M37710S4BFP

16-BIT CMOS MICROCOMPUTER

DESCRIPTION

The M37710S4BFP is a microcomputer designed with high-performance CMOS silicon gate technology. This is housed in a 80-pin plastic molded QFP. This microcomputer has a large 16M bytes address space, three instruction queue buffers, and two data buffers for high-speed instruction execution. The CPU is a 16-bit parallel processor that can also be switched to perform 8-bit parallel processing. This microcomputer is suitable for office, business, and industrial equipment controller that require high-speed processing of large data.

FEATURES

- Number of basic instructions 103
- Memory size RAM ..... 2048 bytes
- Instruction execution time
  The fastest instruction at 25MHz frequency 160ns
- Single power supply ..... 5V±10%
- Low power dissipation (at 25MHz frequency)
  ..... 95mW (Typ.)
- Interrupts ..... 19 types 7 levels
- Multiple function 16-bit timer ..... 5+3
  (Pulse motor drive waveform can be output.)
- UART (may also be synchronous) 2
- 10-bit A-D converter ..... 8-channel inputs
- 8-bit D-A converter ..... 2-channel outputs
- 12-bit watchdog timer
- Programmable input/output
  (ports P4, P5, P6, P7, P8) 37

APPLICATION

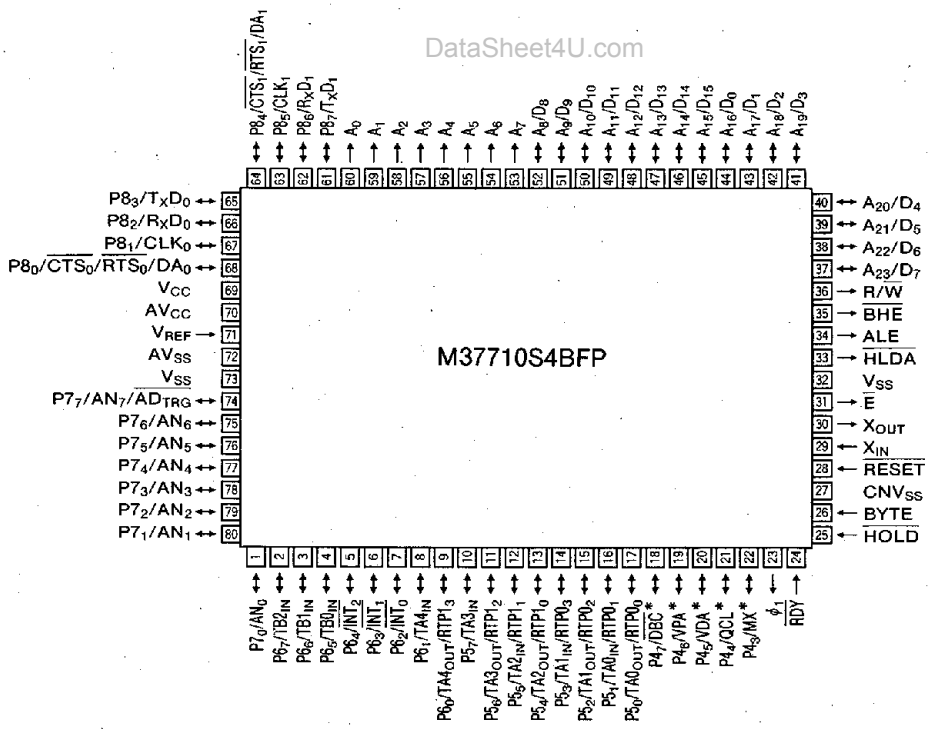
Control devices for office equipment such as copiers, printers, typewriters, facsimiles, word processors, and personal computers

Control devices for industrial equipment such as communication and measuring instruments

NOTE

Refer to "Chapter 5 PRECAUTIONS" when using this microcomputer.

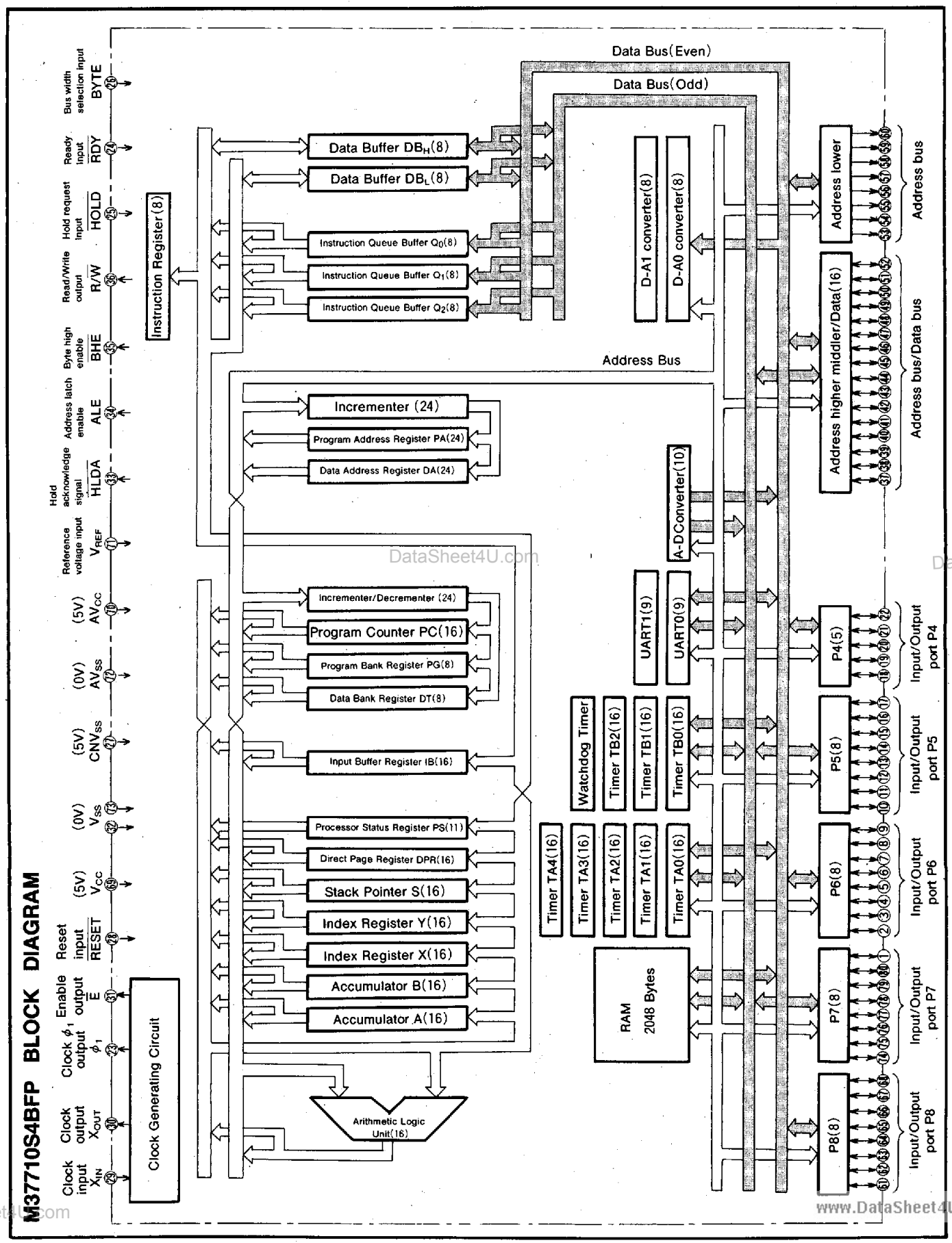
PIN CONFIGURATION (TOP VIEW)



Outline 80P6N-A

*: Used in the evaluation chip mode only

M37710S4BFP BLOCK DIAGRAM



FUNCTIONS OF M37710S4BFP

Parameter		Functions
Number of basic instructions		103
Instruction execution time		160ns (the fastest instruction at external clock 25MHz frequency)
Memory size	RAM	2048 bytes
Input/Output ports	P5~P8	8 -bit× 4
	P4	5 -bit× 1
Multi-function timers	TA0, TA1, TA2, TA3, TA4	16-bit× 5
	TB0, TB1, TB2	16-bit× 3
Serial I/O		(UART or clock synchronous serial I/O)×2
A-D converter		10-bit× 1 (8 channels)
D-A converter		8 -bit× 2
Watchdog timer		12-bit× 1
Interrupts		3 external types, 16 internal types (Each interrupt can be set the priority levels to 0 ~ 7.)
Clock generating circuit		Built-in(externally connected to a ceramic resonator or quartz crystal resonator)
Supply voltage		5 V±10%
Power dissipation		95mW(at external clock 25MHz frequency)
Input/Output characteristic	Input/Output voltage	5 V
	Output current	5 mA
Memory expansion		Maximum 16M bytes
Operating temperature range		−20~85℃
Device structure		CMOS high-performance silicon gate process
Package		80-pin plastic molded QFP

PIN DESCRIPTION

Pin	Name	Input/Output	Functions
V_{CC}, V_{SS}	Power supply		Supply 5 V±10% to V_{CC} and 0 V to V_{SS}.
CNV_{SS}	CNV_{SS} input	Input	Connect to V_{CC}.
$\overline{RESET}$	Reset input	Input	To enter the reset state, this pin must be kept at a "L" condition should be maintained for the required time.
X_{IN}	Clock input	Input	These are I/O pins of internal clock generating circuit. Connect a ceramic or quartz crystal resonator between X_{IN} and X_{OUT}. When an external clock is used, the clock source should be connected to the X_{IN} pin and the X_{OUT} pin should be left open.
X_{OUT}	Clock output	Output	
$\overline{E}$	Enable output	Output	Data or instruction read and data write are performed when output from this pin is "L".
BYTE	Bus width selection input	Input	This pin determines whether the external data bus is 8-bit width or 16-bit width. The width is 16 bits when "L" signal inputs and 8 bits when "H" signal inputs.
AV_{CC}, AV_{SS}	Analog supply input		Power supply for A-D converter. AV_{SS} is also used for D-A converter. Connect AV_{CC} to V_{CC}, and AV_{SS} to V_{SS} externally.
V_{REF}	Reference voltage input	Input	This is reference voltage input pin for A-D converter and D-A converter.
ϕ_1	Clock output	Output	This pin outputs the clock ϕ_1 which is divided the clock to X_{IN} pin by 2.
$\overline{RDY}$	Ready	Input	This is ready input pin. This is an input pin for the $\overline{RDY}$ signal. Internal clock stops while this signal is "L".
$\overline{HOLD}$	Hold request input	Input	This is an input pin for $\overline{HOLD}$ request signal. The microcomputer enters into hold state while this signal is "L".
$\overline{HLDA}$	Hold acknowledge output	Output	This is an output pin for $\overline{HLDA}$ signal, indicates the hold state.
R/$\overline{W}$	Read/Write output	Output	"H" indicates the read status and "L" indicates the write status.
$\overline{BHE}$	Byte high enable output	Output	"L" is output when an odd-numbered address is accessed.
ALE	Address latch enable output	Output	This is used to retrieve only the address data from address data and data multiplex signal.
A_0~A_7	Address (low-order) output	Output	Address (A_7~A_0) is output.
A_8/D_8~A_{15}/D_{15}	Address (middle-order) output/Data (high-order) I/O	I/O	In case the BYTE pin is "L" and an external data bus is 16-bit width, high-order data (D_{15}~D_8) is input or output when $\overline{E}$ output is "L" and an address (A_{15}~A_8) is output when $\overline{E}$ output is "H". In case the BYTE pin is "H" and an external data bus is 8-bit width, only address (A_{15}~A_8) is output.
A_{16}/D_0~A_{23}/D_7	Address (high-order) output/Data (low-order) I/O	I/O	Low-order data (D_7~D_0) is input or output when $\overline{E}$ output is "L", and an address (A_{23}~A_{16}) is output when $\overline{E}$ output is "H".
$P4_3$~$P4_7$	I/O port P4	I/O	Port P4 is a 5-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset.
$P5_0$~$P5_7$	I/O port P5	I/O	Port P5 is a 8-bit I/O port. This port has an data direction register and each pin can be programmed for input or output. This port is in input mode when reset. These pins also function as I/O pins for timer A0, timer A1, timer A2 and timer A3. $P5_0$ to $P5_6$ also function as output pins for pulse motor drive waveform.
$P6_0$~$P6_7$	I/O port P6	I/O	In addition to having the same I/O functions as port P5, these pins also function as I/O pins for timer A4, external interrupt input $\overline{INT_0}$, $\overline{INT_1}$ and $\overline{INT_2}$ pins, and input pin for timer B0, timer B1 and timer B2. $P6_0$ also functions as an output pin for pulse motor drive waveform.
$P7_0$~$P7_7$	I/O port P7	I/O	These pins also function as analog input AN_0~AN_7 input pins in addition to having the same I/O functions as port P5. $P7_7$ also has an A-D conversion trigger input function.
$P8_0$~$P8_7$	I/O port P8	I/O	These pins also function as RxD, TxD, CLK, $\overline{CTS}/\overline{RTS}$ pins for UART0 and UART1 in addition to having the same I/O functions as port P5.

BASIC FUNCTION BLOCKS

The M37710S4BFP contains the following devices on a single chip: RAM for storing instructions and data, CPU for processing, bus interface unit (which controls instruction prefetch and data read/write between CPU and memory), timers, UART, A-D converter, D-A converter, and other peripheral devices such as I/O ports. Each of these devices are described below.

MEMORY

The memory map is shown in Figure 1. The address space is 16M bytes from addresses 0_{16} to $FFFFFF_{16}$. The address space is divided into 64K bytes units called banks. The banks are numbered from 0_{16} to FF_{16}.

Built-in RAM and control registers for built-in peripheral devices are assigned to bank 0_{16}.

Addresses $FFD6_{16}$ to $FFFF_{16}$ are the RESET and interrupt vector addresses and contain the interrupt vectors. Use ROM for memory of this address. Refer to the section on interrupts for details.

The 2048 bytes area from addresses 80_{16} to $87F_{16}$ contains the built-in RAM. In addition to storing data, the RAM is used as stack during a subroutine call, or interrupts.

Assigned to addresses 0_{16} to $7F_{16}$ are peripheral devices such as I/O ports, D-A converter, A-D converter, UART, timer, and interrupt control registers.

A 256 bytes direct page area can be allocated anywhere in bank 0_{16} using the direct page register DPR. In direct page addressing mode, the memory in the direct page area can be accessed with two words thus reducing program steps.

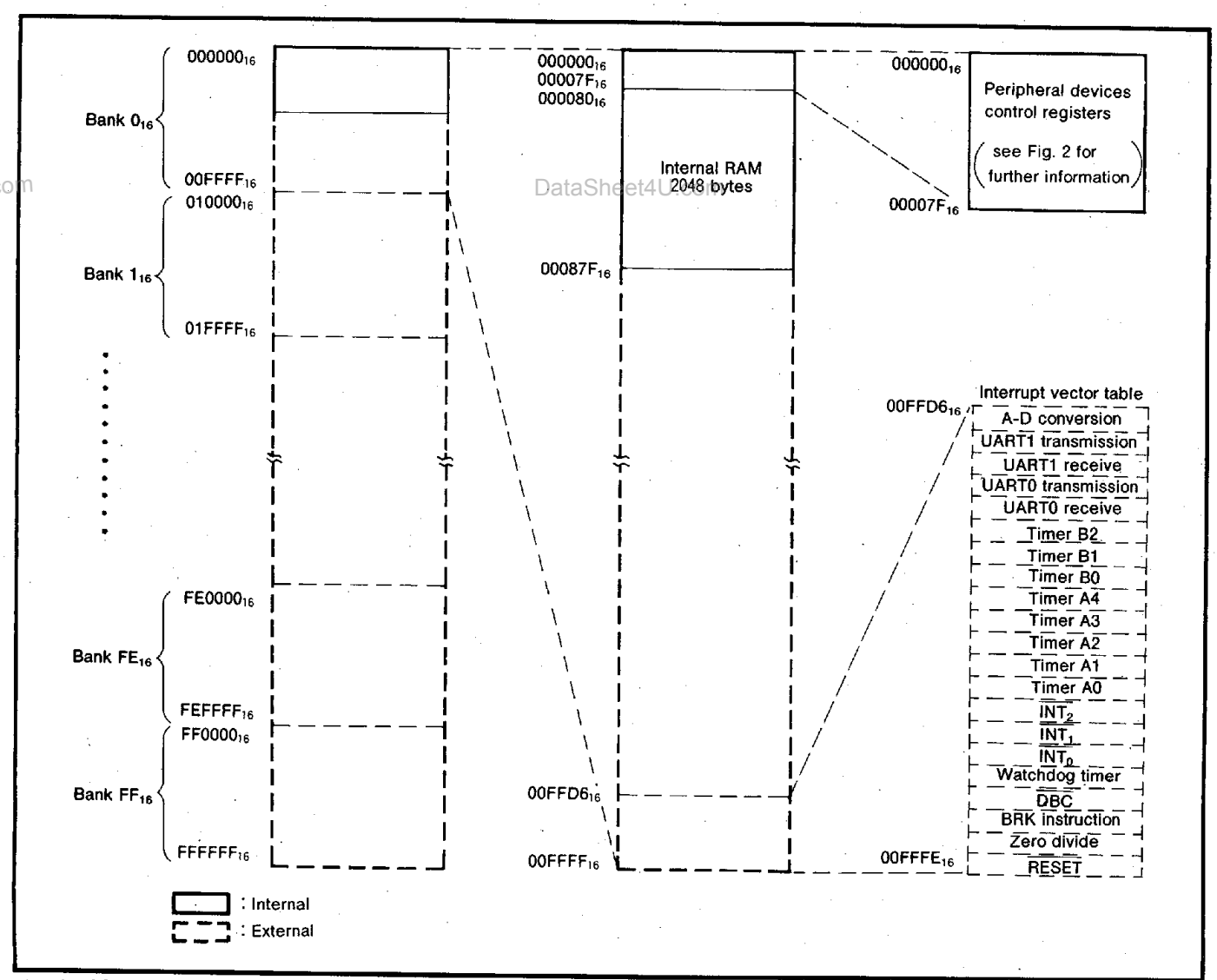


Fig. 1 Memory map

Address (Hexadecimal notation)	
000000	
000001	
000002	
000003	
000004	
000005	
000006	
000007	
000008	
000009	
00000A	Port P4
00000B	Port P5
00000C	Port P4 data direction register
00000D	Port P5 data direction register
00000E	Port P6
00000F	Port P7
000010	Port P6 data direction register
000011	Port P7 data direction register
000012	Port P8
000013	
000014	Port P8 data direction register
000015	
000016	
000017	
000018	
000019	
00001A	D-A register 0
00001B	
00001C	D-A register 1
00001D	
00001E	A-D control register 0
00001F	A-D control register 1
000020	A-D register 0
000021	
000022	A-D register 1
000023	
000024	A-D register 2
000025	
000026	A-D register 3
000027	
000028	A-D register 4
000029	
00002A	A-D register 5
00002B	
00002C	A-D register 6
00002D	
00002E	A-D register 7
00002F	
000030	UART 0 transmit/receive mode register
000031	UART 0 bit rate generator
000032	UART 0 transmission buffer register
000033	
000034	UART 0 transmit/receive control register 0
000035	UART 0 transmit/receive control register 1
000036	UART 0 receive buffer register
000037	
000038	UART 1 transmit/receive mode register
000039	UART 1 bit rate generator
00003A	UART 1 transmission buffer register
00003B	
00003C	UART 1 transmit/receive control register 0
00003D	UART 1 transmit/receive control register 1
00003E	UART 1 receive buffer register
00003F	

Address (Hexadecimal notation)	
000040	Count start flag
000041	
000042	One-shot start flag
000043	
000044	Up-down flag
000045	
000046	Timer A0
000047	
000048	Timer A1
000049	
00004A	Timer A2
00004B	
00004C	Timer A3
00004D	
00004E	Timer A4
00004F	
000050	Timer B0
000051	
000052	Timer B1
000053	
000054	Timer B2
000055	
000056	Timer A0 mode register
000057	Timer A1 mode register
000058	Timer A2 mode register
000059	Timer A3 mode register
00005A	Timer A4 mode register
00005B	Timer B0 mode register
00005C	Timer B1 mode register
00005D	Timer B2 mode register
00005E	Processor mode register 0
00005F	Processor mode register 1
000060	Watchdog timer
000061	Watchdog timer frequency selection flag
000062	Waveform output mode register
000063	
000064	Pulse output data register 1
000065	Pulse output data register 0
000066	Reserved area (Note)
000067	
000068	
000069	
00006A	
00006B	
00006C	
00006D	
00006E	
00006F	
000070	A-D conversion interrupt control register
000071	UART0 transmission interrupt control register
000072	UART0 receive interrupt control register
000073	UART1 transmission interrupt control register
000074	UART1 receive interrupt control register
000075	Timer A0 interrupt control register
000076	Timer A1 interrupt control register
000077	Timer A2 interrupt control register
000078	Timer A3 interrupt control register
000079	Timer A4 interrupt control register
00007A	Timer B0 interrupt control register
00007B	Timer B1 interrupt control register
00007C	Timer B2 interrupt control register
00007D	$\overline{INT_0}$ interrupt control register
00007E	$\overline{INT_1}$ interrupt control register
00007F	$\overline{INT_2}$ interrupt control register

Note : Do not write to this address.

Fig. 2. Location of peripheral devices and interrupt control registers

CENTRAL PROCESSING UNIT (CPU)

The CPU has ten registers and is shown in Figure 3. 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 lower 8 bits can be used separately. The 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., is executed mainly through the accumulator.

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.

INDEX REGISTER X (X)

Index register X consists of 16 bits and the lower 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 mode, register X is used as the index register and the contents of this address is added to obtain the real address.

Also, when executing a block transfer instruction MVP or MVN, the contents of index register X indicate the low-order 16 bits of the source data address. The third byte of the MVP and MVN is the high-order 8 bits of the source data address.

INDEX REGISTER Y (Y)

Index register Y consists of 16 bits and the lower 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 mode, register Y is used as the index register and the contents of this address is added to obtain the real address.

Also, when executing a block transfer instruction MVP or MVN, the contents of index register Y indicate the low-order 16 bits of the destination address. The second byte of the MVP and MVN is the high-order 8 bits of the destination data address.

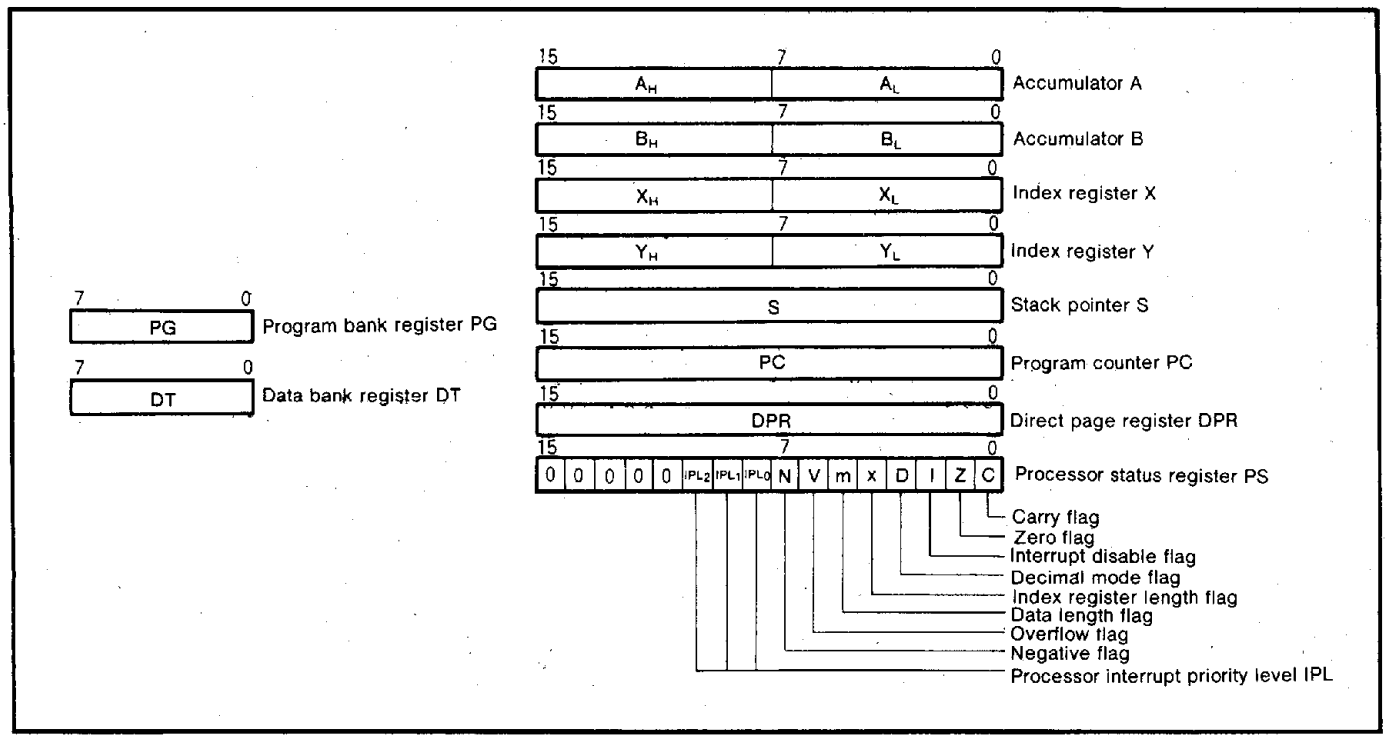


Fig. 3 Register structure

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 cotents of the program bank register (PG) is incremented 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 branch instruction, the contents of the program bank register (PG) is incremented or decremented 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, a part of the data bank register (DT) is used to specify a 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 REGISTER (DPR)

Direct page register (DPR) is a 16-bit register. Its contents is used as the base address of a 256-byte direct page area. The direct page area is allocated in bank 0_{16}, but when the contents of DPR is $FF01_{16}$ or greater, the direct page area spans across bank 0_{16} and bank 1_{16}. All direct addressing modes use the contents of the direct page register (DPR) to generate the data address. If the low-order 8 bits of the direct page register (DPR) is "00_{16}", the number of cycles required to generate an address is minimized. Normally the low-order 8 bits of the direct page register (DPR) is set to "00_{16}".

PROCESSOR STATUS REGISTER (PS)

Processor status register (PS) is an 11-bit register. It consists of a flag 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 processor status register bit 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)

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

3. Interrupt disable flag (I)

When the interrupt disable flag is set to "1", all interrupts except watchdog timer, $\overline{DBC}$, and software interrupt are disabled. This flag is set to "1" automatically when there is an interrupt. 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 two or four digit decimal. Arithmetic operation is performed using four digits when the data length flag m is "0" and with two digits when it is "1". (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-bit when flag m is "0" and 8-bit 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 has meaning when addition or subtraction is performed a word as signed binary number. When the 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. When the 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.

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", when data bit 15 is "1". If data length flag m is "1", when data 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 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.

BUS INTERFACE UNIT

The CPU operates on an internal clock frequency which is obtained by dividing the external clock frequency $f_{(X_{IN})}$ by two. This frequency is twice the bus cycle frequency. In order to speed-up processing, a bus interface unit is used to pre-fetch instructions when the data bus is idle. The bus interface unit synchronizes the CPU and the bus and pre-fetches instructions. Figure 4 shows the relationship between the CPU and the bus interface unit. The bus interface unit has a program address register, a 3-byte instruction queue buffer, a data address register, and a 2-byte data buffer.

The bus interface unit obtains an instruction code from memory and stores it in the instruction queue buffer, obtains data from memory and stores it in the data buffer, or writes the data from the data buffer to the memory.

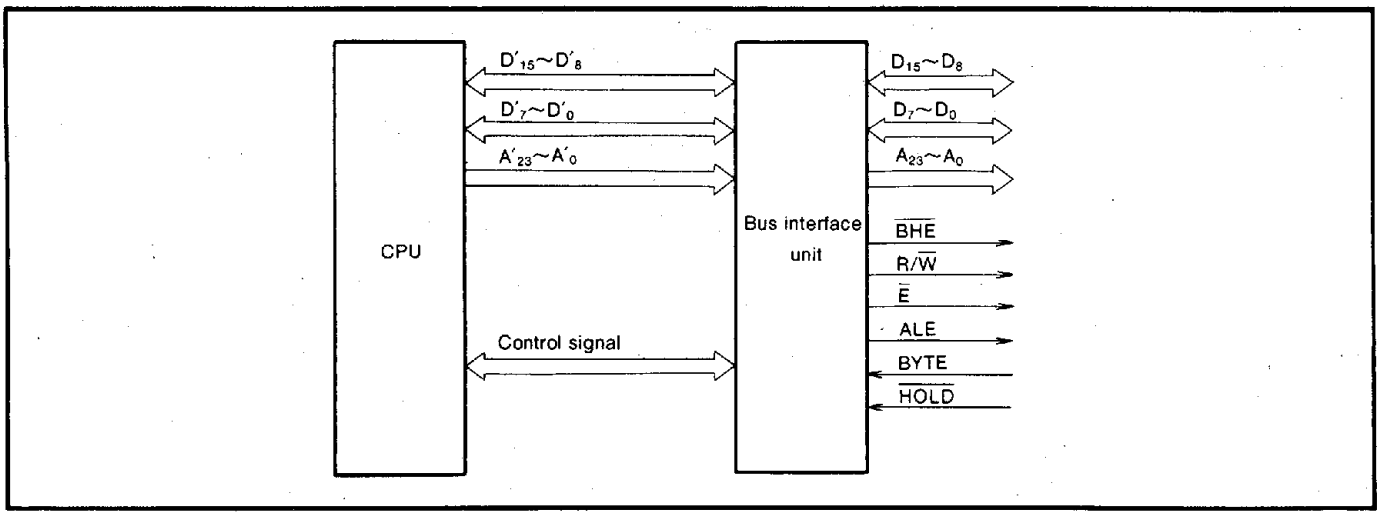


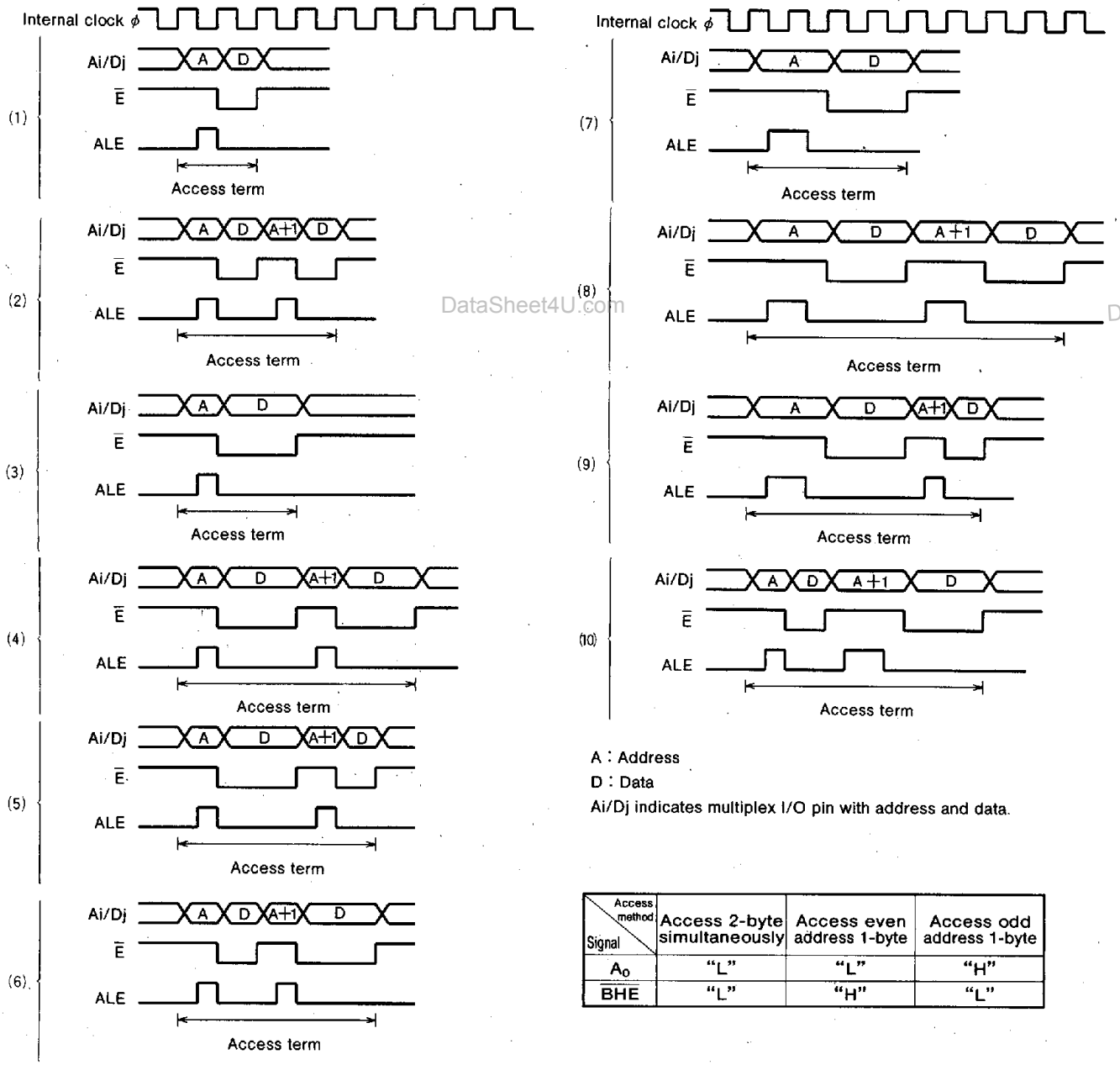
Fig. 4 Relationship between the CPU and the bus interface unit

The bus interface unit operates using one of the waveforms (1) to (10) shown in Figure 5. The standard waveforms are (1) and (2).

