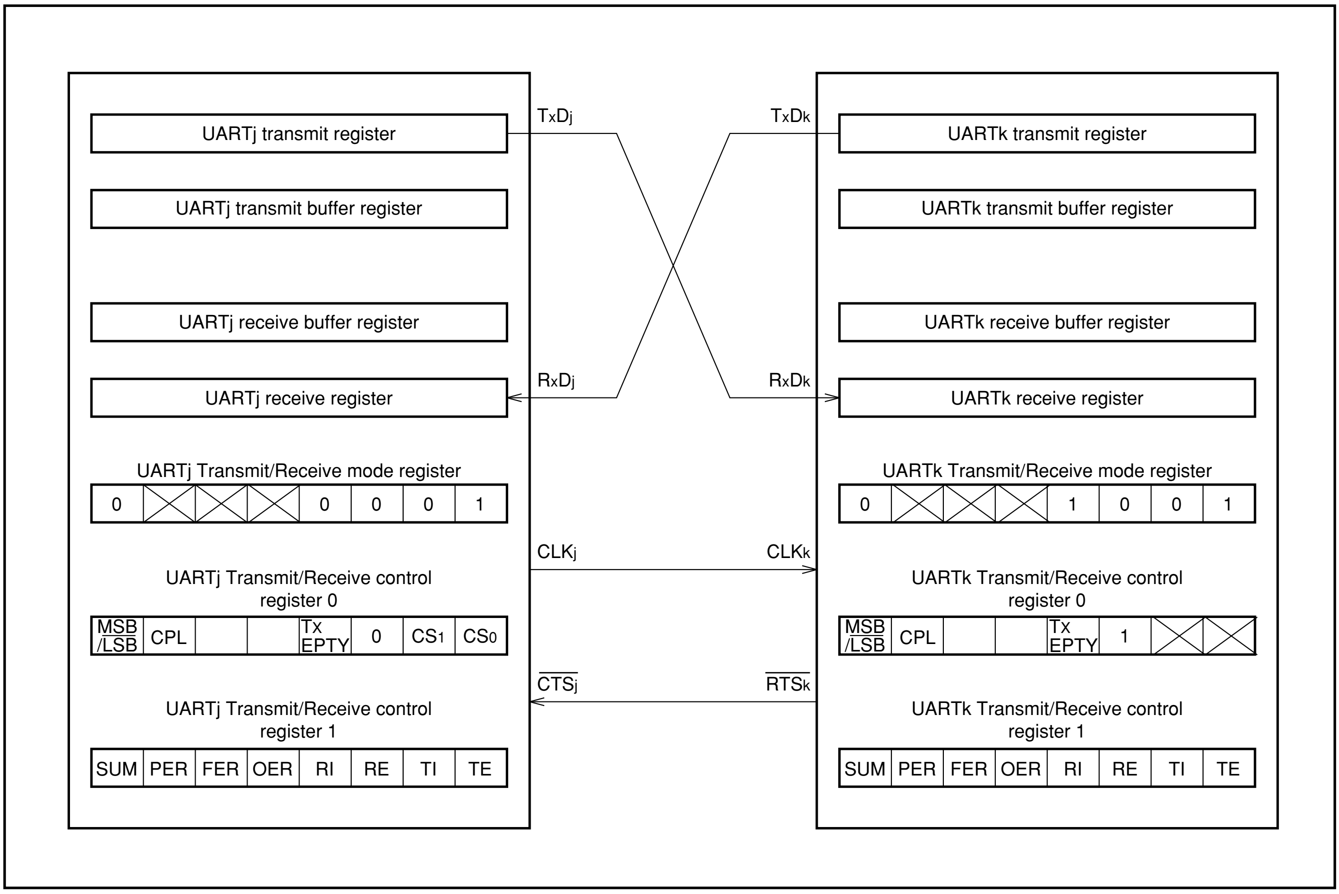

UARTj transmit register
UARTj transmit buffer register
UARTj receive buffer register
UARTj receive register
UARTj Transmit/Receive mode register
0
0
0
0
1
UARTj Transmit/Receive control register 0
MSB /LSB
CPL
Tx EPTY
0
CS1
CS0
UARTj Transmit/Receive control register 1
SUM
PER
FER
OER
RI
RE
TI
TE
TxDj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

Fig. 59 Clock synchronous serial communication

Transmission

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

When all of the transmit conditions are satisfied, the transmit data in the transmit buffer register are transferred to the transmit register, and transmission starts. As shown in Figure 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, TIj flag, and $\overline{\text{CTS}_j}$ signals are ignored until data transmission completes. Therefore, transmission is not interrupted when $\overline{\text{CTS}_j}$ input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, TIj flag, and $\overline{\text{CTS}_j}$ is checked while the TENDj signal (shown in Figure 60) is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmit buffer register and TIj flag is cleared to "0" before theTENDj signal goes "H".

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

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

Receive

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

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

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

As shown in Figure 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.

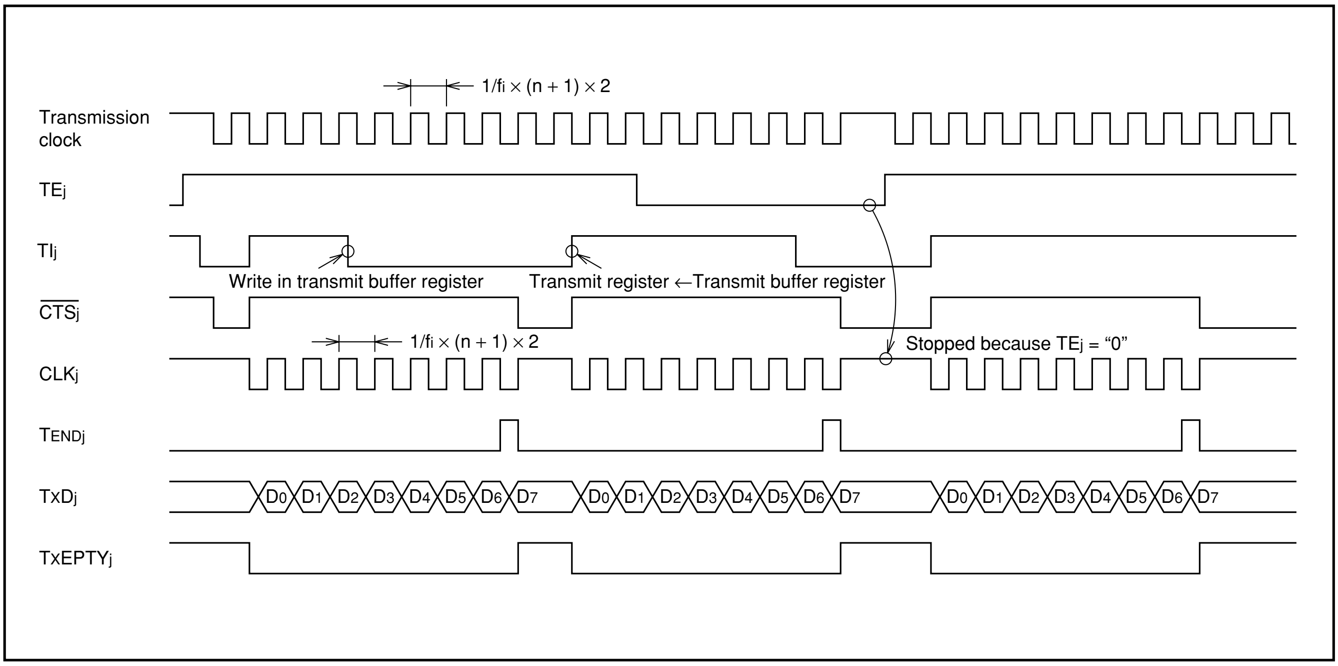

1/fi × (n + 1) × 2
Transmission clock
TEj
TIj
Write in transmit buffer register
Transmit register ←Transmit buffer register
CTSj
1/fi × (n + 1) × 2
Stopped because TEj = "0"
CLKj
TENDj
TxDj
D0 D1 D2 D3 D4 D5 D6 D7
TxEPTYj

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 UARTk receive interrupt control register can be set to "1".

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

- Each reception
- When an error occurs at reception

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

Polarity of transfer clock

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

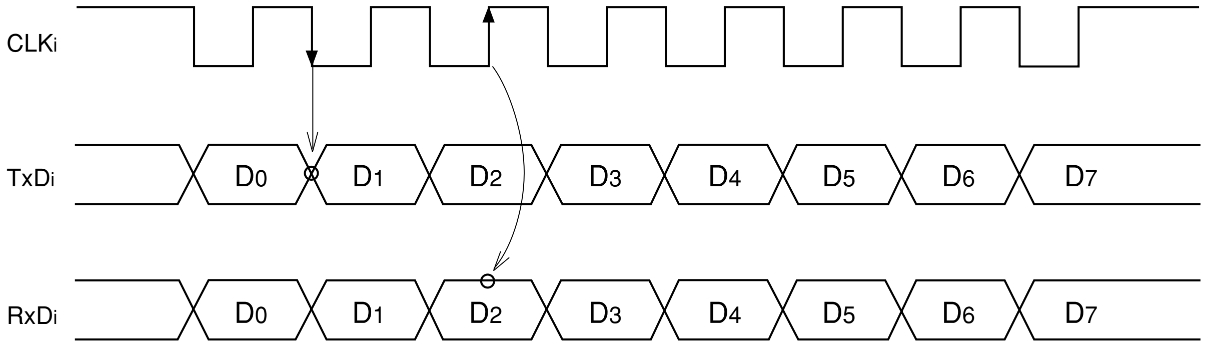
As shown in Figure 61, when bit 6 = "0", the polarity is as follows:

- In transmission, transmit data is output at the falling edge of CLK$_j$.
- In reception, receive data is input at the rising edge of CLK$_k$.
- When not in transfer, CLK$_i$ is at "H" level.

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

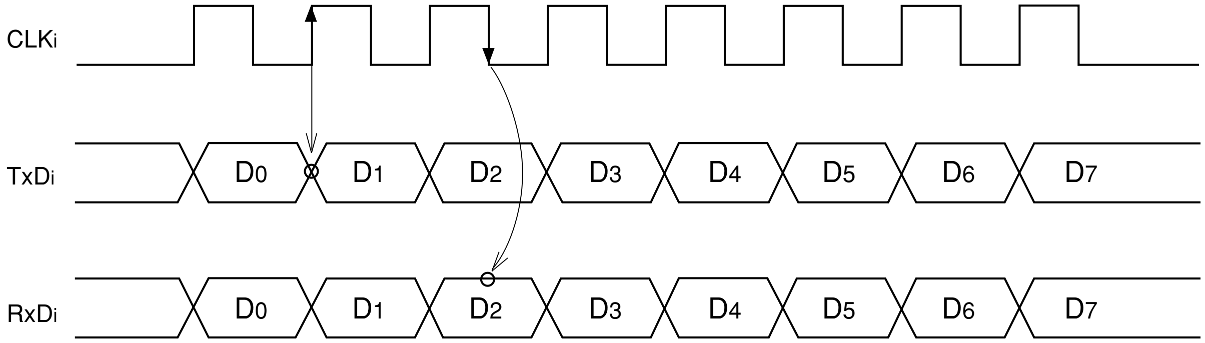
- In transmission, transmit data is output at the rising edge of CLK$_j$.
- In reception, receive data is input at the rising edge of CLK$_k$.
- When not in transfer, CLK$_i$ is at "L" level.

■ CLK polarity select bit = 0



❊ Transmit data is output to pin TxD$_i$ at the falling edge of transfer clock, and receive data is input from pin RxD$_i$ at the rising edge of transfer clock.
When not in transfer, pin CLK$_i$'s level is "H".

■ CLK polarity select bit = 1



❊ Transmit data is output to pin TxD$_i$ at the rising edge of transfer clock, and receive data is input from pin RxD$_i$ at the falling edge of transfer clock.
When not in transfer, pin CLK$_i$'s level is "L".

Fig. 61 Polarity of transfer clock

Selection of transfer format

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

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

Figure 62 shows the connection relation.

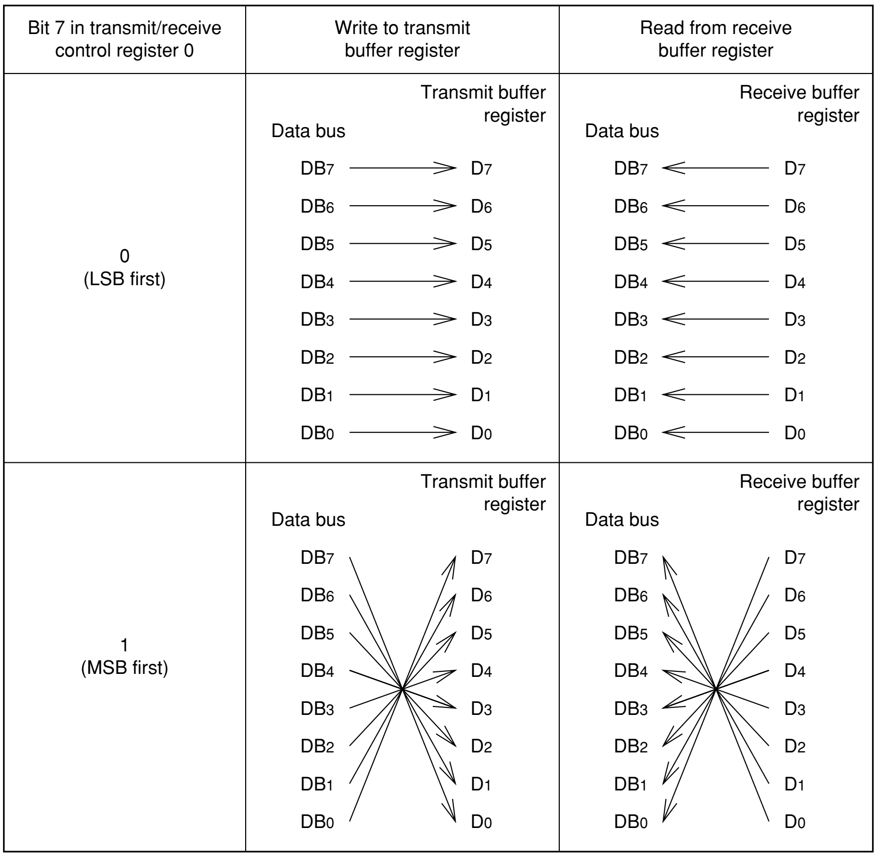

Bit 7 in transmit/receive control register 0
Write to transmit buffer register
Read from receive buffer register
0
(LSB first)
1
(MSB first)
Data bus
Transmit buffer register
Receive buffer register
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
D7
D6
D5
D4
D3
D2
D1
D0

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 $\overline{\text{CTS}_i}/\overline{\text{RTS}_i}$ cannot be selected. Furthermore, when an internal clock is selected, $\overline{\text{RTS}}$ output is undefined. Therefore, do not use the $\overline{\text{RTS}}$ function.

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

ASYNCHRONOUS SERIAL COMMUNICATION

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

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

Bit 3 is used to select an internal clock or an external clock. If bit 3 is "0", an internal clock is selected and if bit 3 is "1", then external clock is selected. If an internal clock is selected, bit 0 (CS$_0$) and bit 1 (CS$_1$) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLK$_i$ pin can be used as a normal I/O pin. The selected internal or external clock is divided by (n + 1), then by 16, and is passed through a control circuit to create the UART transmission clock or UART receive clock.

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

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

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

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

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

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

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

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

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

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 ($\overline{CTS_i}$/$\overline{RTS_i}$ separate select bits), the function of the $\overline{CTS}$/$\overline{RTS}$ pin can be separated into two functions, and each function can be assigned to two different pins. When bits 0 and 1 = "11", the above separation is performed. When bits 0 and 1 = "00", no separation is performed.

Table 8 lists the selection methods of the $\overline{CTS}$/$\overline{RTS}$ function.