The ALE signal is used to latch only the address signal from the multiplexed signal containing data and address.

The $\overline{E}$ signal becomes "L" when the bus interface unit reads an instruction code or data from memory or when it writes data to memory. Whether to perform read or write is controlled by the R/$\overline{W}$ signal. Read is performed when the R/$\overline{W}$ signal is "H" state and write is performed when it is "L" state.

Waveform (1) in Figure 5 is used to access a single byte or two bytes simultaneously. To read or write two bytes simultaneously, the first address accessed must be even. Furthermore, when accessing an external memory area, set the bus width selection input pin BYTE to "L". (external data bus width to 16 bits) The internal memory area is always treated as 16-bit bus width regardless of BYTE.



Access method / Signal	Access 2-byte simultaneously	Access even address 1-byte	Access odd address 1-byte
A_0	"L"	"L"	"H"
$\overline{BHE}$	"L"	"H"	"L"

Fig. 5 Relationship between access method and signals A_0 and $\overline{BHE}$

When performing 16-bit data read or write, if the conditions for simultaneously accessing two bytes are not satisfied, waveform (2) is used to access each byte one by one. However, when prefetching the instruction code, if the address of the instruction code is odd, waveform (1) is used, and only one byte is read in the instruction queue buffer.

The signals A_0 and $\overline{BHE}$ in Figure 5 are used to control these cases: 1-byte read from even address, 1-byte read from odd address, 2-byte simultaneous read from even and odd addresses, 1-byte write to even address, 1-byte write to odd address, or 2-byte simultaneous write to even and odd addresses. The A_0 signal that is the address bit 0 is "L" when an even number address is accessed. The $\overline{BHE}$ signal becomes "L" when an odd number address is accessed.

The bit 2 of processor mode register 0 (address $5E_{16}$) is the wait bit. When external memory area is accessed with this bit set to "0", the width of $\overline{E}$ signal is extended and access time can be extended.

There are two ways to extend the access time and they are selected with the processor mode register 1 (address $5F_{16}$) bit 0.

When this bit is set to "1", the "L" width of $\overline{E}$ signal in (1) becomes twice as long as in (3) and the access time becomes 1.5 times (wait 1). When this bit is set to "0", the ALE signal in (1) and $\overline{E}$ signal are extended as in (7) and the access time is doubled (wait 0).

However, these signals are not extended when accessing internal memory area.

When the wait bit is set to "1", these signals are not extended when accessing either memory area regardless of the value of processor mode register 1 bit 0.

Waveform (4), (5), and (6) show the entire waveform, first half, and last half respectively of waveform (2) for wait 1.

Waveform (8), (9), and (10) show the entire waveform, first half, and last half respectively of waveform (2) for wait 0.

Instruction code read, data read, and data write are described below.

Instruction code read will be described first.

The CPU obtains instruction codes from the instruction queue buffer and executes them. The CPU notifies the bus interface unit that it is requesting an instruction code during an instruction code request cycle. If the requested instruction code is not yet stored in the instruction queue buffer, the bus interface unit halts the CPU until it can store more instructions than requested in the instruction queue buffer.

Even if there is no instruction code request from the CPU, the bus interface unit reads instruction codes from memory and stores them in the instruction queue buffer when the instruction queue buffer is empty or when only one instruction code is stored and the bus is idle on the next cycle. This is referred to as instruction pre-fetching.

Normally, when reading an instruction code from memory, if the accessed address is even the next odd address is read together with the instruction code and stored in the instruction queue buffer.

However, if the bus width switching pin BYTE is "H", external data bus width is 8 bits and the address to be read is in external memory area is odd, only one byte is read and stored in the instruction queue buffer. Therefore, waveform (1), (3) or (7) in Figure 5 is used for instruction code read.

Data read and write are described below.

The CPU notifies the bus interface unit when performing data read or write. At this time, the bus interface unit halts the CPU if the bus interface unit is already using the bus or if there is a request with higher priority. When data read or write is enabled, the bus interface unit uses one of the waveforms from (1) to (10) in Figure 5 to perform the operation.

During data read, the CPU waits until the entire data is stored in the data buffer. The bus interface unit sends the address received from the CPU to the address bus. Then it reads the memory when the $\overline{E}$ signal is "L" and stores the result in the data buffer.

During data write, the CPU writes the data in the data buffer and the bus interface unit writes it to memory. Therefore, the CPU can proceed to the next step without waiting for write to complete. The bus interface unit sends the address received from the CPU to the address bus. Then when the $\overline{E}$ signal is "L", the bus interface unit sends the data in the data buffer to the data bus and writes it to memory.

INTERRUPTS

Table 1 shows the interrupt types and the corresponding interrupt vector addresses. Reset is also treated as a type of interrupt and is discussed in this section, too.

$\overline{DBC}$ is an interrupt used during debugging.

Interrupts other than reset, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction all have interrupt control registers. Table 2 shows the addresses of the interrupt control registers and Figure 6 shows the bit configuration of the interrupt control register.

Use the SEB and CLB instructions when setting each interrupt control register.

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

$\overline{INT_2}$ to $\overline{INT_0}$ are external interrupts and whether to cause an interrupt at the input level (level sense) or at the edge (edge sense) can be selected with the level sense/edge sense selection bit. Furthermore, the polarity of the interrupt input can be selected with polarity selection bit.

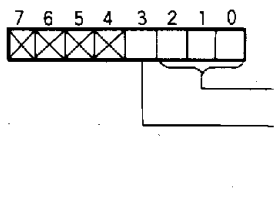
Timer and UART interrupts are described in the respective section.

The priority of interrupts when multiple interrupts are caused simultaneously is partially fixed by hardware, but, it can also be adjusted by software as shown in Figure 7. The hardware priority is fixed the following:

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

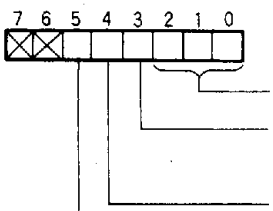
Table 1. Interrupt types and the interrupt vector addresses

Interrupts	Vector addresses
A-D conversion	$00FFD6_{16}$ $00FFD7_{16}$
UART1 transmit	$00FFD8_{16}$ $00FFD9_{16}$
UART1 receive	$00FFDA_{16}$ $00FFDB_{16}$
UART0 transmit	$00FFDC_{16}$ $00FFDD_{16}$
UART0 receive	$00FFDE_{16}$ $00FFDF_{16}$
Timer B2	$00FFE0_{16}$ $00FFE1_{16}$
Timer B1	$00FFE2_{16}$ $00FFE3_{16}$
Timer B0	$00FFE4_{16}$ $00FFE5_{16}$
Timer A4	$00FFE6_{16}$ $00FFE7_{16}$
Timer A3	$00FFE8_{16}$ $00FFE9_{16}$
Timer A2	$00FFEA_{16}$ $00FFEB_{16}$
Timer A1	$00FFEC_{16}$ $00FFED_{16}$
Timer A0	$00FFEE_{16}$ $00FFEF_{16}$
$\overline{INT_2}$ external interrupt	$00FFF0_{16}$ $00FFF1_{16}$
$\overline{INT_1}$ external interrupt	$00FFF2_{16}$ $00FFF3_{16}$
$\overline{INT_0}$ external interrupt	$00FFF4_{16}$ $00FFF5_{16}$
Watchdog timer	$00FFF6_{16}$ $00FFF7_{16}$
$\overline{DBC}$ (unusable)	$00FFF8_{16}$ $00FFF9_{16}$
Break instruction	$00FFFA_{16}$ $00FFFB_{16}$
Zero divide	$00FFFC_{16}$ $00FFFD_{16}$
Reset	$00FFFE_{16}$ $00FFFF_{16}$



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

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



Interrupt priority
Interrupt request bit
0 : No interrupt
1 : Interrupt
Polarity selection bit
0 : Set interrupt request bit at "H" level for level sense and when changing from "H" to "L" level for edge sense.
1 : Set interrupt request bit at "L" level for level sense and when changing from "L" to "H" level for edge sense.
Level sense/edge sense selection bit
0 : Edge sense
1 : Level sense

Interrupt control register configuration for $\overline{INT_2}$~$\overline{INT_0}$.

Fig. 6 Interrupt control register configuration

Table 2. Addresses of interrupt control registers

Interrupt control registers	Addresses
A-D conversion interrupt control register	000070_{16}
UART0 transmit interrupt control register	000071_{16}
UART0 receive interrupt control register	000072_{16}
UART1 transmit interrupt control register	000073_{16}
UART1 receive interrupt control register	000074_{16}
Timer A0 interrupt control register	000075_{16}
Timer A1 interrupt control register	000076_{16}
Timer A2 interrupt control register	000077_{16}
Timer A3 interrupt control register	000078_{16}
Timer A4 interrupt control register	000079_{16}
Timer B0 interrupt control register	$00007A_{16}$
Timer B1 interrupt control register	$00007B_{16}$
Timer B2 interrupt control register	$00007C_{16}$
$\overline{INT_0}$ interrupt control register	$00007D_{16}$
$\overline{INT_1}$ interrupt control register	$00007E_{16}$
$\overline{INT_2}$ interrupt control register	$00007F_{16}$

Interrupts caused by a BRK instruction and when dividing by zero are software interrupts and are not included in this list.

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

Figure 8 shows a diagram of the interrupt priority resolution circuit. When an interrupt is caused, the each interrupt device compares its own priority with the priority from above and if its own priority is higher, then it sends the priority below and requests the interrupt. If the priorities are the same, the one above has priority.

This comparison is repeated to select the interrupt with the highest priority among the interrupts that are being requested. Finally the selected interrupt is compared with the processor interrupt priority level (IPL) contained in the processor status register (PS) and the request is accepted if it is higher than IPL and the interrupt disable flag I is "0". The request is not accepted if flag I is "1". The reset, $\overline{DBC}$, 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, $\overline{DBC}$, watchdog timer, zero divide, and BRK instruction interrupts, which do not have an interrupt control register, the processor interrupt level (IPL) is set as shown in Table 3.

Priority resolution is performed by latching the interrupt request bit and interrupt priority level so that they do not change. They are sampled at the first half and latched at the last half of the operation code fetch cycle.

Because priority resolution takes some time, no sampling pulse is generated for a certain interval even if it is the next operation code fetch cycle.

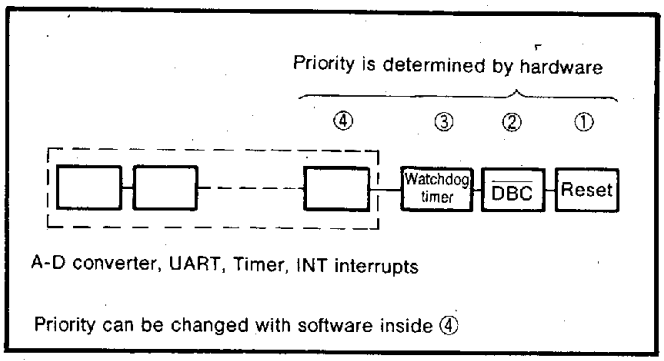


Fig. 7 Interrupt priority

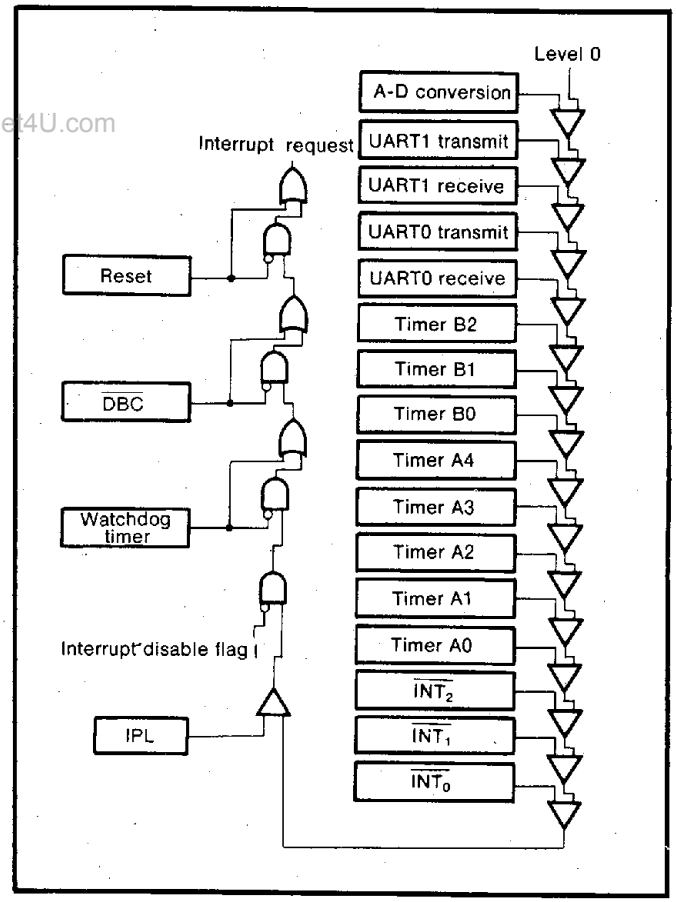


Fig. 8 Interrupt priority resolution

As shown in Figure 9, there are three different interrupt priority resolution 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 10. Table 4 shows the relationship between these bits and the number of cycles. After a reset, the processor mode register is initialized to "00_{16}" and therefore, the longest time is selected.

However, the shortest time should be selected by software.

Table 3. Value set in processor interrupt level (IPL) during an interrupt

Interrupt types	Setting value
Reset	0
DBC	7
Watchdog timer	7
Zero divide	Not change value of IPL.
BRK instruction	Not change value of IPL.

Table 4. Relationship between priority level resolution time selection bit and number of cycles

Priority level resolution time selection bit		Number of cycles
Bit 5	Bit 4	
0	0	7 cycles of ϕ
0	1	4 cycles of ϕ
1	0	2 cycles of ϕ

ϕ : internal clock

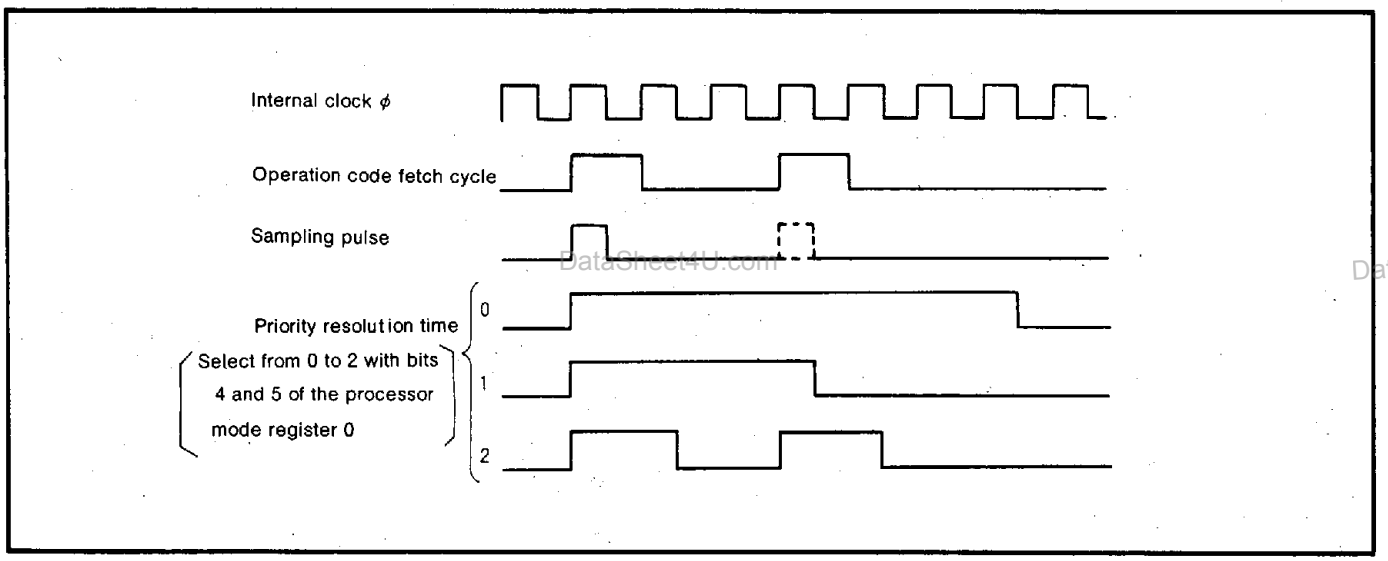


Fig. 9 Interrupt priority resolution time

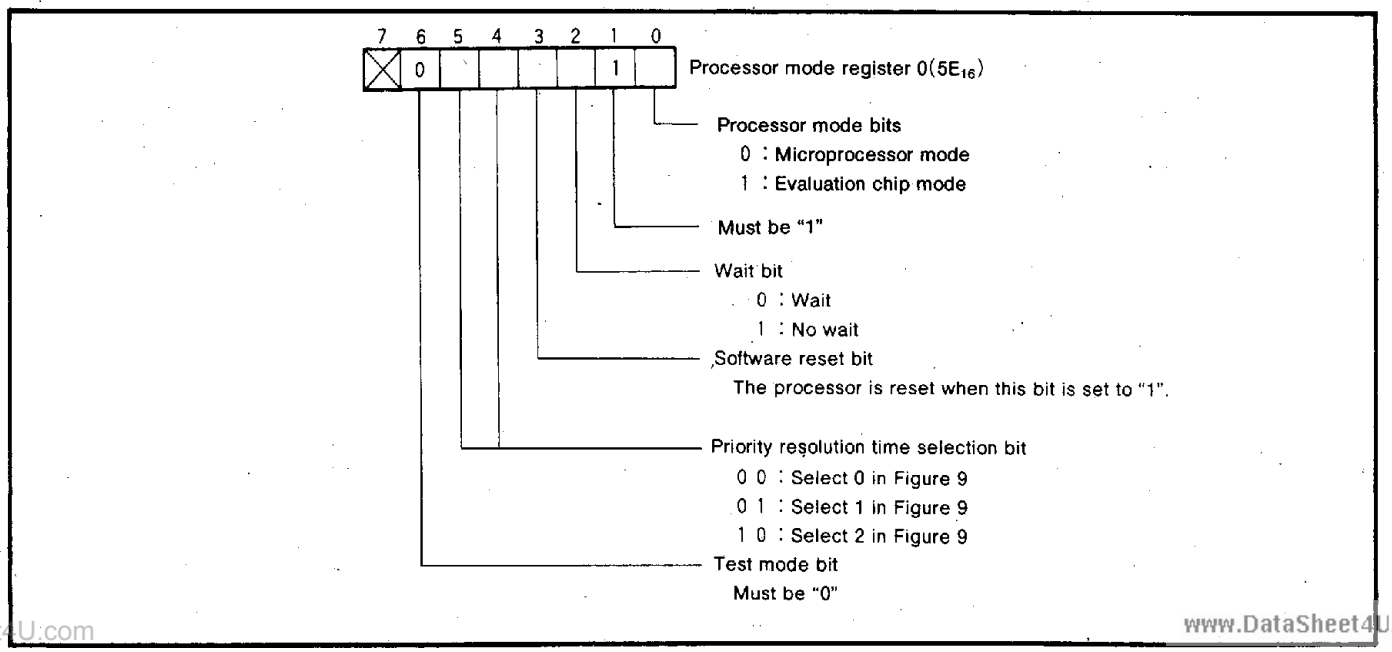


Fig. 10 Processor mode register 0 configuration

TIMER

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

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

TIMER A

Figure 11 shows a block diagram of timer A.

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

(1) Timer mode [00]

Figure 12 shows the bit configuration of the timer Ai mode register during timer mode. Bits 0, 1, and 5 of the timer Ai mode register must always be "0" in timer mode.

Bit 3 is ignored if bit 4 is "0".

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

Figure 13 shows the bit configuration of the count start flag. The counter is decremented, an interrupt is caused and the interrupt request bit in the timer Ai interrupt control register is set when the contents becomes 0000_{16}. At the same time, the contents of the reload register is transferred to the counter and count is continued.

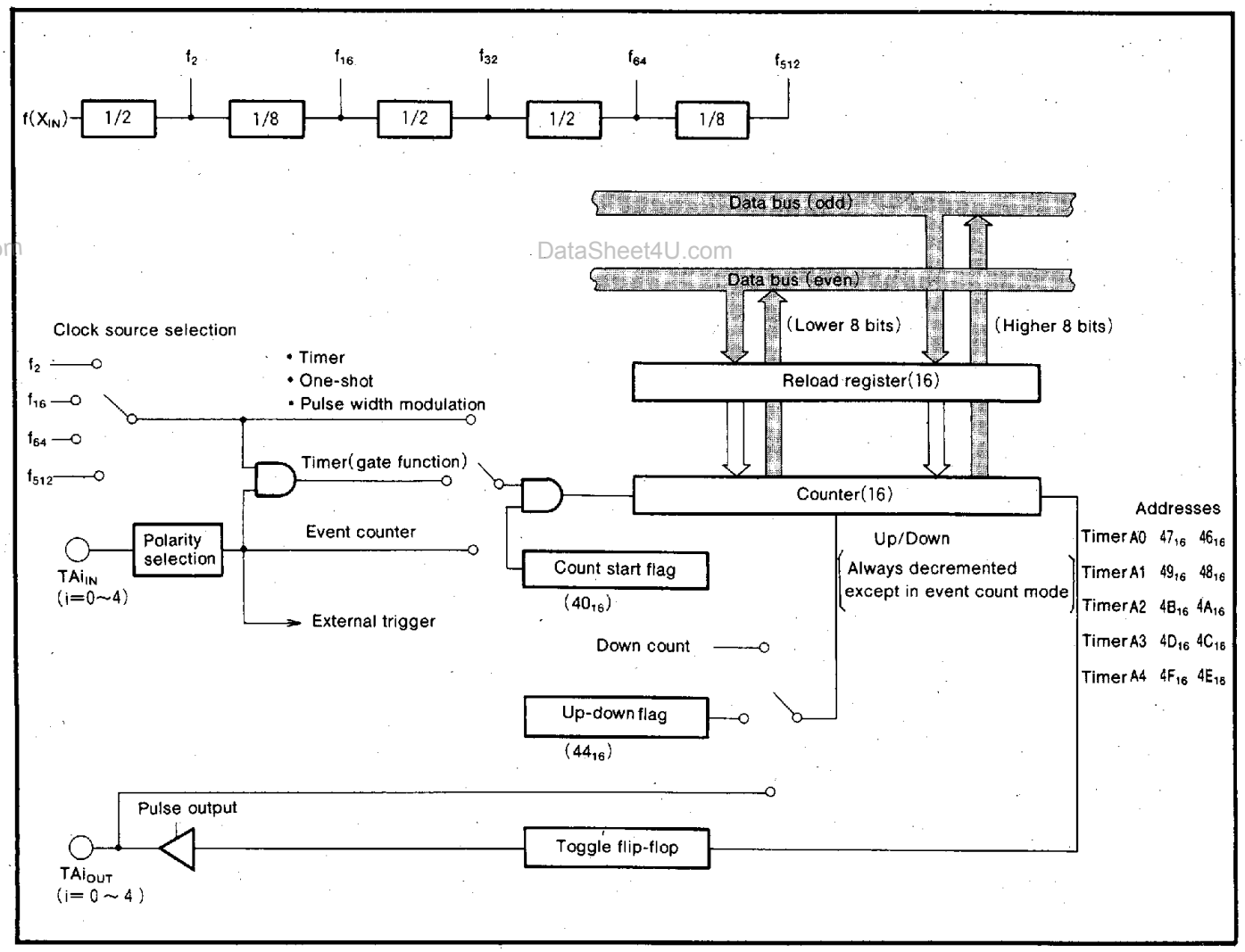


Fig. 11 Block diagram of timer A

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

When bit 2 is "0", TAi_{OUT} pin can be used as a normal port pin.

When bit 4 is "0", TAi_{IN} pin can be used as a normal port pin.

When bit 4 is "1", counting is performed only while the input signal from the TAi_{IN} pin is "H" or "L" as shown in Figure 14. Therefore, this can be used to measure the pulse width of the input signal to TAi_{IN} pin. Whether to count while the input signal is "H" or while it is "L" is determined by bit 3. If bit 3 is "1", counting is performed while the TAi_{IN} pin input signal is "H" and if bit 3 is "0", counting is performed while it is "L".

Note that the duration of "H" or "L" on the TAi_{IN} pin must be two 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 counter is reloaded with new data from the reload register at the next reload time. 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 dividing ratio is $1/(n+1)$.

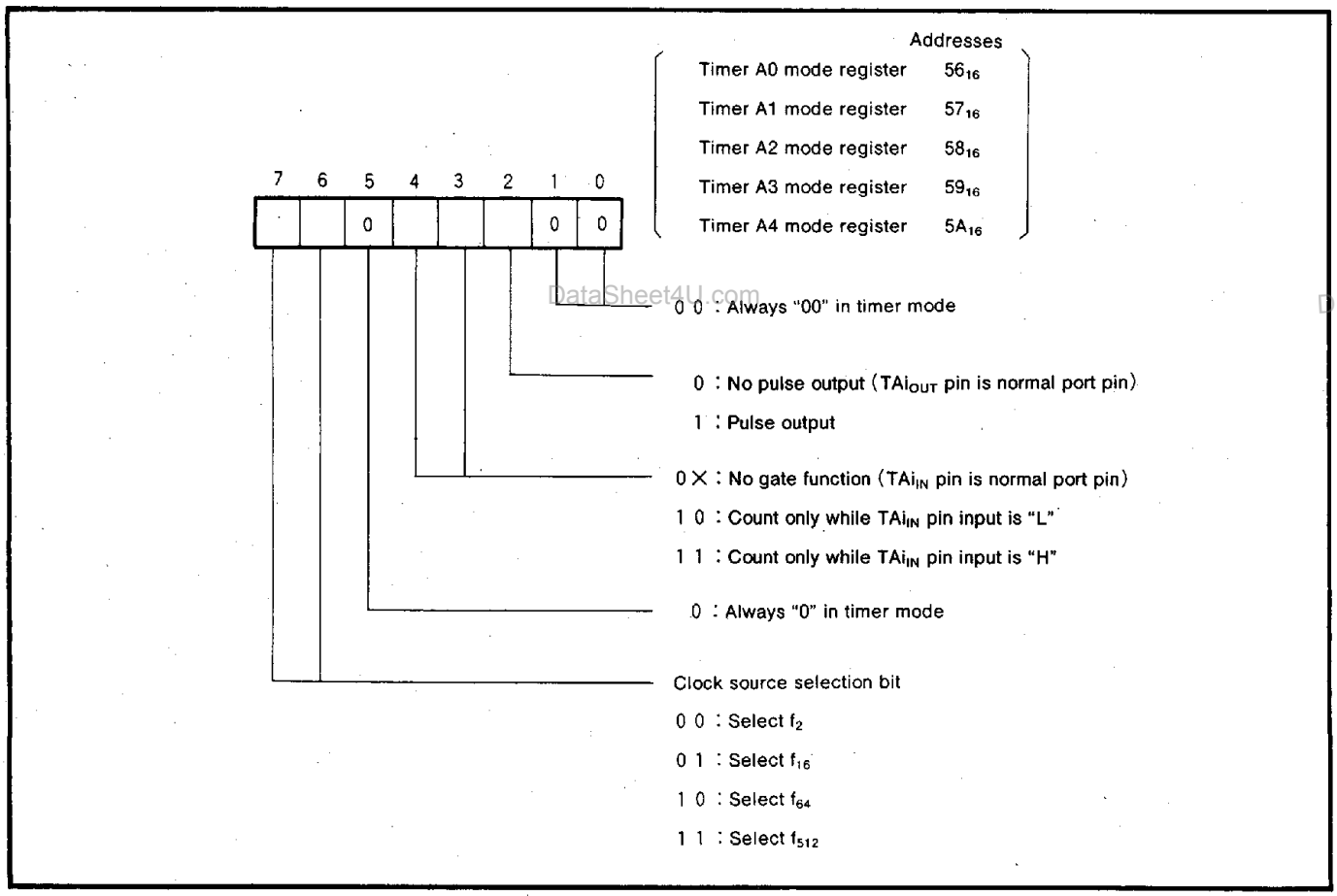


Fig. 12 Timer Ai mode register bit configuration during timer mode

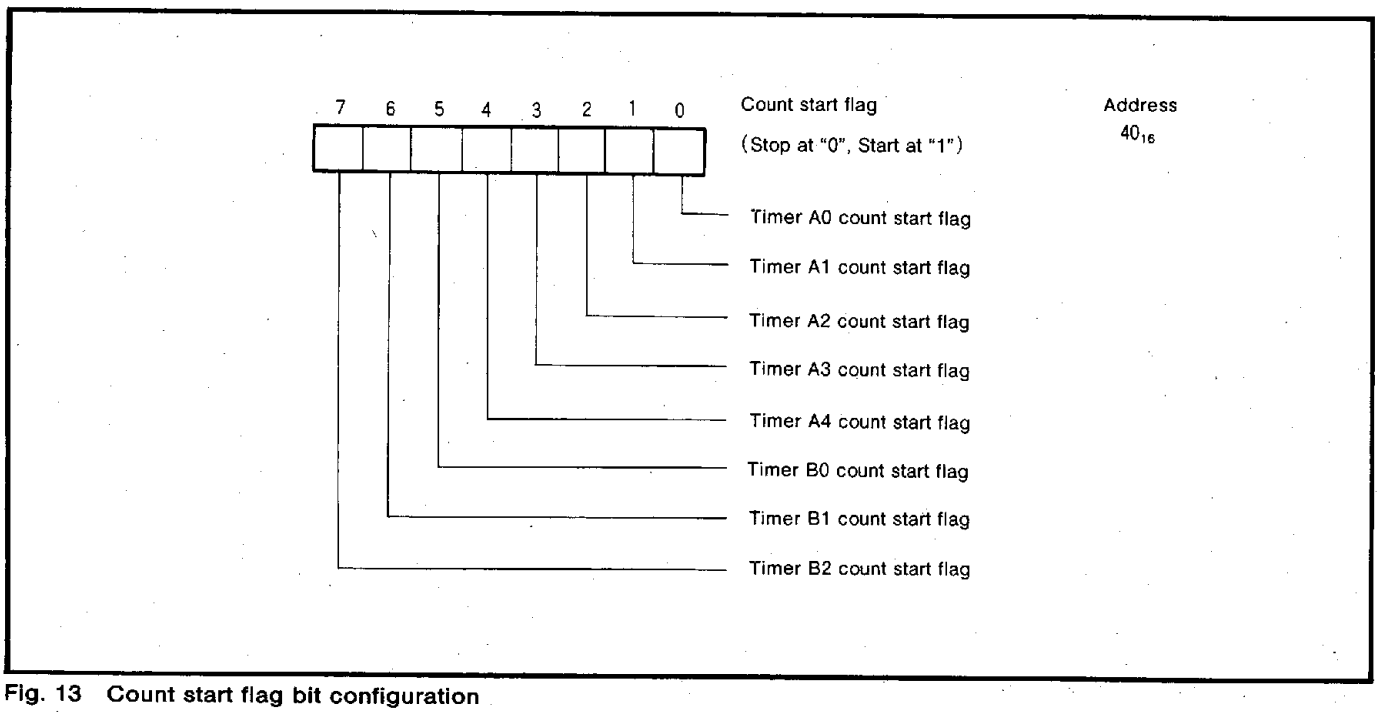


Fig. 13 Count start flag bit configuration

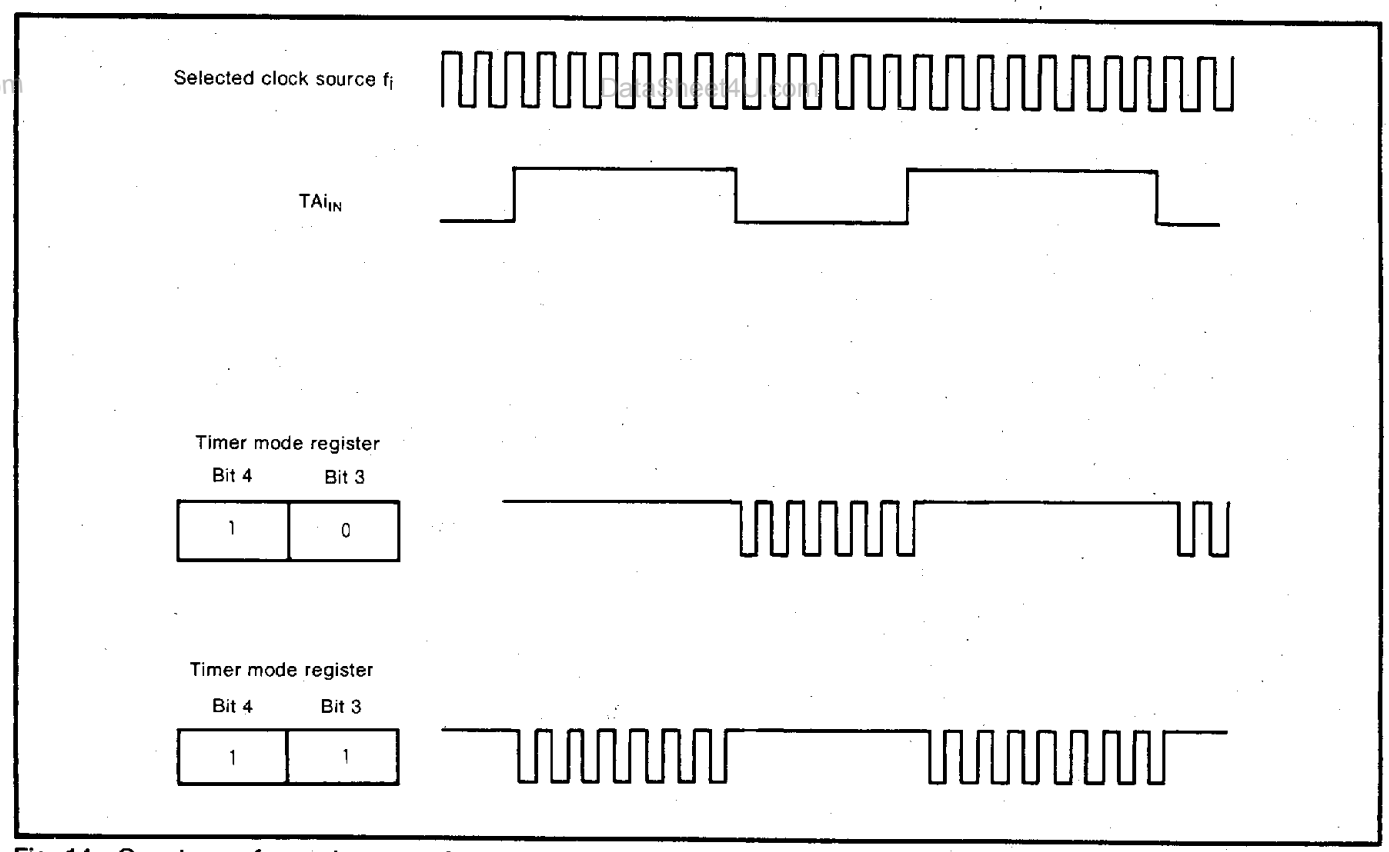


Fig. 14 Count waveform when gate function is available

(2) Event counter mode [01]

Figure 15 shows the bit configuration of the timer Ai mode register during event counter mode. In event counter mode, the bit 0 of the timer Ai mode register must be "1" and bit 1 and 5 must be "0".