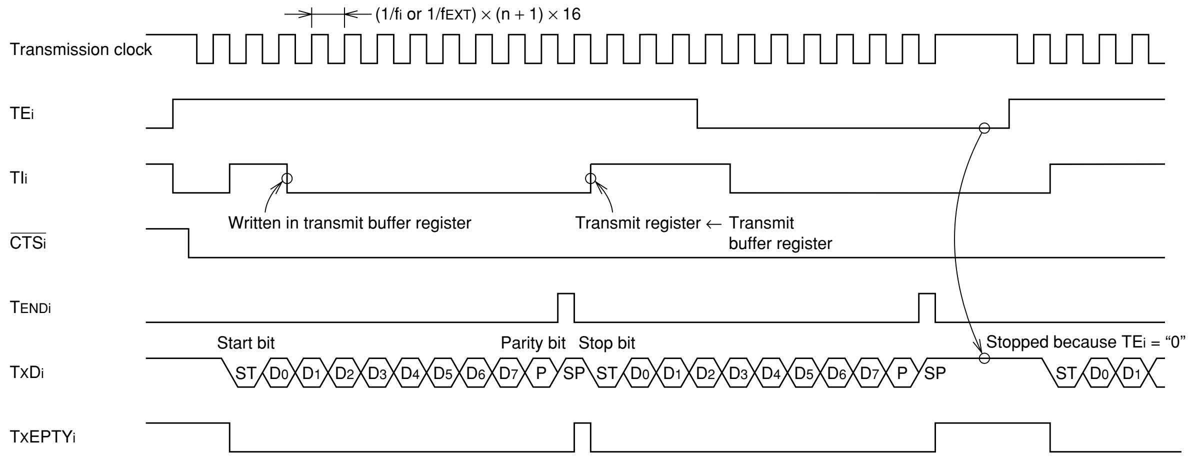

(1/fi or 1/fEXT) × (n + 1) × 16
Transmission clock
TEi
TIi
Written in transmit buffer register
Transmit register ← Transmit buffer register
CTSi
TENDi
Start bit
Parity bit
Stop bit
Stopped because TEi = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
TxEPTYi

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

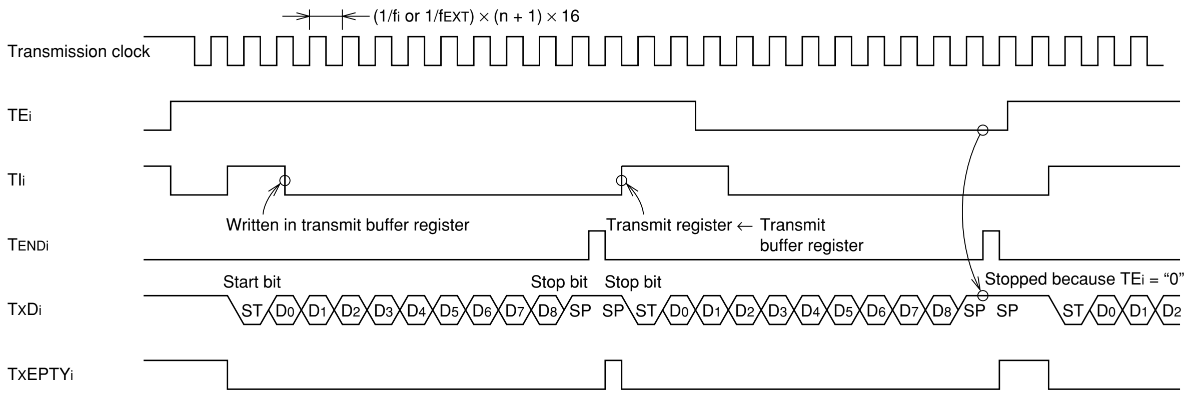

(1/fi or 1/fEXT) × (n + 1) × 16
Transmission clock
TEi
TIi
Written in transmit buffer register
Transmit register ← Transmit buffer register
TENDi
Start bit
Stop bit
Stop bit
Stopped because TEi = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2
TxEPTYi

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

Transmission

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

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

As shown in Figures 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 succeedingly.

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

Therefore, transmission does not stop until it completes event if, during transmission, the TEi flag is cleared to “0” or CTSi input is set to “1”.

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

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

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

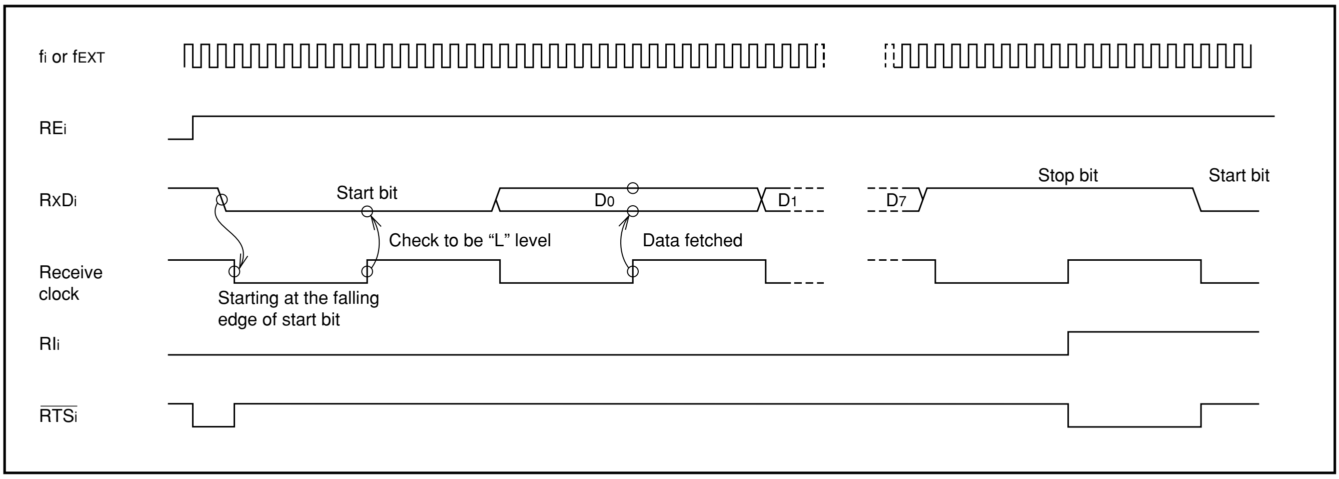


Fig. 65 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit selected

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

$\overline{\text{CTS}}/\overline{\text{RTS}}$ enable bit	$\overline{\text{CTS}}_i/\overline{\text{RTS}}_i$ separate select bit	$\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit	Functions			
			Pin $P1_0/\overline{\text{CTS}}_0/\overline{\text{RTS}}_0$	Pin $P1_1/\overline{\text{CTS}}_0/\text{CLK}_0$	Pin $P1_4/\overline{\text{CTS}}_1/\overline{\text{RTS}}_1$	Pin $P1_5/\overline{\text{CTS}}_1/\text{CLK}_1$
0	0	0	$\overline{\text{CTS}}_0$	$P1_1$ or CLK_0	$\overline{\text{CTS}}_1$	$P1_5$ or CLK_1
		1	$\overline{\text{RTS}}_0$	$P1_1$ or CLK_0	$\overline{\text{RTS}}_1$	$P1_5$ or CLK_1
	1	×	$\overline{\text{RTS}}_0$	$\overline{\text{CTS}}_0$ **(Notes 1 and 2)**	$\overline{\text{RTS}}_1$	$\overline{\text{CTS}}_1$ **(Notes 1 and 2)**
1	×	×	$P1_0$	$P1_1$ or CLK_0	$P1_4$	$P1_5$ or CLK_1

×: 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 CLKi 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.

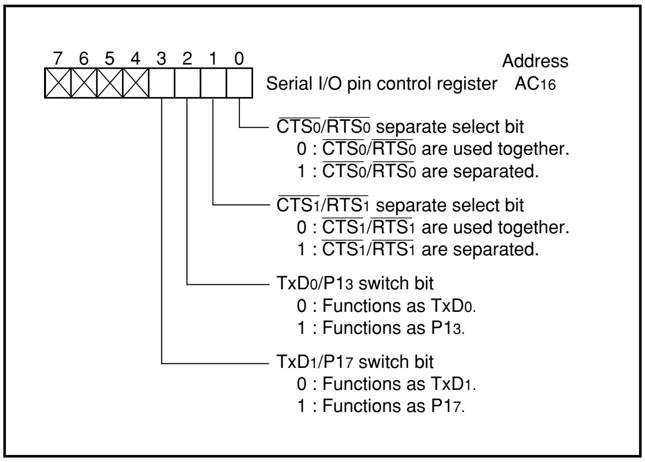

7 6 5 4 3 2 1 0
Address
Serial I/O pin control register AC16
CTS0/RTS0 separate select bit
0 : CTS0/RTS0 are used together.
1 : CTS0/RTS0 are separated.
CTS1/RTS1 separate select bit
0 : CTS1/RTS1 are used together.
1 : CTS1/RTS1 are separated.
TxD0/P13 switch bit
0 : Functions as TxD0.
1 : Functions as P13.
TxD1/P17 switch bit
0 : Functions as TxD1.
1 : Functions as P17.

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 (RIi flag) of UARTi transmit/receive control register 1 is set to "1." In other words, the RIi flag indicates that the receive buffer register contains data when it is set to "1." At this time, when the low-order byte of the UARTk receive buffer register is read out, $\overline{\text{RTS}}$i output goes back to "L" to indicate that the register is ready to receive the next data.

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

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

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

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

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

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

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

Interrupt request at completion of reception

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

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

- Each reception
- When an error occurs at reception

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

Sleep mode

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

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

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

When sleep mode is selected, the contents of the receive register is not transferred to the receive buffer register if bit 7 (bit 6 if 7-bit asynchronous communication and bit 8 if 9-bit asynchronous communication) of the received data is “0”. Also the RIi, OERi, FERi, PERi, and the SUMi flags are unchanged. Therefore, the interrupt request bit of the UARTi receive interrupt control register is also unchanged. Normal receive operation takes place when bit 7 of the received data is “1”.

The following is an example of how the sleep mode can be used.

The main microcomputer first sends data: bit 7 is “1” and bits 0 to 6 are set to the address of the subordinate microcomputer to be communicated with. Then all subordinate microcomputers receive this data. Each subordinate microcomputer checks the received data, clears the sleep bit to “0” if bits 0 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 $\overline{\text{CTSi}}$ and $\overline{\text{RTSi}}$ are separated, pin CLKi cannot be used. Therefore, when $\overline{\text{CTSi}}$ and $\overline{\text{RTSi}}$ are separated in UART mode, be sure to select an internal clock.

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

A-D CONVERTER

The A-D converter is a 10-bit successive approximation converter. 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.

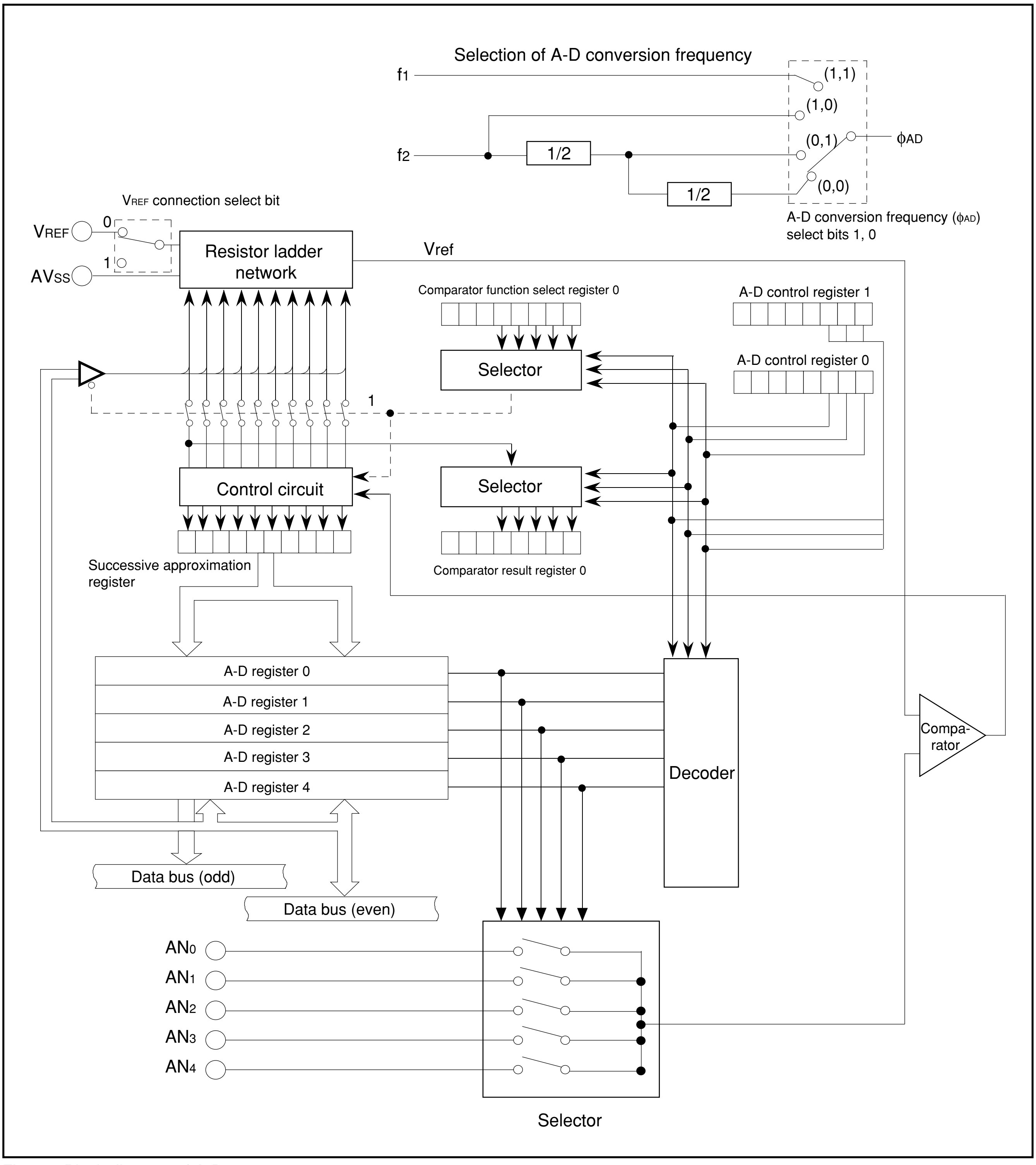

Selection of A-D conversion frequency
f1
f2
1/2
1/2
(1,1)
(1,0)
(0,1)
(0,0)
ϕAD
A-D conversion frequency (ϕAD) select bits 1, 0
VREF connection select bit
VREF
AVSS
0
1
Resistor ladder network
Vref
Comparator function select register 0
A-D control register 1
A-D control register 0
Selector
Selector
1
Control circuit
Successive approximation register
Comparator result register 0
A-D register 0
A-D register 1
A-D register 2
A-D register 3
A-D register 4
Decoder
Compa-rator
Data bus (odd)
Data bus (even)
AN0
AN1
AN2
AN3
AN4
Selector

Fig. 67 Block diagram of A-D converter

Figure 68 shows the bit configuration of comparator function select register 0 (address DC_{16}). 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_2." 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 DE_{16}) 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 $1E_{16}$), and Figure 71 shows that of the A-D control register 1 (address $1F_{16}$).

The operation clock of the A-D converter, ϕ_{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", ϕ_{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", ϕ_{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.

ϕ_{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_2" 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 "00_{16}" when read.

Whether to connect the reference voltage input (V_{REF}) with the ladder network or not depends on bit 5 of the A-D control register 1. The V_{REF} 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 V_{REF} pin to the ladder network can be cut off by disconnecting ladder network from the V_{REF} pin.

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

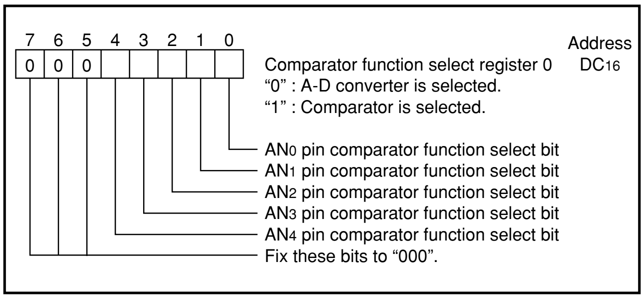
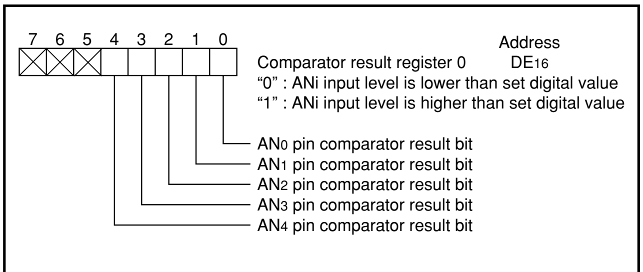

7 6 5 4 3 2 1 0
0 0 0
Address
Comparator function select register 0
DC_{16}
"0" : A-D converter is selected.
"1" : Comparator is selected.
AN_0 pin comparator function select bit
AN_1 pin comparator function select bit
AN_2 pin comparator function select bit
AN_3 pin comparator function select bit
AN_4 pin comparator function select bit
Fix these bits to "000".

Fig. 68 Bit configuration of comparator function select register 0


7 6 5 4 3 2 1 0
Address
Comparator result register 0
DE_{16}
"0" : ANi input level is lower than set digital value
"1" : ANi input level is higher than set digital value
AN_0 pin comparator result bit
AN_1 pin comparator result bit
AN_2 pin comparator result bit
AN_3 pin comparator result bit
AN_4 pin comparator result bit

Note: Do not access with the **ORAM(ORAMB)** or **ANDM(ANDMB)** instruction.

Fig. 69 Bit configuration of comparator result register 0

Operation mode

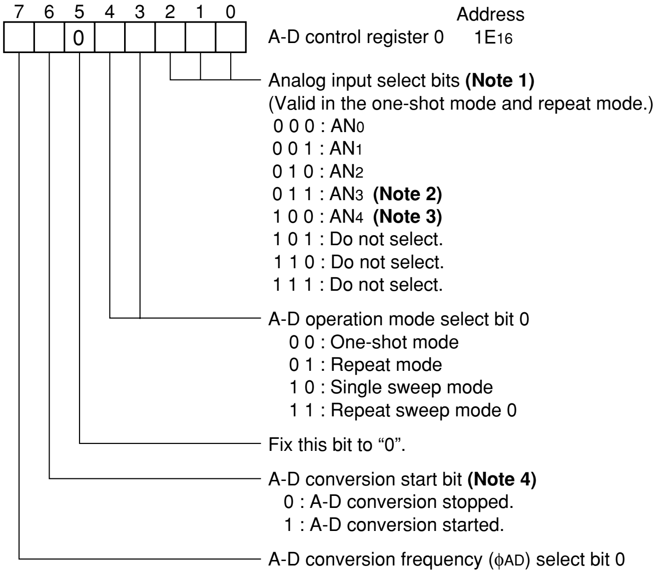
The operation mode is selected by bits 3 and 4 of A-D control register 0 and bit 2 of A-D control register 1. The available operation modes are one-shot, repeat, single sweep, and repeat sweep 0. Either an A-D converter or a comparator can be selected respectively for every pin in the following 4 modes. 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".

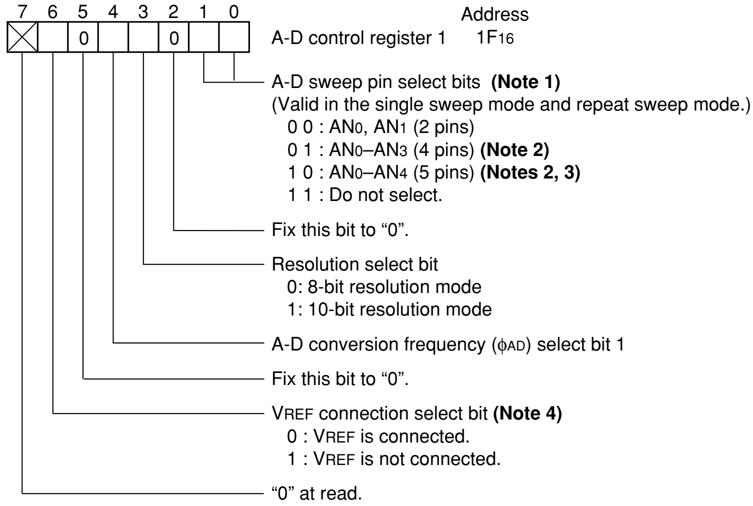
When the ANi (i = 4 through 0) comparator function select bit (bits 4 through 0) of the comparator function select register 0 = "0" and bit 3 of the A-D control register 1 = "1", A-D conversion ends 59 ϕ_{AD} cycles after, and the interrupt request bit of the A-D conversion interrupt control register 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 ANi (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_{AD} 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.


7 6 5 4 3 2 1 0
0
Address
A-D control register 0 $1E_{16}$
Analog input select bits **(Note 1)**
(Valid in the one-shot mode and repeat mode.)
0 0 0 : AN_0
0 0 1 : AN_1
0 1 0 : AN_2
0 1 1 : AN_3 **(Note 2)**
1 0 0 : AN_4 **(Note 3)**
1 0 1 : Do not select.
1 1 0 : Do not select.
1 1 1 : Do not select.
A-D operation mode select bit 0
0 0 : One-shot mode
0 1 : Repeat mode
1 0 : Single sweep mode
1 1 : Repeat sweep mode 0
Fix this bit to "0".
A-D conversion start bit **(Note 4)**
0 : A-D conversion stopped.
1 : A-D conversion started.
A-D conversion frequency (ϕ_{AD}) select bit 0

Notes 1: Invalid in the single sweep mode and repeat sweep mode 0. (Each of these bits may be "0" or "1".)
2: When using pin AN_3, make sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = "0" (output disabled).
3: When using pin AN_4, make sure that the D-A_1 output enable bit (bit 1 at address 96_{16}) = "0" (output disabled).
4: Use the **MOVM (MOVMB)** or **STA (STAB or STAD)** instruction for rewriting to this bit.
5: Rewriting to each bit of the A-D control register 0 (except for bit 6) must be performed while A-D conversion is stopped.

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



A-D control register 1 — 1F16

A-D sweep pin select bits **(Note 1)**
(Valid in the single sweep mode and repeat sweep mode.)
0 0 : AN_0, AN_1 (2 pins)
0 1 : AN_0–AN_3 (4 pins) **(Note 2)**
1 0 : AN_0–AN_4 (5 pins) **(Notes 2, 3)**
1 1 : Do not select.

Fix this bit to “0”.

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

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

Fix this bit to “0”.

V_{REF} connection select bit **(Note 4)**
0 : V_{REF} is connected.
1 : V_{REF} is not connected.

“0” at read.

A-D conversion frequency (ϕ_{AD}) select bit		ϕ_{AD}
Bit 1	Bit 0	
0	0	$f_2/4$
0	1	$f_2/2$
1	0	f_2
1	1	f_1 (Selectable only in 8-bit resolution mode)

Notes 1: Invalid in the one-shot mode and repeat mode. (Each of these bits may be “0” or “1”.)
2: When using pin AN_3, make sure that the D-A_0 output enable bit (bit 0 at address 96_{16}) = “0” (output disabled).
3: When using pin AN_4, make sure that the D-A_1 output enable bit (bit 1 at address 96_{16}) = “0” (output disabled).
4: Once this bit is cleared from “1” to “0”, it is necessary to wait for 1 µs or more before the A-D conversion starts.
5: Rewriting to each bit of the A-D control register 1 must be performed while A-D conversion is stopped.

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

(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 $1F_{16}$). 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 AN_0 pin, the conversion result is stored in A-D register 0, and in the same way, A-D conversion is performed for selected pins one after another. After A-D conversion is performed for all selected pins, the sweep is stopped.

A-D conversion 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 AN_0 pin. The repeat is performed among the selected pins. Also, no interrupt request is generated. Furthermore, the A-D convension 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

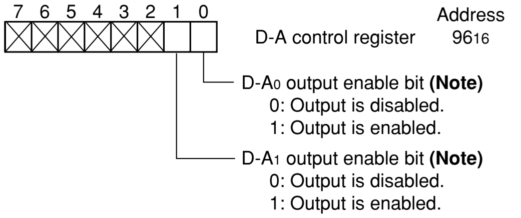
Clear the interrupt request bit of the A-D conversion interrupt control register (bit 3 at address 70_{16}) before using an A-D conversion interrupt. It is because this interrupt request bit is undefined just after reset.

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$_{16}$).
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$_0$ or D-A$_1$) outputs the analog voltage.
This analog voltage (V) is determined according to value n. (“n” = decimal number. This has been set in the D-A register.)

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

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-Ai (i = 0, 1) is multiplexed with I/O port pins, analog input pins, and external interrupt input pins. When a D-A$_i$ output enable bit = “1” (in other words, output is enabled.), however, the corresponding pin cannot function as another I/O pin, which is multiplexed with pin D-Ai.
Also, when not using the D-A converter, be sure to clear the contents of the corresponding D-A output enable bit and D-A register to “0”.


7 6 5 4 3 2 1 0
D-A control register
Address
96$_{16}$
D-A$_0$ output enable bit **(Note)**
0: Output is disabled.
1: Output is enabled.
D-A$_1$ output enable bit **(Note)**
0: Output is disabled.
1: Output is enabled.

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

Fig. 73 Bit configuration of D-A control register

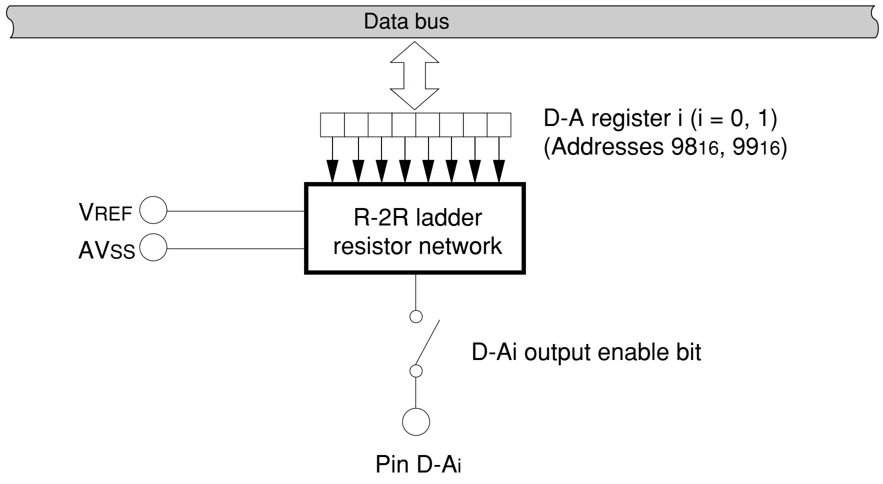

Data bus
D-A register i (i = 0, 1)
(Addresses 98$_{16}$, 99$_{16}$)
V$_{REF}$
AV$_{SS}$
R-2R ladder
resistor network
D-Ai output enable bit
Pin D-A$_i$

Fig. 72 Block diagram of D-A converter

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 Wf_{32}, which is obtained by dividing the peripheral devices' clock f_2 by 16; or clock Wf_{512}, which is obtained by doing it by 256. Bit 0 of the watchdog timer frequency select register (watchdog timer frequency select bit) shown in Figure 75 selects which clock is to be counted.

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

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

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

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

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

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

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

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

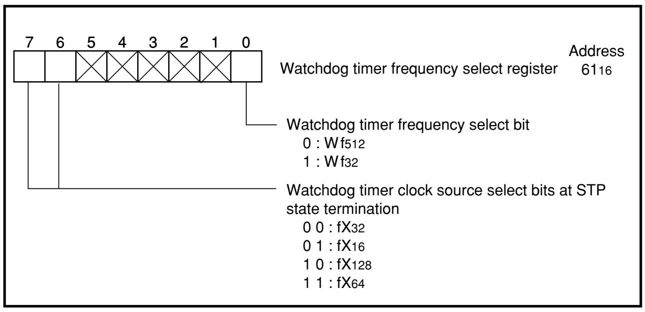
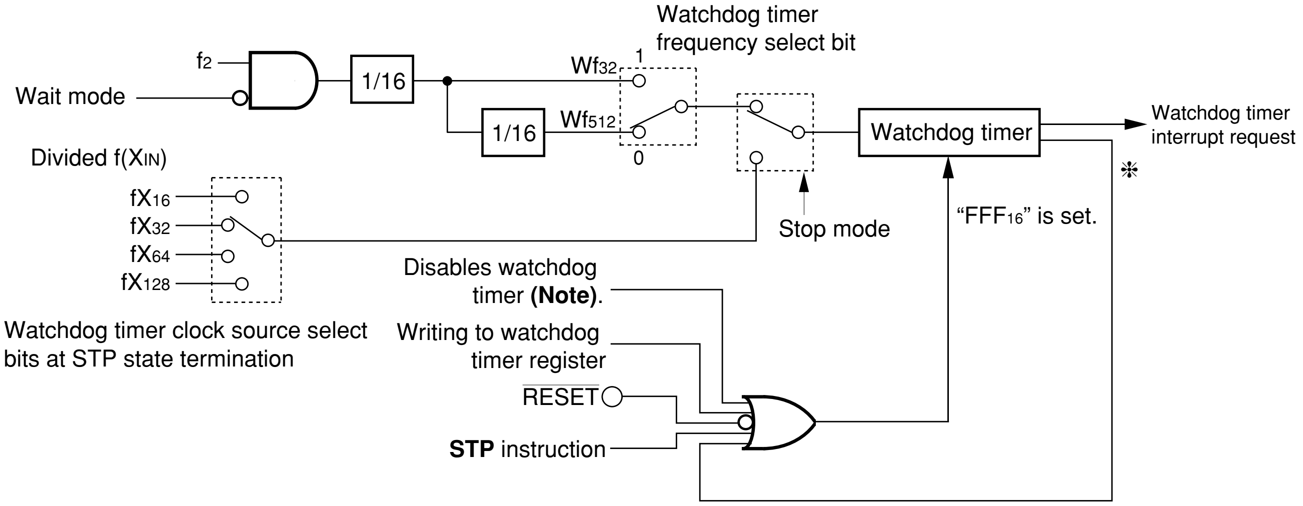

7 6 5 4 3 2 1 0
Address
Watchdog timer frequency select register
61_{16}
Watchdog timer frequency select bit
0 : Wf_{512}
1 : Wf_{32}
Watchdog timer clock source select bits at STP state termination
0 0 : fX_{32}
0 1 : fX_{16}
1 0 : fX_{128}
1 1 : fX_{64}

Fig. 75 Bit configuration of watchdog timer frequency select register


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

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

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

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

Fig. 74 Block diagram of watchdog timer

How to disable watchdog timer

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

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

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

INPUT/OUTPUT PINS

Ports 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 $\overline{\text{P6OUT}_{\text{CUT}}}$ 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 flooting state, and the value input to the pin can be read. When a pin is programmed as an input pin, the data is written only in the port latch and the pin remains floating.

Each of Figures 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”.

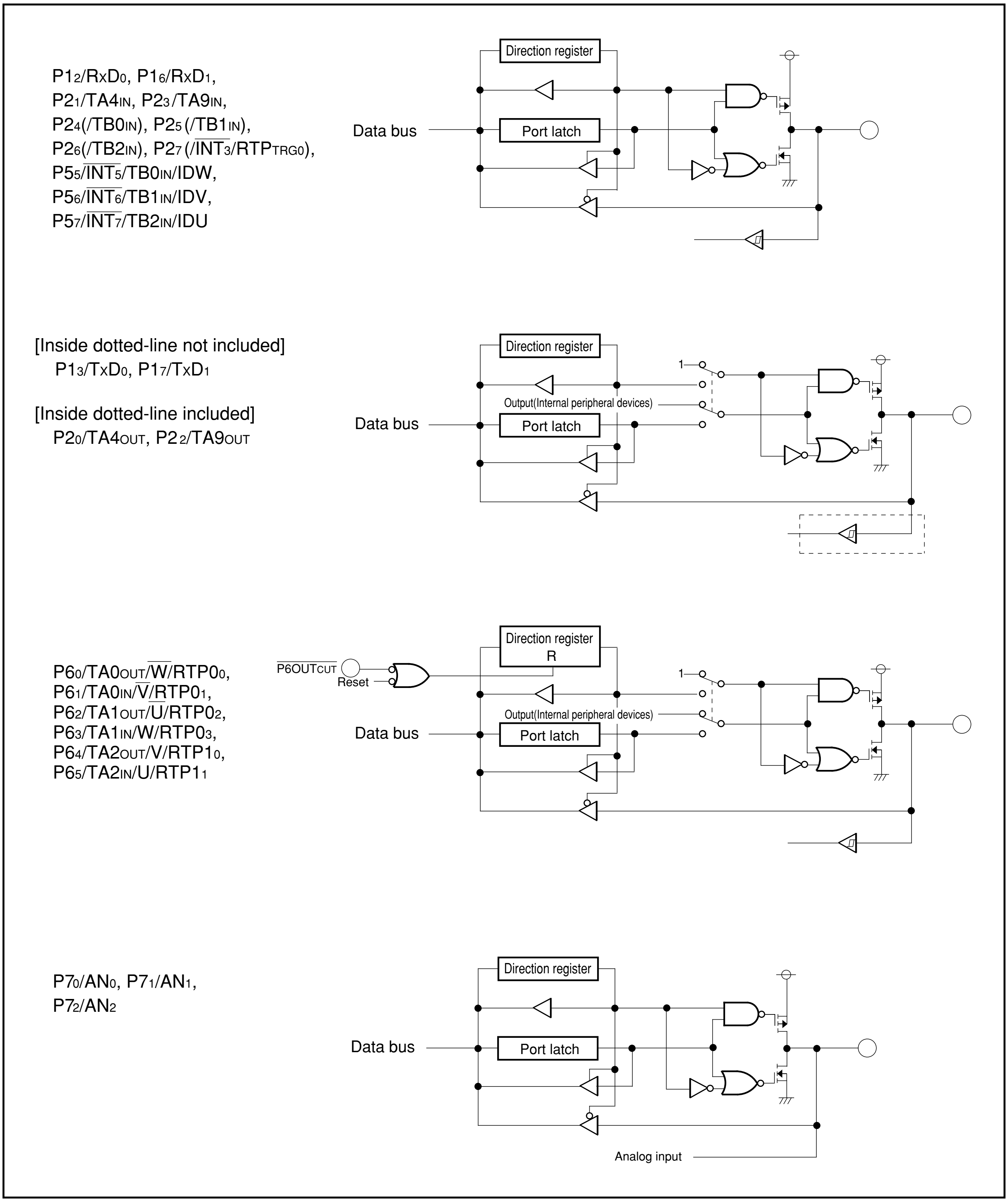

P12/RxD0, P16/RxD1,
P21/TA4IN, P23/TA9IN,
P24(/TB0IN), P25(/TB1IN),
P26(/TB2IN), P27(/INT3/RTPTRG0),
P55/INT5/TB0IN/IDW,
P56/INT6/TB1IN/IDV,
P57/INT7/TB2IN/IDU
Direction register
Data bus
Port latch
[Inside dotted-line not included]
P13/TxD0, P17/TxD1
[Inside dotted-line included]
P20/TA4OUT, P22/TA9OUT
Direction register
1
Output(Internal peripheral devices)
Data bus
Port latch
P60/TA0OUT/W/RTP00,
P61/TA0IN/V/RTP01,
P62/TA1OUT/U/RTP02,
P63/TA1IN/W/RTP03,
P64/TA2OUT/V/RTP10,
P65/TA2IN/U/RTP11
P6OUTCUT
Reset
Direction register
R
1
Output(Internal peripheral devices)
Data bus
Port latch
P70/AN0, P71/AN1,
P72/AN2
Direction register
Data bus
Port latch
Analog input