The input signal from the TAi_{IN} pin is counted when the count start flag shown in Figure 13 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 flag or the input signal from the TAi_{OUT} pin.

When bit 4 of the timer Ai mode register is "0", the up-down flag is used to determine whether to increment or decrement the count (decrement when the flag is "0" and increment when it is "1"). Figure 16 shows the bit configuration of the up-down flag.

When bit 4 of the timer Ai mode register is "1", the input signal from the TAi_{OUT} 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" because if bit 2 is "1", TAi_{OUT} pin becomes an output pin with pulse output.

The count is decremented when the input signal from the TAi_{OUT} pin is "L" and incremented when it is "H". Determine the level of the input signal from the TAi_{OUT} pin before valid edge is input to the TAi_{IN} 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, timers A0 and A1 transfer the contents of the reload register to the counter and continue counting.

Timers A2, A3, and A4 transfer the contents of the reload register to the counter and continue count when bit 6 of the corresponding timer Ai mode register is "0", but when bit 6 is "1", they continue counting without transferring the contents of the reload register to the counter.

When bit 2 is "1", the waveform reversing polarity is output from TAi_{OUT} pin each time the counter reaches 0000_{16} (decrement count) or $FFFF_{16}$ (increment count). If bit 2 is "0", TAi_{OUT} pin can be used as a normal port pin. However, if bit 4 is "1" and the TAi_{OUT} 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 TAi_{OUT} pin is not 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 which is 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 and continues counting. For timer

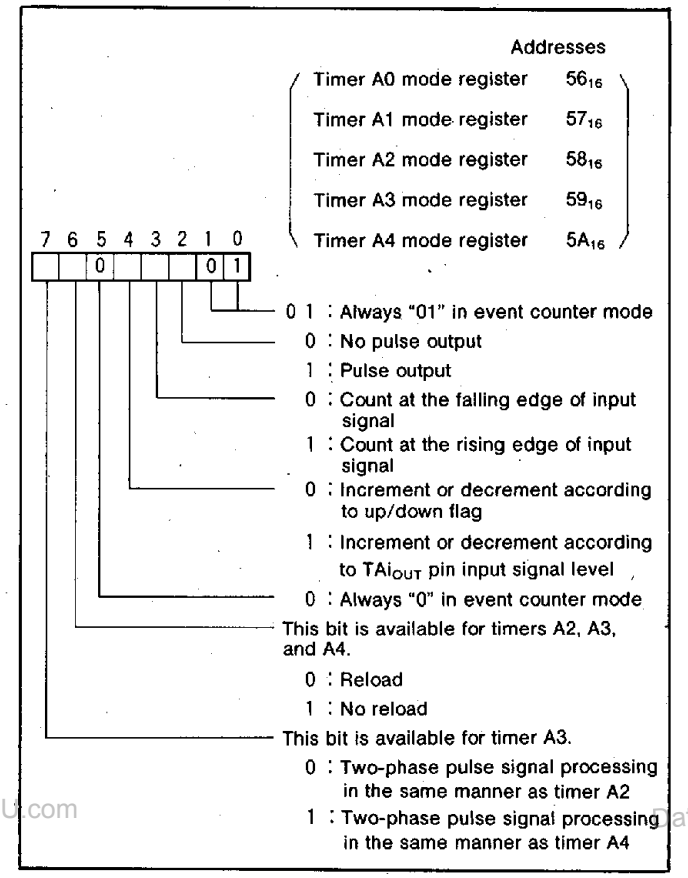


Fig. 15 Timer Ai mode register bit configuration during event counter mode

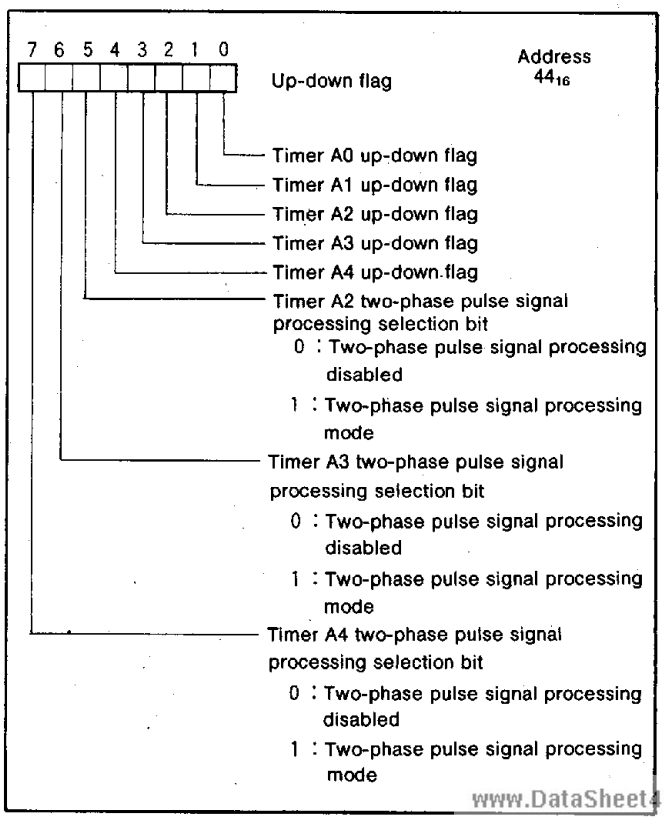


Fig. 16 Up-down flag bit configuration

A2, A3, and A4, the contents of the reload register are not reloaded in the counter when bit 6 of the corresponding timer Ai mode register is "1". The contents of the counter can be read at any time.

Furthermore, in event counter mode, whether to increment or decrement the counter can also be determined by supplying two-phase pulse input with phase shifted by 90° to timer A2, A3, or A4. There are two types of two-phase pulse processing operations. One uses timer A2 and the other uses timer A4. Timer A3 can select one of these two operations with bit 7 of the timer A3 mode register. In either processing operation, two-phase pulse is input in the same way, that is, pulses out of phase by 90° are input at the TAj_{OUT} (j = 2 to 4) pin and TAj_{IN} pin.

When timer A2 is used, as shown in Figure 17, the count is incremented when a rising edge is input to the $TA2_{IN}$ pin after the level of $TA2_{OUT}$ pin changes from "L" to "H", and when the falling edge is input, the count is decremented.

For timer A4, as shown in Figure 18, when a phase related pulse with a rising edge input to the $TA4_{IN}$ pin is input after the level of $TA4_{OUT}$ pin changes from "L" to "H", the count is incremented at the respective rising edge and falling edge of the $TA4_{OUT}$ pin and $TA4_{IN}$ pin. When a phase related pulse with a falling edge input to the $TA4_{OUT}$ pin is input after the level of $TA4_{IN}$ pin changes from "H" to "L", the count is decremented at the respective rising edge and falling edge of the $TA4_{IN}$ pin and $TA4_{OUT}$ pin.

When performing this two-phase pulse signal processing, timer Aj mode register bit 0 and bit 4 must be set to "1" and bits 1, 2, 3, and 5 must be set to "0" as shown in Figure 19.

Bit 7 is used to select whether to perform two-phase pulse signal processing for timer A3 in the same manner as timer A2 or as timer A4. When this bit is "0", two-phase pulse signal processing for timer A3 is performed in the same manner as timer A2 and when it is "1", it is performed in the same manner as timer A4. This bit is ignored for timer A2 and A4.

Note that bits 5, 6, and 7 of the up-down flag (address 44_{16}) are the two-phase pulse signal processing selection bit for timer A2, A3, and A4 respectively.

Each timer operates in normal event counter mode when the corresponding bit is "0" and performs two-phase pulse signal processing when it is "1".

Count is started by setting the count start flag 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-phase pulse signal is input. Also, there can be no pulse output in this mode.

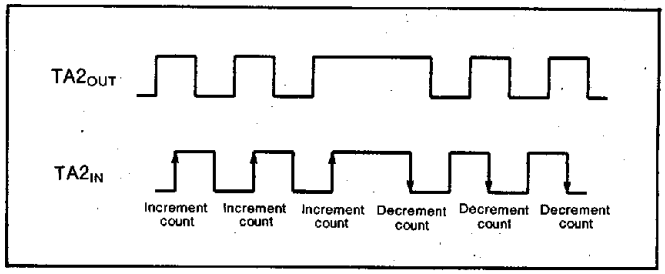


Fig. 17 Two-phase pulse processing operation of timer A2

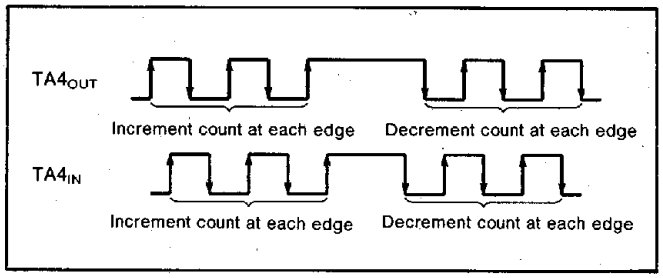


Fig. 18 Two-phase pulse processing operation of timer A4

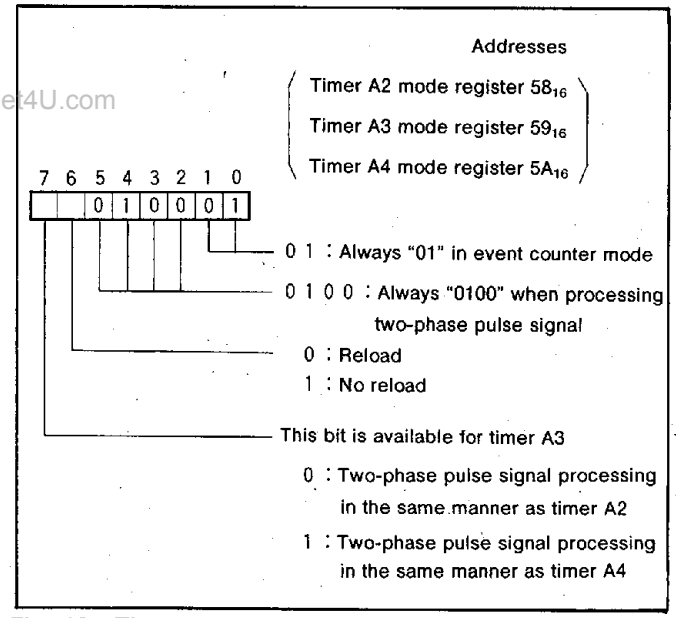


Fig. 19 Timer Aj mode register bit configuration when performing two-phase pulse signal processing in event counter mode

(3) One-shot pulse mode (10)

Figure 20 shows the bit configuration of the timer Ai mode register during one-shot pulse mode. In one-shot pulse mode, bit 0 and bit 5 must be "0" and bit 1 and bit 2 must be "1".

The trigger is enabled when the count start flag is "1". The trigger can be generated by software or it can be input from the TAi_{IN} pin. Software trigger is selected when bit 4 is "0" and the input signal from the TAi_{IN} 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 the bit in the one-shot start flag corresponding to each timer.

Figure 21 shows the bit configuration of the one-shot start flag.

As shown in Figure 22, when a trigger signal is received, the counter counts the clock selected by bits 6 and 7.

If the contents of the counter is not 0000_{16}, the TAi_{OUT} pin goes "H" when a trigger signal is received. The count direction is decrement.

When the counter reaches 0001_{16}, The TAi_{OUT} 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", TAi_{OUT} goes "L". Therefore, the value corresponding to the desired pulse width must be written to timer Ai before setting the timer Ai count start flag.

As shown in Figure 23, 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 is 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 to the same way as for timer mode. When data is written in timer Ai halted, it is also written to the reload register and the counter.

When data is written to timer Ai which is busy, the data is written to the reload register, but not to the counter. The counter is reloaded with new data from the reload register at the next reload time.

Undefined data is read when timer Ai is read.

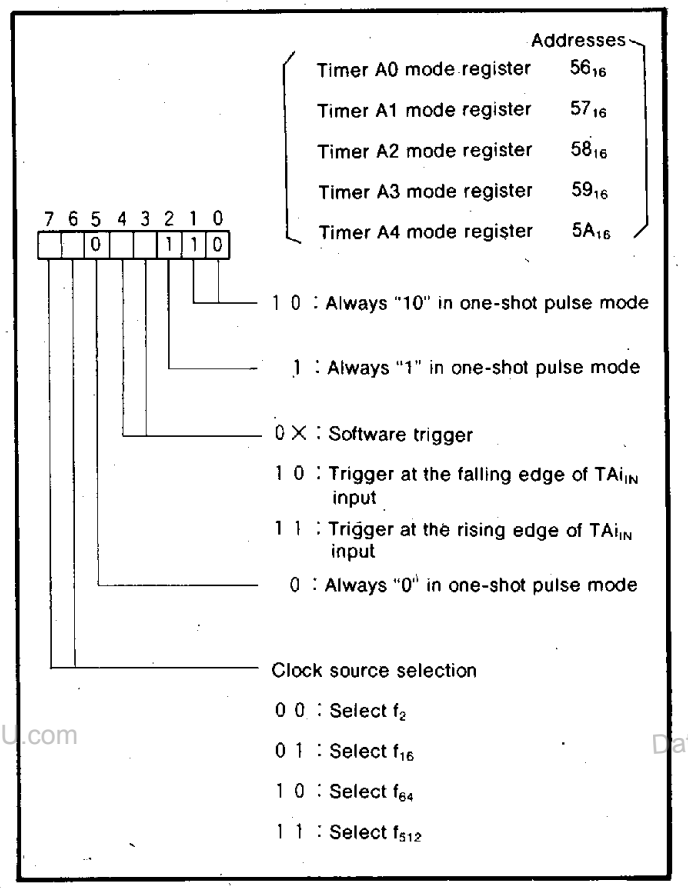


Fig. 20 Timer Ai mode register bit configuration during one-shot pulse mode

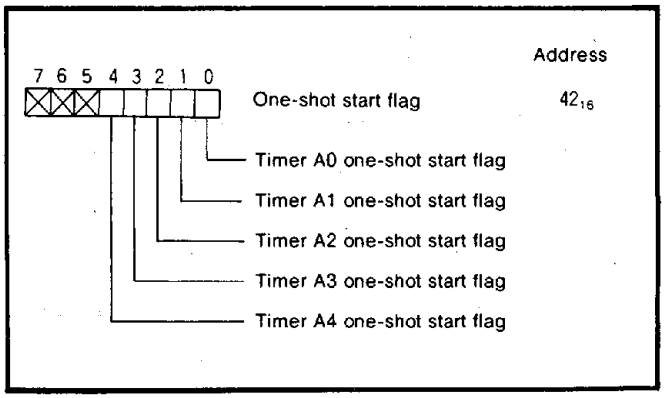


Fig. 21 One-shot start flag bit configuration

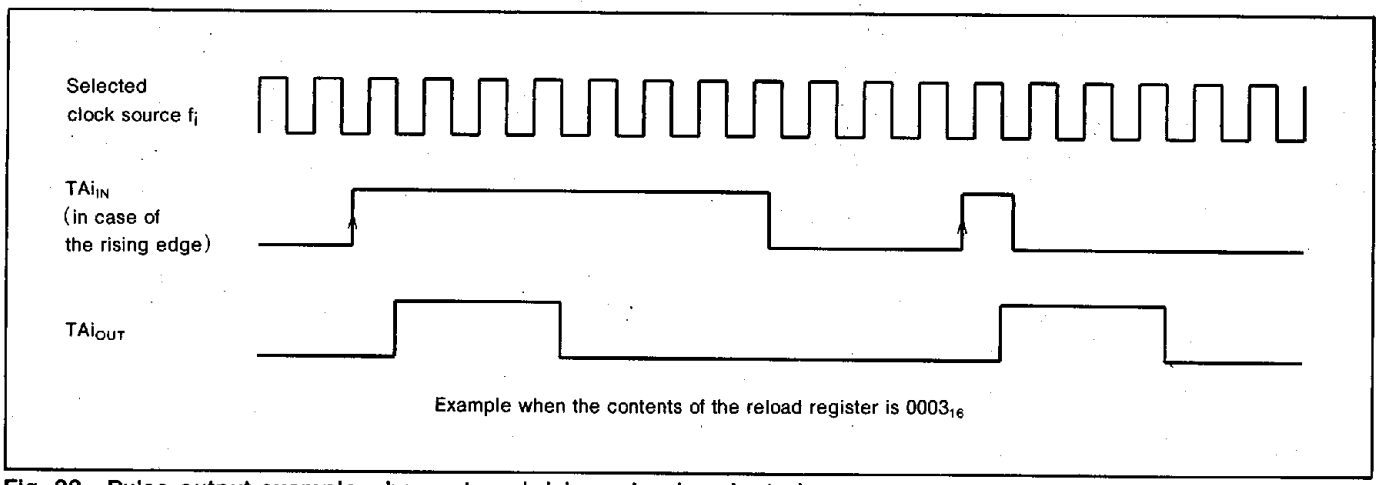


Fig. 22 Pulse output example when external rising edge is selected

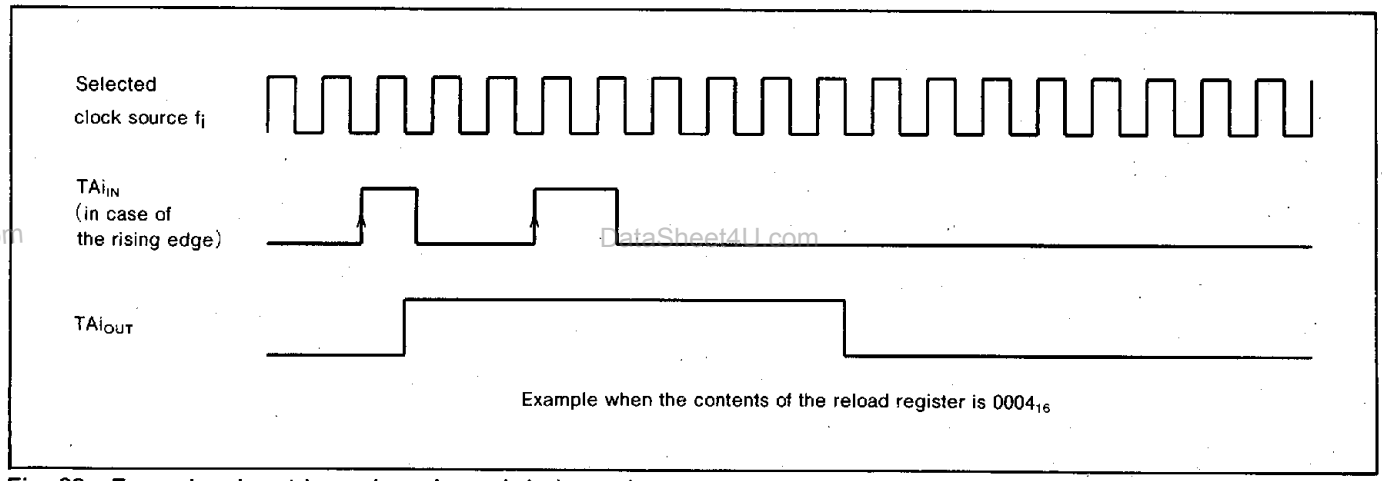


Fig. 23 Example when trigger is re-issued during pulse output

(4) Pulse width modulation mode (11)

Figure 24 shows the bit configuration of the timer Ai mode register during pulse width modulation mode. In pulse width modulation mode, bits 0, 1, and 2 must be set to "1".

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

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

The software trigger mode is selected when bit 4 is "0". Pulse width modulator is started and pulse is output from TAi_{OUT} when the timer Ai start flag is set to "1".

The external trigger mode is selected when bit 4 is "1". Pulse width modulator starts when a trigger signal is input from the TAi_{IN} pin when the timer Ai start flag 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 timer Ai start flag is set to "1" and a software trigger or an external trigger is issued to start modulation, the waveform shown in Figure 25 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 and 7. A pulse is generated when the counter reaches 0000_{16} as shown in Figure 26. At the same time, the contents of the reload register is transferred to the counter and count is continued.

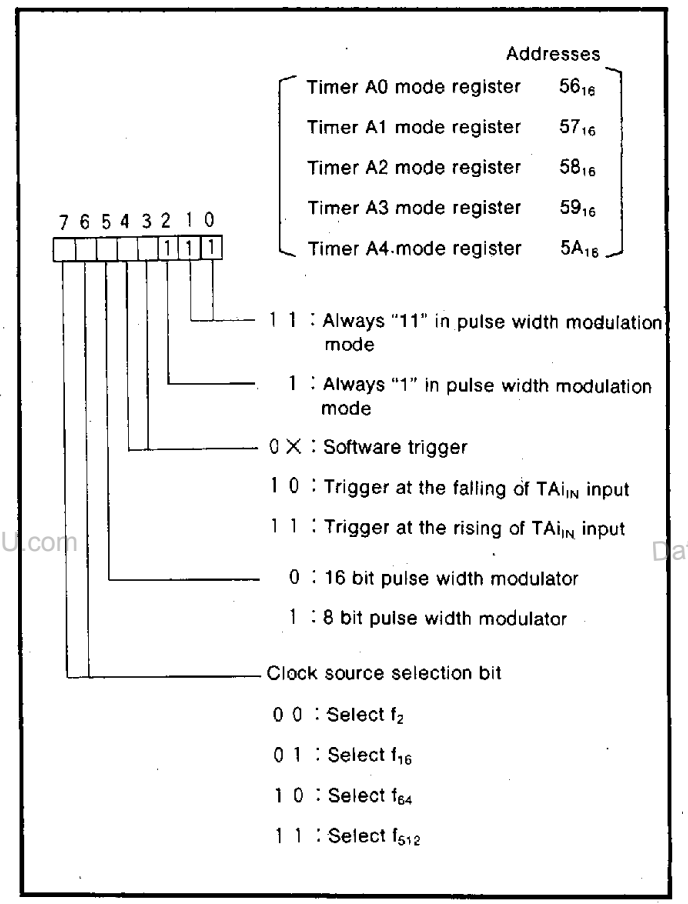


Fig. 24 Timer Ai mode register bit configuration during pulse width modulation mode

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

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

The high order 8-bit 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-bit of the reload register is m, the duration "H" of pulse is

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

And the output pulse period is

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

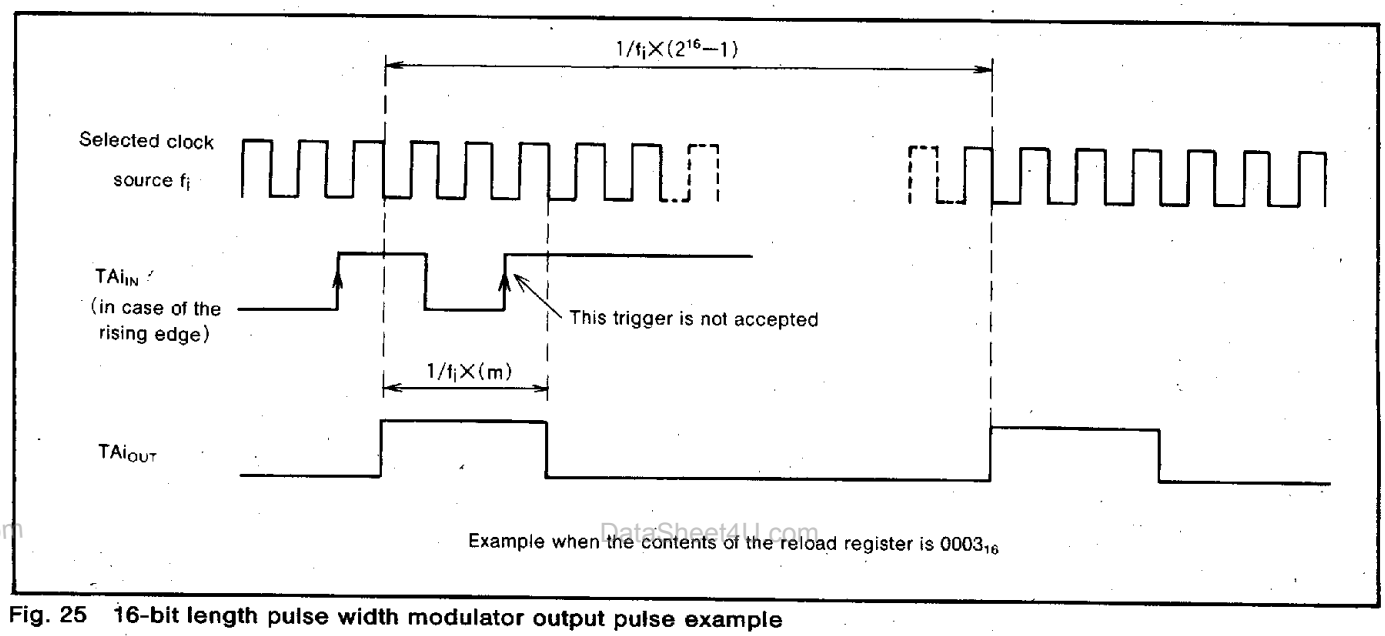


Fig. 25 16-bit length pulse width modulator output pulse example

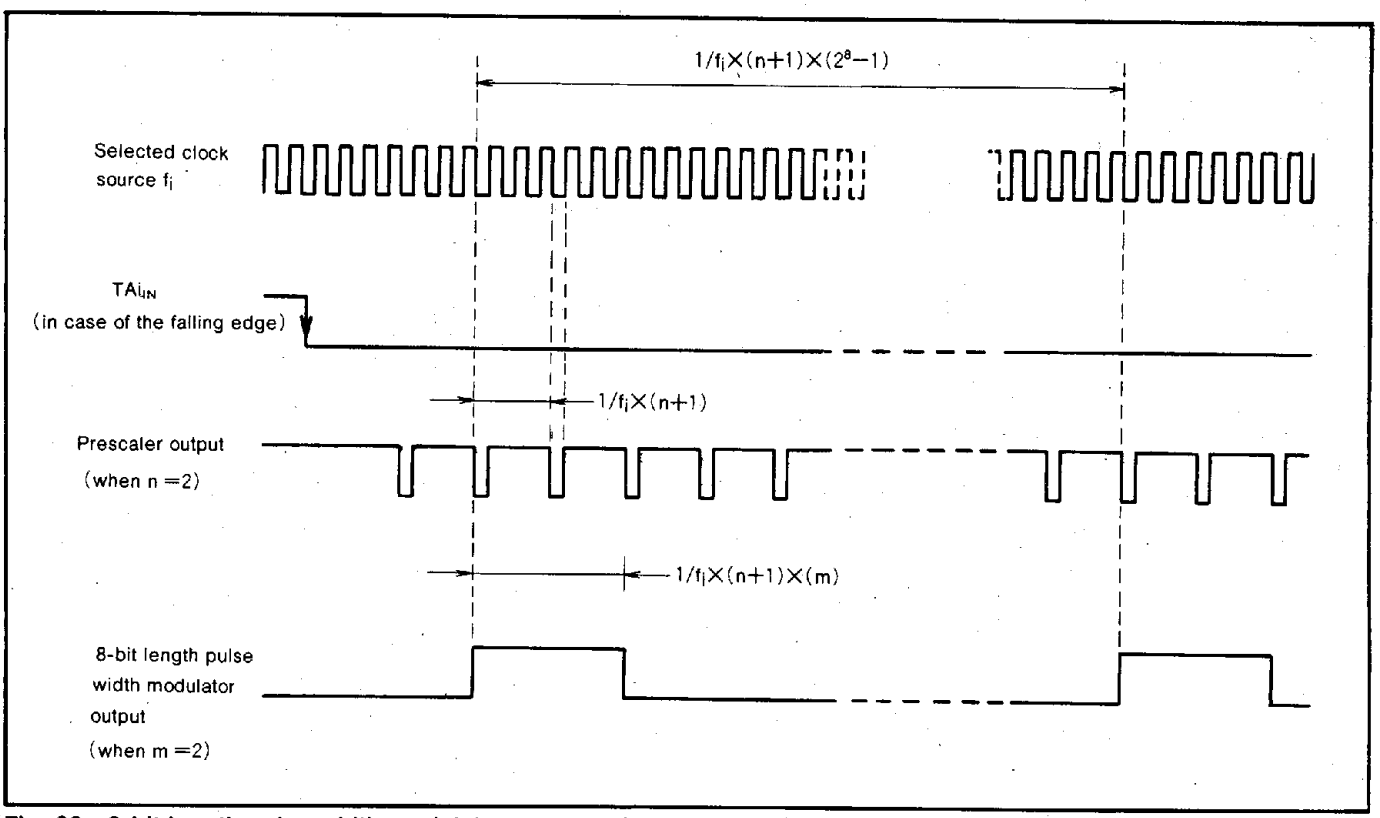


Fig. 26 8-bit length pulse width modulator output pulse example

TIMER B

Figure 27 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 28 shows the bit configuration of the timer Bi mode register during timer mode. Bits 0, and 1 of the timer Bi mode register must always be "0" in timer mode.

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

As shown in Figure 13, the timer Bi count start flag is at the same address as the timer Ai count start flag. 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 counter is reloaded with new data from the reload register at the next reload time. The contents of the counter can be read at any time.

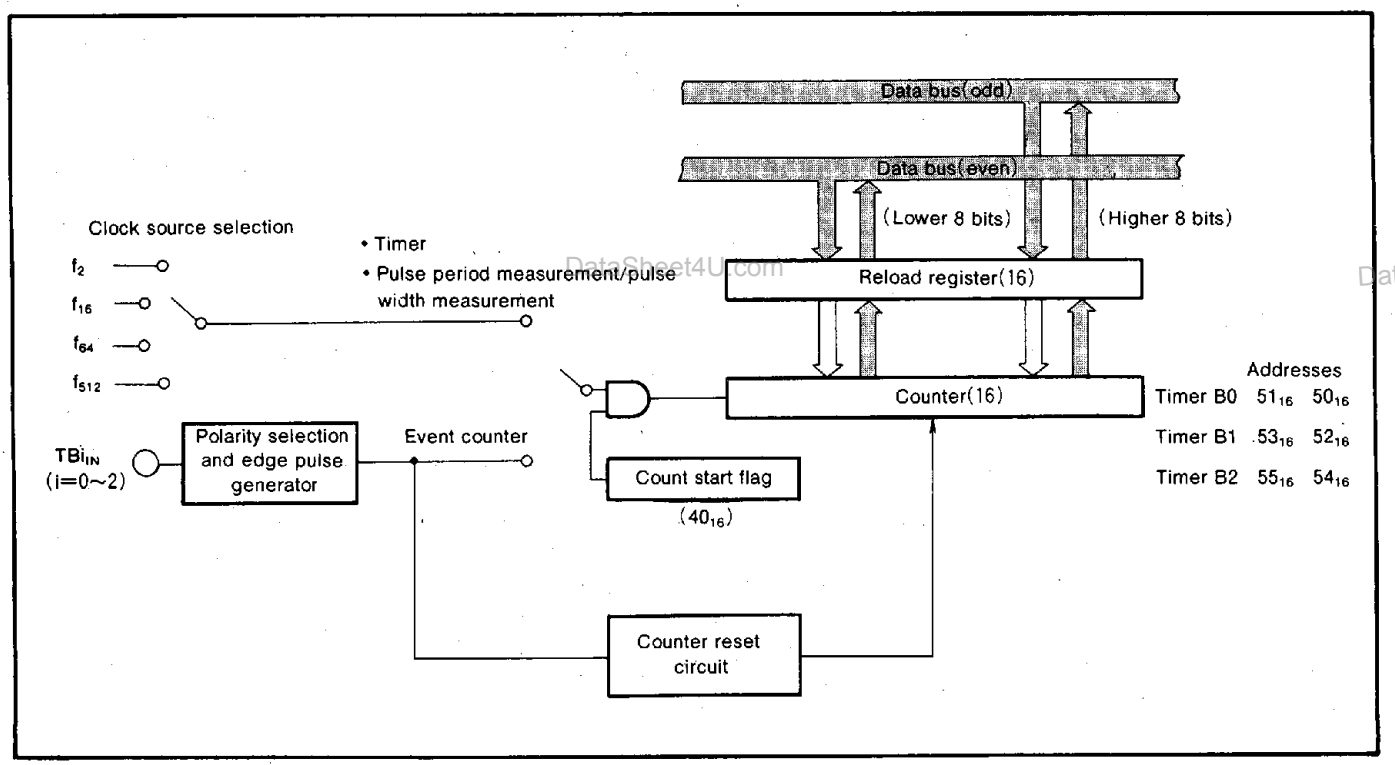


Fig. 27 Timer B block diagram

(2) Event counter mode [01]

Figure 29 shows the bit configuration of the timer Bi mode register during event counter mode. In event counter mode, the 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 flag is "1" and counting is stopped when it is "0". Count is performed at the fall of the input signal when bits 2, and 3 are "0" and at the rise of the input signal when bit 3 is "0" and bit 2 is "1".