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

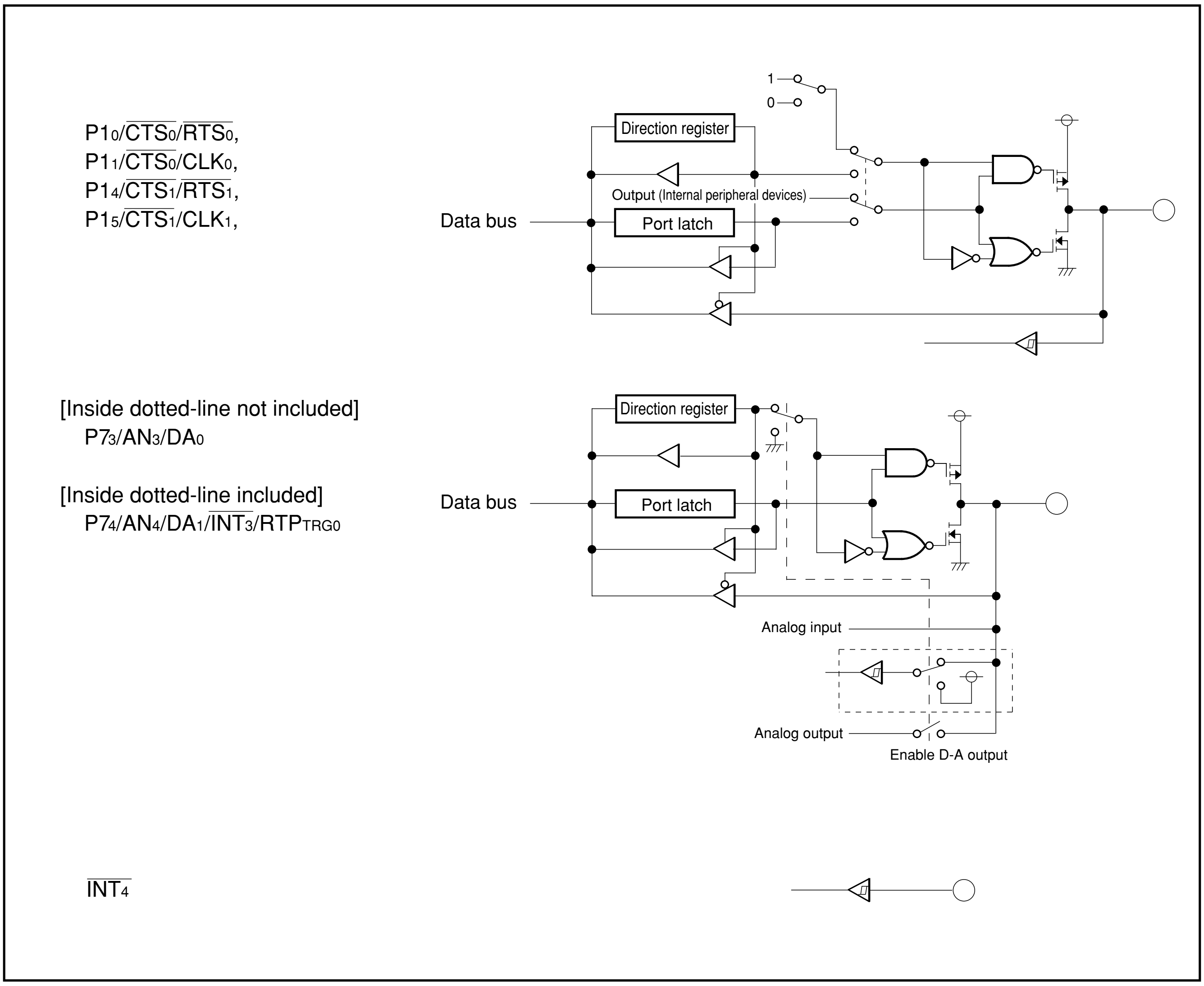

$P1_0/\overline{CTS_0}/\overline{RTS_0}$,
$P1_1/\overline{CTS_0}/CLK_0$,
$P1_4/\overline{CTS_1}/\overline{RTS_1}$,
$P1_5/\overline{CTS_1}/CLK_1$,
1
0
Direction register
Output (Internal peripheral devices)
Data bus
Port latch
[Inside dotted-line not included]
$P7_3/AN_3/DA_0$
[Inside dotted-line included]
$P7_4/AN_4/DA_1/\overline{INT_3}/RTP_{TRG0}$
Direction register
Data bus
Port latch
Analog input
Analog output
Enable D-A output
$\overline{INT_4}$

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

RESET CIRCUIT

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

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

Figures 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 μs or more) under the following conditions:

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

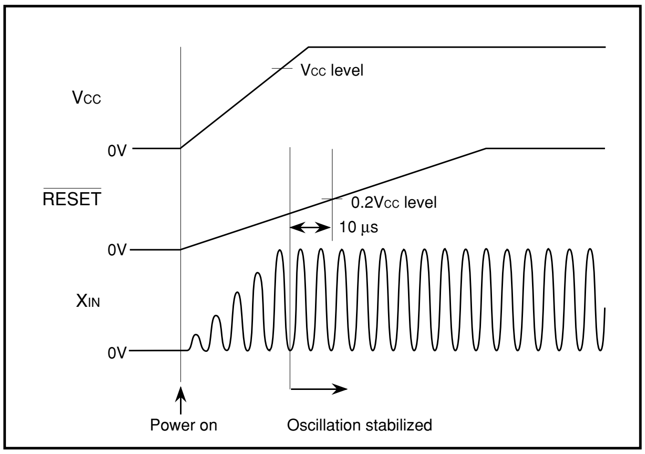


Fig. 80 Operation example of reset circuit (Note that proper evaluation is necessary in the system development stage.)

	Address	
Waveform output mode register	($A6_{16}$)···	00_{16}
Three-phase output data register 0	($A8_{16}$)···	00_{16}
Three-phase output data register 1	($A9_{16}$)···	00_{16}
Position-data-retain function control register	(AA_{16})···	× × × × 0 0 0 0
Serial I/O pin control register	(AC_{16})···	× × × × 0 0 0 0
Port P2 pin function control register	(AE_{16})···	× × × × 0 0 0 0
Clock control register 0	(BC_{16})···	0 0 0 1 0 1 1 1
Up-down register 1	($C4_{16}$)···	00_{16}
Timer A5 mode register	($D6_{16}$)···	00_{16}
Timer A6 mode register	($D7_{16}$)···	00_{16}
Timer A7 mode register	($D8_{16}$)···	00_{16}
Timer A8 mode register	($D9_{16}$)···	00_{16}
Timer A9 mode register	(DA_{16})···	00_{16}
Comparator function select register 0	(DC_{16})···	00_{16}
Comparator result register 0	(DE_{16})···	00_{16}

	Address	
Timer A5 interrupt control register	($F5_{16}$)···	× × × × 0 0 0 0
Timer A6 interrupt control register	($F6_{16}$)···	× × × × 0 0 0 0
Timer A7 interrupt control register	($F7_{16}$)···	× × × × 0 0 0 0
Timer A8 interrupt control register	($F8_{16}$)···	× × × × 0 0 0 0
Timer A9 interrupt control register	($F9_{16}$)···	× × × × 0 0 0 0
$\overline{\text{INT}_5}$ interrupt control register	(FD_{16})···	× × 0 0 0 0 0 0
$\overline{\text{INT}_6}$ interrupt control register	(FE_{16})···	× × 0 0 0 0 0 0
$\overline{\text{INT}_7}$ interrupt control register	(FF_{16})···	× × 0 0 0 0 0 0
Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
Program bank register PG		00_{16}
Program counter PC_H		Contents at address $FFFF_{16}$
Program counter PC_L		Contents at address $FFFE_{16}$
Direct page registers DPR0 to DPR3		0000_{16}
Data bank register DT		00_{16}
Stack pointer		FFF_{16}

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)

	Address	
Port P1 direction register	(05_{16})···	00_{16}
Port P2 direction register	(08_{16})···	00_{16}
Port P5 direction register	($0D_{16}$)···	0 0 0 × × × × ×
Port P6 direction register	(10_{16})···	× × 0 0 0 0 0 0
Port P7 direction register	(11_{16})···	× × × 0 0 0 0 0
A-D control register 0	($1E_{16}$)···	0 0 0 0 0 ? ? ?
A-D control register 1	($1F_{16}$)···	× 0 0 0 0 0 ? ?
UART 0 transmit/receive mode register	(30_{16})···	00_{16}
UART 1 transmit/receive mode register	(38_{16})···	00_{16}
UART 0 transmit/receive control register 0	(34_{16})···	0 0 0 0 1 0 0 0
UART 1 transmit/receive control register 0	($3C_{16}$)···	0 0 0 0 1 0 0 0
UART 0 transmit/receive control register 1	(35_{16})···	0 0 0 0 0 0 1 0
UART 1 transmit/receive control register 1	($3D_{16}$)···	0 0 0 0 0 0 1 0
Count start register 0	(40_{16})···	00_{16}
Count start register 1	(41_{16})···	× × × 0 0 0 0 0
One-shot start register 0	(42_{16})···	0 × × 0 0 0 0 0
One-shot start register 1	(43_{16})···	0 × × 0 0 0 0 0
Up-down register 0	(44_{16})···	00_{16}
Timer A clock frequency select register	(45_{16})···	× × × × × × 0 0
Timer A0 mode register	(56_{16})···	00_{16}
Timer A1 mode register	(57_{16})···	00_{16}
Timer A2 mode register	(58_{16})···	00_{16}
Timer A3 mode register	(59_{16})···	00_{16}
Timer A4 mode register	($5A_{16}$)···	00_{16}
Timer B0 mode register	($5B_{16}$)···	0 0 ? 0 0 0 0 0
Timer B1 mode register	($5C_{16}$)···	0 0 ? 0 0 0 0 0
Timer B2 mode register	($5D_{16}$)···	0 0 ? 0 0 0 0 0

	Address	
Processor mode register 0	($5E_{16}$)···	0 0 0 0 1 0 0 0
Processor mode register 1	($5F_{16}$)···	0 0 0 0 0 0 0 1
Watchdog timer	(60_{16})···	FFF_{16}
Watchdog timer frequency select register	(61_{16})···	0 0 × × × × × 0
Particular function select register 0	(62_{16})···	0 × 0 × × × 0 0
Particular function select register 1	(63_{16})···	× 0 × 0 0 0 **(Note 2)**
Debug control register 0	(66_{16})···	1 **(Note 2)**
Debug control register 1	(67_{16})···	0 0 0 × 0 0 0 **(Note 2)**
$\overline{\mathrm{INT}}_3$ interrupt control register	($6E_{16}$)···	× × × 0 0 0 0 0
$\overline{\mathrm{INT}}_4$ interrupt control register	($6F_{16}$)···	× × × 0 0 0 0 0
A-D conversion interrupt control register	(70_{16})···	× × × × ? 0 0 0
UART 0 transmit interrupt control register	(71_{16})···	× × × × 0 0 0 0
UART 0 receive interrupt control register	(72_{16})···	× × × × 0 0 0 0
UART 1 transmit interrupt control register	(73_{16})···	× × × × 0 0 0 0
UART 1 receive interrupt control register	(74_{16})···	× × × × 0 0 0 0
Timer A0 interrupt control register	(75_{16})···	× × × × 0 0 0 0
Timer A1 interrupt control register	(76_{16})···	× × × × 0 0 0 0
Timer A2 interrupt control register	(77_{16})···	× × × × 0 0 0 0
Timer A3 interrupt control register	(78_{16})···	× × × × 0 0 0 0
Timer A4 interrupt control register	(79_{16})···	× × × × 0 0 0 0
Timer B0 interrupt control register	($7A_{16}$)···	× × × × 0 0 0 0
Timer B1 interrupt control register	($7B_{16}$)···	× × × × 0 0 0 0
Timer B2 interrupt control register	($7C_{16}$)···	× × × × 0 0 0 0
D-A control register	(96_{16})···	× × × × × × 0 0
D-A register 0	(98_{16})···	00_{16}
D-A register 1	(99_{16})···	00_{16}

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)

OSCILLATION CIRCUIT

An oscillation circuit locates between pins X_{IN} and X_{OUT}, 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 X_{OUT} open. Also, in this case, when the external clock input select bit (bit 1 of the particular function select register 0; See Figure 86.) is set to "1", the oscillation circuit stops it's 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 X_{IN} and generates a multiplied clock. When using the PLL circuit, be sure to connect pin V_{CONT} with an external filter circuit. (See Figure 83.) When not using the PLL circuit, be sure to leave pin V_{CONT} open.

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

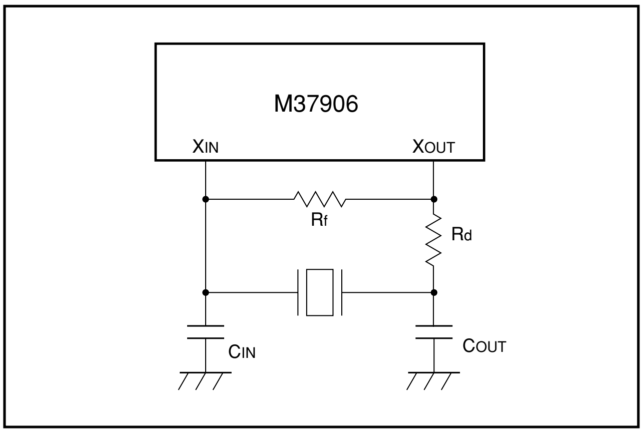


Fig. 81 Circuit example with external ceramic resonator or quartz-crystal oscillator

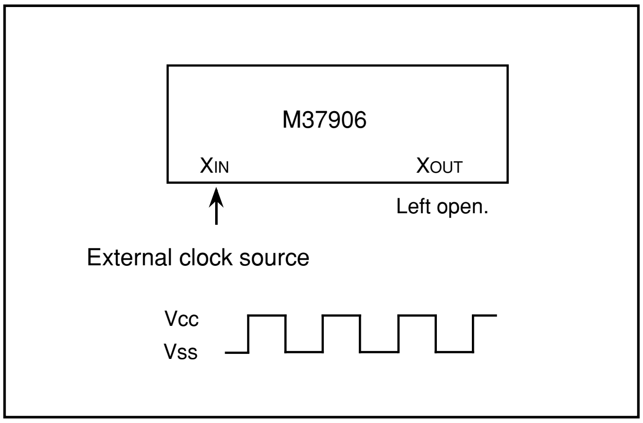
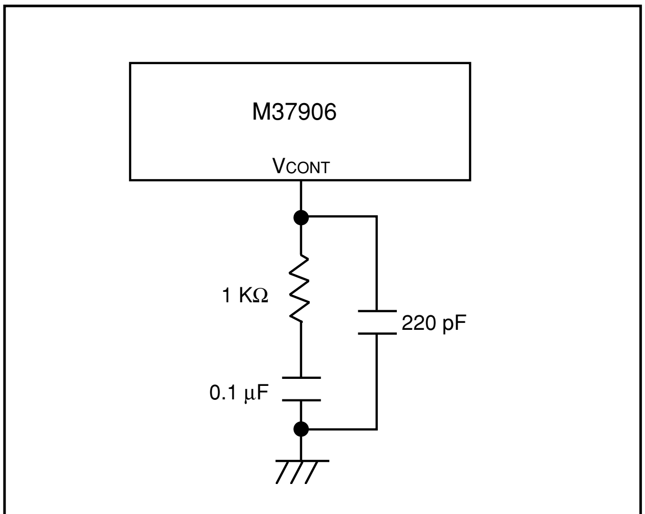


Fig. 82 Circuit example with external clock source



Note: Make the wiring length as short as possible, and shield it with the GND line which surrounds this circuit. Also, for the clock supply to pin X_{IN}, see Figures 81 and 82.

Fig. 83 Circuit example of connection with pin V_{CONT} when PLL circuit used

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 BC_{16}), particular function select register 0 (address 62_{16}) 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 ϕ_{CPU}, ϕ_{BIU}, f_1 to f_{4096}, Wf_{32}, Wf_{512}) are made from system clock f_{sys}. System clock f_{sys} can be selected between fX_{IN} (in other words, a clock input from pin X_{IN}) and f_{PLL} (in other words, an output clock generated by the PLL circuit).

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 becomes valid, and the PLL circuit will operate. At reset, the PLL circuit operation enable bit becomes "1". (In this case, the PLL circuit operates 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}/fX_{IN}. 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" (× 2). The change of the multiplication ratio must be performed while input clock fX_{IN} is set as system clock. (In this case, bit 5 of the clock control register 0 = "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 fX_{IN} 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	——	——	fX_{IN}	$f(X_{IN})$
1	1	01 (× 2)	f_{PLL}	$f(X_{IN})$ × 2
		10 (× 3)	f_{PLL}	$f(X_{IN})$ × 3
		11 (× 4)	f_{PLL}	$f(X_{IN})$ × 4

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

Table 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 = "01_2", be sure that system clock f_{sys} does not exceed 10 MHz.

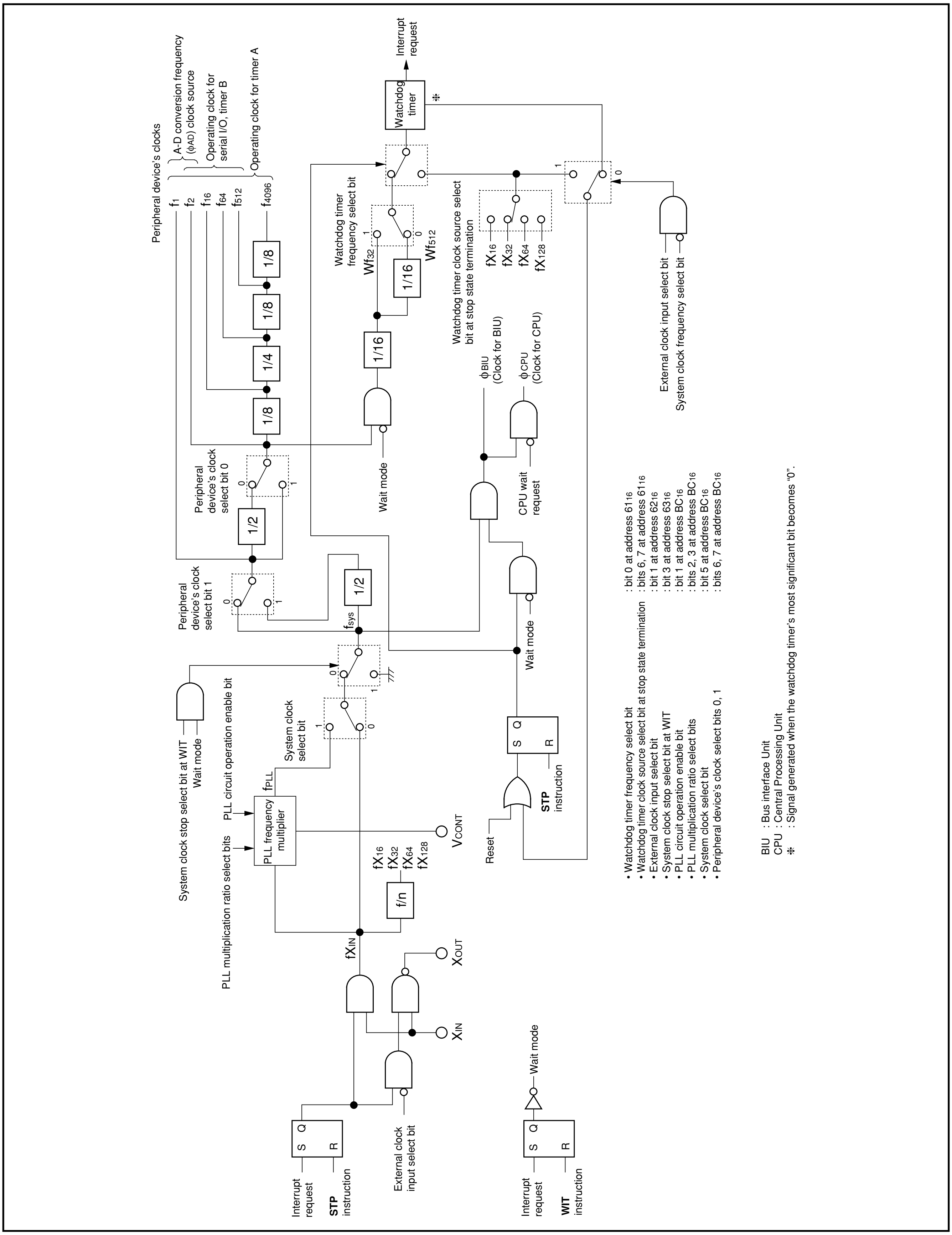
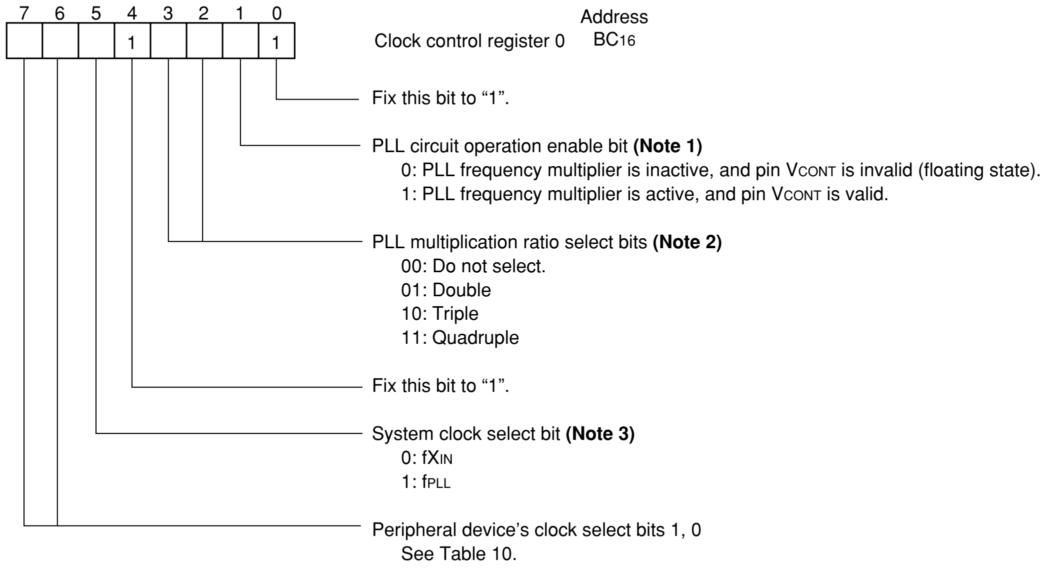

Peripheral device's clocks
f1
f2
f16
f64
f512
f4096
A-D conversion frequency (φAD) clock source
Operating clock for serial I/O, timer B
Operating clock for timer A
Interrupt request
Watchdog timer
Watchdog timer frequency select bit
Wf32
Wf512
1/16
1/16
1/8
1/8
1/4
1/8
1/2
1/2
Watchdog timer clock source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

Fig. 84 Block diagram of clock generating circuit



Clock control register 0 — Address BC$_{16}$

Fix this bit to “1”.

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

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

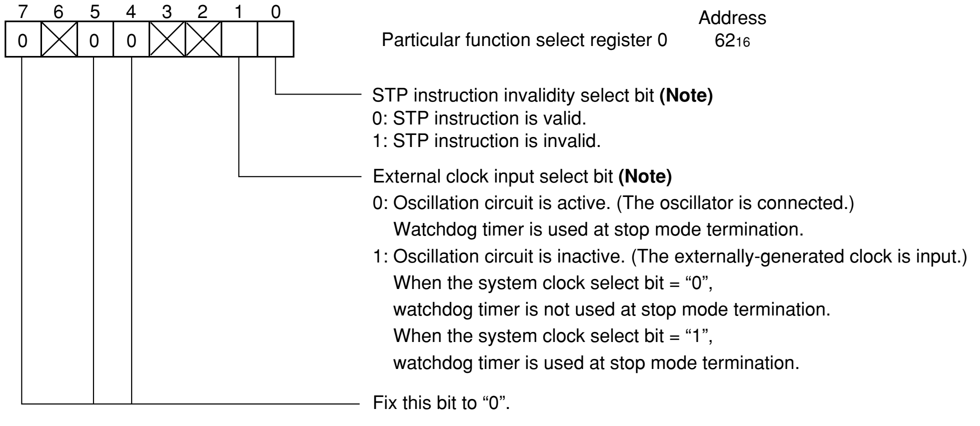
Fix this bit to “1”.

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

Peripheral device’s clock select bits 1, 0
See Table 10.

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

Fig. 85 Bit configuration of clock control register 0



Particular function select register 0 — Address 62$_{16}$

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

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

Fix this bit to “0”.

Note: Writing to these bits requires the following procedure:
- Write “55$_{16}$” to this register. (The bit status does not change only by this writing.)
- Succeedingly, write “0” or “1” to each bit.

Also, use the **MOVM (MOVMB)** instruction or **STA (STAB, STAD)** instruction

Fig. 86 Bit configuration of particular function select register 0

STANDBY FUNCTION

The standby function provides the stop (hereafter called STP) and the wait (hereafter called WIT) mode. These modes are used to save the power dissipation of the system, by 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 f_{XIN}, system clock f_{sys}, ϕ_{BIU}, ϕ_{CPU}, and peripheral devices' clocks f_1 to f_{4096}, Wf_{32} and Wf_{512} with the "L" state, and divide clocks fX_{16} to fX_{128} with the "H" state. In the watchdog timer, "FFF_{16}" is automatically set. As shown in Figure 84, any one of divide clocks fX_{16} to fX_{128}, which is selected by the watchdog timer clock source select bits at STP termination (bits 6 and 7 of the watchdog timer frequency select register), becomes the watchdog timer's clock source.

In the STP mode, the A-D converter and watchdog timer, which uses peripheral devices' clocks f_1 to f_{4096}, Wf_{32} and Wf_{512}, are stopped. At this time, timers A and B 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 f_{XIN}, system clock f_{sys}, and peripheral devices' clocks f_1 to f_{4096}, Wf_{32} and Wf_{512} are also supplied.

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

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

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

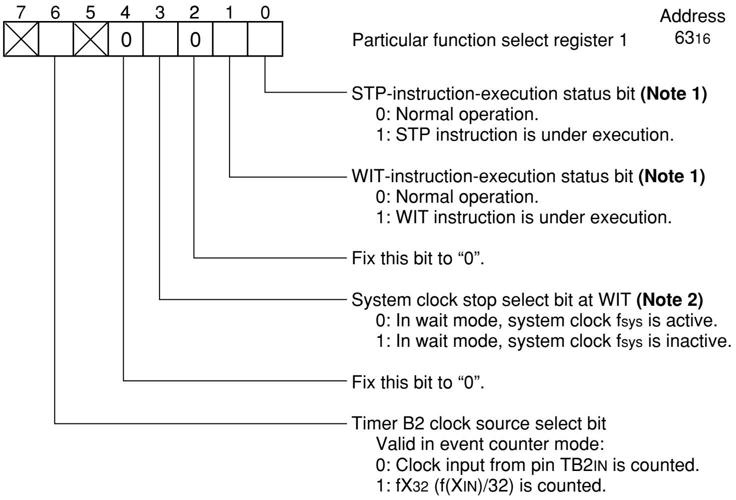
(2) The supply of ϕ_{BIU} and ϕ_{CPU} is started immediately after the operation restart of the oscillation circuit and PLL circuit.

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	f_{sys}, ϕ_1, f_1 to f_{4096}	Wf_{32}, Wf_{512}	ϕ_{BIU}, ϕ_{CPU}	Peripheral devices using f_1 to f_{4096}, Wf_{32}, Wf_{512}
STP	—	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 X_{IN} is allowed.
2: When the PLL operation enable bit = "0", the PLL circuit is inactive.

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


7 6 5 4 3 2 1 0
0
0
Address
63₁₆
Particular function select register 1
STP-instruction-execution status bit (Note 1)
0: Normal operation.
1: STP instruction is under execution.
WIT-instruction-execution status bit (Note 1)
0: Normal operation.
1: WIT instruction is under execution.
Fix this bit to "0".
System clock stop select bit at WIT (Note 2)
0: In wait mode, system clock fsys is active.
1: In wait mode, system clock fsys is inactive.
Fix this bit to "0".
Timer B2 clock source select bit
Valid in event counter mode:
0: Clock input from pin TB2IN is counted.
1: fX32 (f(XIN)/32) is counted.

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

Fig. 87 Bit configuration of particular function select register 1

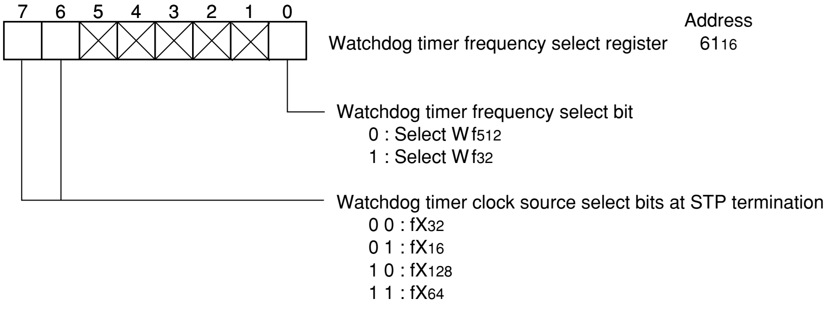

7 6 5 4 3 2 1 0
Address
61₁₆
Watchdog timer frequency select register
Watchdog timer frequency select bit
0 : Select Wf512
1 : Select Wf32
Watchdog timer clock source select bits at STP termination
0 0 : fX32
0 1 : fX16
1 0 : fX128
1 1 : fX64

Fig. 88 Bit configuration of watchdog timer frequency select register

WIT mode

When the WIT instruction is executed with the system clock stop select bit at WIT (bit 3 of the particular function select register 1 in Figure 87) being “0”, ϕ_{BIU}, ϕ_{CPU}, and divide clocks Wf_{32} and Wf_{512} are inactive with the “L“ state. However, the oscillation circuit, PLL circuit, input clock fX_{IN}, system clock f_{sys}, ϕ_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 fX_{IN} are active, while system clock f_{sys}, ϕ_{BIU}, ϕ_{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 ϕ_{BIU} and ϕ_{CPU} will restart. Since the oscillation circuit, PLL circuit, and clock input fX_{IN} are active in the WIT mode, an interrupt processing can be executed just after the WIT mode termination.

POWER SAVING FUNCTION

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

(1) 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 X_{IN}, the power dissipation can be saved if both of the following conditions are met:

- the external clock input select bit (bit 1 of the particular function select register 0) = "1".
- the oscillation driver circuit between pins X_{IN} and X_{OUT} is inactive.

At this time, the output level at pin X_{OUT} is fixed to "H". When not using f_{PLL}, also, the supply of ϕ_{BIU} and ϕ_{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 V_{REF}

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

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.

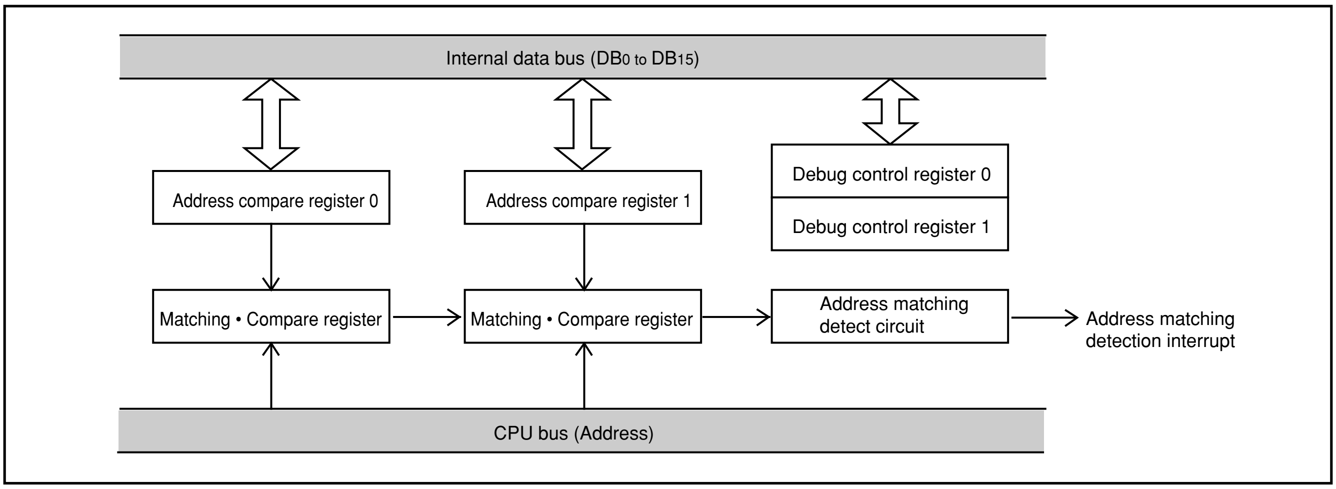
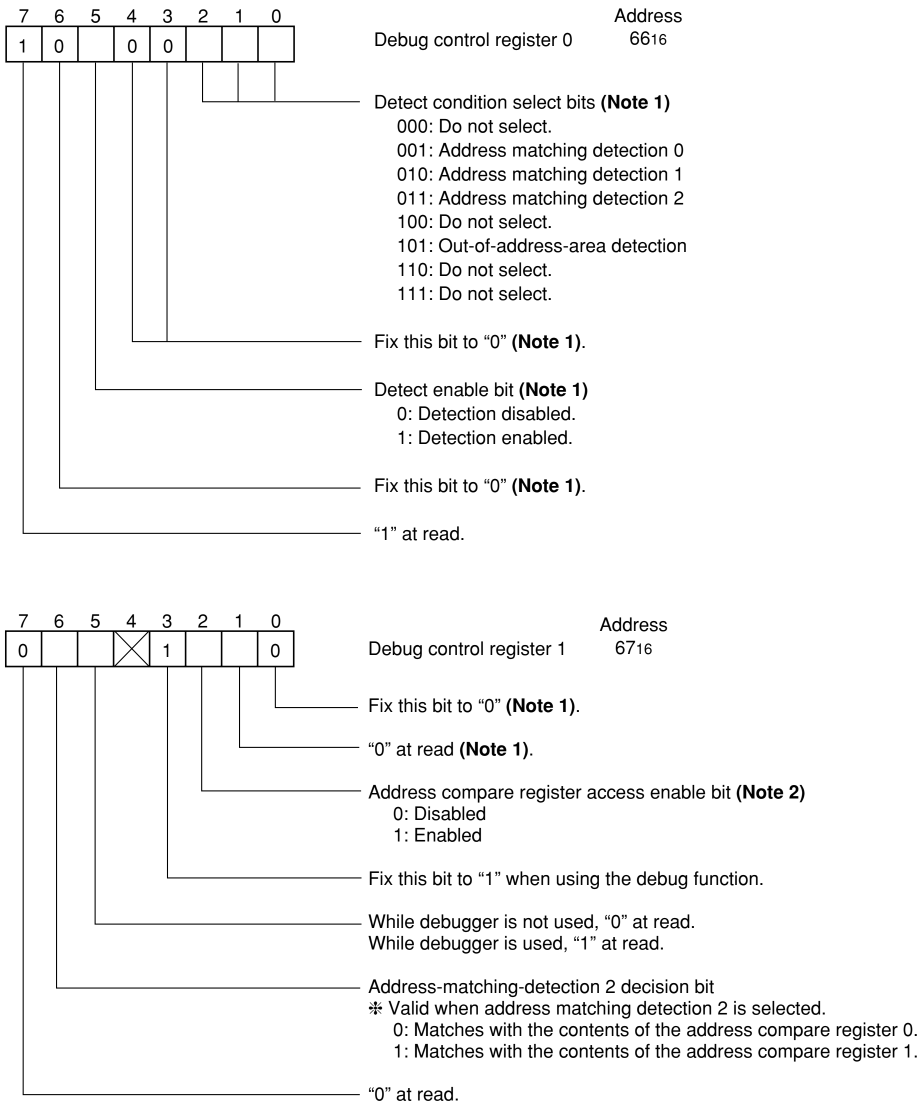

Internal data bus (DB0 to DB15)
Address compare register 0
Address compare register 1
Debug control register 0
Debug control register 1
Matching • Compare register
Matching • Compare register
Address matching detect circuit
Address matching detection interrupt
CPU bus (Address)

Fig. 89 Block diagram of debug function



Debug control register 0 — Address 66_{16}

Detect condition select bits **(Note 1)**
- 000: Do not select.
- 001: Address matching detection 0
- 010: Address matching detection 1
- 011: Address matching detection 2
- 100: Do not select.
- 101: Out-of-address-area detection
- 110: Do not select.
- 111: Do not select.