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

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

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

Figure 30 shows the bit configuration of the timer Bi mode register during pulse period measurement/pulse width measurement mode.

In pulse period measurement/pulse width measurement mode, bit 0 must be "0" and bit 1 must be "1". Bits 6 and 7 are used to select the clock source. The selected clock is counted when the count start flag is "1" and counting stops when it is "0".

The pulse period measurement mode is selected when bit 3 is "0". In pulse period measurement mode, the selected clock is counted during the interval starting at the fall of the input signal from the TBi_{IN} pin to the next fall or at the rise of the input signal to the next rise and 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 31, 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 time to the reload register after the count start flag is set to "1".

When bit 3 is "1", the pulse width measurement mode is selected. Pulse width measurement mode is similar to 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 32.

When timer Bi is read, the contents of the reload register is read.

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

Timer Bi overflow flag which is bit 5 of timer Bi mode register is set to "1" when the timer Bi counter reaches 0000_{16}. This flag is cleared by writing to corresponding timer Bi mode register. This bit is set to "1" at reset.

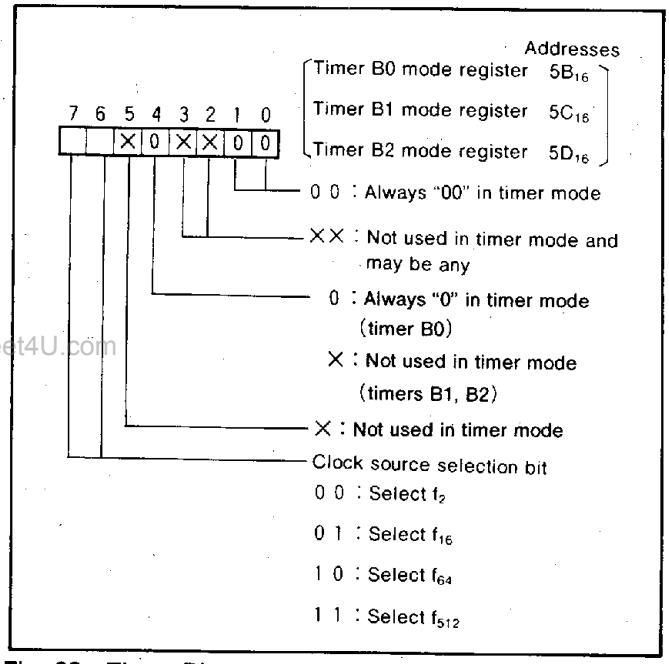


Fig. 28 Timer Bi mode register bit configuration during timer mode

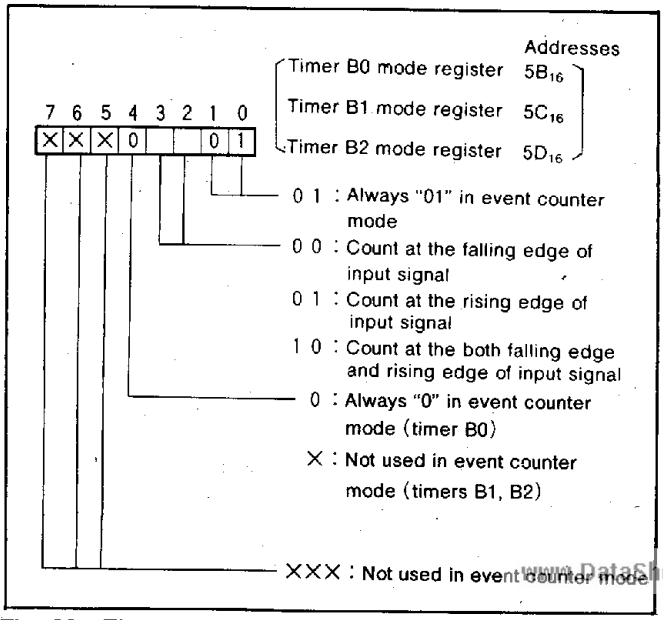


Fig. 29 Timer Bi mode register bit configuration during event counter mode

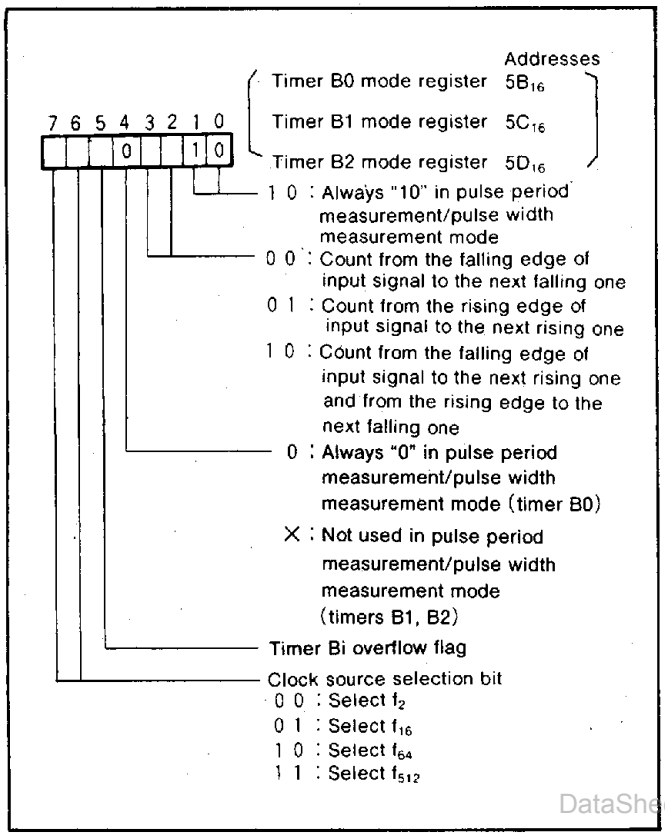


Fig. 30 Timer Bi mode register bit configuration during pulse period measurement/pulse width measurement mode

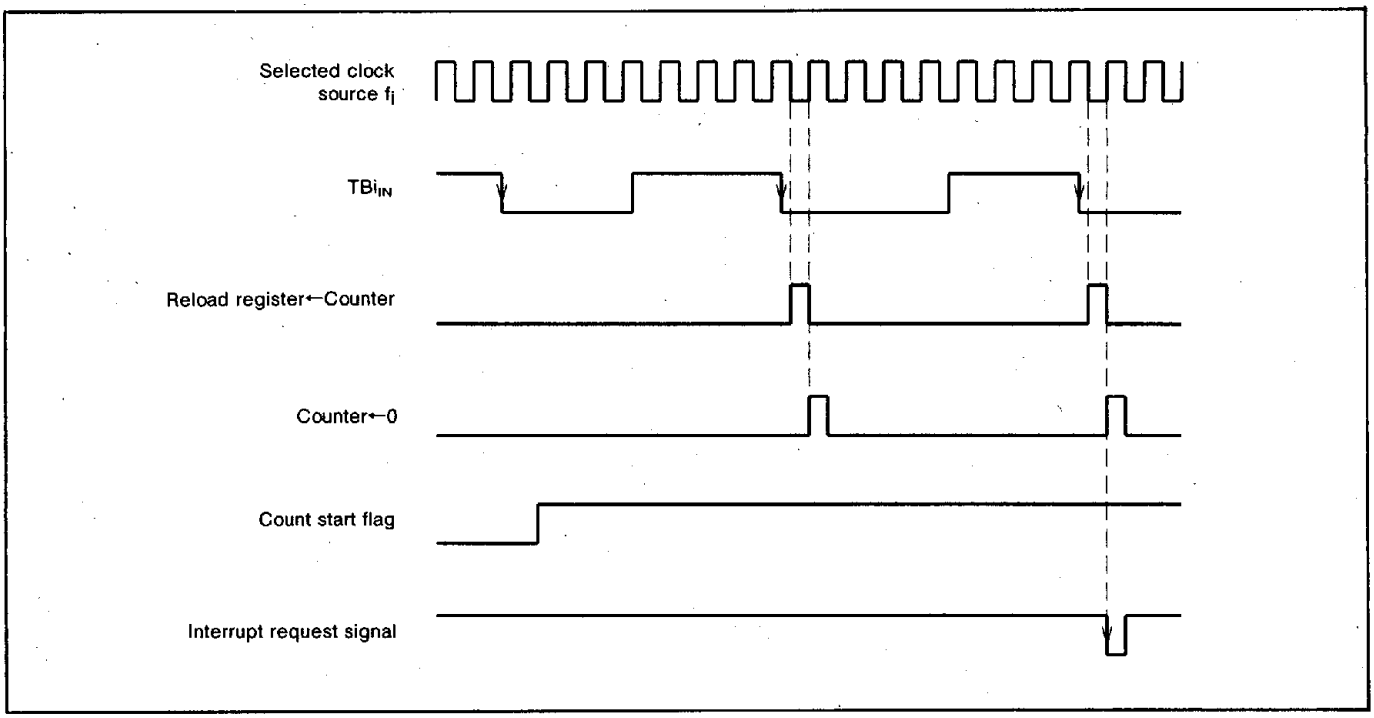


Fig. 31 Pulse period measurement mode operation (example of measuring the interval between the falling edge to next falling one)

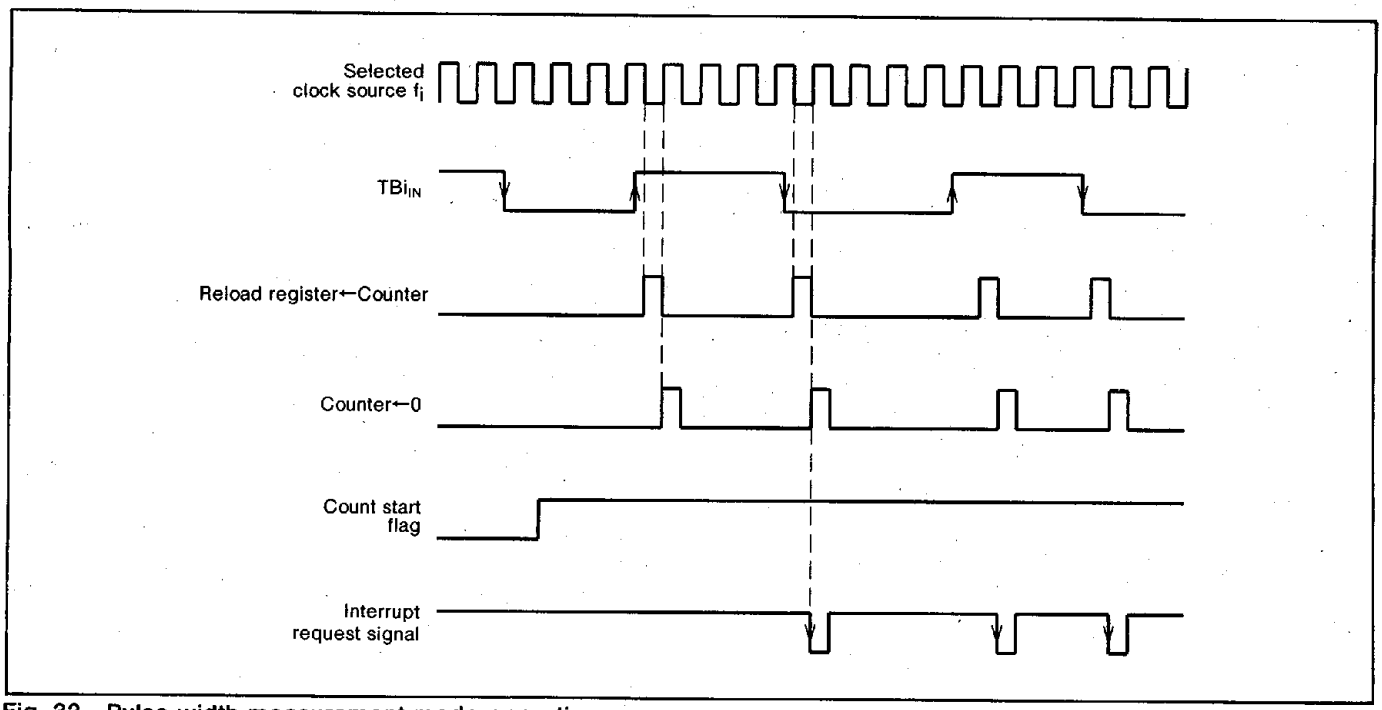


Fig. 32 Pulse width measurement mode operation

PULSE OUTPUT PORT MODE

The M37710 can use several built-in timer As to output pulse motor drive waveform. This mode is described below. Figure 33 shows a block diagram for pulse output port mode. In the pulse output port mode, two pairs of 4-bit pulse output ports are used. There are three combinations of pulse output ports. When bit 0 of the waveform output selection bit in waveform output mode register (address 62_{16}) shown in Figure 34 is set to "1" and bit 1 is set to "0", $RTP1_3$, $RTP1_2$, $RTP1_1$, and $RTP1_0$ become pulse output ports. When bit 1 of waveform output selection bit in waveform output mode register is set to "1" and bit 0 is set to "0", $RTP0_3$, $RTP0_2$, $RTP0_1$, and $RTP0_0$ become pulse output ports. The ports not used as pulse output ports can be used as normal parallel ports or timer I/O pins. When bit 0 and bit 1 of pulse output selection bit are set to "1", the two four-line sets $RTP1_3$, $RTP1_2$, $RTP1_1$, and $RTP1_0$ and $RTP0_3$, $RTP0_2$, $RTP0_1$, and $RTP0_0$ become pulse output ports.

In the pulse output port mode, set timers A2 and A0 to timer mode as timers A2 and A0 are used. In addition, set bit 2 of the corresponding timer Ai mode register to "1" when using $RTP0_0$, $RTP0_2$, $RTP1_0$, $RTP1_2$, and $RTP1_3$ as pulse output port because they are shared with TAi_{OUT} (i = 0 to 4) pins. Figure 35 shows the bit configuration of timer A0, A2 mode registers in pulse output port mode.

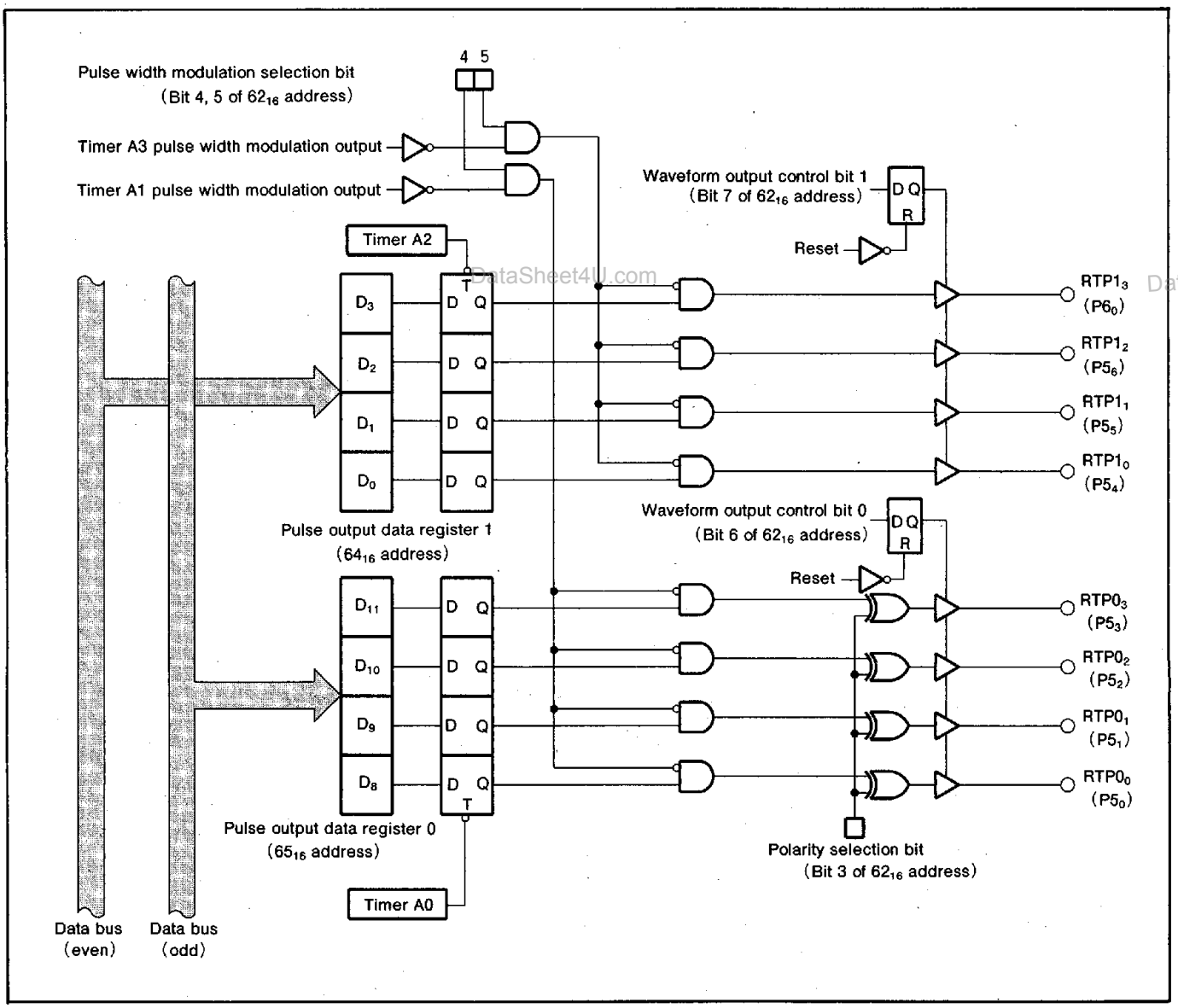


Fig. 33 Block diagram for pulse output port mode

Data can be set in each bit of the pulse output data register corresponding to four ports selected as pulse output ports. Figure 36 shows the bit configuration of the pulse output data register.

The contents of the pulse output data register 1 (low-order four bits of address 64_{16}) corresponding to ports $RTP1_3$, $RTP1_2$, $RTP1_1$, and $RTP1_0$ are output to the ports each time the counter of timer A2 becomes 0000_{16}. The contents of the pulse output data register 0 (low-order four bits of address 65_{16}) corresponding to ports $RTP0_3$, $RTP0_2$, $RTP0_1$, and $RTP0_0$ are output to the ports each time the counter of timer A0 becomes 0000_{16}.

Each timer starts counting when the content of the corresponding timer count start flag (address 40_{16}) is set to "1" and stops counting when it is set to "0". When "0" is written to a specified bit of the pulse output data register, "L" level is output to the corresponding pulse output port when the counter of corresponding timer becomes 0000_{16}, and when "1" is written, "H" level is output to the pulse output port.

Pulse width modulation can be applied to each pulse output port. Since pulse width modulation involves the use of timers A3 and A1, activate these timers in pulse width modulation mode. When a certain bit of the pulse output register is "1", pulse width modulation is output from the pulse output port when the counter of the corresponding timer becomes 0000_{16}.

Ports $RTP1_3$, $RTP1_2$, $RTP1_1$, and $RTP1_0$ are applied pulse width modulation by setting the pulse width modulation selection bit by timer A3 (bit 5) of the waveform output mode register to "1".

Ports $RTP0_3$, $RTP0_2$, $RTP0_1$, and $RTP0_0$ are applied pulse width modulation by setting the pulse width modulation selection bit by timer A1 (bit 4) of the waveform output mode register to "1".

When outputting to pulse output ports $RTP0_3$, $RTP0_2$, $RTP0_1$, and $RTP0_0$, the output data can be reversed by the polarity selection bit (bit 3) of the waveform output mode register. When the polarity selection bit is "0", the contents of the pulse output data register 0 is output unchanged When the polarity selection bit is "1", the reversed data of the contents of the pulse output data register 0 is output. When pulse width modulation is applied, likewise the polarity reverse to pulse width modulation can be selected by the polarity selection bit. Figure 37 shows example of waveforms in pulse output port mode.

Ports selecting pulse output port mode can control the output of $RTP0_3$, $RTP0_2$, $RTP0_1$, and $RTP0_0$ with the waveform output control bit 0 (bit 6) and the output of $RTP1_3$, $RTP1_2$, $RTP1_1$, and $RTP1_0$ with waveform output control bit 1 (bit 7). When the waveform output control bit is set to "1", waveform is output from the corresponding port. When it is set to "0", the waveform output from the corresponding port is stopped and the port becomes floating. These bits can be cleared with an instruction and cleared by reset.

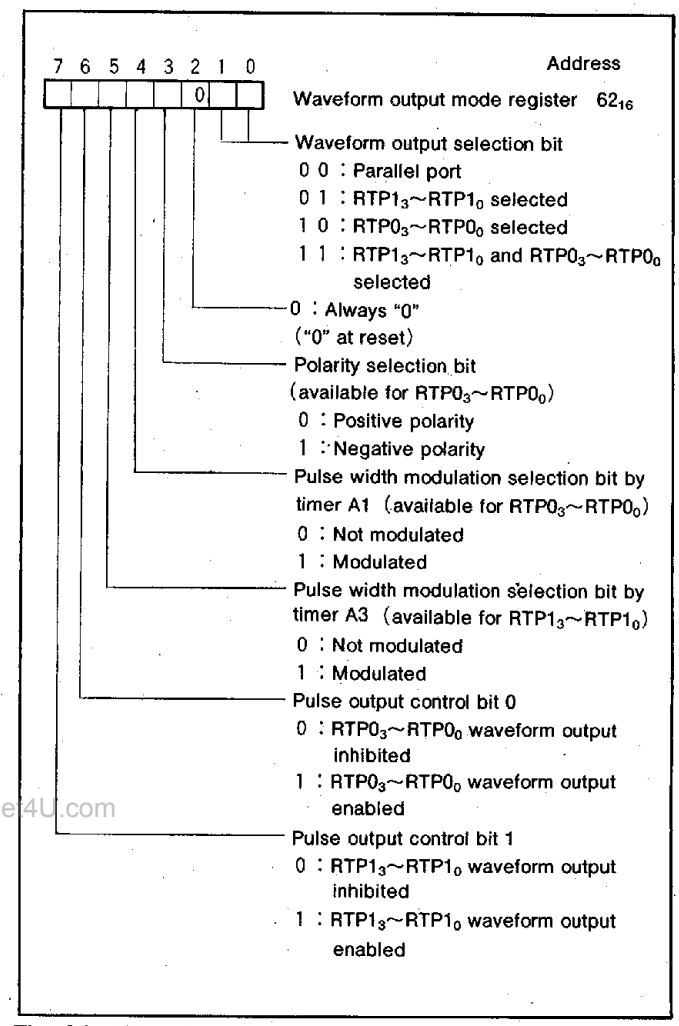


Fig. 34 Waveform output mode register bit configuration

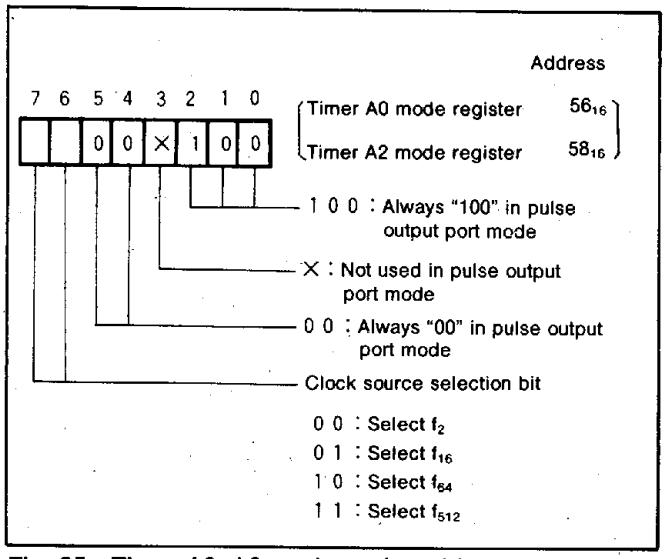


Fig. 35 Timer A0, A2 mode register bit configuration in pulse output port mode

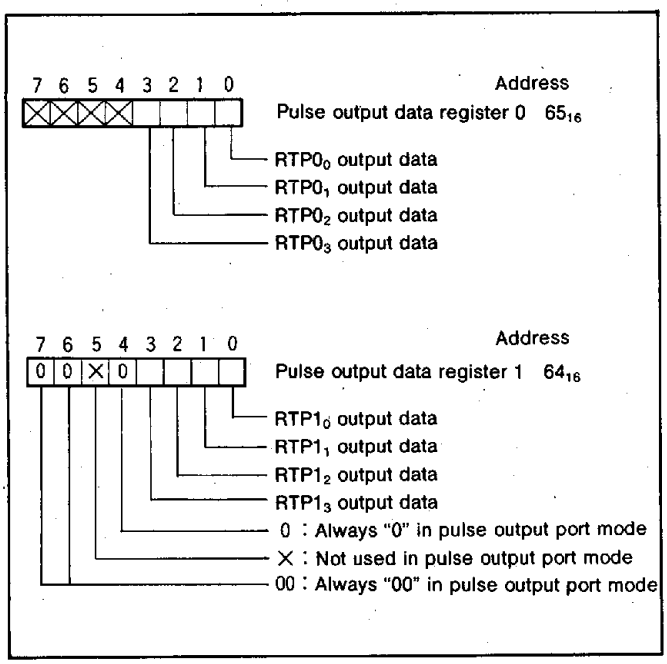


Fig. 36 Pulse output data register bit configuration

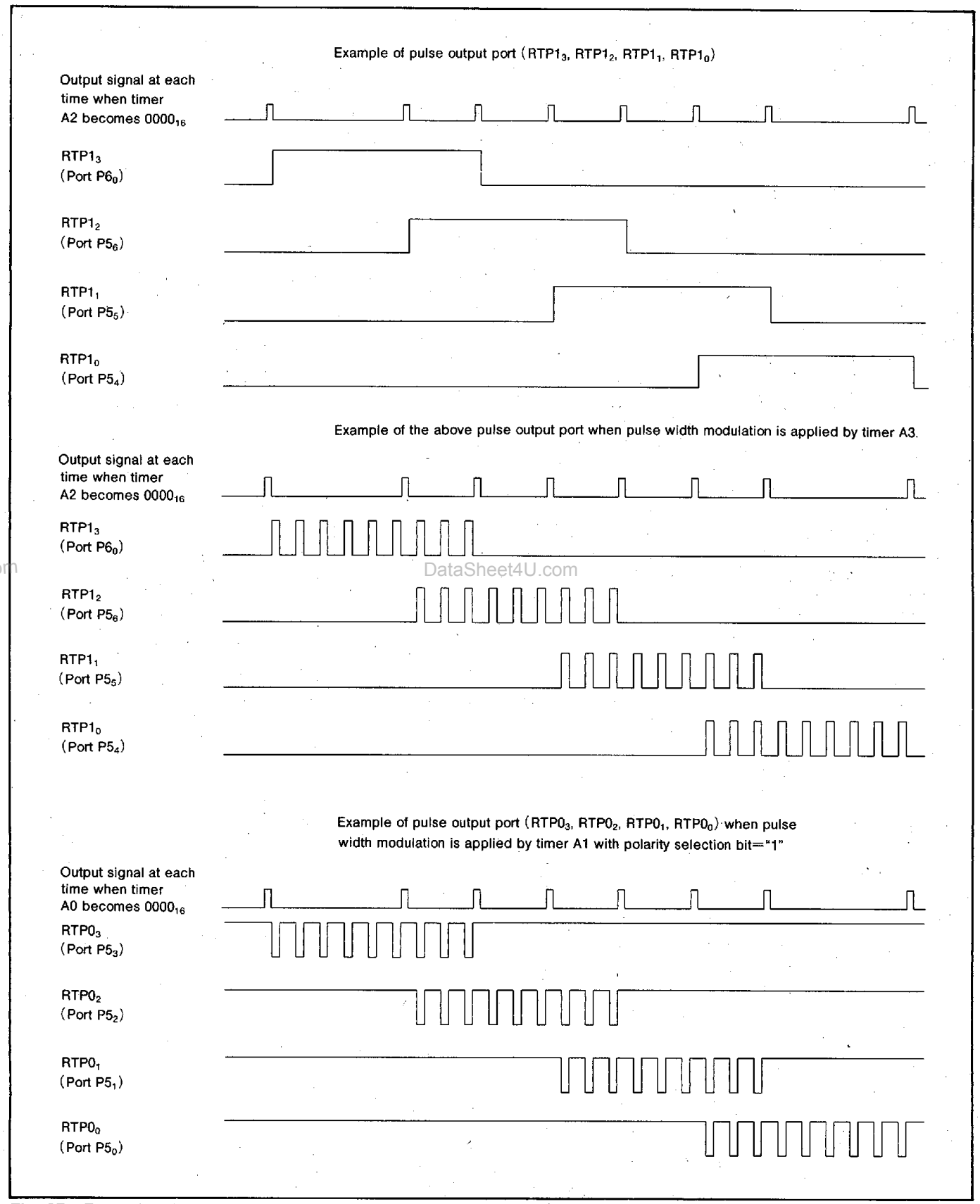


Fig. 37 Example of waveforms in pulse output port mode

SERIAL I/O PORTS

Two independent serial I/O ports are provided. Figure 38 shows a block diagram of the serial I/O ports.

Bits 0, 1, and 2 of the UARTi (i = 0, 1) Transmit/Receive mode register shown in Figure 39 are used to determine whether to use port P8 as parallel port, clock synchronous serial I/O port, or asynchronous (UART) serial I/O port using start and stop bits.

Figures 40 and 41 show the connections of receiver/transmitter according to the mode.

Figure 42 shows the bit configuration of the UARTi transmit/receive control register.

Each communication method is described below.