Fix this bit to “0” **(Note 1)**.

Detect enable bit **(Note 1)**
- 0: Detection disabled.
- 1: Detection enabled.

Fix this bit to “0” **(Note 1)**.

“1” at read.

Debug control register 1 — Address 67_{16}

Fix this bit to “0” **(Note 1)**.

“0” at read **(Note 1)**.

Address compare register access enable bit **(Note 2)**
- 0: Disabled
- 1: Enabled

Fix this bit to “1” when using the debug function.

While debugger is not used, “0” at read.
While debugger is used, “1” at read.

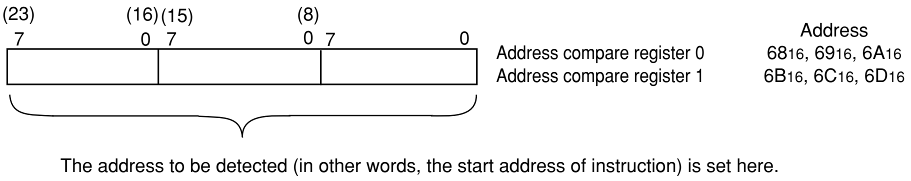
Address-matching-detection 2 decision bit
✻ Valid when address matching detection 2 is selected.
- 0: Matches with the contents of the address compare register 0.
- 1: Matches with the contents of the address compare register 1.

“0” at read.

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

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

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



	Address
Address compare register 0	68_{16}, 69_{16}, $6A_{16}$
Address compare register 1	$6B_{16}$, $6C_{16}$, $6D_{16}$

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

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

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Ratings	Unit
V_{CC}	Power source voltage	–0.3 to 6.5	V
AV_{CC}	Analog power source voltage	–0.3 to 6.5	V
V_I	Input voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$, $\overline{P6OUT_{CUT}}$, V_{CONT}, V_{REF}, X_{IN}, $\overline{RESET}$, BYTE, MD0, MD1	–0.3 to V_{CC}+0.3	V
V_O	Output voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$, X_{OUT}	–0.3 to V_{CC}+0.3	V
P_d	Power dissipation	300	mW
T_{opr}	Operating ambient temperature	–20 to 85	°C
T_{stg}	Storage temperature	–40 to 150	°C

RECOMMENDED OPERATING CONDITIONS (Vcc = 5 V, Ta = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Power source voltage	4.5	5.0	5.5	V
AV_{CC}	Analog power source voltage		V_{CC}		V
V_{SS}	Power source voltage		0		V
AV_{SS}	Analog power source voltage		0		V
V_{IH}	High-level input voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{RESET}$, MD0, MD1	0.8 Vcc		Vcc	V
V_{IL}	Low-level input voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{RESET}$, MD0, MD1	0		0.2 V_{CC}	V
$I_{OH(peak)}$	High-level peak output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$			–10	mA
$I_{OH(avg)}$	High-level average output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$			–5	mA
$I_{OL(peak)}$	Low-level peak output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P7_0$–$P7_4$			10	mA
$I_{OL(peak)}$	Low-level peak output current $P6_0$–$P6_5$			20	mA
$I_{OL(avg)}$	Low-level average output current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P7_0$–$P7_4$			5	mA
$I_{OL(avg)}$	Low-level average output current $P6_0$–$P6_5$			15	mA
$f(X_{IN})$	External clock input frequency **(Note 1)**			20	MHz
$f(f_{sys})$	System clock frequency			20	MHz

Notes 1: When using the PLL frequency multiplier, be sure that $f(f_{sys})$ = 20 MHz or less.
2: The average output current is the average value of an interval of 100 ms.
3: The sum of $I_{OL(peak)}$ must be 110 mA or less, the sum of $I_{OH(peak)}$ must be 80 mA or less.

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

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
V_{OH}	High-level output voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_5$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$	I_{OH} = –10 mA		3			V
V_{OL}	Low-level output voltage $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_5$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$	I_{OL} = 10 mA				2	V
$V_{T+}-V_{T-}$	Hysteresis $TA0_{IN}$–$TA2_{IN}$, $TA4_{IN}$, $TA9_{IN}$, $TA0_{OUT}$–$TA2_{OUT}$, $TA4_{OUT}$, $TA9_{OUT}$, $TB0_{IN}$–$TB2_{IN}$, $\overline{INT_3}$–$\overline{INT_7}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1, RxD_0, RxD_1, RTP_{TRG0}, $\overline{P6OUT_{CUT}}$			0.4		1	V
$V_{T+}-V_{T-}$	Hysteresis $\overline{RESET}$			0.5		1.5	V
$V_{T+}-V_{T-}$	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{RESET}$, MD0, MD1	V_I = 5.0 V				5	μA
I_{IL}	Low-level input current $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P5_0$–$P5_7$, $P6_0$–$P6_5$, $P7_0$–$P7_4$, $\overline{P6OUT_{CUT}}$, X_{IN}, $\overline{RESET}$, MD0, MD1	V_I = 0 V				–5	μA
V_{RAM}	RAM hold voltage	When clock is inactive.		2			V
I_{CC}	Power source current	Output-only pins are open, and the other pins are connected to Vss or Vcc. An external square-waveform clock is input. (Pin X_{OUT} is open.) The PLL frequency multiplier is inactive.	f(f_{sys}) = 20 MHz. CPU is active.		25	50	mA
			Ta = 25 °C when clock is inactive.			1	μA
			Ta = 85 °C when clock is inactive.			20	

A-D CONVERTER CHARACTERISTICS

(V_{CC} = AV_{CC} = 5 V ± 0.5 V, V_{SS} = AV_{SS} = 0 V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits			Unit
				Min.	Typ.	Max.	
————	Resolution	V_{REF} = V_{CC}	A-D converter			10	Bits
			Comparator			$\frac{1}{256}V_{REF}$	V
————	Absolute accuracy	V_{REF} = V_{CC}	10-bit resolution mode			± 3	LSB
			8-bit resolution mode			± 2	LSB
			Comparater			± 40	mV
R_{LADDER}	Ladder resistance	V_{REF} = V_{CC}		5			kΩ
t_{CONV}	Conversion time	$f(f_{sys}) \leq 20$ MHz	10-bit resolution mode	5.9			µs
			8-bit resolution mode	2.45 **(Note)**			
			Comparater	0.7 **(Note)**			
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Note: This is applied when A-D conversion freguency (ϕ_{AD}) = f_1 (ϕ).

D-A CONVERTER CHARACTERISTICS

(V_{CC} = 5 V, V_{SS} = AV_{SS} = 0 V, V_{REF} = 5 V, T_a = –20 to 85 °C, unless otherwise noted)

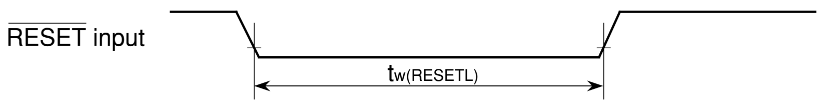
Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
——	Resolution				8	Bits
——	Absolute accuracy				± 1.0	%
t_{su}	Set time				3	µs
R_O	Output resistance		2	3.5	4.5	kΩ
I_{VREF}	Reference power source input current	**(Note)**			3.2	mA

Note: The test conditions are as follows:
- One D-A converter is used.
- The D-A register value of the unused D-A converter is "00_{16}."
- The reference power source input current for the ladder resistance of the A-D converter is excluded.

RESET INPUT

Reset input timing requirements (V_{CC} = 5 V ± 0.5 V, V_{SS} = 0V, T_a = –20 to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{w(RESETL)}$	$\overline{\text{RESET}}$ input low-level pulse width	10			µs



PERIPHERAL DEVICE INPUT/OUTPUT TIMING

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

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

Timer A input (Count input in event counter mode)

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

Timer A input (Gating input in timer mode)

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

Note : The TAi_{IN} input cycle time requires 4 or more cycles of a count source. The TAi_{IN} input high-level pulse width and the TAi_{IN} input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at $f(f_{sys}) \leq 20$ MHz.

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

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

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

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

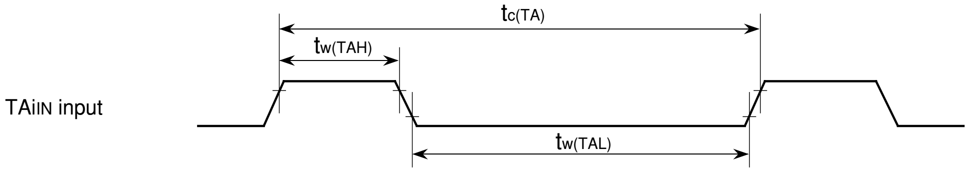
Timer A input (Up-down input and Count input in event counter mode)

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

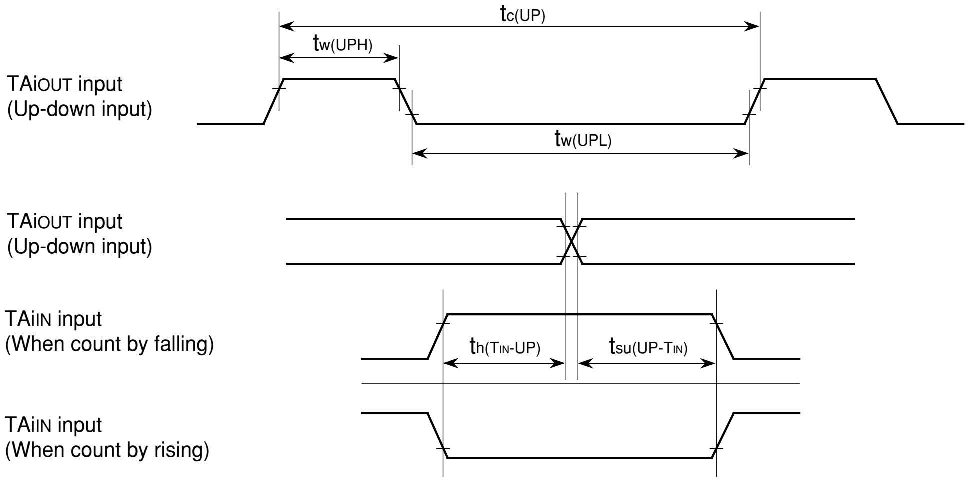
Timer A input (Two-phase pulse input in event counter mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{c(TA)}$	TAjIN input cycle time	800		ns
$t_{su(TAjIN-TAjOUT)}$	TAjIN input setup time	200		ns
$t_{su(TAjOUT-TAjIN)}$	TAjOUT input setup time	200		ns

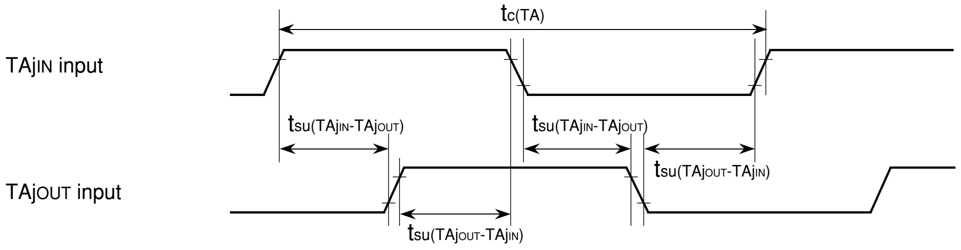
- Gating input in timer mode
- Count input in event counter mode
- External trigger input in one-shot pulse mode
- External trigger input in pulse width modulation mode


$t_{c(TA)}$
$t_{w(TAH)}$
TAiIN input
$t_{w(TAL)}$

- Up-down and Count input in event counter mode


$t_{c(UP)}$
$t_{w(UPH)}$
TAiOUT input
(Up-down input)
$t_{w(UPL)}$
TAiOUT input
(Up-down input)
TAiIN input
(When count by falling)
$t_{h(TIN-UP)}$
$t_{su(UP-TIN)}$
TAiIN input
(When count by rising)

- Two-phase pulse input in event counter mode


$t_{c(TA)}$
TAjIN input
$t_{su(TAjIN-TAjOUT)}$
$t_{su(TAjIN-TAjOUT)}$
$t_{su(TAjOUT-TAjIN)}$
TAjOUT input
$t_{su(TAjOUT-TAjIN)}$

Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V

Timer B input (Count input in event counter mode)

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

Timer B input (Pulse period measurement mode)

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

Note: The TB$_{i\mathrm{IN}}$ input cycle time requires 4 or more cycles of a count source. The TB$_{i\mathrm{IN}}$ input high-level pulse width and the TB$_{i\mathrm{IN}}$ input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at $f(f_{sys}) \leq 20$ MHz.

Timer B input (Pulse width measurement mode)

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

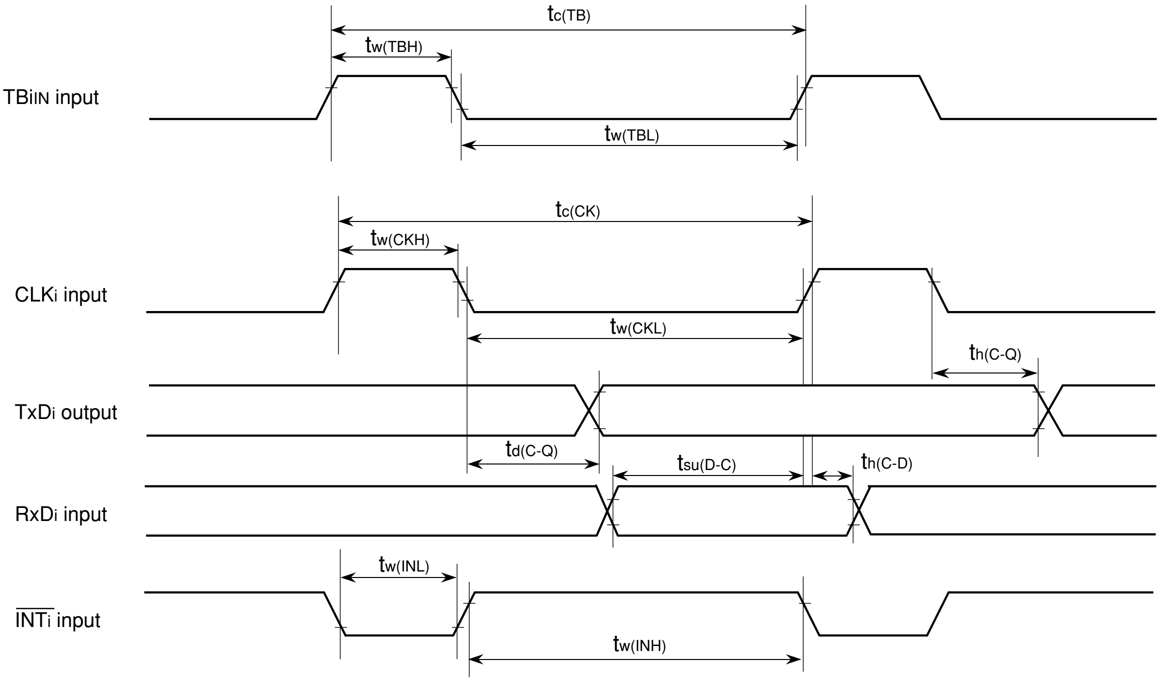
Note: The TB$_{i\mathrm{IN}}$ input cycle time requires 4 or more cycles of a count source. The TB$_{i\mathrm{IN}}$ input high-level pulse width and the TB$_{i\mathrm{IN}}$ input low-level pulse width respectively require 2 or more cycles of a count source. The limits in this table are applied when the count source = f_2 at $f(f_{sys}) \leq 20$ MHz.