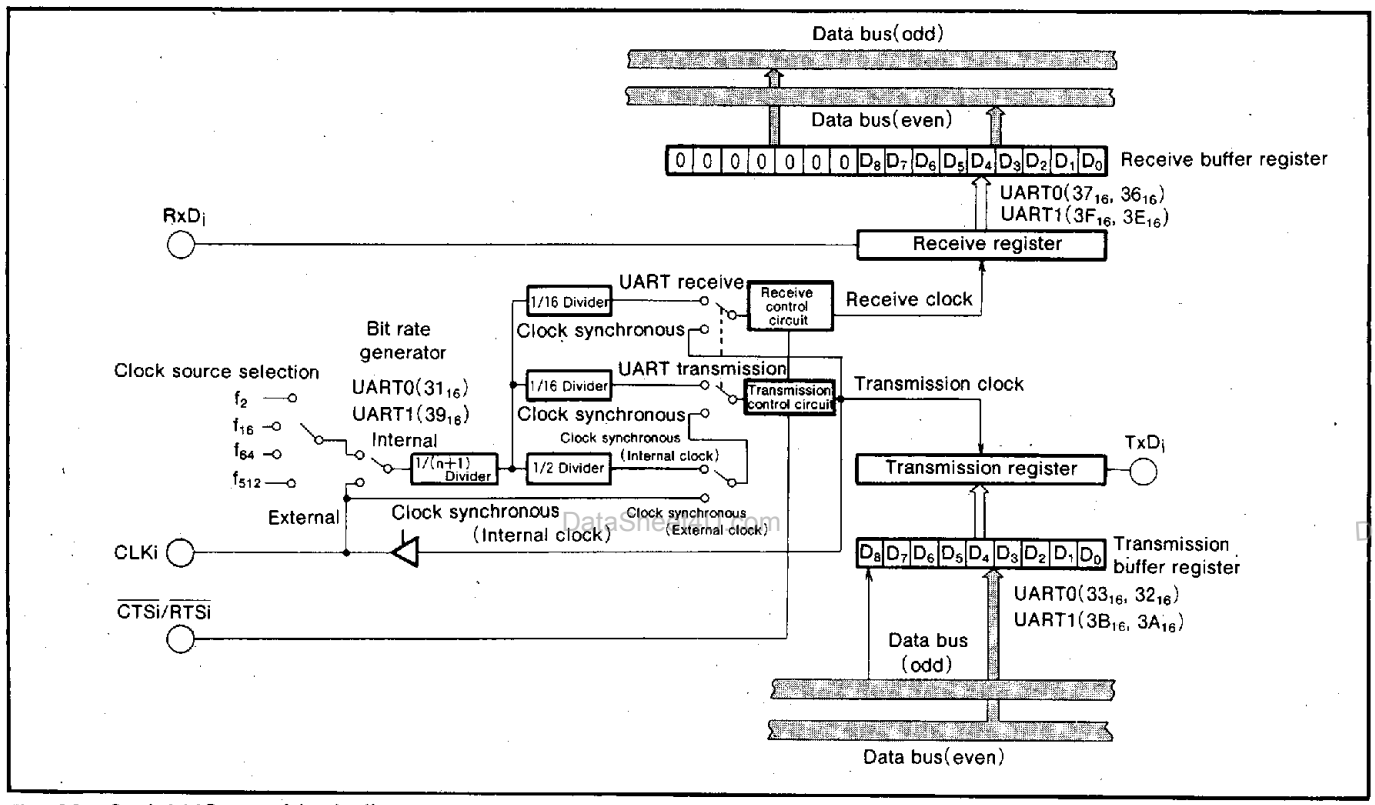


Fig. 38 Serial I/O port block diagram

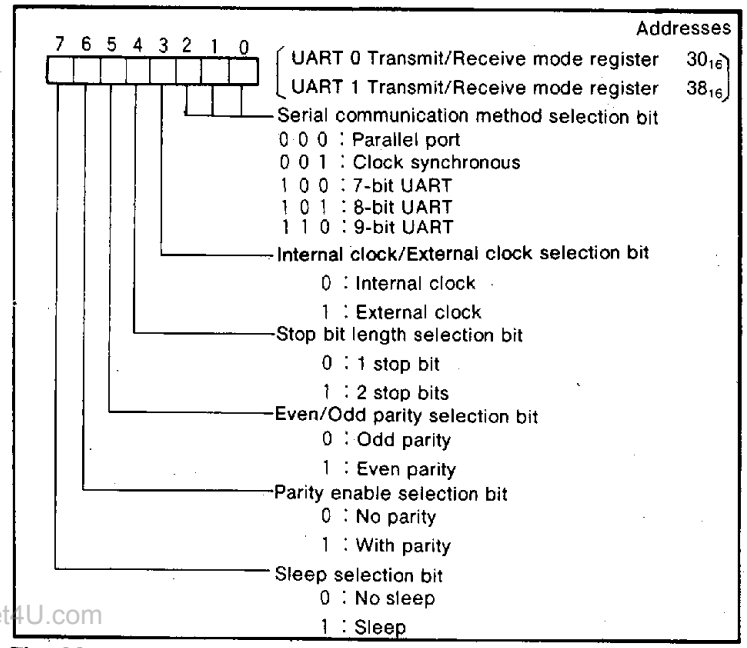


Fig. 39 UART i Transmit/Receive mode register bit configuration

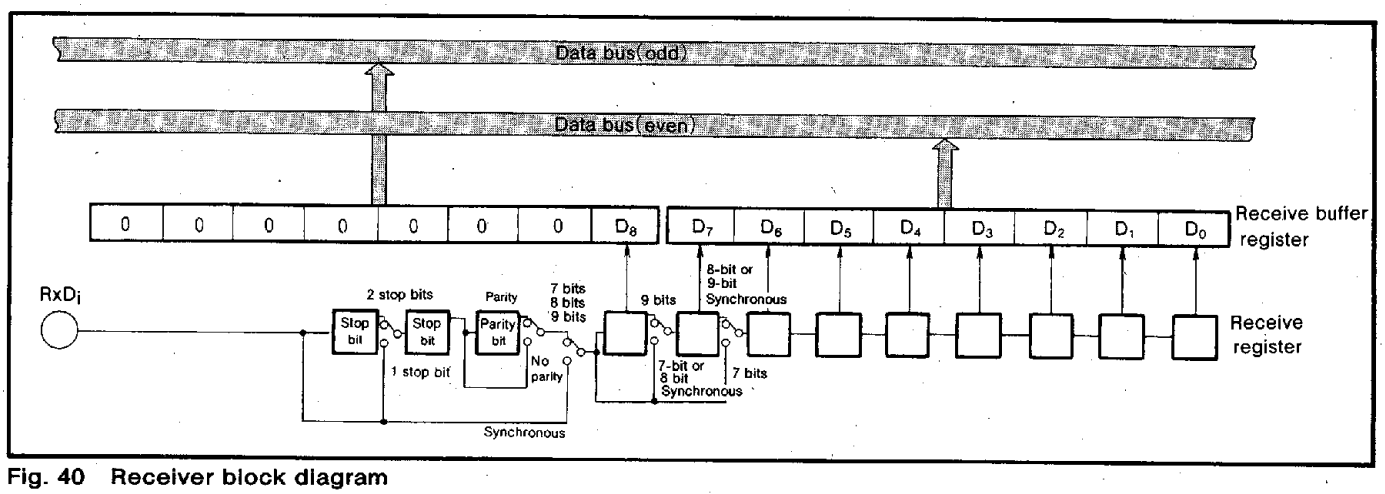


Fig. 40 Receiver block diagram

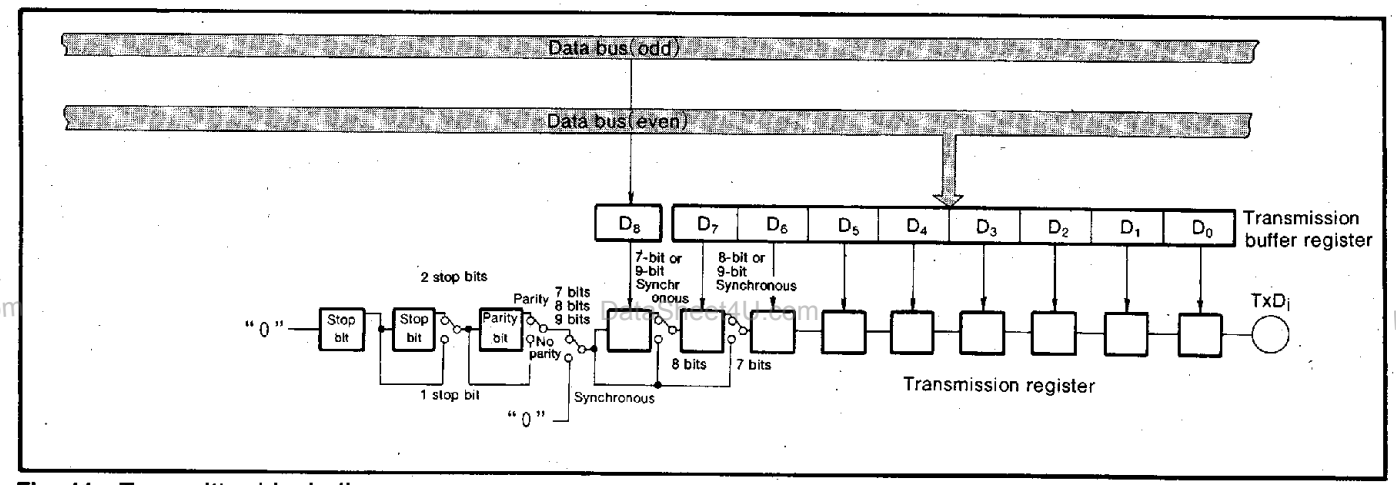


Fig. 41 Transmitter block diagram

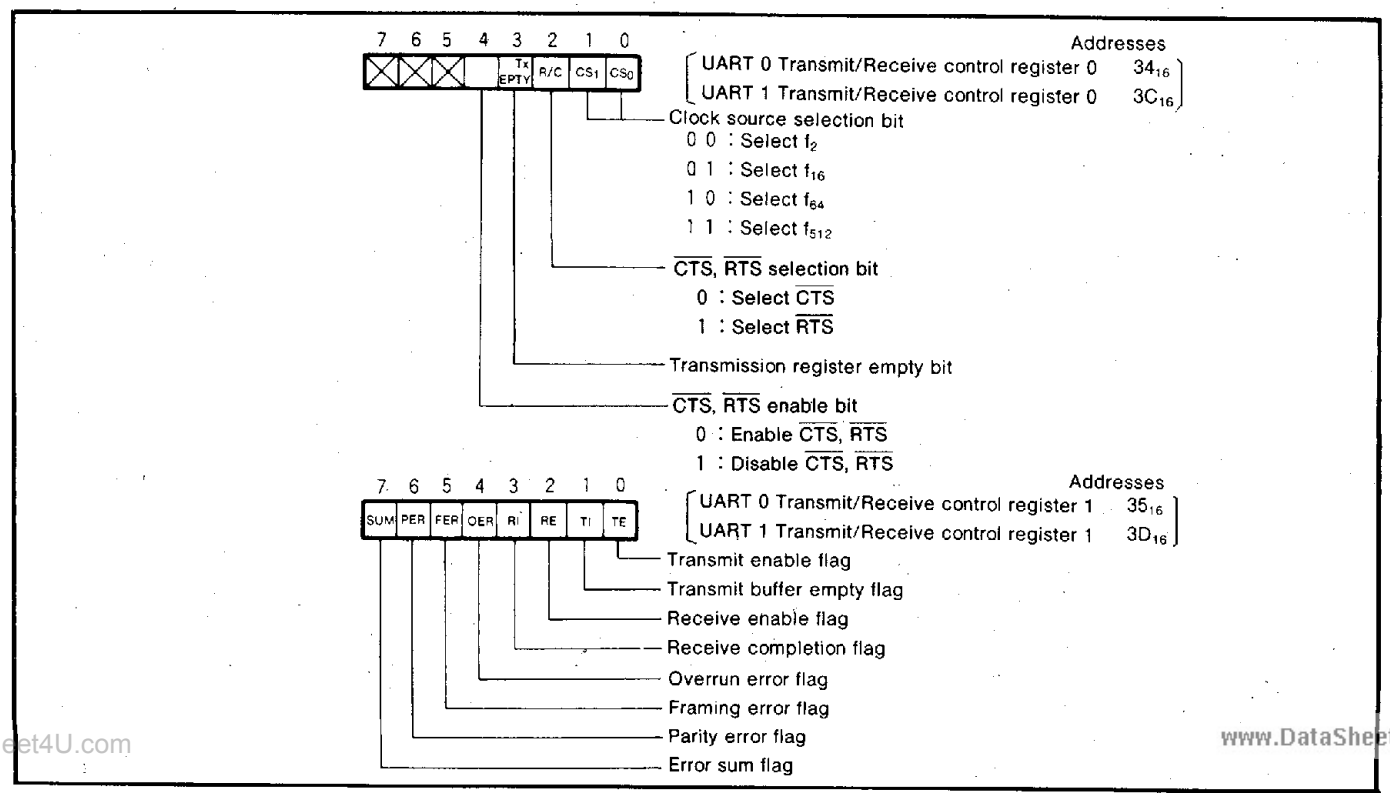


Fig. 42 UARTi Transmit/Receive control register bit configuration

CLOCK SYNCHRONOUS SERIAL COMMUNICATION

A case where communication is performed between two clock synchronous serial I/O ports as shown in Figure 43 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 38, the selected clock is divided by $(n+1)$, then by 2, passed through a transmisson control circuit, and output as transmisson clock CLKj. Therefore, when the selected clock is fi,

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

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

The bit 2 of the clock sending side UARTj transmit/receive control register 0 is cleared to "0" to select $\overline{CTS_j}$ input. The bit 2 of the clock receiving side is set to "1" to select $\overline{RTS_k}$ output.

Whether to use the $\overline{CTS}$ and $\overline{RTS}$ signals is determined by bit 4 of the UART transmit/receive control register 0. Set bit 4 to "0" when $\overline{CTS}$ and $\overline{RTS}$ signals are used, and to "1" when they are not used.

When $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS}/\overline{RTS}$ pin can be used as normal port. The following describes the case when the $\overline{CTS}$ and $\overline{RTS}$ signals are used. If $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS_j}$ input condition is unnecessary and there is no $\overline{RTS_k}$ output.

Transmission

Transmission is started when the bit 0 (TEj flag) of UARTj transmit/receive control register 1 is "1", bit 1 (Tlj flag) of one is "0", and $\overline{CTS_j}$ input is "L". As shown in Figure 44, data is output from TxDj pin when transmission clock CLKj changes from "H" to "L". The data is output from the least significant bit.

The Tlj flag indicates whether the transmission buffer register is empty or not. It is cleared to "0" when data is written in the transmission buffer register and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.

When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically from the transmission buffer register to the transmission register if the next transmission start condition is satisfied. If the bit 2 of UARTj transmit/receive control register 0 is "1", $\overline{CTS_j}$ input is ignored and transmission start is controlled only by the TEj flag and Tlj flag. Once transmission has started, the TEj flag, Tlj flag, and $\overline{CTS_j}$ signals are ignored until data transmission completes. Therefore, transmission is not interrupt when $\overline{CTS_j}$ input is changed to "H" during transmission.

The transmission start condition indicated by TEj flag, Tlj flag, and $\overline{CTS_j}$ is checked while the T_{ENDj} signal shown in Figure 44 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission buffer register and Tlj flag is cleared to "0" before the T_{ENDj} signal goes "H".

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

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

Receive

Receive starts when the bit 2 (RE_k flag) of $UART_k$ transmit/receive control register 1 is set to "1".

The $\overline{RTS_k}$ output is "H" when the RE_k flag is "0" and goes "L" when the RE_k flag changed to "1". It goes back to "H" when receive starts. Therefore, the $\overline{RTS_k}$ output can be used to determine whether the receive register is ready to receive. It is ready when $\overline{RTS_k}$ output is "L".

The data from the RxD_k pin is retrieved and the contents of the receive register is shifted by 1 bit each time the transmission clock CLKj changes from "L" to"H". When an 8-bit data is received, the contents of the receive register is transferred to the receive buffer register and the bit 3 (RI_k flag) of $UART_k$ transmit/receive control register 1 is set to "1". In other words, the setting of the RI_k flag indicates that the receive buffer register contains the received data. At this point, $\overline{RTS_j}$ output goes "L" to indicate that the next data can be received. When the RI_k flag changes from "0" to "1", the interrupt request bit in the $UART_k$ receive interrupt control register is set to "1". Bit 4 (OER_k flag) of $UART_k$ transmit/receive control register is set to "1" when the next data is transferred from the receive register to the receive buffer register while RI_k 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. RI_k and OER_k flags are cleared automatically to "0" when the low-order byte of the receive buffer register is read. The OER_k flag is also cleared when the RE_k flag is

cleared. Bit 5 (FER_K flag), bit 6 (PER_K flag), and bit 7 (SUM_K flag) are ignored in clock synchronous mode.
As shown in Figure 38, 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 data to be sent from $UART_K$ to UARTj.

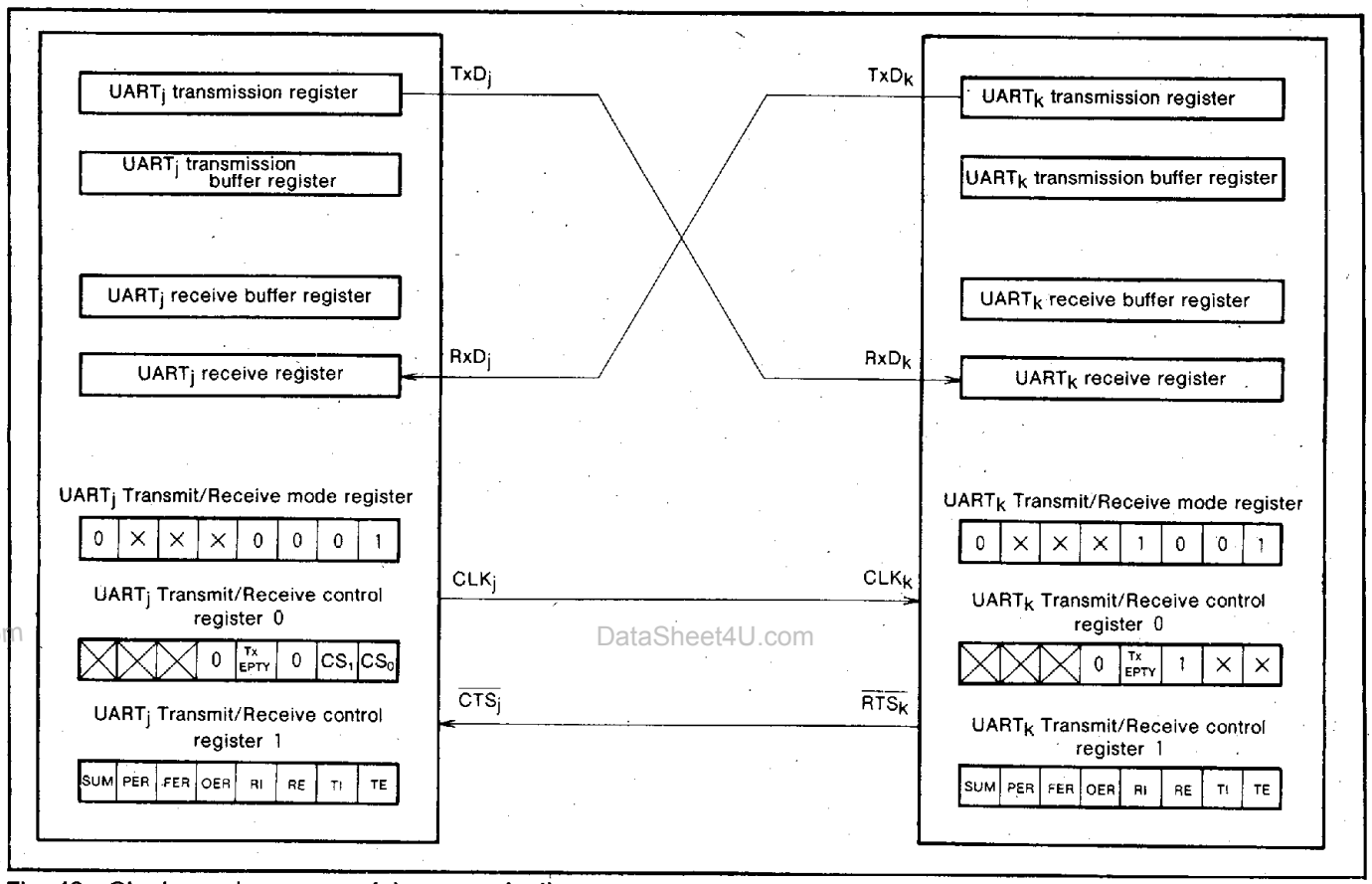


Fig. 43 Clock synchronous serial communication

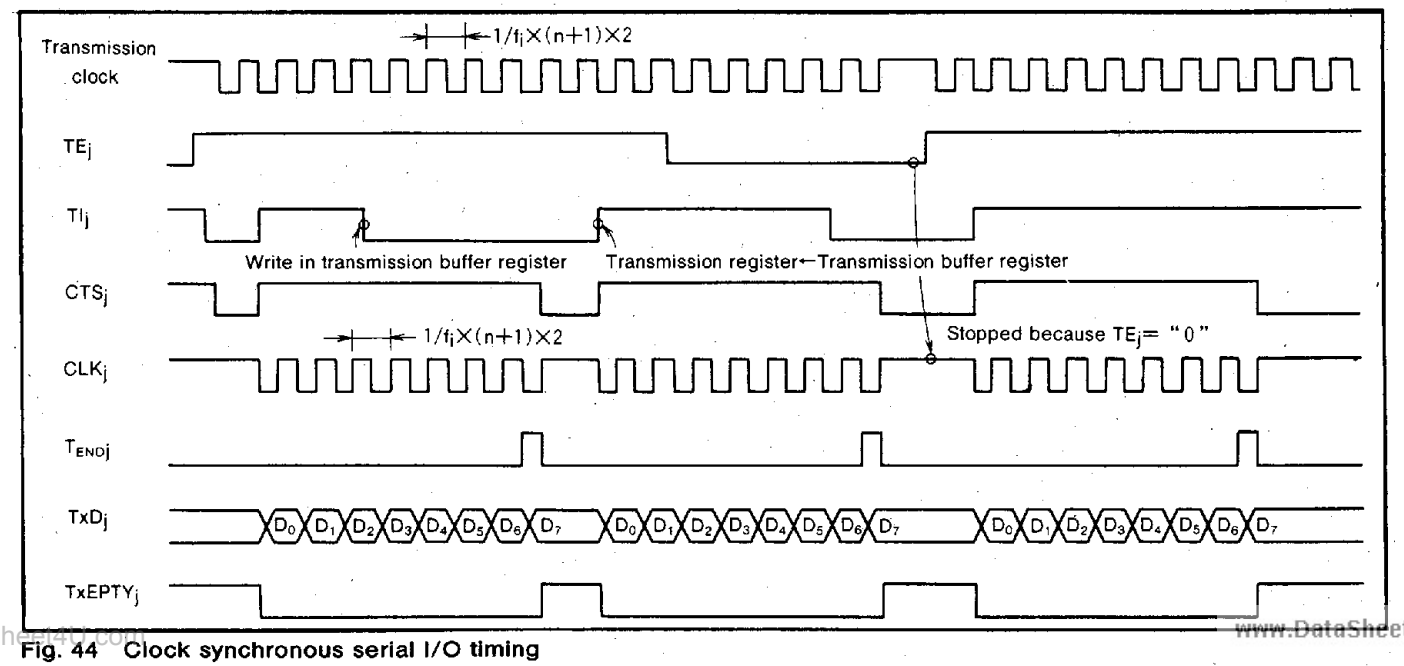


Fig. 44 Clock synchronous serial I/O timing

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, the bit 0 of UARTi transmit/receive mode register is "1", the bit 1 is "0", and the 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, the bit 0 (CS_0) and bit 1 (CS_1) of UARTi transmit/receive control register 0 are used to select the clock source. When an internal clock is selected for asynchronous serial communication, the CLKi pin can be used as a normal I/O pin.

The selected internal or external clock is divided by (n+1), then by 16, and 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 fi 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 selection bit to select 1 stop bit or 2 stop bits.

The bit 5 is a selection bit of odd parity or even parity.

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

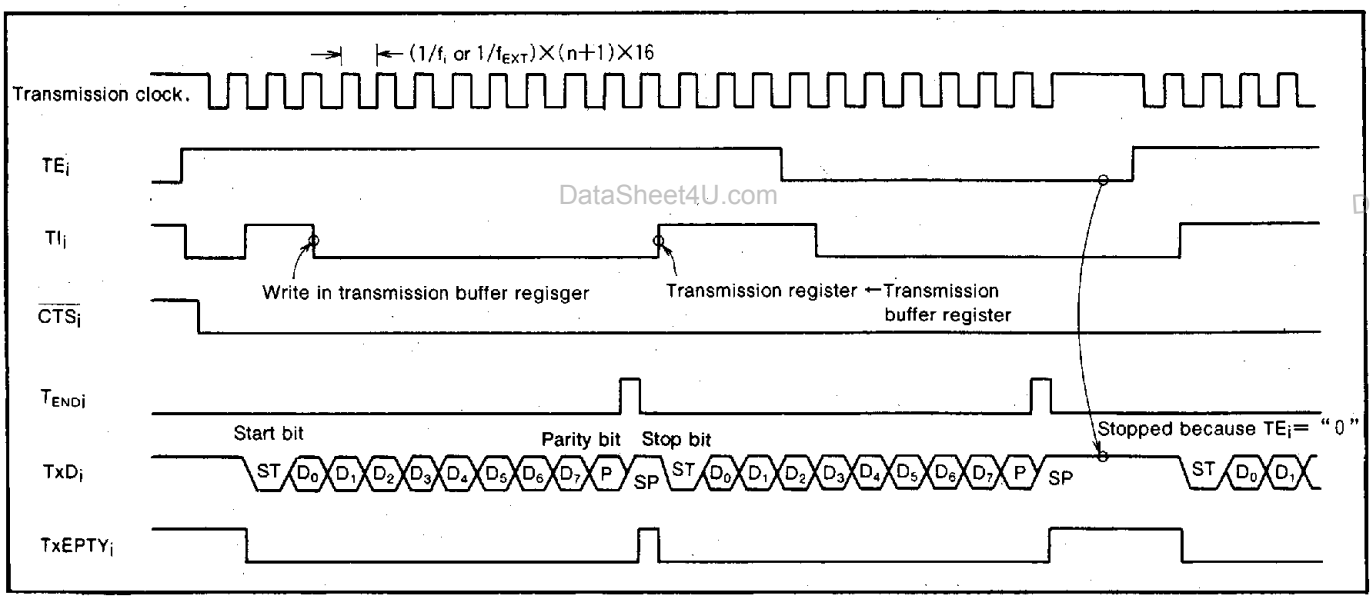


Fig. 45 Transmit timing example when 8-bit asynchronous communication with parity and 1 stop bit is selected

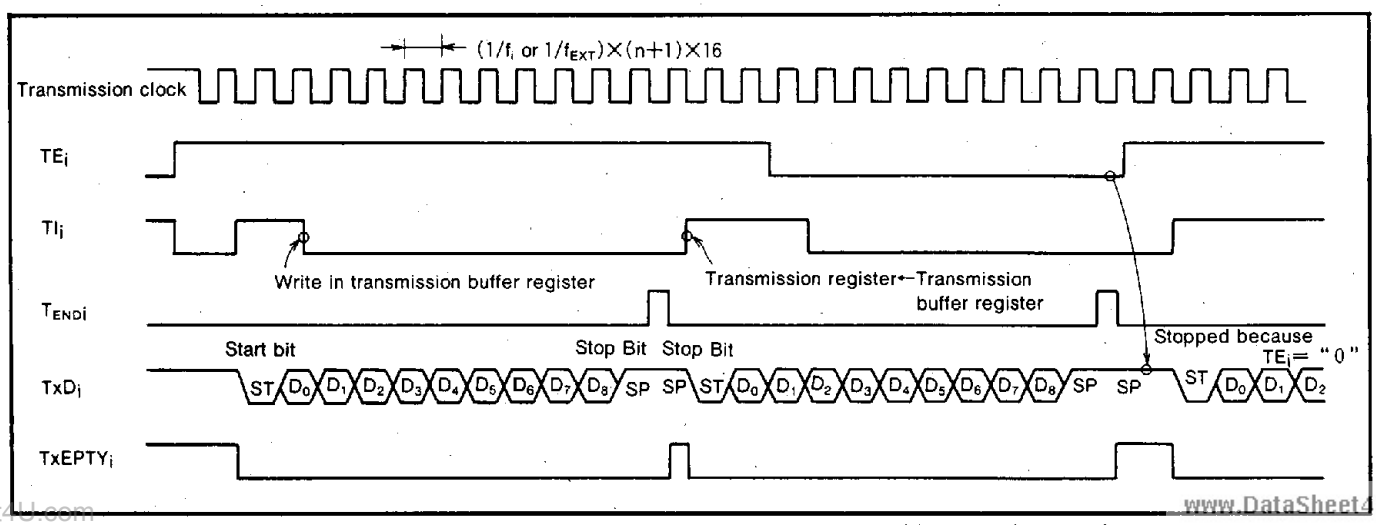


Fig. 46 Transmit timing example when 9-bit asynchronous communication with no parity and 2 stop bits is selected

Bit 6 is the parity bit selection bit which indicates whether to add parity bit or not.
Bits 4 to 6 should be set or reset according to the data format of the communicating devices.
Bit 7 is the sleep selection bit. The sleep mode is described later.
The UART$_i$ transmit/receive control regisger 0 bit 2 is used to determine whether to use $\overline{CTS}_i$ input or $\overline{RTS}_i$ output. $\overline{CTS}_i$ input is used if bit 2 is "0" and $\overline{RTS}_i$ output is used if bit 2 is "1".
If $\overline{CTS}_i$ input is selected, the user can control whether to stop or start transmission by external $\overline{CTS}_i$ input.
Whether to use the $\overline{CTS}$ and $\overline{RTS}$ signals is determined by bit 4 of the UART transmit/receive control register 0. Set bit 4 to "0" when $\overline{CTS}$ and $\overline{RTS}$ signals are used, and to "1" when they are not used.
When $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS}/\overline{RTS}$ pin can be used as normal port. The following describes the case when the $\overline{CTS}$ and $\overline{RTS}$ signals are used. If $\overline{CTS}$ and $\overline{RTS}$ signals are not used, the $\overline{CTS}_i$ input condition is unnecessary and there is no $\overline{RTS}_i$ output.

Transmission

Transmission is started when the bit 0 (TEi flag) of UART$_i$ transmit/receive control register 1 is "1", the bit 1 (TI$_i$ flag) is "0", and $\overline{CTS}_i$ input is "L" if $\overline{CTS}_i$ input is selected. As shown in Figure 45 and 46, data is output from the TxD$_i$ pin with the stop bit and parity bit specified by the bits 4 to 6 of UART$_i$ transmit/receive mode register. The data is output from the least significant bit.
The TI$_i$ flag indicates whether the transmission butter is empty or not. It is cleared to "0" when data is written in the transmission buffer and set to "1" when the contents of the transmission buffer register is transferred to the transmission register.
When the transmission register becomes empty after the contents has been transmitted, data is transferred automatically form the transmission buffer register to the transmission register if the next transmission start condition is satisfied.
Once transmission has started, the TE$_i$ flag, TI$_i$ flag, and $\overline{CTS}_i$ signal (if $\overline{CTS}_i$ input is selected) are ignored until data transmission is completed.
Therefore, transmission does not stop until it completes even if the TE$_i$ flag is cleared during transmission.
The transmission start condition indicated by TE$_i$ flag, TI$_i$ flag, and $\overline{CTS}_i$ is checked while the T_{ENDi} signal shown in Figure 45 is "H". Therefore, data can be transmitted continuously if the next transmission data is written in the transmission butter register and TI$_i$ flag is cleared to 0 before the T_{ENDi} signal goes "H".
The bit 3 (TxEPTY$_i$ flag) of UART$_i$ transmit/receive control register 0 changes to "1" at the next cycle after the T_{ENDi} signal goes "H" and changes to "0" when transmission starts. Therefore, this flag can be used to determine whether data transmision is completed.
When the TI$_i$ flag changes from "0" to "1", the interrupt request bit in the UART$_i$ transmissoin interrupt control register is set to "1".

Receive

Receive is enabled when the bit 2 (RE$_i$ flag) of UART$_i$ transmit/receive control register 1 is set. As shown in Figure 47, the frequency divider circuit at the receiving end begin to work when a start bit is arrived and the data is received.

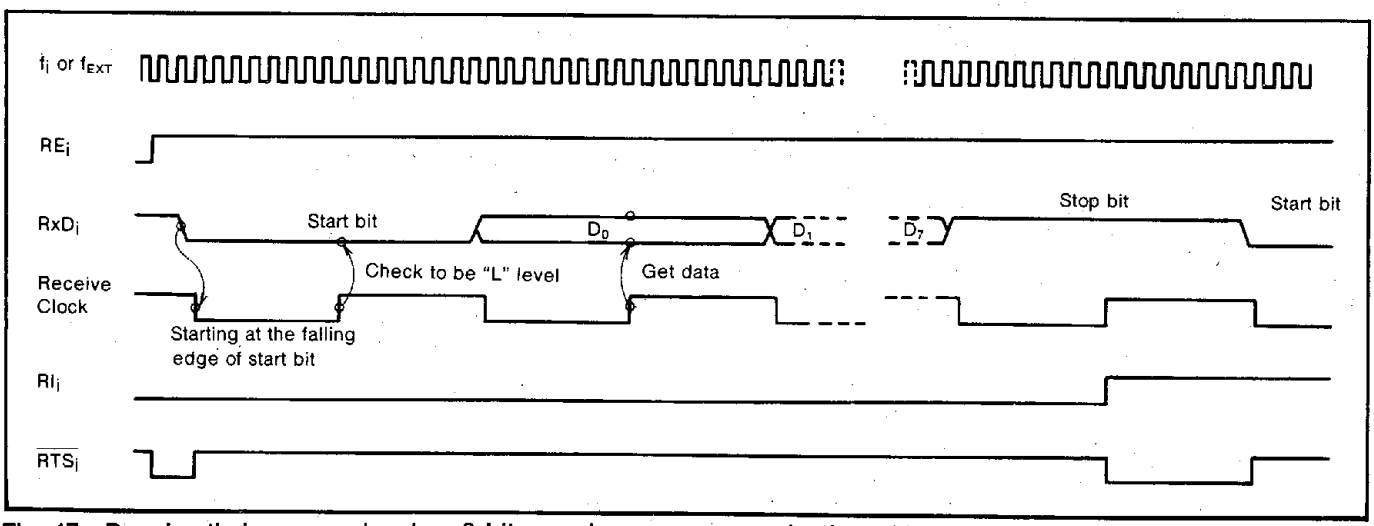


Fig. 47 Receive timing example when 8-bit asynchronous communication with no parity and 1 stop bit is selected

If $\overline{RTS_i}$ output is selected by setting the bit 2 of UART$_i$ transmit/receive control register 0 to "1", the $\overline{RTSi}$ output is "H" when the RE$_i$ flag is "0". When the RE$_i$ flag changes to "1", the $\overline{RTSi}$ output goes "L" to indicate receive ready and returns to "H" once receive has started. In other words, $\overline{RTS_i}$ output can be used to determine externally whether the receive register is ready to receive.

The entire transmission data bits are received when the start bit passes the final bit of the receive block shown in Figure 40. At this point, the contents of the receive register is transferred to the receive buffer register and the bit 3 of UART$_i$ transmit/receive control register 1 is set. In other words, the RI$_i$ flag indicates that the receive buffer register contains data when it is set. If $\overline{RTS_i}$ output is selected, $\overline{RTS_i}$ output goes "L" to indicate that the register is ready to receive the next data.

The interrupt request bit in the UART$_i$ receive interrupt control register is set when the RI$_i$ flag changes from "0" to "1".

The bit 4 (OER$_i$ flag) of UART$_i$ transmission control register 1 is set when the next data is transferred from the receive register to the receive buffer register while the RI$_i$ flag is "1". In other words when an overrun error occurs. If the OER$_i$ flag is "1", it indicates that the next data has been transferred to the receive buffer register before the contents of the receive butter register has been read.

Bit 5 (FER$_i$ flag) is set when the number of stop bits is less than required (framing error).

Bit 6 (PER$_i$ flag) is set when a parity error occurs.

Bit 7 (SUM$_i$ flag) is set when either the OER$_i$ flag, FER$_i$ flag, or the PER$_i$ flag is set. Therefore, the SUM$_i$ flag can be used to determine whether there is an error.

The setting of the RI$_i$ flag, OER$_i$ flag, FER$_i$ flag, and the PERi flag is performed while transferring the contents of the receive register to the receive buffer register. The RI$_i$ OER$_i$, FER$_i$, PER$_i$, and SUM$_i$ flags are cleared when the low order byte of the receive buffer register is read or when the RE$_i$ flag is cleared.