Serial I/O

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

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

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


$t_{c(TB)}$
$t_{w(TBH)}$
TB_{iIN} input
$t_{w(TBL)}$
$t_{c(CK)}$
$t_{w(CKH)}$
CLK_i input
$t_{w(CKL)}$
$t_{h(C-Q)}$
TxD_i output
$t_{d(C-Q)}$
$t_{su(D-C)}$
$t_{h(C-D)}$
RxD_i input
$t_{w(INL)}$
$\overline{\text{INT}}_i$ input
$t_{w(INH)}$

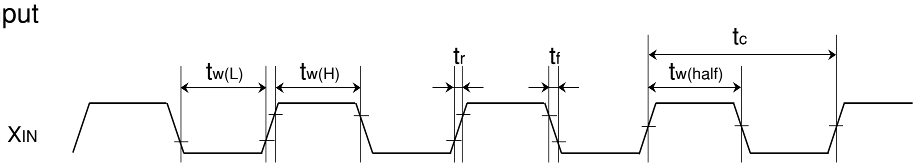
Test conditions
- V_{CC} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V
- Output timing voltage : V_{OL} = 0.8 V, V_{OH} = 2.0 V, C_L = 50 pF

External clock input

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

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_c	External clock input cycle time	50		ns
$t_{w(half)}$	External clock input pulse width with half input-voltage	0.45 tc	0.55 tc	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



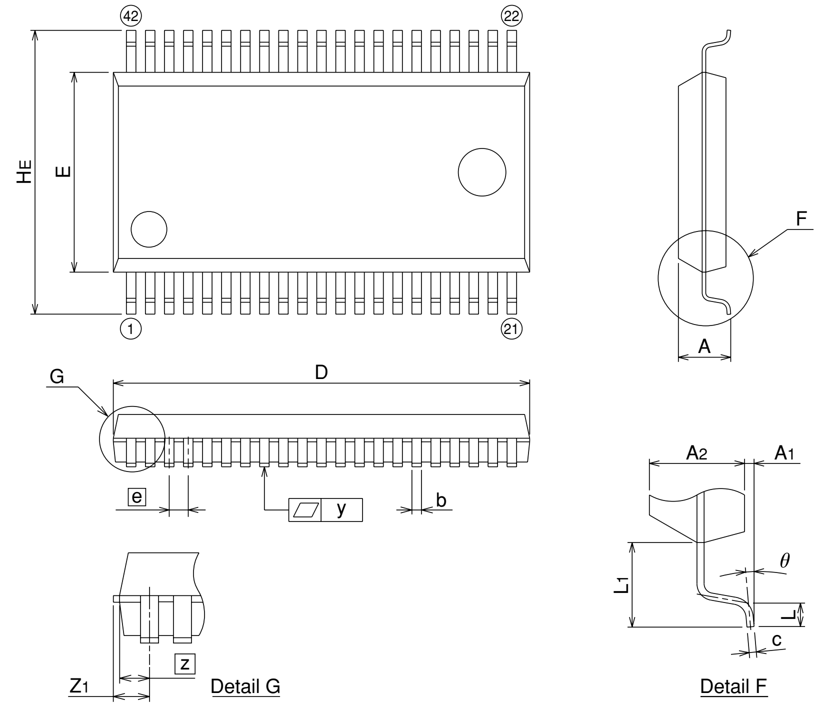
Test conditions
- V_{cc} = 5 V ± 0.5 V, Ta = –20 to 85 °C
- Input timing voltage : V_{IL} = 1.0 V, V_{IH} = 4.0 V ($t_{w(H)}$, $t_{w(L)}$, t_r, t_f)
- Output timing voltage : 2.5 V (t_c, $t_{w(half)}$)

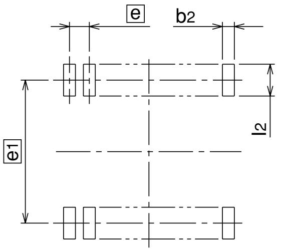
PACKAGE OUTLINE

42P2R-E

Plastic 42pin 450mil SSOP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SSOP42-P-450-0.80	–	0.63	Alloy 42/Cu Alloy


42
22
HE
E
1
21
F
A
G
D
e
y
b
A2
A1
L1
θ
L
c
z
Z1
Detail G
Detail F



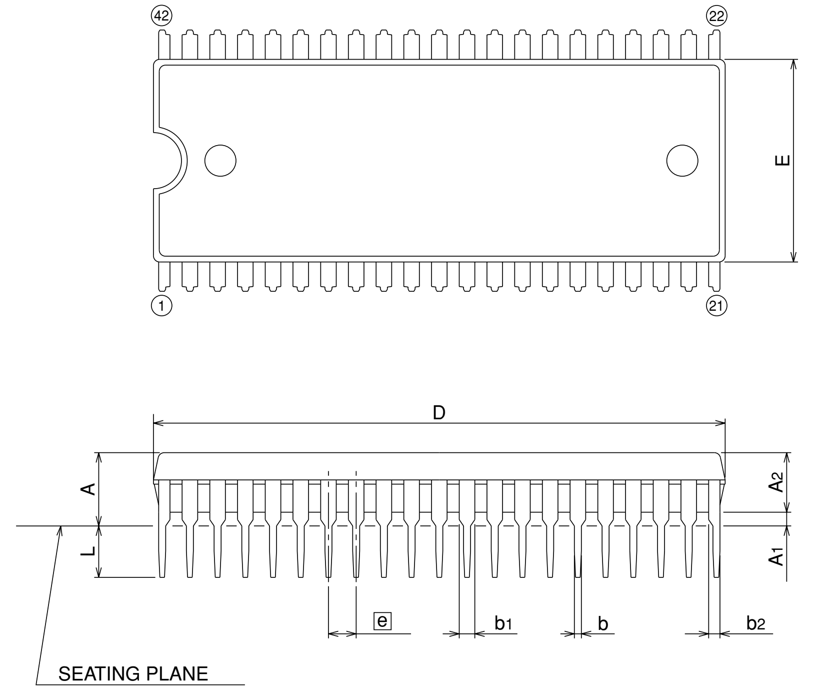
Recommended Mount Pad

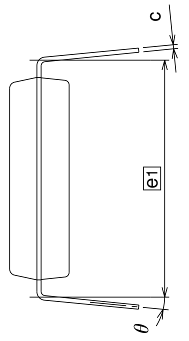
Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	2.4
A1	0.05	–	–
A2	–	2.0	–
b	0.35	0.4	0.5
c	0.13	0.15	0.2
D	17.3	17.5	17.7
E	8.2	8.4	8.6
e	–	0.8	–
HE	11.63	11.93	12.23
L	0.3	0.5	0.7
L1	–	1.765	–
z	–	0.75	–
Z1	–	–	0.9
y	–	–	0.15
θ	0°	–	10°
b2	–	0.5	–
e1	–	11.43	–
I2	1.27	–	–

42P4B

Plastic 42pin 600mil SDIP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
SDIP42-P-600-1.78	–	4.1	Alloy 42/Cu Alloy


42
22
E
1
21
D
A
L
A2
A1
e
b1
b
b2
SEATING PLANE



Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	5.5
A1	0.51	–	–
A2	–	3.8	–
b	0.35	0.45	0.55
b1	0.9	1.0	1.3
b2	0.63	0.73	1.03
c	0.22	0.27	0.34
D	36.5	36.7	36.9
E	12.85	13.0	13.15
e	–	1.778	–
e1	–	15.24	–
L	3.0	–	–
θ	0°	–	15°

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: 2-2-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, 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.

© 2001 MITSUBISHI ELECTRIC CORP.
New publication, effective Jul. 2001.
Specifications subject to change without notice.

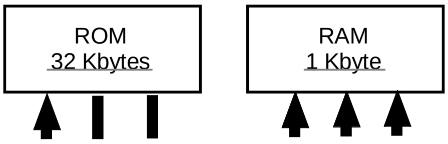
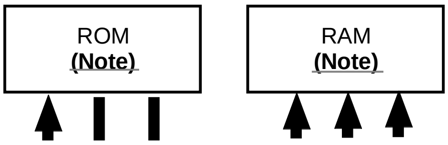
MITSUBISHI
ELECTRIC

Revision History	M37906MxC-XXXFP/SP Datasheet

Rev. No.	Revision Description	Rev. date
1.0	First Edition. The following are released. • DESCRIPTION • Timer function for motor control • Pulse output port mode	000331
2.0	(1) The following are revised: • Figure 42 in page 36; the description for bits 6 and 7 of the timer Ax (x = 0 to 3) mode register <Error> (See the table below.) <Correction> (See Table 7.) • Figure 50 in page 43; the description for bits 6 and 7 of the timer Ax (x = 0, 3) mode register <Error> (See Fig. 42.) <Correction> (See Table 7.) (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 **Corrections and Supplementary Explanation for “M37906MxC Datasheet (Rev.A)”**.	000807
4.0	Refer to **“Corrections and Supplementary Explanation for M37906MxC-XXXFP/SP Datasheet (Rev.B)”**. **Note :** ★ represents the new information added in Rev.4.0.	010706

Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.1

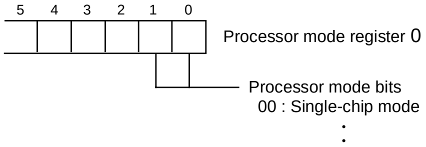
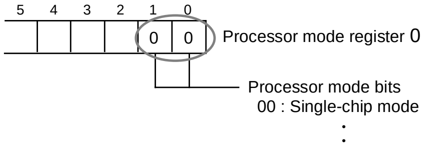
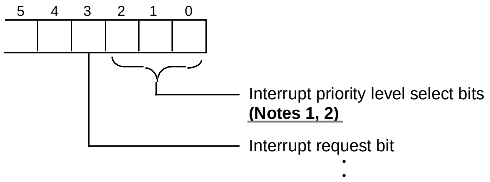
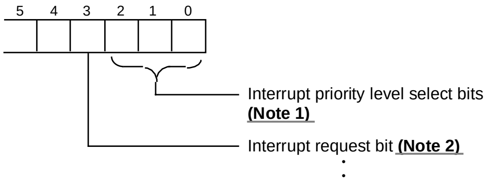
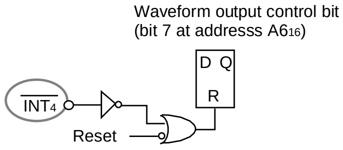
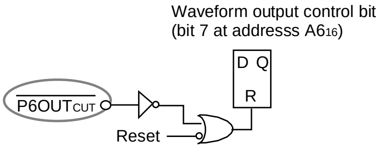
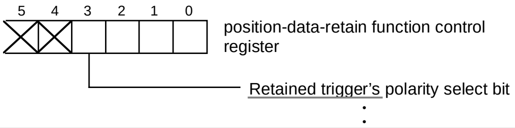
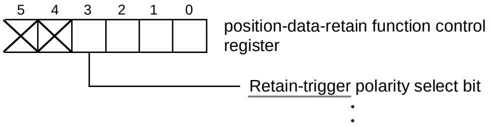
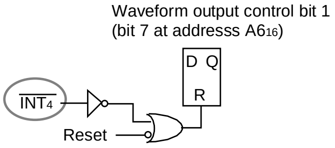
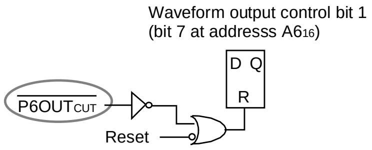
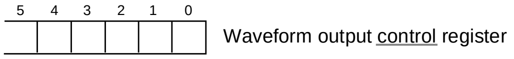
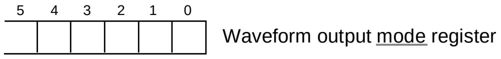
<table>
<tr><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td>All pages,
Header</td><td>M37906M4C-XXXFP, M37906M4C-XXXSP</td><td>M37906M4C-XXXFP, M37906M4C-XXXSP
M37906M6C-XXXFP, M37906M6C-XXXSP
M37906M8C-XXXFP, M37906M8C-XXXSP</td></tr>
<tr><td rowspan="4">Page 1,
DISTINCTIVE FEATURES;
Memory

★</td><td>ROM..32 Kbytes
RAM..1 Kbyte</td><td>[M37906M4C-XXXFP, M37906M4C-XXXSP]
ROM..32 Kbytes
RAM..1 Kbyte</td></tr>
<tr><td>—</td><td>[M37906M6C-XXXFP, M37906M6C-XXXSP]
ROM..48 Kbytes
RAM..3 Kbytes</td></tr>
<tr><td>—</td><td>[M37906M8C-XXXFP, M37906M8C-XXXSP]
ROM..60 Kbytes
RAM..3 Kbytes</td></tr>
<tr><td>• 8-bit A-D converter...........................2-channel outputs</td><td>• 8-bit D-A converter...........................2-channel outputs</td></tr>
<tr><td rowspan="3">Page 1,
PIN CONFIGURATION
of outline 42P2R-E</td><td>M37906M4C-XXXFP PIN CONFIGURATION (TOP VIEW)</td><td>M37906MxC-XXXFP PIN CONFIGURATION (TOP VIEW)</td></tr>
<tr><td>(Type) M37906M4C-XXXFP</td><td>(Type) M37906M4C-XXXFP
M37906M6C-XXXFP
M37906M8C-XXXFP</td></tr>
<tr><td>$\overline{INT_4}$</td><td>$\overline{P6OUT_{CUT}}/\overline{INT_4}$</td></tr>
<tr><td rowspan="3">Page 2,
PIN CONFIGURATION
of outline 42P4B</td><td>M37906M4C-XXXSP PIN CONFIGURATION (TOP VIEW)</td><td>M37906MxC-XXXFP PIN CONFIGURATION (TOP VIEW)</td></tr>
<tr><td>(Type) M37906M4C-XXXSP</td><td>(Type) M37906M4C-XXXSP
M37906M6C-XXXSP
M37906M8C-XXXSP</td></tr>
<tr><td>$\overline{INT_4}$</td><td>$\overline{P6OUT_{CUT}}/\overline{INT_4}$</td></tr>
<tr><td rowspan="4">★ Pages 1, 2
PIN CONFIGURATION of outlines
42P2R-E and
42P4B</td><td>$P7_4/AN_4/DA_1/\overline{INT_3}/RTP_{TRG0}$</td><td>(Note) $P7_4/AN_4/DA_1/\overline{INT_3}/RTP_{TRG0}$</td></tr>
<tr><td>$P5_7/\overline{INT_7}/TB2_{IN}/IDU$
$P5_6/\overline{INT_6}/TB1_{IN}/IDV$
$P5_5/\overline{INT_5}/TB0_{IN}/IDW$</td><td>(Note) $P5_7/\overline{INT_7}/TB2_{IN}/IDU$
$P5_6/\overline{INT_6}/TB1_{IN}/IDV$
$P5_5/\overline{INT_5}/TB0_{IN}/IDW$</td></tr>
<tr><td>$P2_4(/TB0_{IN})$
$P2_5(/TB1_{IN})$
$P2_6(/TB2_{IN})$
$P2_6(/\overline{INT_3}/RTP_{TRG0})$</td><td>$P2_4(/TB0_{IN})$
$P2_5(/TB1_{IN})$
$P2_6(/TB2_{IN})$
$P2_6(/\overline{INT_3}/RTP_{TRG0})$
(Note)</td></tr>
<tr><td>( ) : Shift function</td><td>**Note:** Allocation of these pins can be switched by software.</td></tr>
<tr><td rowspan="3">Page 3,
BLOCK
DIAGRAM</td><td>$\overline{INT_4}$</td><td>$\overline{P6OUT_{CUT}}$</td></tr>
<tr><td> </td><td> </td></tr>
<tr><td>—</td><td>**Note:**
<table>
<tr><th></th><th>ROM</th><th>RAM</th></tr>
<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>
</table></td></tr>
<tr><td>Page 4,
FUNCTIONS</td><td>
<table>
<tr><td rowspan="2">Memory size</td><td>ROM</td><td>32 Kbytes</td></tr>
<tr><td>RAM</td><td>1 Kbyte</td></tr>
</table></td><td>
<table>
<tr><td rowspan="2">Memory size</td><td>ROM</td><td>**(Note 1)**</td></tr>
<tr><td>RAM</td><td>**(Note 1)**</td></tr>
</table></td></tr>
</table>

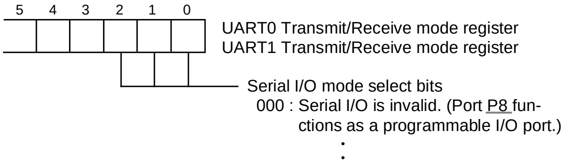
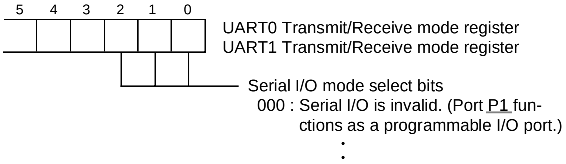
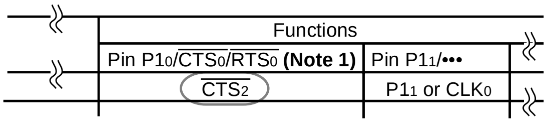
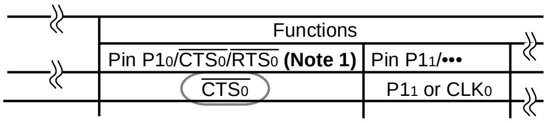
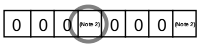
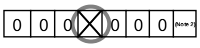
Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.2