Sleep mode

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

The sleep mode is entered when the bit 7 of UART$_i$ transmit/receive mode register is set.

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 asychronous communication) of the received data is "0". Also the RI$_i$, OER$_i$, FER$_i$, PER$_i$, and the SUM$_i$ flag are unchanged. Therefore, the interrupt request bit of the UART$_i$ 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 with bit 7 set to "1" and bits 0 to 6 set to the address of the subordinate microcomputer which wants to communicate with. Then all subordinate microcomputers receive the same data. Each subordinate microcomputer checks the received data, clears the sleep bit if bits 0 to 6 are its own address and sets the sleep bit if not. Next the main microcomputer sends data with bit 7 cleared. Then the microcomputer with the sleep bit cleared will receive the data, but the microcomputer with the sleep bit set will not. In this way, the main microcomputer is able to communicate with only the designated microcomputer.

A-D CONVERTER

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

Figure 48 shows a block diagram of the A-D converter and Figure 49 shows the configuration of the A-D control register 0 (address $1E_{16}$) and A-D control register 1 (address $1F_{16}$).

The frequency of the A-D converter operating clock ϕ_{AD} is selected by bit 7 of the A-D control register 0. When bit 7 is "0", ϕ_{AD} is the clock frequency divided by 8. That is, $\phi_{AD}=f(X_{IN})/8$. When bit 7 is "1", ϕ_{AD} is the clock frequency divided by 4 and $\phi_{AD}=f(X_{IN})/4$.

The ϕ_{AD} during A-D conversion must be 250kHz minimum because the comparator consists of a capacity coupling amplifier.

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

When the conversion result is used as 10 bits, 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 two bits are stored in bits 0 and 1 of the odd address of the corresponding A-D register. Bits 2 to 7 of the A-D register odd address return "000000_2" when read.

When the conversion result is used as 8 bits, the high-order 8 bits of the 10-bit A-D conversion are stored in even address of the corresponding A-D register. In this case, the A-D register odd address returns "00_{16}" when read.

The operating mode is selected by the bits 3 and 4 of A-D control register 0 and bits 1 and 2 of A-D control register 1. The available operating modes are one-shot, repeat, single sweep, repeat sweep 0, and repeat sweep 1.

The bit of data direction register bit corresponding to the A-D converter pin must be "0" (input mode) because the analog input port is shared with port P7.

The operation of each mode is described below.

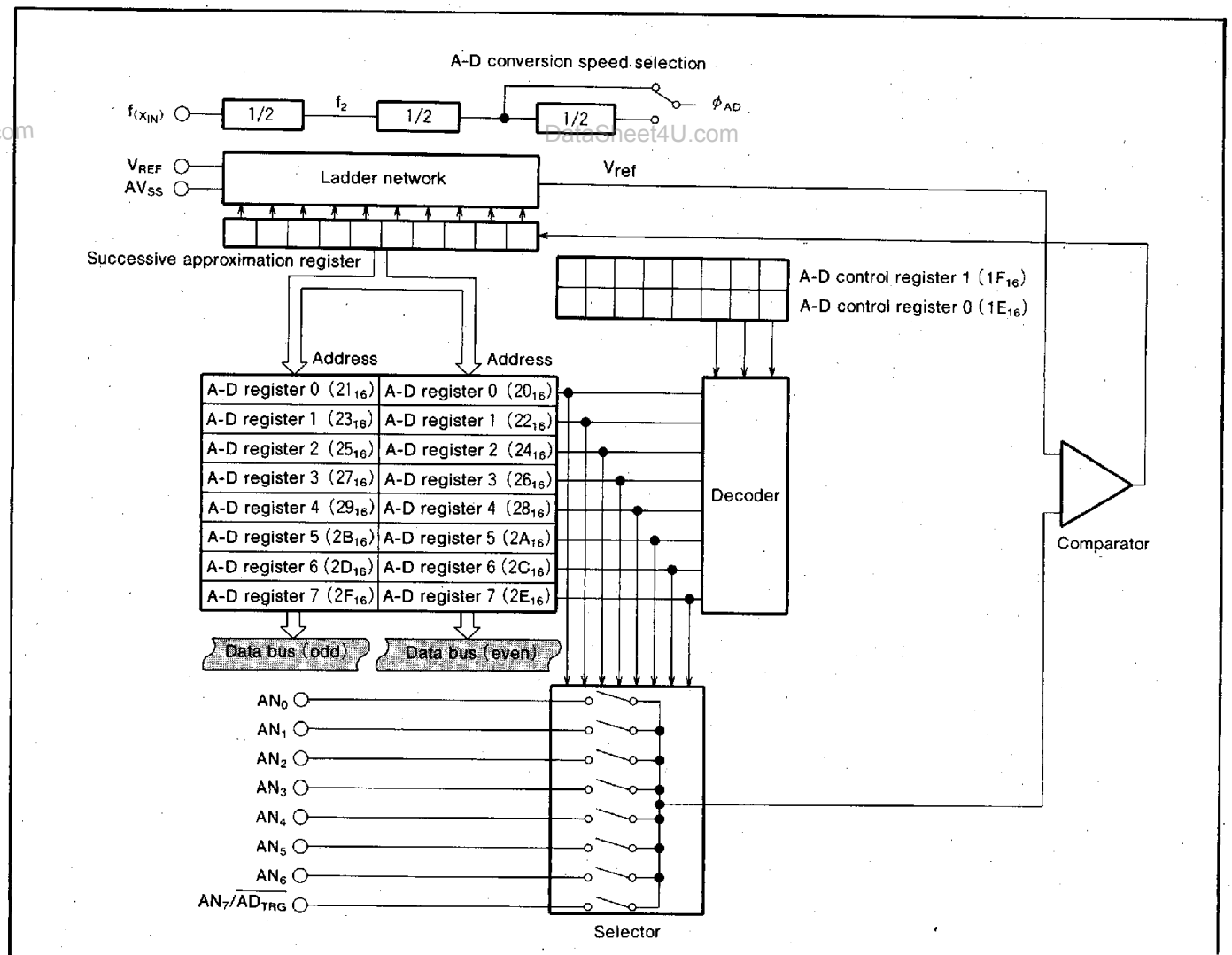


Fig. 48 A-D converter block diagram

(1) One-shot mode

One-shot mode is selected when bits 3 and 4 of A-D control register 0 are "0" and bit 2 of A-D control register 1 is "0". The A-D conversion pins are selected with bits 0 to 2 of A-D control register 0. A-D conversion can be started by a software trigger or by an external trigger.

A software trigger is selected when bit 5 of A-D control register 0 is "0" and an external trigger is selected when it is "1". When a software trigger is selected, A-D conversion is started when bit 6 (A-D conversion start flag) is set. When bit 3 of A-D control register 1 is "1", A-D conversion ends after 59 ϕ_{AD} cycles and an interrupt request bit of the A-D conversion interrupt control register is set. At the same time, A-D control register 0 bit 6 (A-D conversion start flag) is cleared and A-D conversion stops. The result of A-D conversion is stored in the A-D register corresponding to the selected pin.

If an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the pins that can be used for A-D conversion are AN_0 to AN_6 because the $\overline{AD_{TRG}}$ pin is shared with the analog voltage input pin AN_7. The operation is the same as with software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(2) Repeat mode

Repeat mode is selected when bit 3 of A-D control register 0 is "1", bit 4 is "0" and bit 2 of A-D control register 1 is "0". The operation of this mode is the same as the operation of one-shot mode except that when A-D conversion of the selected pin is complete and the result is stored in the A-D register, conversion does not stop, but is repeated.

No interrupt request is issued in this mode. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared.

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

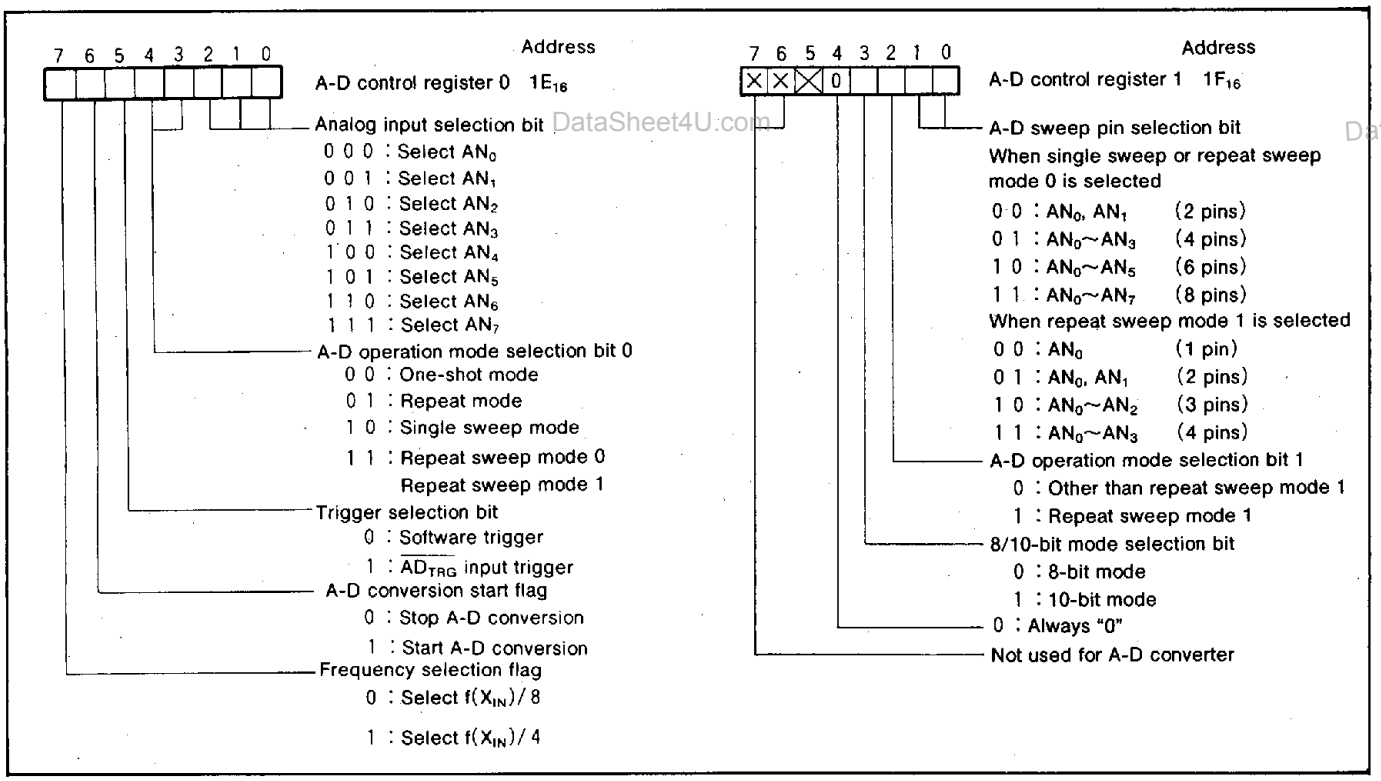


Fig. 49 A-D control register bit configuration

(3) Single sweep mode

Single sweep mode is selected when bit 3 of A-D control register 0 is "0", bit 4 is "1" and bit 2 of A-D control register 1 is "0".

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

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

A-D conversion can be started with a software trigger or with an external trigger input. A software trigger is selected when bit 5 is "0" and an external trigger is selected when it is "1".

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

When an external trigger is selected, A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{AD_{TRG}}$ input changes from "H" to "L". In this case, the A-D conversion result which is stored in the A-D register 7 becomes invalid because the $\overline{AD_{TRG}}$ pin is shared with AN_7 pin.

The operation by external trigger is the same as that done by software trigger except that the A-D conversion start flag is not cleared after A-D conversion and a retrigger can be available during A-D conversion.

(4) Repeat sweep mode 0

Repeat sweep mode 0 is selected when bit 3 of A-D control register 0 is "1", bit 4 is "1" and bit 2 of A-D control register 1 is "0".

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

(5) Repeat sweep mode 1

Repeat sweep mode 1 is selected when bit 3 of A-D control register 0 is "1", bit 4 is "1" and bit 2 of A-D control register 1 is "1".

The difference with the repeat sweep mode 0 is that A-D conversion of one unselected pin is performed and A-D conversion is repeated once again from AN_0 pin after A-D conversion of selected pins is performed in sequence from AN_0 pin. The number of analog input pins to be swept is also different.

The analog input pins for repeat sweep are selected with bits 1 and 0 of A-D control register 1. The contents of these pins are used to select one pin, two pins, three pins, or four pins.

The unselected pins are converted in sequence after the conversion of pins selected as repeat sweep pins. No interrupt request is generated. Furthermore, if software trigger is selected, the A-D conversion start flag is not cleared.

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

D-A CONVERTER

The D-A converter is an 8-bit R-2R method D-A converter. Two independent D-A converters are built in. Figure 50 shows the block diagram of the D-A converter and Figure 51 shows the bit configuration of A-D control register 1.

D-A conversion is performed by writing a value in the corresponding D-A register. The conversion result is output by bits 6 and 7 of A-D control register 1 (address $1F_{16}$). When bit 7 is "1", the conversion result is output from DA_0 pin. When bit 6 is "1", the conversion result is output from DA_1 pin.

The output analog voltage V is determined from the value n (decimal) set in the D-A register.

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

V_{REF} : Reference voltage

The D-A output enable bit is cleared to "0" at reset. An external buffer should be connected when connecting to a low impedance load because there is no built-in D-A output buffer.

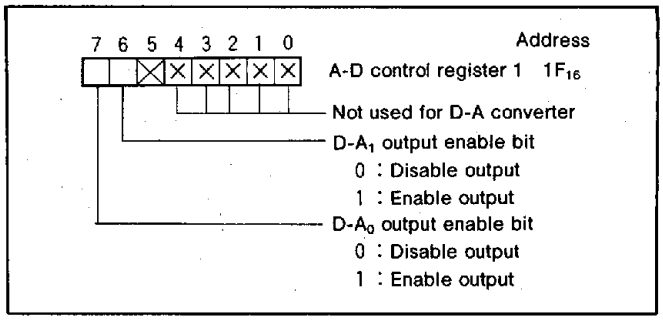


Fig. 51 A-D control register 1 bit configuration

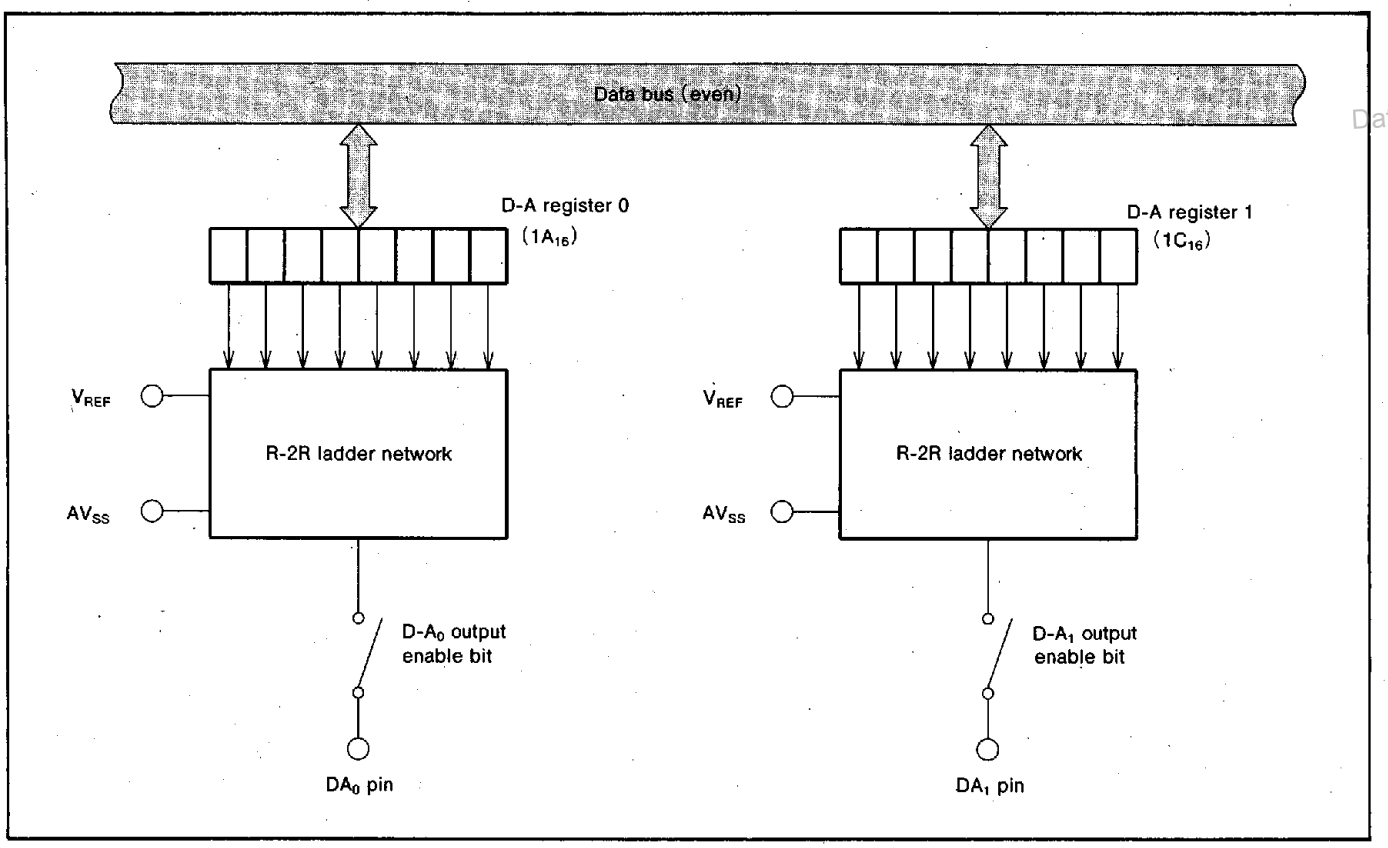


Fig. 50 D-A converter block diagram

WATCHDOG TIMER

The watchdog timer is used to detect unexpected execution sequence caused by software run-away.

Figure 52 shows a block diagram of the watchdog timer. The watchdog timer consists of a 12-bit binary counter.

The watchdog timer counts the clock frequency divided by 32 (f_{32}) or by 512 (f_{512}). Whether to count f_{32} or f_{512} is determined by the watchdog timer frequency selection flag shown in Figure 53. f_{512} is selected when the flag is "0" and f_{32} is selected when it is "1". The flag is cleared after reset.

FFF_{16} is set in the watchdog timer when "L" or $2 \cdot V_{CC}$ is applied to the $\overline{RESET}$ pin, STP instruction is executed, data is written to the watchdog timer, or the most significant bit of the watchdog timer become "0".

After FFF_{16} is set in the watchdog timer, the contents of watchdog timer is decremented by one at every cycle of selected frequency f_{32} or f_{512}, and after 2048 counts, the most significant bit of watchdog timer become "0", and a watchdog timer interrupt request bit is set, and FFF_{16} is preset in the watchdog timer.

Normally, a program is written so that data is written in the watchdog timer before the most significant bit of the watchdog timer become "0". If this routine is not executed due to unexpected program execution, the most significant bit of the watchdog timer become eventually "0" and an interrupt is generated.

The processor can be reset by setting the bit 3 (software reset bit) of processor mode register 0 described in Figure 10 in the interrupt section and generating.

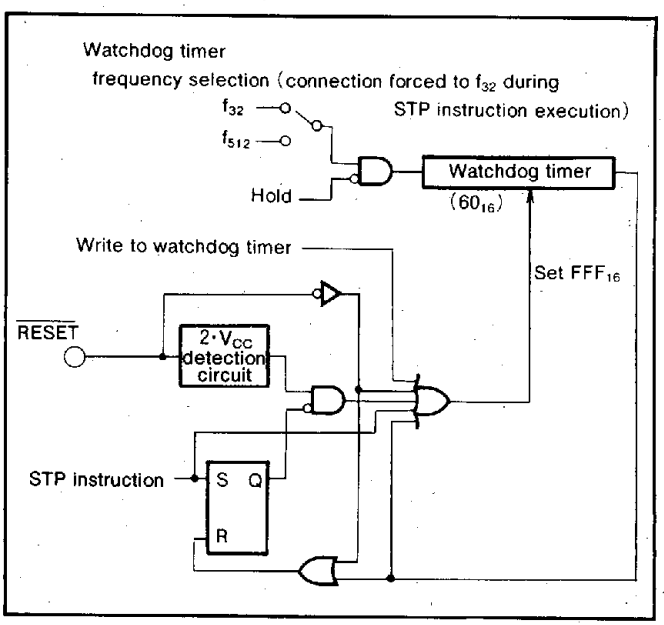


Fig. 52 Watchdog timer block diagram

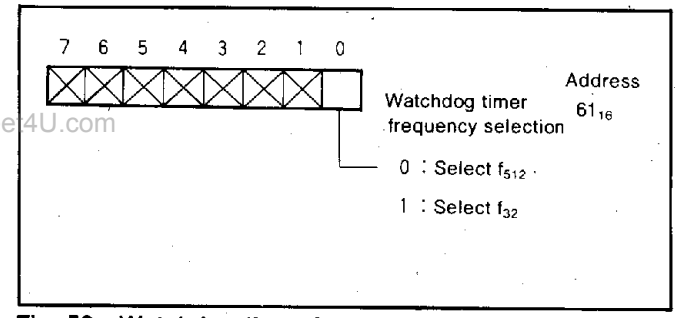


Fig. 53 Watchdog timer frequency selection flag

RESET CIRCUIT

Reset occurs when the $\overline{\text{RESET}}$ pin is returned to "H" level after holding it at "L" level when the power voltage is at 5V ±10%. Program execution starts at the address formed by setting the address pins $A_{23} \sim A_{16}$ to 00_{16}, $A_{15} \sim A_8$ to the contents of address $FFFF_{16}$, and $A_7 \sim A_0$ to the contents of address $FFFE_{16}$.

Figure 54 shows the status of the internal registers when a reset occurs.

Figure 55 shows an example of a reset circuit. The reset input voltage must be held 0.9V or lower when the power voltage reaches 4.5V.

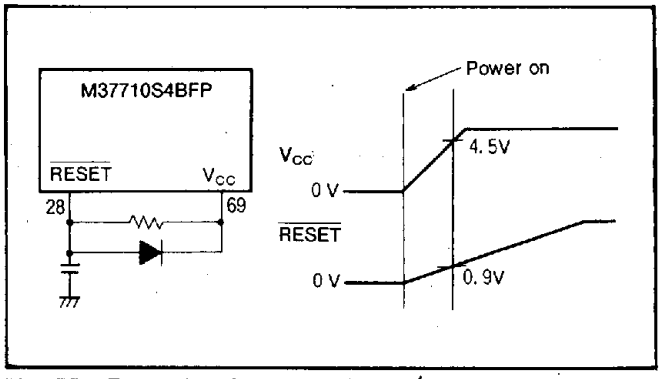


Fig. 55 Example of a reset circuit (perform careful evaluation at the system design level before using)

		Address	
(1)	Port P4 data direction register	$(0C_{16})\cdots$	00_{16}
(2)	Port P5 data direction register	$(0D_{16})\cdots$	00_{16}
(3)	Port P6 data direction register	$(10_{16})\cdots$	00_{16}
(4)	Port P7 data direction register	$(11_{16})\cdots$	00_{16}
(5)	Port P8 data direction register	$(14_{16})\cdots$	00_{16}
(6)	A-D control register 0	$(1E_{16})\cdots$	0 0 0 0 0 ? ? ?
(7)	A-D control register 1	$(1F_{16})\cdots$	0 0 X 0 0 0 1 1
(8)	UART 0 Transmit/Receive mode register	$(30_{16})\cdots$	00_{16}
(9)	UART 1 Transmit/Receive mode register	$(38_{16})\cdots$	00_{16}
(10)	UART 0 Transmit/Receive control register 0	$(34_{16})\cdots$	X X X 0 1 0 0 0
(11)	UART 1 Transmit/Receive control register 0	$(3C_{16})\cdots$	X X X 0 1 0 0 0
(12)	UART 0 Transmit/Receive control register 1	$(35_{16})\cdots$	0 0 0 0 0 0 1 0
(13)	UART 1 Transmit/Receive control register 1	$(3D_{16})\cdots$	0 0 0 0 0 0 1 0
(14)	Count start flag	$(40_{16})\cdots$	00_{16}
(15)	One-shot start flag	$(42_{16})\cdots$	X X X 0 0 0 0 0
(16)	Up-down flag	$(44_{16})\cdots$	00_{16}
(17)	Timer A0 mode register	$(56_{16})\cdots$	00_{16}
(18)	Timer A1 mode register	$(57_{16})\cdots$	00_{16}
(19)	Timer A2 mode register	$(58_{16})\cdots$	00_{16}
(20)	Timer A3 mode register	$(59_{16})\cdots$	00_{16}
(21)	Timer A4 mode register	$(5A_{16})\cdots$	00_{16}
(22)	Timer B0 mode register	$(5B_{16})\cdots$	0 0 1 0 0 0 0 0
(23)	Timer B1 mode register	$(5C_{16})\cdots$	0 0 1 X 0 0 0 0
(24)	Timer B2 mode register	$(5D_{16})\cdots$	0 0 1 X 0 0 0 0
(25)	Processor mode register 0	$(5E_{16})\cdots$	X 0 0 0 0 0 1 0
(26)	Processor mode register 1	$(5F_{16})\cdots$	X X X X X X X 0

		Address	
(27)	Watchdog timer	$(60_{16})\cdots$	FFF_{16}
(28)	Watchdog timer frequency selection flag	$(61_{16})\cdots$	X X X X X X X 0
(29)	Waveform output mode register	$(62_{16})\cdots$	00_{16}
(30)	Reserved area (Do not write to this address)	$(66_{16})\cdots$	00_{16}
(31)	A-D conversion interrupt control register	$(70_{16})\cdots$	X X X X 0 0 0 0
(32)	UART 0 transmission interrupt control register	$(71_{16})\cdots$	X X X X 0 0 0 0
(33)	UART 0 receive interrupt control register	$(72_{16})\cdots$	X X X X 0 0 0 0
(34)	UART 1 transmission interrupt control register	$(73_{16})\cdots$	X X X X 0 0 0 0
(35)	UART 1 receive interrupt control register	$(74_{16})\cdots$	X X X X 0 0 0 0
(36)	Timer A0 interrupt control register	$(75_{16})\cdots$	X X X X 0 0 0 0
(37)	Timer A1 interrupt control register	$(76_{16})\cdots$	X X X X 0 0 0 0
(38)	Timer A2 interrupt control register	$(77_{16})\cdots$	X X X X 0 0 0 0
(39)	Timer A3 interrupt control register	$(78_{16})\cdots$	X X X X 0 0 0 0
(40)	Timer A4 interrupt control register	$(79_{16})\cdots$	X X X X 0 0 0 0
(41)	Timer B0 interrupt control register	$(7A_{16})\cdots$	X X X X 0 0 0 0
(42)	Timer B1 interrupt control register	$(7B_{16})\cdots$	X X X X 0 0 0 0
(43)	Timer B2 interrupt control register	$(7C_{16})\cdots$	X X X X 0 0 0 0
(44)	$\overline{\text{INT}}_0$ interrupt control register	$(7D_{16})\cdots$	X X 0 0 0 0 0 0
(45)	$\overline{\text{INT}}_1$ interrupt control register	$(7E_{16})\cdots$	X X 0 0 0 0 0 0
(46)	$\overline{\text{INT}}_2$ interrupt control register	$(7F_{16})\cdots$	X X 0 0 0 0 0 0
(47)	Processor status register PS		0 0 0 ? ? 0 0 0 1 ? ?
(48)	Program bank register PG		00_{16}
(49)	Program counter PC_H		Content of $FFFF_{16}$
(50)	Program counter PC_L		Content of $FFFE_{16}$
(51)	Direct page register DPR		0000_{16}
(52)	Data bank register DT		00_{16}

Contents of other registers and RAM are not initialized and should be initialized by software.

Fig. 54 Microcomputer internal status during reset

INPUT/OUTPUT PINS

Ports P8 to P4 all have a data direction register and each bit can be programmed for input or output. A pin becomes an output pin when the corresponding data direction register is set and an input pin when it is cleared.

When pin programmed for output, the data is written to the 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. Therefore, a previously output value can be read correctly even when the output "L" voltage is raised due to reasons such as directly driving an LED.

A pin programmed for input is floating and the value input to the pin can be read. When a pin is programmed for input, the data is written only in the port latch and the pin stays floating.

If an input/output pin is not used as an output port, clear the bit of the corresponding data direction register so that the pin become input mode.

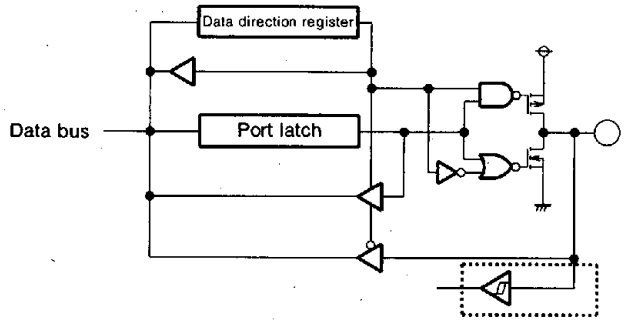
Figures 56 and 57 show block diagrams of ports P8 to P4 and the $\overline{E}$ pin output.

In evaluation chip mode, ports P4 is also used as control signal pin.

Refer to the section on processor modes for more details.

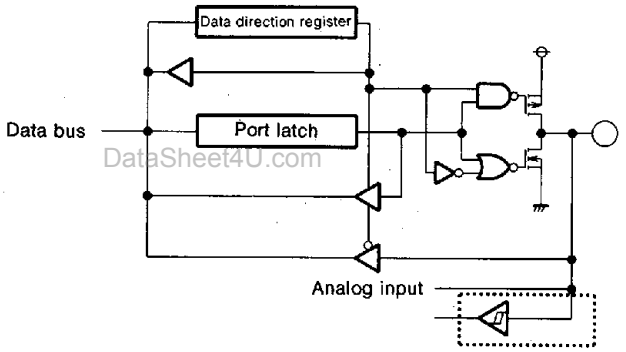
• Port $P4_3$~$P4_6$ (Inside dotted-line not included)

Port $P4_7$, $P5_7$, $P6_1$~$P6_7$, $P8_2$, $P8_6$ (Inside dotted-line included)



• Port $P7_0$~$P7_6$ (Inside dotted-line not included)

• Port $P7_7$ (Inside dotted-line included)



• Port $P8_3$, $P8_7$ (Inside dotted-line not included)

Port $P5_0$~$P5_6$, $P6_0$ (Inside dotted-line included)

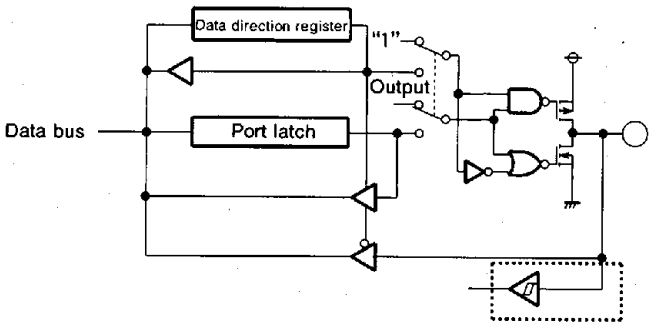
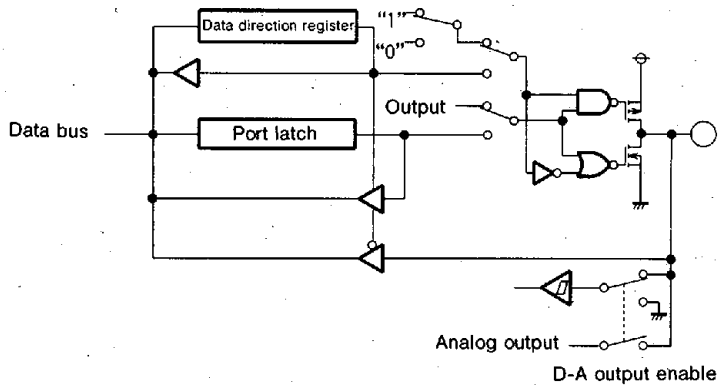
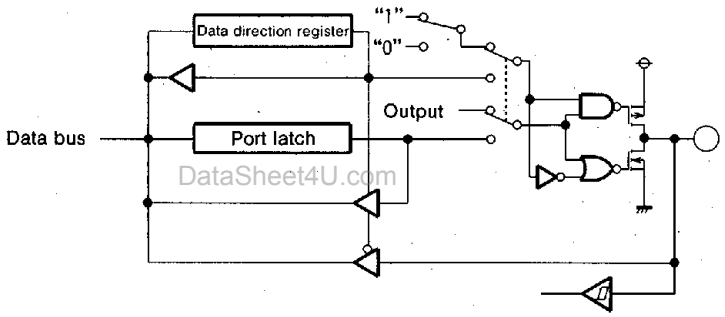


Fig. 56 Block diagram for ports P8 to P4 and the $\overline{E}$ pin output (1)

• Port $P8_0$, $P8_4$



• Port $P8_1$, $P8_5$



• $\overline{E}$

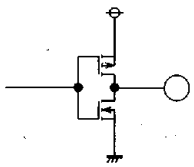


Fig. 57 Block diagram for ports P8 to P4 and $\overline{E}$ pin output (2)

PROCESSOR MODE

The bit 0 of processor mode register 0 as shown in Figure 58 is used to select which mode of microprocessor mode, and evaluation chip mode.

Figure 59 shows the functions of A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins, and ports P4 in each mode.

The external memory area also changes when the mode changes.

Figure 60 shows the memory map for each mode.