<table>
<tr><th>Page</th><th>Error</th><th>Correction</th></tr>
<tr><td>Page 4, FUNCTIONS</td>
<td><table><tr><td>Interrupts</td><td>Maskable interrupts</td><td>5 external types, 18 internal types. •••••</td></tr><tr><td></td><td>Non-maskable interrupts</td><td>3 internal types.</td></tr></table></td>
<td><table><tr><td>Interrupts</td><td>Maskable interrupts</td><td>5 external sources, 18 internal sources. •••••</td></tr><tr><td></td><td>Non-maskable interrupts</td><td>3 internal sources.</td></tr></table></td></tr>
<tr><td>★</td>
<td><table><tr><td>Clock generating circuit</td><td>Built-in (externally connected to a ceramic resonator or quartz crystal resonator).</td></tr></table></td>
<td><table><tr><td>Clock generating circuit</td><td>Built-in (externally connected to a ceramic resonator or quartz-crystal oscillator).</td></tr></table></td></tr>
<tr><td>★</td>
<td><table><tr><td>PLL frequency multiplier</td><td>The following multiplication methods are avaiable : doble, triple, and quadruple.</td></tr></table></td>
<td><table><tr><td>PLL frequency multiplier</td><td>The following multiplication ratios are avaiable : × 2, × 3, × 4.</td></tr></table></td></tr>
<tr><td>★</td>
<td><table><tr><td>Power dissipation</td><td>125 mW (at $f(X_{IN})$ = 20 MHz, typ.,PLL frequency multiplier stopped.)</td></tr></table></td>
<td><table><tr><td>Power dissipation</td><td>125 mW (at $f(f_{sys})$ = 20 MHz, typ., the PLL frequency multiplier is inactive.)</td></tr></table></td></tr>
<tr><td></td>
<td><table><tr><td>Package</td><td>(Note)</td></tr></table></td>
<td><table><tr><td>Package</td><td>(Note 2)</td></tr></table></td></tr>
<tr><td></td>
<td>―――</td>
<td>Note 1:<table><tr><td>ROM</td><td>M37906M4C-XXXFP, M37906M4C-XXXSP</td><td>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>RAM</td><td>M37906M4C-XXXFP, M37906M4C-XXXSP</td><td>1 Kbyte</td></tr><tr><td></td><td>M37906M6C-XXXFP, M37906M6C-XXXSP</td><td>3 Kbytes</td></tr><tr><td></td><td>M37906M8C-XXXFP, M37906M8C-XXXSP</td><td>3 Kbytes</td></tr></table></td></tr>
<tr><td></td>
<td>Note:<table><tr><td>Packages</td><td>M37906M4C-XXXFP</td><td>42-pin plastic •••••</td></tr><tr><td></td><td>M37906M4C-XXXSP</td><td>42-pin shrink •••••</td></tr></table></td>
<td>2:<table><tr><td>Packages</td><td>M37906M4C-XXXFP,
M37906M6C-XXXFP,
M37906M8C-XXXFP</td><td>42-pin plastic •••••</td></tr><tr><td></td><td>M37906M4C-XXXSP,
M37906M6C-XXXSP,
M37906M8C-XXXSP</td><td>42-pin shrink •••••</td></tr></table></td></tr>
<tr><td>Page 5</td><td></td><td></td></tr>
<tr><td>$\overline{\mathrm{P6OUT_{CUT}}}$
[Pin], [Name]</td>
<td><table><tr><td>$\overline{\mathrm{INT_4}}$</td><td>$\overline{\mathrm{INT_4}}$ input</td><td>input</td><td>•••••</td></tr></table></td>
<td><table><tr><td>$\overline{\mathrm{P6OUT_{CUT}}}$</td><td>$\overline{\mathrm{P6OUT_{CUT}}}$ input</td><td>input</td><td>•••••</td></tr></table></td></tr>
<tr><td>[Functions]</td>
<td>This pin functions as an input pin for $\overline{\mathrm{INT_4}}$. Also, this pin functions as an input pin for a forcibly-cut-off signal in the motor drive waveform output mode.</td>
<td>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 $\overline{\mathrm{INT_4}}$, and as an input pin for the port-output-cutoff function in the motor drive wave-form output mode.</td></tr>
<tr><td>Page 6, Memory</td>
<td>(Line 5)
The internal ROM is allocated to addresses 8000_{16} through $FFFF_{16}$, and its address space is 32 Kbytes.</td>
<td>(Line 5)
The internal ROM and RAM are allocated as shown in Figures 1 (1) through (3).</td></tr>
<tr><td></td>
<td>The internal RAM is allocated to addresses $C00_{16}$ through FFF_{16}, 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.</td>
<td>(Deleted)</td></tr>
<tr><td>Page 6, Right column Line 8</td>
<td>Allocated to addresses 0_{16} through FF_{16} are ••••••
A-D converter, D-A converter, UART, timers, interrupt control registers, etc.</td>
<td>Allocated to addresses 0_{16} through FF_{16} are ••••••
A-D converter, D-A converter, serial I/O, timers, interrupt control registers, etc.</td></tr>
<tr><td>Page 6</td>
<td>Figure 1. Memory map of M37906M4C-XXXFP (Single-chip mode)</td>
<td>Figure 1 (1). Memory map of M37906M4C-XXXFP/SP (Single-chip mode)</td></tr>
<tr><td>Page 7</td>
<td>―――</td>
<td>Memory map of M37906M6C-XXXFP/SP (Single-chip mode)</td></tr>
</table>

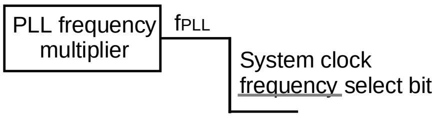
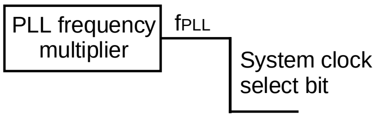
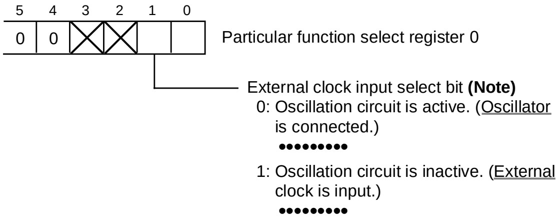
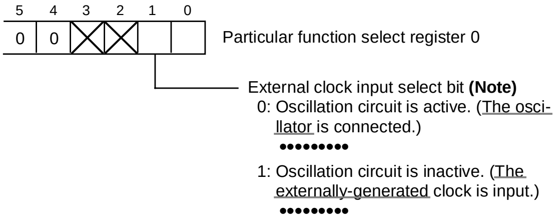
Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.3

Page	Error	Correction
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, NMI, and software interrupt ••••	(Line 3) •••• all interrupts except watchdog timer and software interrupts ••••
Page 20, Fig. 11	 5 4 3 2 1 0 Processor mode register 0 Processor mode bits 00 : Single-chip mode :	 5 4 3 2 1 0 0 0 Processor mode register 0 Processor mode bits 00 : Single-chip mode :
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 INT3 is to be allocated can be selected ••••	The position where pin INT3/RTPTRG0 is allocated can be selected ••••
★ Page 22, Fig. 13	(Interrupt control register bit configuration for INT3–INT7)  5 4 3 2 1 0 Interrupt priority level select bits (Notes 1, 2) Interrupt request bit :	(Interrupt control register bit configuration for INT3–INT7)  5 4 3 2 1 0 Interrupt priority level select bits (Note 1) Interrupt request bit (Note 2) :
Page 39, Fig. 43	 Waveform output control bit (bit 7 at addresss A616) D Q R INT4 Reset	 Waveform output control bit (bit 7 at addresss A616) D Q R P6OUTCUT Reset
Page 40, Right column Lines 5, 6	••• This bit is cleared to “0” by inputting falling edge to the INT4 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	 5 4 3 2 1 0 position-data-retain function control register Retained trigger’s polarity select bit :	 5 4 3 2 1 0 position-data-retain function control register Retain-trigger polarity select bit :
★ 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 addresss A616) D Q R INT4 Reset	 Waveform output control bit 1 (bit 7 at addresss A616) D Q R P6OUTCUT Reset
★ Page 45, Fig. 49	 5 4 3 2 1 0 Waveform output control register	 5 4 3 2 1 0 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.

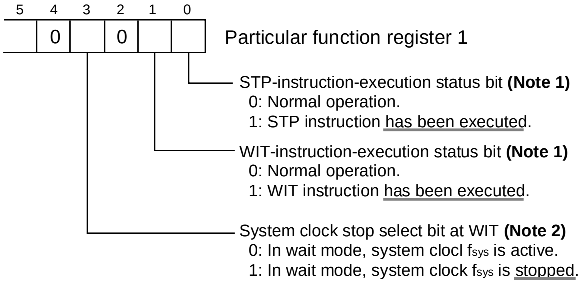
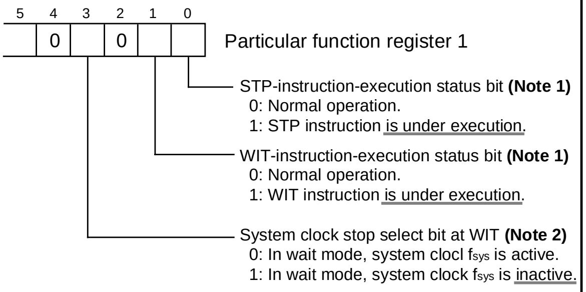
Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.4

Page	Error	Correction
Page 46, Right column Last line	••• by inputting a falling edge to an external interrupt pin, $\overline{INT_4}$.	••• by inputting a falling edge to pin $\overline{P6OUT_{CUT}}$.
Page 47, Left column Line 45	••• by inputting a falling edge to an external interrupt pin, $\overline{INT_4}$, or clearing with instructions.	••• by inputting a falling edge to pin $\overline{P6OUT_{CUT}}$, or by executing instructions.
Page 49, Left column Line 5	••• shown in Figure 55 are used to determine whether to use port P8 as a ••••	••• shown in Figure 55 are used to determine whether to use port P1 as a ••••
Page 49, Fig. 55	 5 4 3 2 1 0 UART0 Transmit/Receive mode register UART1 Transmit/Receive mode register Serial I/O mode select bits 000 : Serial I/O is invalid. (Port P8 functions as a programmable I/O port.) :	 5 4 3 2 1 0 UART0 Transmit/Receive mode register UART1 Transmit/Receive mode register Serial I/O mode select bits 000 : Serial I/O is invalid. (Port P1 functions as a programmable I/O port.) :
Page 59, Table 8	 Functions Pin $P1_0/\overline{CTS_0}/\overline{RTS_0}$ **(Note 1)** Pin $P1_1$/••• $\overline{CTS_2}$ $P1_1$ or CLK_0	 Functions Pin $P1_0/\overline{CTS_0}/\overline{RTS_0}$ **(Note 1)** Pin $P1_1$/••• $\overline{CTS_0}$ $P1_1$ or CLK_0
★ Page 63, Operation mode Line 8	••• the comparator function select register is "0" and ••••	••• the comparator function select register 0 is "0" and ••••
★ Page 63, Operation mode Last line	••• the comparator result register.	••• the comparator result register 0.
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 $\overline{P6OUT_{CUT}}$ or by executing instructions. When a pin is programmed for output, the data ••••
Page 70, Fig. 76	[Inside dotted-line included] $P2_0/TA4_{OUT}$, $P2_2/TA9_{OUT}$, $P6_0/TA0_{OUT}/\overline{W}/RTP0_0$, $P6_1/TA0_{IN}/\overline{V}/RTP0_1$, $P6_2/TA1_{OUT}/\overline{U}/RTP0_2$, $P6_3/TA1_{IN}/W/RTP0_3$, $P6_4/TA2_{OUT}/V/RTP1_0$, $P6_5/TA2_{IN}/U/RTP1_1$	[Inside dotted-line included] $P2_0/TA4_{OUT}$, $P2_2/TA9_{OUT}$
	———	The block diagram for port P6 pins (the 3rd block diagram) is added.
★ Page 72, Fig. 79	Comparison result register 0 (DE_{16})••• 00_{16}	Comparator result register 0 (DE_{16})••• 00_{16}
Page 73, Fig. 78	Debug control register 1 (81_{16})•••  0 0 0 (Note 2) 0 0 0 (Note 2)	Debug control register 1 (81_{16})•••  0 0 0 0 0 0 (Note 2)
★ 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 0; 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 V_{CONT} and PLL circuit	Fig. 83 Circuit example of connection with pin V_{CONT} when PLL circuit used
★ Page 75, Left column Lines 7, 27 Right column Lines 15, 10, 21	••• the clock control register ••••	••• the clock control register 0 ••••

Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.5

	Page	Error	Correction
★	Page 75, Left column Line 20	••• selects the PLL circuit's operation (stopped/active).	••• selects the PLL circuit's operation (inactive/active).
★	Page 75, Left column Line 23	••• (In this case, the PLL circuit operates.)	••• (In this case, the PLL circuit is active.)
★	Page 75, Left column Line 25	At the STP instruction execution, the PLL circuit stops its operation, and ••••	At the STP instruction execution, the PLL circuit is active, and ••••
★	Page 75, Right column Lines 1, 7, 12	••• the frequency of the PLL output clock (f_{PLL}) •••	••• the frequency of f_{PLL} •••
★	Page 75, Right column Line 11	•••, and fX_{IN} is select as the system clock when bit 5 = "0".	•••, and input clock fX_{IN} is select as the system clock when bit 5 = "0".
★	Page 75, Right column Line 22	•••, and these bits select the multiplication ratio of (f_1 to f_{4096})/(f_{sys}).	•••, and these bits select the division ratio of (f_1 to f_{4096})/(f_{sys}).
★	Page 75, Table 9 Note	**Note:** The PLL multiplication must be set so that the frequency of the PLL output clock (f_{PLL}) must be •••	**Note:** The PLL multiplication must be set so that the frequency of f_{PLL} must be •••
★	Page 76, Fig. 84	 PLL frequency multiplier f_{PLL} System clock frequency select bit	 PLL frequency multiplier f_{PLL} System clock select bit
★	Page 77, Fig. 86	 5 4 3 2 1 0 0 0 Particular function select register 0 External clock input select bit (Note) 0: Oscillation circuit is active. (Oscillator is connected.) ••••••••• 1: Oscillation circuit is inactive. (External clock is input.) •••••••••	 5 4 3 2 1 0 0 0 Particular function select register 0 External clock input select bit (Note) 0: Oscillation circuit is active. (The oscillator is connected.) ••••••••• 1: Oscillation circuit is inactive. (The externally-generated clock is input.) •••••••••
★	Page 78, Left column Lines 4, 5	These modes are used to save the power dissipation of the system, by stopping oscillation or system clock in the case that the CPU needs not be operating.	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.
★	Page 78, Left column Line 12	The interrupt priority level of this interrupt is required to be higher than the •••	The interrupt priority level of this interrupt needs 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 stops the oscillation circuit and PLL circuit. It also stops input clock fX_{IN}, system clock f_{sys}, •••	The execution of the STP instruction makes the oscillation circuit and PLL circuit inactive. It also makes the following inactive: input clock fX_{IN}, system clock f_{sys}, •••
★	Page 78, Right column Lines 5, 6	•••, Wf_{32}, and Wf_{512} in the "L" state, and divide clocks fX_{16} to fX_{128} in the "H" state.	•••, Wf_{32}, and Wf_{512} with the "L" state, and divide clocks fX_{16} to fX_{128} with the "H" state.
★	Page 78, Right column Line 13	At this time, timers A and B operate only in the event counter mode, •••	At this time, timers A and B can be active only in the event counter mode, •••

Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.6

	Page	Error	Correction
★	Page 78, Table 11	Stopped	Inactive
		Notes 1: ••••, the oscillation circuit stops. Also, clock input from pin X_{IN} available. **2:** ••••, the PLL circuit stops.	**Notes 1:** ••••, the oscillation circuit is inactive. Also, clock input from pin X_{IN} is allowed. **2:** ••••, the PLL circuit is inactive.
★	Page 79, Left column Line 3	•••• (bit 5 of the clock control register) = “1”, ••••	•••• (bit 5 of the clock control register 0) = “1”, ••••
★	Page 79, Left column Line 10	••• after the oscillation circuit has been restarted ••••	••• after the oscillation circuit and PLL circuit have been restarted ••••
★	Page 79, Fig. 87	 5 4 3 2 1 0 0 0 Particular function register 1 STP-instruction-execution status bit **(Note 1)** 0: Normal operation. 1: STP instruction has been executed. WIT-instruction-execution status bit **(Note 1)** 0: Normal operation. 1: WIT instruction has been executed. System clock stop select bit at WIT **(Note 2)** 0: In wait mode, system clocl f_{sys} is active. 1: In wait mode, system clock f_{sys} is stopped.	 5 4 3 2 1 0 0 0 Particular function register 1 STP-instruction-execution status bit **(Note 1)** 0: Normal operation. 1: STP instruction is under execution. WIT-instruction-execution status bit **(Note 1)** 0: Normal operation. 1: WIT instruction is under execution. System clock stop select bit at WIT **(Note 2)** 0: In wait mode, system clocl f_{sys} is active. 1: In wait mode, system clock f_{sys} is inactive.
		Notes 1: •••• Even when “1” is written, the bit status will not change.	**Notes 1:** •••• Even when “1” is try to be written, the bit status will not change.
★	Page 80, Left column Line 5	••••, and divide clocks Wf_{32} and Wf_{512} are stopped in the “L” state.	••••, and divide clocks Wf_{32} and Wf_{512} are inactive with the “L” state.
★	Page 80, Left column Lines 6, 7	••••, and peripheral devices’ clock f_1 to f_{4096} remain operating. Therefore, BIU and CPU are stopped, ••••	••••, and peripheral devices’ clocks f_1 to f_{4096} remain active. Therefore, BIU and CPU are inactive, ••••
★	Page 80, Left column Lines 9, 10	••••, which use the peripheral devices’ clocks f_1 to f_{4096}, are still operating. Note that the watchdog timer is stopped.	••••, which use the peripheral devices’ clocks f_1 to f_{4096}, are still active. Note that the watchdog timer is inactive.
★	Page 80, Left column Lines 13, 14	••••, and input clock fX_{IN} are operating, while system clock f_{sys}, ϕ_{BIU}, ϕ_{CPU}, and peripheral devices’ clocks stop operating.	••••, and input clock fX_{IN} are active, while system clock f_{sys}, ϕ_{BIU}, ϕ_{CPU}, and peripheral devices’ clocks are inactive.
★	Page 80, Left column Lines 16, 17	••••, Wf_{32} and Wf_{512}, are stopped. At this time, timers A and B operate only in the event counter mode, ••••	••••, Wf_{32} and Wf_{512}, become inactive. At this time, timers A and B are active only in the event counter mode, ••••
★	Page 80, Left column Line 21	•••• the system clocks stop select bit at WIT is to be set to “1” ••••	•••• the system clocks stop select bit at WIT needs to be set to “1” ••••
★	Page 80, Left column Line 26	••••, and clock input fX_{IN} are operating in the WIT mode, ••••	••••, and clock input fX_{IN} are active in the WIT mode, ••••
★	Page 81, Left column Line 4	**(1) Stop of system clock in wait mode**	**(1) Inactive system clock in wait mode**
★	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, when the system clock stop select bit ••••
★	Page 81, Left column Line 8	•••• and peripheral devices’ clock stop their operations, and ••••	•••• and peripheral devices’ clock are inactive, and ••••
★	Page 81, Left column Line 11	**(2) Stop of oscillation circuit**	**(2) Inactive oscillation circuit**
★	Page 81, Left column Line 12	When an externally-generated-stable clock is input to pin X_{IN}, ••••	When an externally-generated stable clock is input to pin X_{IN}, ••••

Corrections and Supplementary Explanation for M37906MxC Datasheet (Rev.B) No.7

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