The accessing of the external memory is affected by the BYTE pin, the bit 2 (wait bit) of processor mode register 0, and bit 0 (wait selection bit) of processor mode register 1. These will be described next.

•BYTE pin

When accessing the external memory, the level of the BYTE pin is used to determine whether to use the data bus as 8-bit width or 16-bit width.

The data bus width is 8 bits when the level of the BYTE pin is "H" and A_{16}/D_0 to A_{23}/D_7 pins become the data I/O pins.

The data bus width is 16 bits when the level of the BYTE pin is "L" and both A_{16}/D_0 to A_{23}/D_7 pins and A_8/D_8 to A_{15}/D_{15} pins become the data I/O pins.

When accessing the internal memory, the data bus width is always 16 bits regardless of the BYTE pin level.

An exclusive mode in the evaluation chip mode allows the BYTE pin level to be set to $2 \cdot V_{CC}$. In this case, the operation is slightly different from the above. This is described in the evaluation chip mode section.

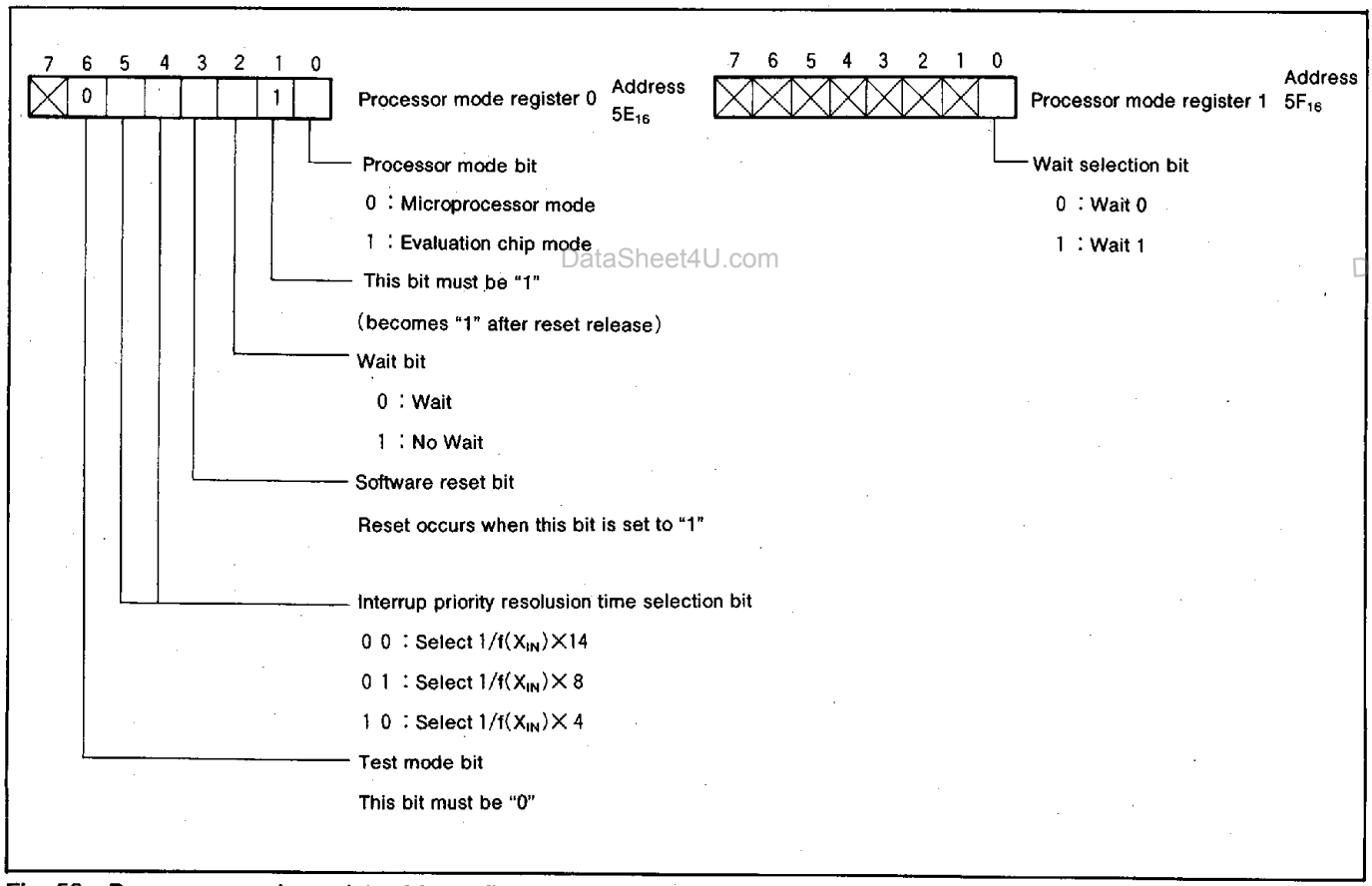


Fig. 58 Processor mode register bit configulation

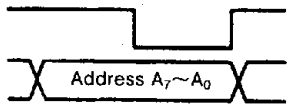
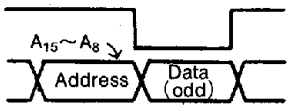
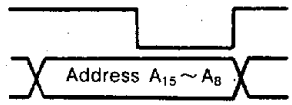
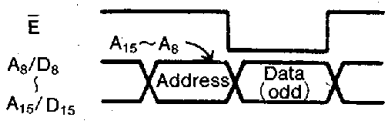
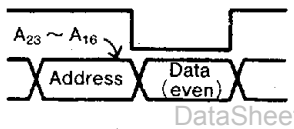
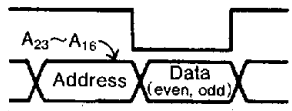
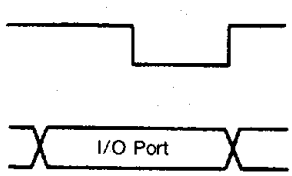
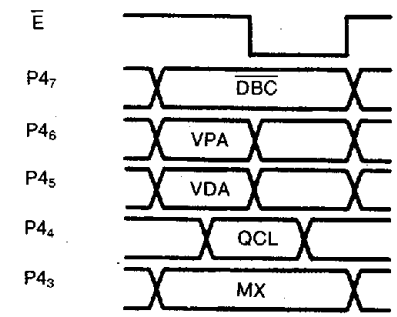
Port \ Mode		CM_1 = 1, CM_0 = 0	CM_1 = 1, CM_0 = 1
		Microprocessor Mode	Evaluation Chip Mode
A_0~A_7			Same as left
A_8/D_8 ∫ A_{15}/D_{15}	BYTE = "L"		Same as left
	BYTE = "H"		 Ports P4, P5 and their direction registers are treated as 16-bit wide bus.
A_{16}/D_0 ∫ A_{23}/D_7	BYTE = "L"		Same as left
	BYTE = "H"		 Same as A_8/D_8 to A_{15}/D_{15}
Port P4			

Fig. 59 Processor mode and A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins and port P4 functions

●Wait bit

As shown in Figure 61, when the external memory area is accessed with the processor mode register 0 (address $5E_{16}$) bit 2 (wait bit) cleared to "0", the access time can be extended compared with no wait (the wait bit is "1").

The access time is extended in two ways and this is selected with bit 0 (wait selection bit) of processor mode register 1 (address $5F_{16}$).

When this bit is "1", the access time is 1.5 times compared to that for no wait. When this bit is "0", the access time is twice compared to that for no wait.

At reset the wait bit is "0" and bit 0 of processor mode register 1 is "0".

The accessing of internal memory area is performed in no wait mode regardless of the wait bit.

The processor modes are described below.

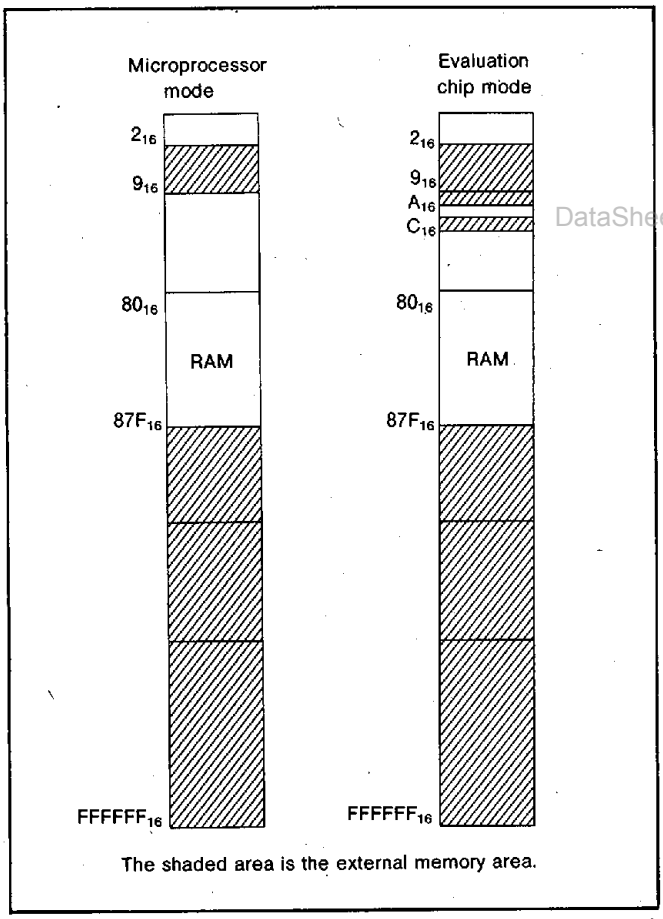


Fig. 60 External memory area for each processor mode

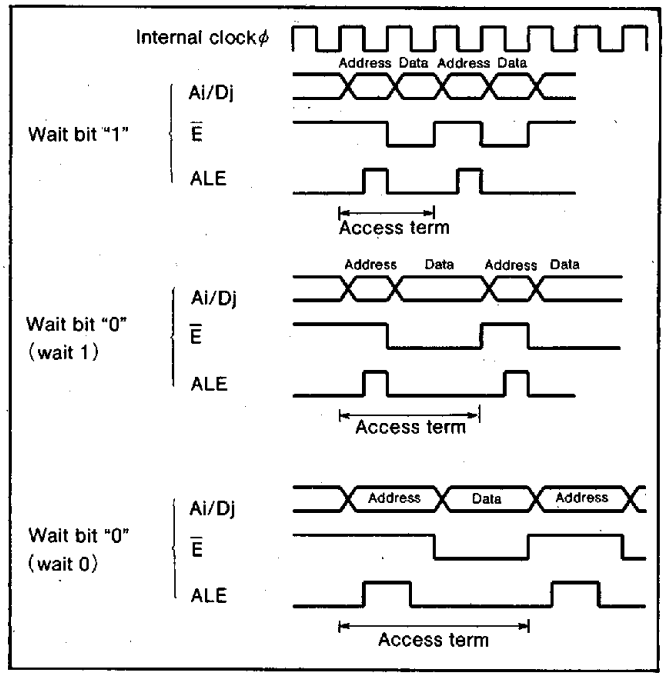


Fig. 61 Relationship between wait bit, wait selection bit, and access time

(1) Microprocessor mode [10]

Microprocessor mode is entered by connecting the CNV_{SS} pin to V_{CC} and starting from reset.

A_0 to A_7 pins become address output pins.

A_8/D_8 to A_{15}/D_{15} pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", A_8/D_8 to A_{15}/D_{15} pins function as an address output pin while $\overline{E}$ is "H" and as an odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level "H", A_8/D_8 to A_{15}/D_{15} pins function as an address output pin.

A_{16}/D_0 to A_{23}/D_7 pins have two functions depending on the level of the BYTE pin.

When the BYTE pin level is "L", A_{16}/D_0 to A_{23}/D_7 pins function as an address output pin while $\overline{E}$ is "H" and as an even address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H", A_{16}/D_0 to A_{23}/D_7 pins function as an address output pin while $\overline{E}$ is "H" and as an even and odd address data I/O pin while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

R/$\overline{W}$ is a read/write signal which indicates a read when it is "H" and a write when it is "L".

$\overline{BHE}$ is a byte high enable signal which indicates that an odd address is accessed when it is "L".

Therefore, two bytes at even and odd addresses are accessed simultaneously if address A_0 is "L" and $\overline{BHE}$ is "L".

ALE is an address latch enable signal used to latch the address signal from a multiplexed signal of address and data. The latch is transparent while ALE is "H" to let the address signal pass through and held while ALE is "L".

ϕ_1 outputs X_{IN} clock divided by 2.

$\overline{HLDA}$ is a hold acknowledge signal and is used to notify externally when the microcomputer receives $\overline{HOLD}$ input and enters into hold state.

$\overline{HOLD}$ is a hold request signal. It is an input signal used to put the microcomputer in hold state. $\overline{HOLD}$ input is accepted when the internal clock ϕ falls from "H" level to "L" level while the bus is not used. A_0 to A_7 pins, A_8/D_8 to A_{23}/D_7 pins, R/$\overline{W}$ pin, and $\overline{BHE}$ pin are floating while the microcomputer stays in hold state. These pins are floating after one cycle of the internal clock ϕ later than $\overline{HLDA}$ signal changes to "L" level. At the removing of hold state, these ports are removed from floating state after one cycle of ϕ later than $\overline{HLDA}$ signal changes to "H" level.

$\overline{RDY}$ is a ready signal. If this signal goes "L", the internal clock ϕ stops at "L". However, ϕ_1 output from clock ϕ_1 output pin does not stop. $\overline{RDY}$ is used when slow external memory is attached.

(2) Evaluation chip mode (11)

Evaluation chip mode is entered by applying voltage twice the V_{CC} voltage to the CNV_{SS} pin. This mode is normally used for evaluation tools.

A_8/D_8 to A_{15}/D_{15} pins function as address output pins while $\overline{E}$ is "H" and as data I/O pin of odd addresses while $\overline{E}$ is "L" regardless of the BYTE pin level. However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

A_{16}/D_0 to A_{23}/D_7 pins function as address output pins while $\overline{E}$ is "H" and as data I/O pin of even addresses while $\overline{E}$ is "L" when the BYTE pin level is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

When the BYTE pin level is "H" or $2 \cdot V_{CC}$, port P2 functions as an address output pin while $\overline{E}$ is "H" and as data I/O pin of even and odd addresses while $\overline{E}$ is "L". However, if an internal memory is read, external data is ignored while $\overline{E}$ is "L".

Port P4 and its data direction register which are located at address $0A_{16}$ and $0C_{16}$ are treated differently in evaluation chip mode. When these addresses are accessed, the data bus width is treated as 16 bits regardless of the BYTE pin level, and the access cycle is treated as internal memory regardless of the wait bit.

Ports $P4_3$ to $P4_6$ become MX, QCL, VDA, and VPA output pins respectively. Port $P4_7$ becomes the $\overline{DBC}$ input pin.

The MX signal normally contains the contents of flag m, but the contents of flag x is output if the CPU is using flag x.

QCL is the queue buffer clear signal. It becomes "H" when the instruction queue buffer is cleared, for example, when a jump instruction is executed.

VDA is the valid data address signal. It becomes "H" while the CPU is reading data from data buffer or writing data to data buffer. It also becomes "H" when the first byte of the instruction (operation code) is read from the instruction queue buffer.

VPA is the valid program address signal. It becomes "H" while the CPU is reading an instruction code from the instruction queue buffer.

$\overline{DBC}$ is the debug control signal and is used for debugging.

Table 5 shows the relationship between the CNV_{SS} pin input levels and processor modes.

Table 5. Relationship between the CNV_{SS} pin input levels and processor modes

CNV_{SS}	Mode	Description
V_{CC}	• Microprocessor • Evaluation chip	Microprocessor mode upon starting after reset. Evaluation chip mode can be selected by changing the processor mode bit by software.
$2 \cdot V_{CC}$	• Evaluation chip	Evaluation chip mode only.

CLOCK GENERATING CIRCUIT

Figure 62 shows a block diagram of the clock generator.

When an STP instruction is executed, the internal clock ϕ stops oscillating at "L" level. At the same time, FFF_{16} is written to watchdog timer and the watchdog timer input connection is forced to f_{32}. This connection is broken and connected to the input determined by the watchdog timer frequency selection flag when the most significant bit of the watchdog timer is cleared or reset.

Oscillation resumes when an interrupt is received, but the internal clock ϕ remains at "L" level until the most significant bit of the watchdog timer is cleared. This is to avoid the unstable interval at the start of oscillation when using a ceramic resonator.

When a WIT instruction is executed, the internal clock ϕ stops at "L" level, but the oscillator does not stop. The clock is restarted when an interrupt is received. Instructions can be executed immediately because the oscillator is not stopped.

The stop or wait state is released when an interrupt is received or when reset is issued. Therefore, interrupts must be enabled before executing a STP or WIT instruction.

Figure 63 shows a circuit example using a ceramic (or quartz crystal) resonator. Use the manufacturer's recommended values for constants such as capacitance which differ for each resonator. Figure 64 shows an example of using an external clock signal.

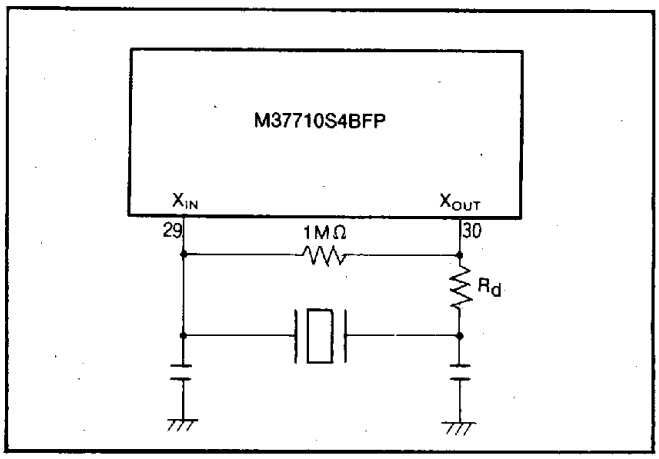


Fig. 63 Circuit using a ceramic resonator

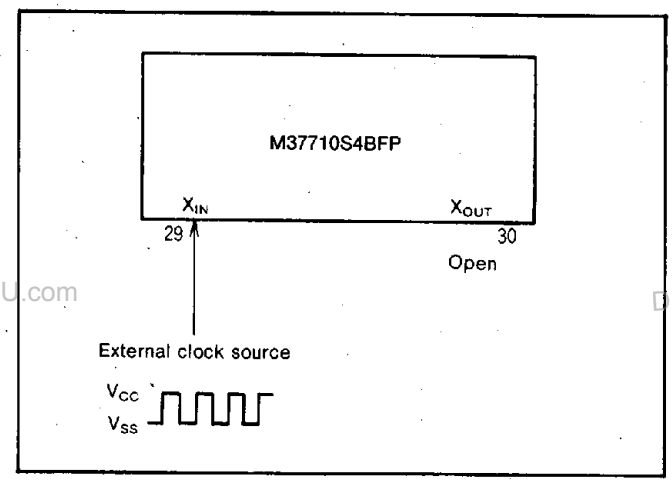


Fig. 64 External clock input circuit

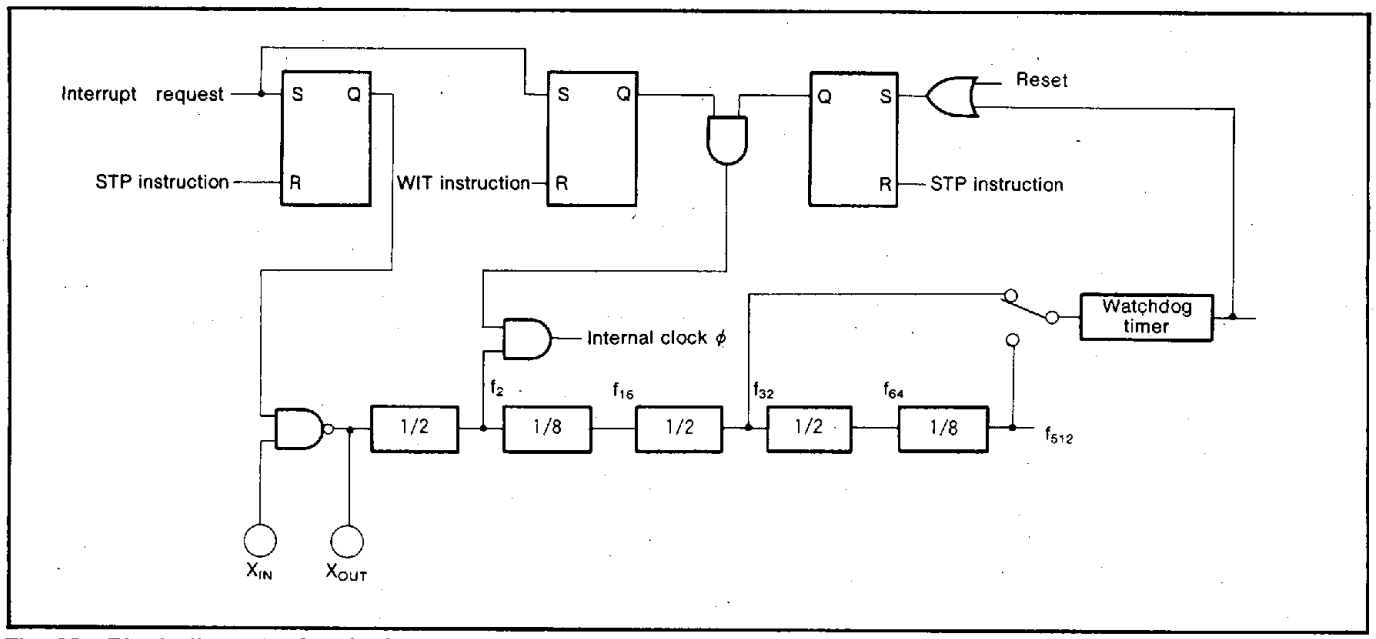


Fig. 62 Block diagram of a clock generator

ADDRESSING MODES

The M37710S4BFP has 28 powerful addressing modes. Refer to the 7700 Family addressing mode description for the details of each addressing mode.

MACHINE INSTRUCTION LIST

The M37710S4BFP has 103 machine instructions. Refer to the 7700 Family machine instruction list for details.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Supply voltage		−0.3∼7	V
AV_{CC}	Analog supply voltage		−0.3∼7	V
V_I	Input voltage $\overline{RESET}$, CNV_{SS}, BYTE		−0.3∼12	V
V_I	Input voltage A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, V_{REF}, X_{IN}, $\overline{HOLD}$, $\overline{RDY}$		−0.3∼V_{CC}+0.3	V
V_O	Output voltage A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{OUT}, $\overline{E}$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$		−0.3∼V_{CC}+0.3	V
P_d	Power dissipation	T_a=25℃	300	mW
Topr	Operating temperature		−20∼85	℃
Tstg	Storage temperature		−40∼150	℃

RECOMMENDED OPERATING CONDITIONS (V_{CC}=5V±10%, T_a=−20∼85℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
V_{CC}	Supply voltage	4.5	5.0	5.5	V
AV_{CC}	Analog supply voltage		V_{CC}		V
V_{SS}	Supply voltage		0		V
AV_{SS}	Analog supply voltage		0		V
V_{IH}	High-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	0.8V_{CC}		V_{CC}	V
V_{IH}	High-level input voltage A_8/D_8∼A_{23}/D_7	0.5V_{CC}		V_{CC}	V
V_{IL}	Low-level input voltage $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	0		0.2V_{CC}	V
V_{IL}	Low-level input voltage A_8/D_8∼A_{23}/D_7	0		0.16V_{CC}	V
$I_{OH(peak)}$	High-level peak output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$			−10	mA
$I_{OH(avg)}$	High-level average output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$			−5	mA
$I_{OL(peak)}$	Low-level peak output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$			10	mA
$I_{OL(avg)}$	Low-level average output current A_0∼A_7, A_8/D_8∼A_{23}/D_7, $P4_3$∼$P4_7$, $P5_0$∼$P5_7$, $P6_0$∼$P6_7$, $P7_0$∼$P7_7$, $P8_0$∼$P8_7$, ϕ_1, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$			5	mA
$f(X_{IN})$	External clock frequency input			25	MHz

Note 1. Average output current is the average value of a 100ms interval.

2. The sum of $I_{OL(peak)}$ for ports A_0∼A_7, A_8/D_8∼A_{23}/D_7, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$, and P8 must be 80mA or less, the sum of $I_{OH(peak)}$ for ports A_0∼A_7, A_8/D_8∼A_{23}/D_7, $\overline{HLDA}$, ALE, $\overline{BHE}$, R/$\overline{W}$, and P8 must be 80mA or less, the sum of $I_{OL(peak)}$ for ports P4, P5, P6, P7, ϕ_1 must be 80mA or less, and the sum of $I_{OH(peak)}$ for ports P4, P5, P6, P7, ϕ_1 must be 80mA or less.

ELECTRICAL CHARACTERISTICS (V_{CC}=5V, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−10mA		3			V
V_{OH}	High-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OH}=−400μA		4.7			V
V_{OH}	High-level output voltage ALE	I_{OH}=−10mA		3.1			V
		I_{OH}=−400μA		4.8			
V_{OH}	High-level output voltage $\overline{E}$	I_{OH}=−10mA		3.4			V
		I_{OH}=−400μA		4.8			
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=10mA				2	V
V_{OL}	Low-level output voltage A_0~A_7, A_8/D_8~A_{23}/D_7, ϕ_1, $\overline{HLDA}$, $\overline{BHE}$, R/$\overline{W}$	I_{OL}=2mA				0.45	V
V_{OL}	Low-level output voltage ALE	I_{OL}=10mA				1.9	V
		I_{OL}=2mA				0.43	
V_{OL}	Low-level output voltage $\overline{E}$	I_{OL}=10mA				1.6	V
		I_{OL}=2mA				0.4	
V_{T+}−V_{T-}	Hysteresis $\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$~$TA4_{IN}$, $TB0_{IN}$~$TB2_{IN}$, $\overline{INT_0}$~$\overline{INT_2}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$, $\overline{CTS_1}$, CLK_0, CLK_1			0.4		1	V
V_{T+}−V_{T-}	Hysteresis $\overline{RESET}$			0.2		0.5	V
V_{T+}−V_{T-}	Hysteresis X_{IN}			0.1		0.3	V
I_{IH}	High-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=5V				5	μA
I_{IL}	Low-level input current A_8/D_8~A_{23}/D_7, $P4_3$~$P4_7$, $P5_0$~$P5_7$, $P6_0$~$P6_7$, $P7_0$~$P7_7$, $P8_0$~$P8_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE, $\overline{HOLD}$, $\overline{RDY}$	V_I=0V				−5	μA
V_{RAM}	RAM hold voltage	When clock is stopped.		2			V
I_{CC}	Power supply current	Output only pin is open and other pins are V_{SS} during reset.	f(X_{IN})=25MHz, square waveform		19	38	mA
			T_a=25℃ when clock is stopped.			1	μA
			T_a=85℃ when clock is stopped.			20	

A-D CONVERTER CHARACTERISTICS (V_{CC}=AV_{CC}=5V, V_{SS}=AV_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution	V_{REF}=V_{CC}			10	Bits
—	Absolute accuracy	V_{REF}=V_{CC}			±3	LSB
R_{LADDER}	Ladder resistance	V_{REF}=V_{CC}	5		20	kΩ
t_{CONV}	Conversion time		9.44			μs
V_{REF}	Reference voltage		2		V_{CC}	V
V_{IA}	Analog input voltage		0		V_{REF}	V

D-A CONVERTER CHARACTERISTICS (V_{CC}=5V, V_{SS}=AV_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
—	Resolution				8	Bits
—	Absolute accuracy				1.0	%
t_{SU}	Set time				3	μs
R_O	Output resistance		1	2.5	4	kΩ
I_{VREF}	Reference power input current	(Note)			3.2	mA

Note. One D-A converter is used, and the value of D-A register for unused D-A converter is "00_{16}".
Current that flows to the ladder resistance of A-D converter is excluded.

TIMING REQUIREMENTS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

External clock input

Symbol	Parameter	Limits		Unit
		Min.	Max.	
t_C	External clock input cycle time	40		ns
$t_{W(H)}$	External clock input high-level pulse width	15		ns
$t_{W(L)}$	External clock input low-level pulse width	15		ns
t_r	External clock rise time		8	ns
t_f	External clock fall time		8	ns

Microprocessor mode

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{SU(DH-E)}$	Data high-order input setup time	30		ns
$t_{SU(DL-E)}$	Data low-order input setup time	30		ns
$t_{SU(P4D-E)}$	Port P4 input setup time	60		ns
$t_{SU(P5D-E)}$	Port P5 input setup time	60		ns
$t_{SU(P6D-E)}$	Port P6 input setup time	60		ns
$t_{SU(P7D-E)}$	Port P7 input setup time	60		ns
$t_{SU(P8D-E)}$	Port P8 input setup time	60		ns
$t_{SU(RDY-\phi_1)}$	$\overline{RDY}$ input setup time	55		ns
$t_{SU(HOLD-\phi_1)}$	$\overline{HOLD}$ input setup time	55		ns
$t_{h(E-DH)}$	Data high-order input hold time	0		ns
$t_{h(E-DL)}$	Data low-order input hold time	0		ns
$t_{h(E-P4D)}$	Port P4 input hold time	0		ns
$t_{h(E-P5D)}$	Port P5 input hold time	0		ns
$t_{h(E-P6D)}$	Port P6 input hold time	0		ns
$t_{h(E-P7D)}$	Port P7 input hold time	0		ns
$t_{h(E-P8D)}$	Port P8 input hold time	0		ns
$t_{h(\phi_1-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(\phi_1-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns

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	320		ns
$t_{W(TAH)}$	TAi_{IN} input high-level pulse width	160		ns
$t_{W(TAL)}$	TAi_{IN} input low-level pulse width	160		ns

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	160		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 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 B input (Count input in event counter mode)

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

Timer B input (Pulse period measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	160		ns

Timer B input (Pulse width measurement mode)

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{C(TB)}$	TBi_{IN} input cycle time	320		ns
$t_{W(TBH)}$	TBi_{IN} input high-level pulse width	160		ns
$t_{W(TBL)}$	TBi_{IN} input low-level pulse width	160		ns

A-D trigger input

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

Serial I/O

Symbol	Parameter	Limits		Unit
		Min.	Max.	
$t_{C(CK)}$	CLK_i input cycle time	200		ns
$t_{W(CKH)}$	CLK_i input high-level pulse width	100		ns
$t_{W(CKL)}$	CLK_i input low-level pulse width	100		ns
$t_{d(C-Q)}$	TxD_i output delay time		80	ns
$t_{h(C-Q)}$	TxD_i hold time	0		ns
$t_{su(D-C)}$	RxD_i input setup time	30		ns
$t_{h(C-D)}$	RxD_i input hold time	90		ns

External interrupt $\overline{INT_i}$ input

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

SWITCHING CHARACTERISTICS (V_{CC}=5V±10%, V_{SS}=0V, T_a=25℃, f(X_{IN})=25MHz, unless otherwise noted)

Microprocessor mode (when wait bit = "1")

Symbol	Parameter	Test conditions	Limits		Unit
			Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 65	12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5	ns
$t_{d(AH-E)}$	Address high-order output delay time		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		ns
$t_{W(ALE)}$	ALE pulse width		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		25		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		25		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		25		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		25		ns
$t_{h(ALE-AH)}$	Address high-order hold time		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time		25		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		25		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		50		ns

Microprocessor mode (when wait bit = "0", wait selection bit = "1", and external memory area is accessed)

Symbol	Parameter	Test conditions	Limits		Unit
			Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 65	12		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		12		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		5		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5	ns
$t_{d(AH-E)}$	Address high-order output delay time		12		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		5		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			50	ns
$t_{d(ALE-E)}$	ALE output delay time		4		ns
$t_{W(ALE)}$	ALE pulse width		22		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		20		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		20		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		25		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		9		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		25		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		25		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		25		ns
$t_{h(ALE-AH)}$	Address high-order hold time		9		ns
$t_{h(E-DLQ)}$	Data low-order hold time		25		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		25		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		130		ns

Microprocessor mode (when wait bit = "0", wait selection bit = "0", and external memory area is accessed)

Symbol	Parameter	Test conditions	Limits		Unit
			Min.	Max.	
$t_{d(AL-E)}$	Address low-order output delay time	Fig. 65	92		ns
$t_{d(E-DHQ)}$	Data high-order output delay time (BYTE="L")			45	ns
$t_{PXZ(E-DHZ)}$	Floating start delay time (BYTE="L")			5	ns
$t_{d(AM-E)}$	Address middle-order output delay time		92		ns
$t_{d(AM-ALE)}$	Address middle-order output delay time		70		ns
$t_{d(E-DLQ)}$	Data low-order output delay time			45	ns
$t_{PXZ(E-DLZ)}$	Floating start delay time			5	ns
$t_{d(AH-E)}$	Address high-order output delay time		92		ns
$t_{d(AH-ALE)}$	Address high-order output delay time		70		ns
$t_{d(\phi_1-HLDA)}$	$\overline{HLDA}$ output delay time			50	ns
$t_{d(ALE-E)}$	ALE output delay time		15		ns
$t_{W(ALE)}$	ALE pulse width		62		ns
$t_{d(BHE-E)}$	$\overline{BHE}$ output delay time		100		ns
$t_{d(R/W-E)}$	R/$\overline{W}$ output delay time		100		ns
$t_{d(E-\phi_1)}$	ϕ_1 output delay time		0	18	ns
$t_{h(E-AL)}$	Address low-order hold time		25		ns
$t_{h(ALE-AM)}$	Address middle-order hold time (BYTE="L")		20		ns
$t_{h(E-DHQ)}$	Data high-order hold time (BYTE="L")		25		ns
$t_{PZX(E-DHZ)}$	Floating release delay time (BYTE="L")		25		ns
$t_{h(E-AM)}$	Address middle-order hold time (BYTE="H")		25		ns
$t_{h(ALE-AH)}$	Address high-order hold time		20		ns
$t_{h(E-DLQ)}$	Data low-order hold time		25		ns
$t_{PZX(E-DLZ)}$	Floating release delay time		25		ns
$t_{h(E-BHE)}$	$\overline{BHE}$ hold time		18		ns
$t_{h(E-R/W)}$	R/$\overline{W}$ hold time		18		ns
$t_{d(E-P4Q)}$	Port P4 data output delay time			80	ns
$t_{d(E-P5Q)}$	Port P5 data output delay time			80	ns
$t_{d(E-P6Q)}$	Port P6 data output delay time			80	ns
$t_{d(E-P7Q)}$	Port P7 data output delay time			80	ns
$t_{d(E-P8Q)}$	Port P8 data output delay time			80	ns
$t_{W(EL)}$	$\overline{E}$ pulse width		130		ns

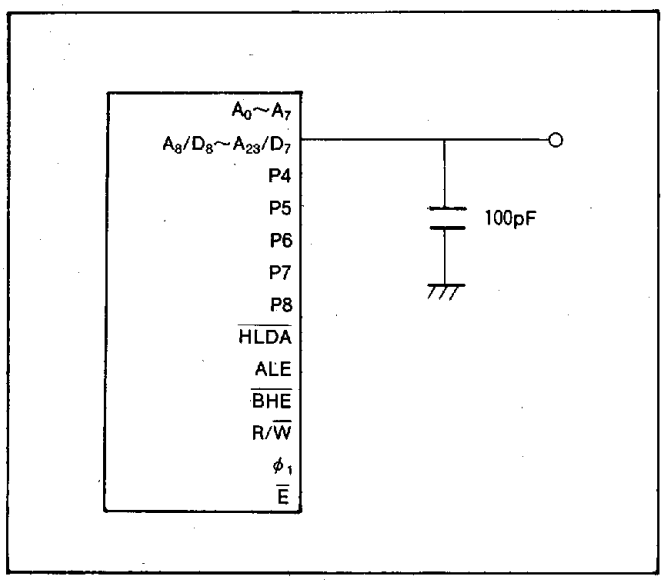
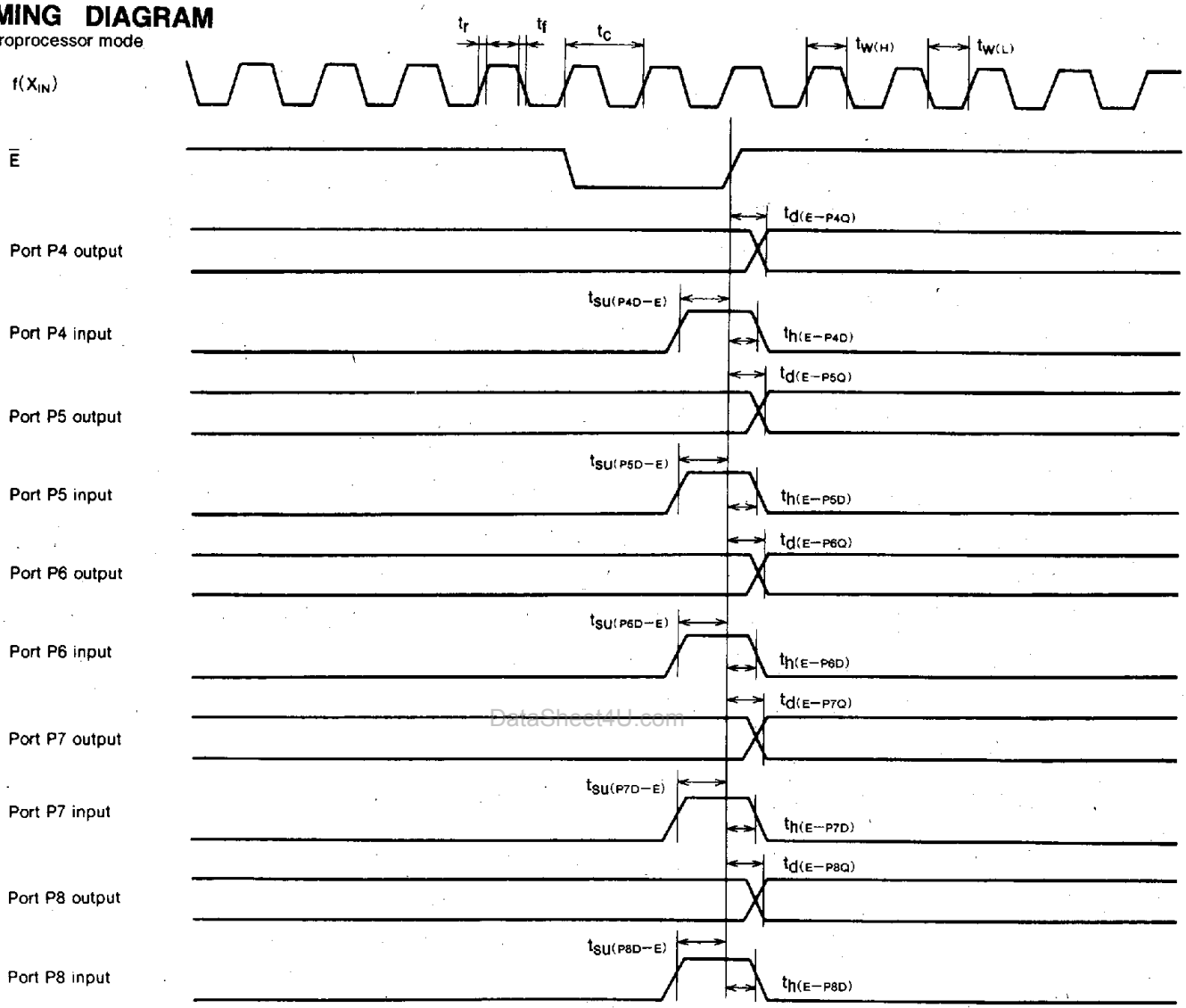
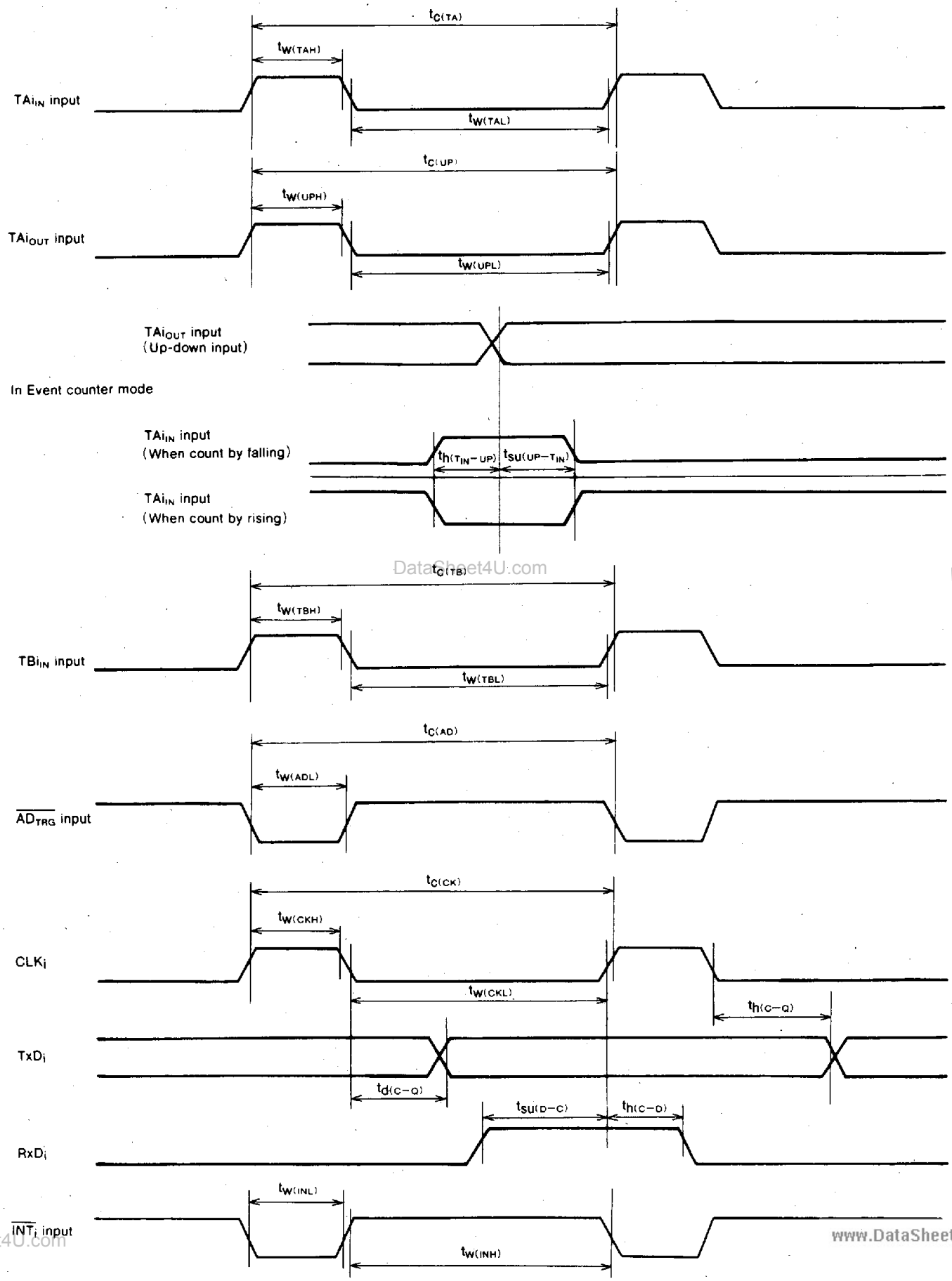


Fig. 65 Testing circuit for each teminal

TIMING DIAGRAM

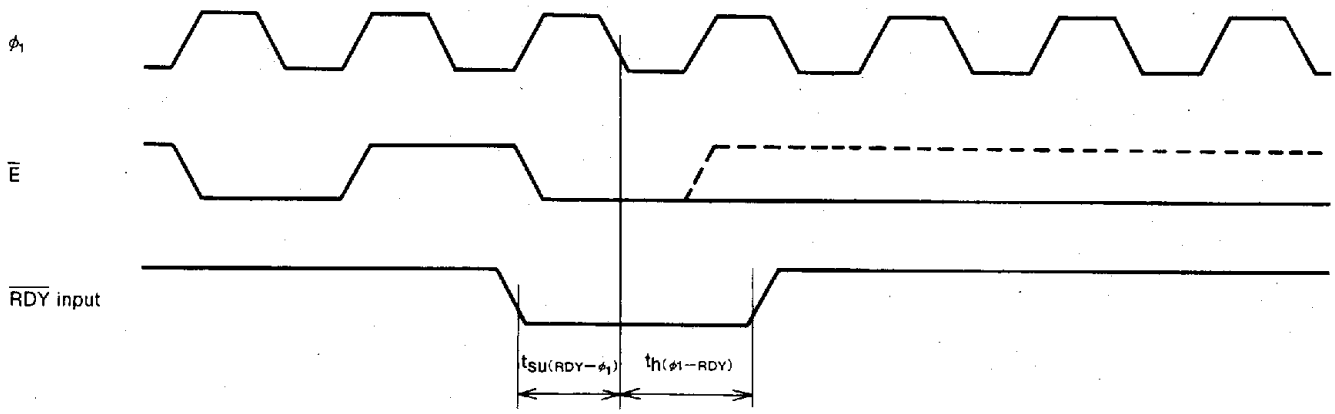
Microprocessor mode



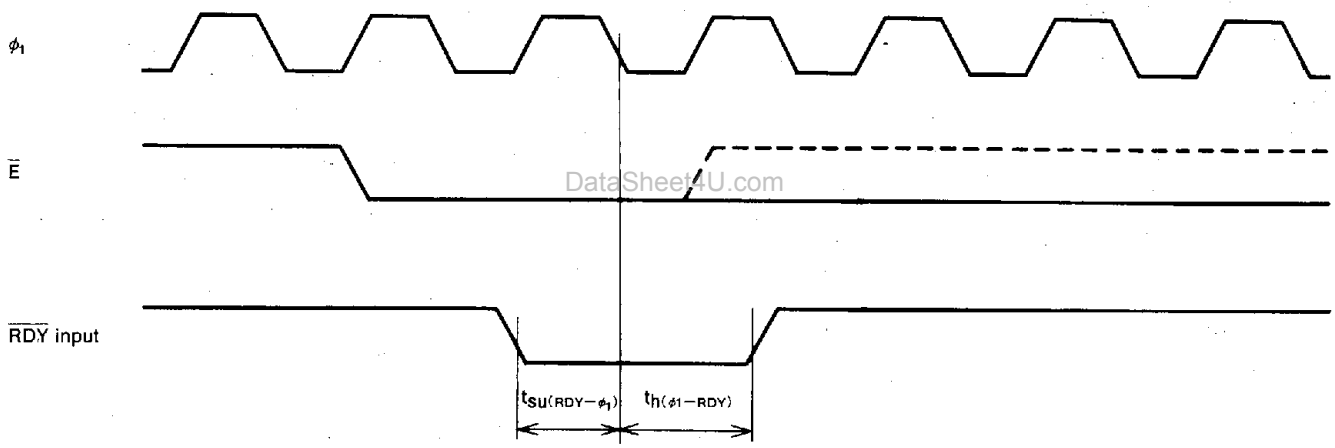


Me ocessor mode

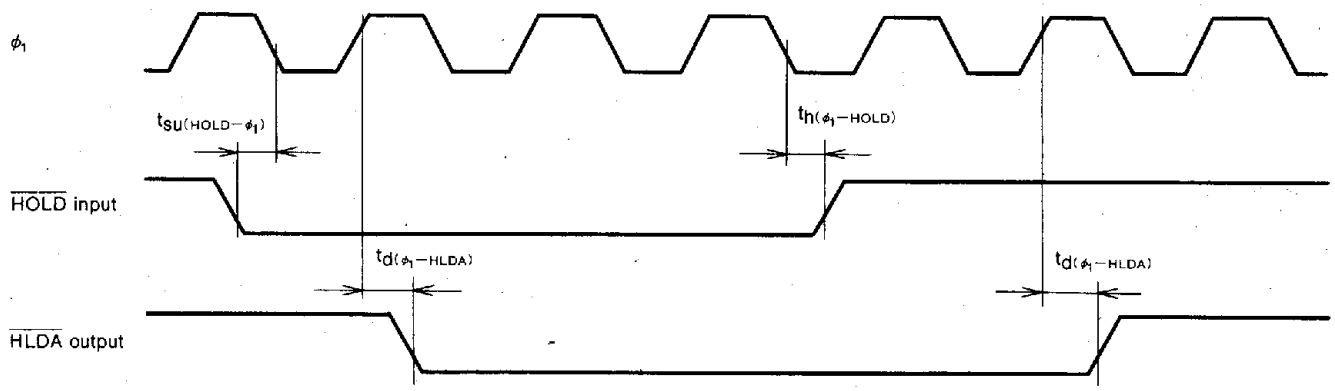
(When wait bit = "1")



(When wait bit = "0")



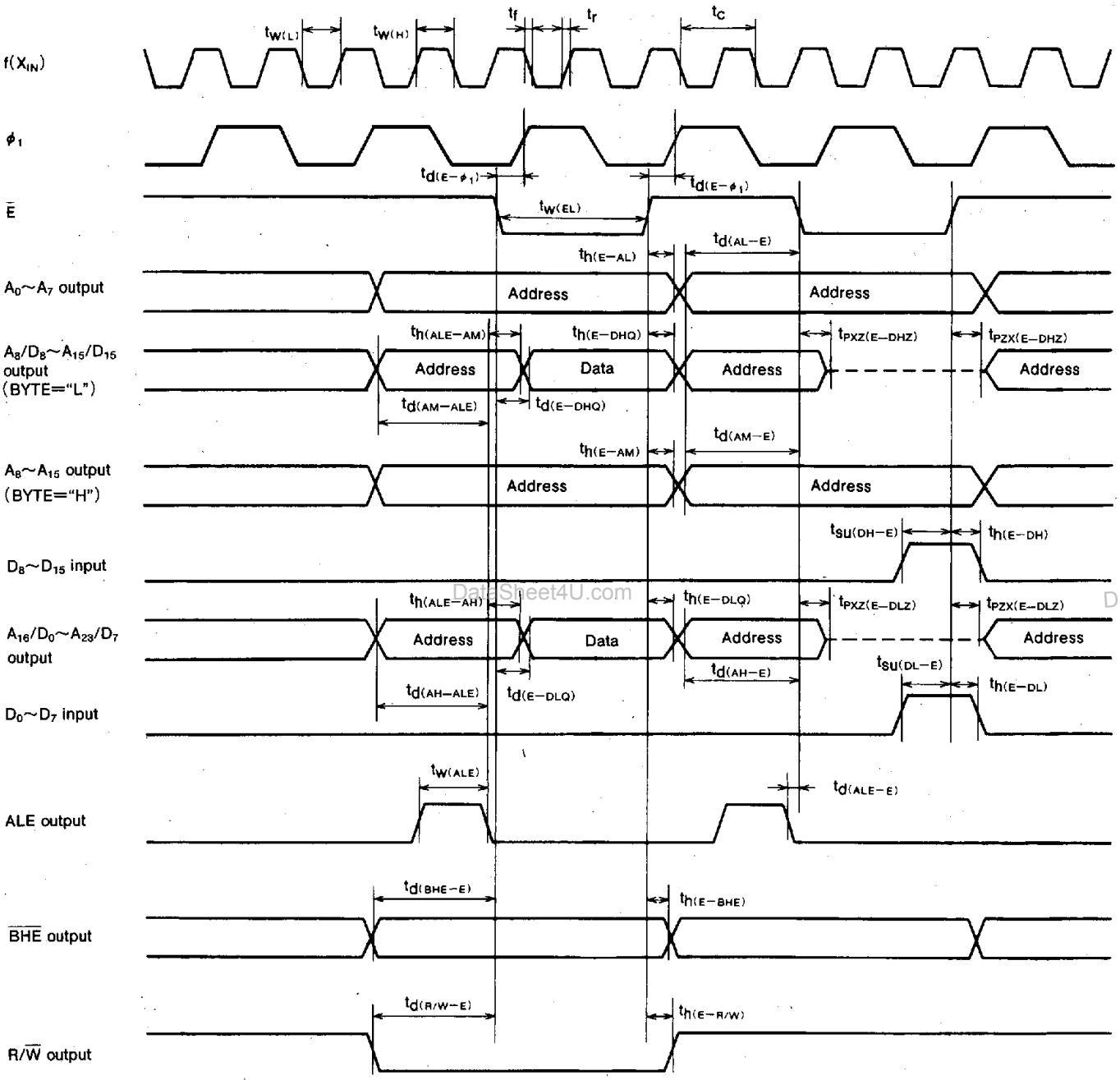
(When wait bit = "1" or "0" in common)



Test conditions

- V_{CC} = 5V±10%
- Input timing voltage : V_{IL} = 1.0V, V_{IH} = 4.0V
- Output timing voltage : V_{OL} = 0.8V, V_{OH} = 2.0V

Microprocessor mode (When wait bit = "1")

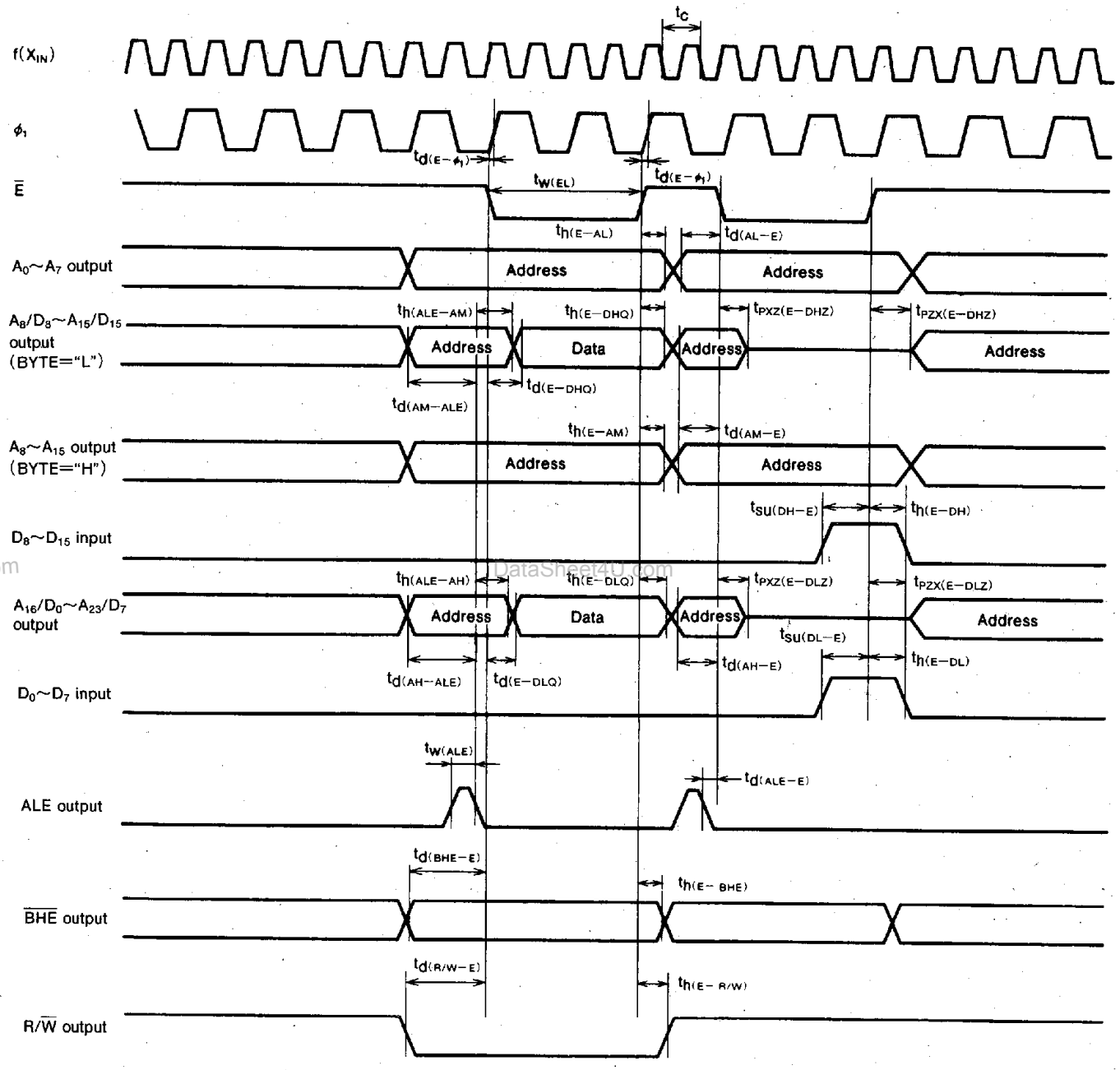


Test conditions

- V_{CC}=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0 to D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

Microprocessor mode

(When wait bit = "0", wait selection bit = "1", and external memory area is accessed)

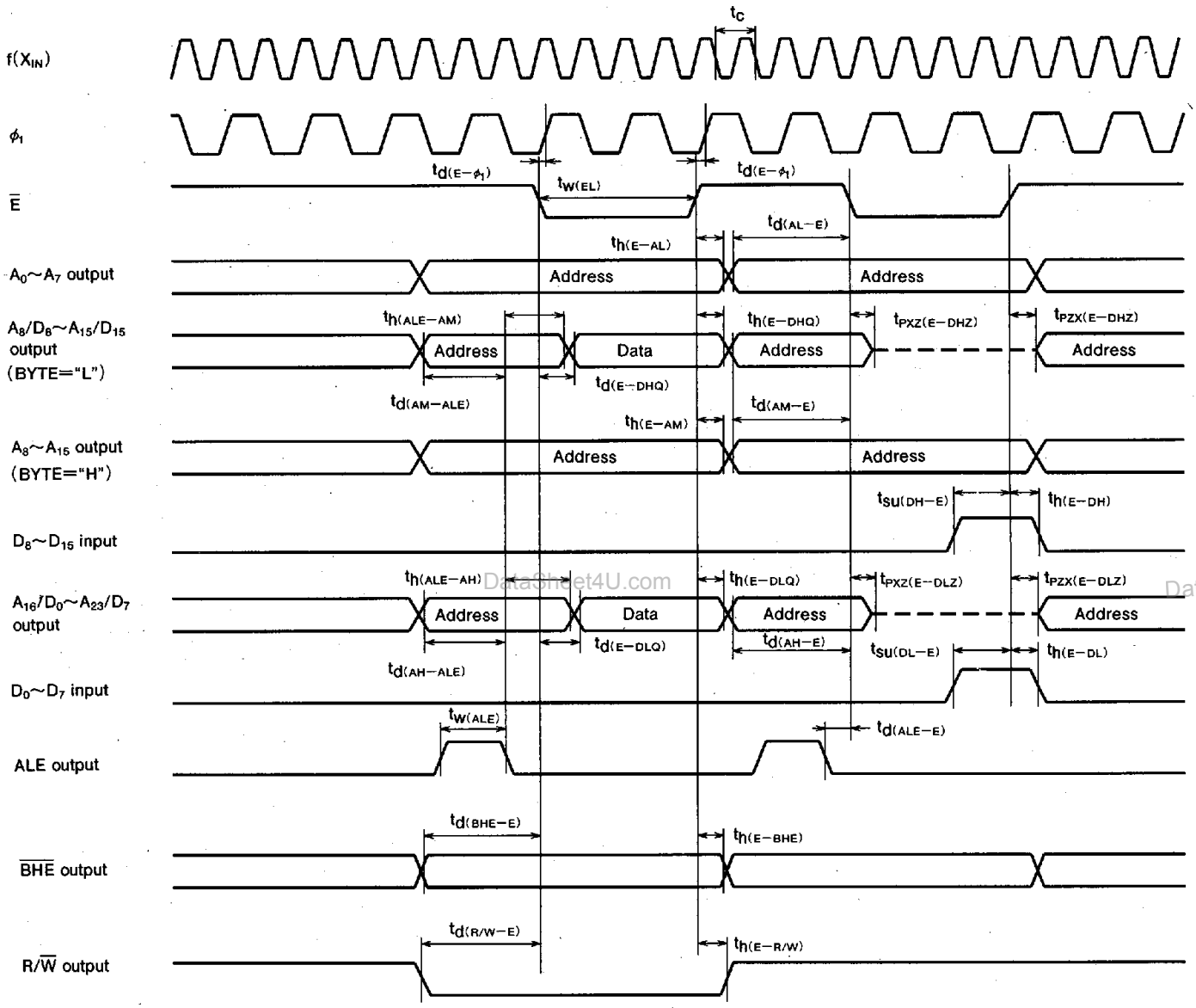


Test conditions

- V_{CC}=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0 to D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

Memory expansion mode and microprocessor mode

(When wait bit = "0", wait selection bit = "0", and external memory area is accessed)



Test conditions

- V_{CC}=5V±10%
- Output timing voltage : V_{OL}=0.8V, V_{OH}=2.0V
- D_0 to D_{15} input : V_{IL}=0.8V, V_{IH}=2.5V

8/16-bit Data Bus Flash Memory Card

MF84M1-G1EATXX

Connector Type

Two-piece 68-pin

DESCRIPTION

The MF84M1-G1EATXX is a flash memory card which uses sixteen two-megabit flash electrically erasable and programmable read only memory IC's.

FEATURES

- 68 pin JEIDA/PCMCIA
- 8/16 controllable data bus width
- Buffered interface
- TTL interface level
- Program/erase operation by software command control
- Program/erase voltage 12V
- 10,000 program/erase cycles
- Write protect switch

APPLICATIONS

- Note book computers
- Printers
- Industrical machines

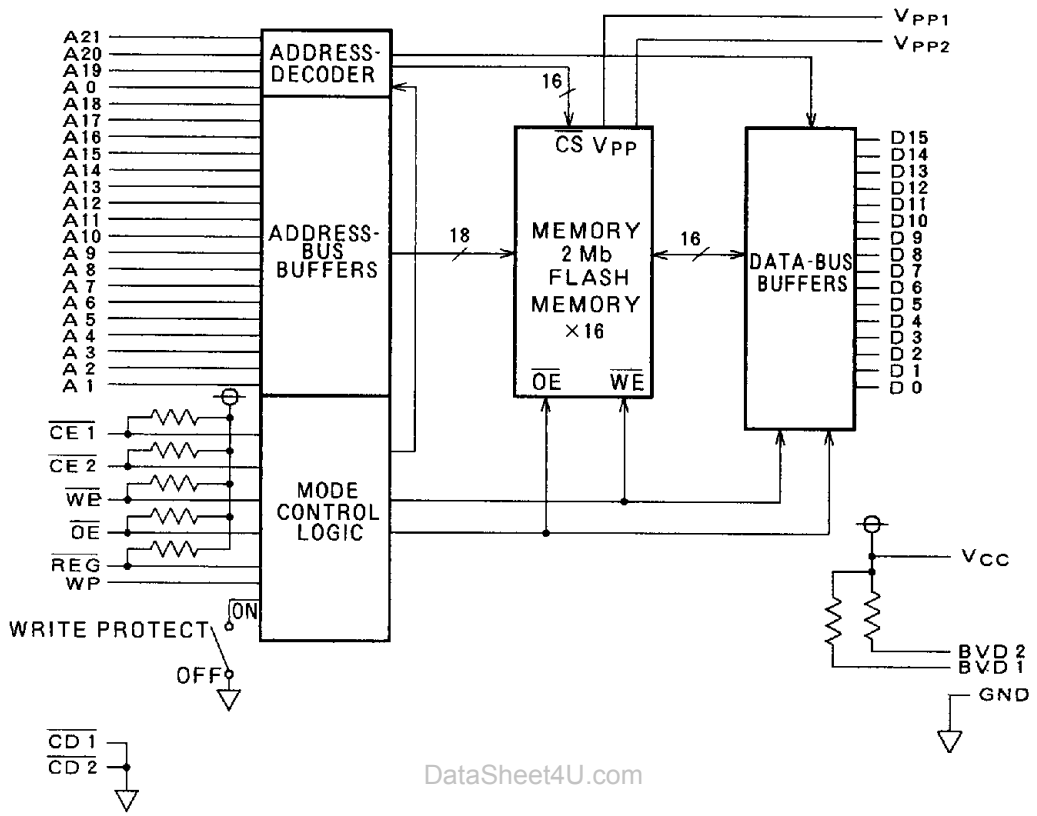
PRODUCT LIST

Item / Type name	Memory capacity	Data bus width (bits)	Access time (ns)	Connector type	Number of pins	Outline drawing
MF84M1-G1EATXX	4 MB	8/16	250	Two-piece	68	68P-002

PIN ASSIGNMENT

Pin No.	Symbol	Function	Pin No.	Symbol	Function
1	GND	Ground	35	GND	Ground
2	D3	Data I/O	36	$\overline{CD1}$	Card detect 1
3	D4		37	D11	Data I/O
4	D5		38	D12	
5	D6		39	D13	
6	D7		40	D14	
7	$\overline{CE1}$	Card enable 1	41	D15	
8	A10	Address input	42	$\overline{CE2}$	Card enable 2
9	$\overline{OE}$	Output enable	43	NC	No connection
10	A11	Address input	44	NC	
11	A9		45	NC	
12	A8		46	A17	Address input
13	A13		47	A18	
14	A14		48	A19	
15	$\overline{WE}$	Write enable	49	A20	
16	NC	No connection	50	A21	
17	V_{CC}	Power supply voltage	51	V_{CC}	Power supply voltage
18	$V_{PP}1$	Programming supply voltage 1	52	$V_{PP}2$	Programming supply voltage 2
19	A16	Address input	53	NC	No connection
20	A15		54	NC	
21	A12		55	NC	
22	A7		56	NC	
23	A6		57	NC	
24	A5		58	NC	
25	A4		59	NC	
26	A3		60	NC	
27	A2		61	$\overline{REG}$	Attribute memory select
28	A1		62	BVD2	Battery voltage detect 2
29	A0		63	BVD1	Battery voltage detect 1
30	D0	Data I/O	64	D8	Data I/O
31	D1		65	D9	
32	D2		66	D10	
33	WP	Write protect	67	$\overline{CD2}$	Card detect 2
34	GND	Ground	68	GND	Ground

BLOCK DIAGRAM (MF84M1-G1EATXX)



FUNCTIONAL DESCRIPTION

The operating mode of the card is determined by five active low control signals ($\overline{REG}$, $\overline{CE1}$, $\overline{CE2}$, $\overline{OE}$, $\overline{WE}$), three supply voltages (V_{CC}, V_{PP1}, V_{PP2}) and control registers located in each memory IC.

Common memory function

When the $\overline{REG}$ signal is set to a high level common memory is selected.

Read only mode

When the voltages applied to both V_{PP1} and V_{PP2} are less than the voltage applied to V_{CC} (i.e. V_{PP} = 0V to V_{CC}), the control registers of each memory IC are set to read only mode.

Operation of the card then depends on the four possible combinations of $\overline{CE1}$ and $\overline{CE2}$ (note/WE should be set to a high level when the device is in read only mode except during combination (4) where it's condition is unimportant) :

(1) If $\overline{CE1}$ is set to a low level and $\overline{CE2}$ is set to a high level, the card will work as an eight bit data bus width card. Data can be accessed via the lower half of the data bus (D0 to D7).

(2) If both $\overline{CE1}$ and $\overline{CE2}$ are set to a low level, data will be accessible via the full sixteen bit data bus width of the card. In this mode LSB of address bus (A0) is ignored.

(3) If $\overline{CE1}$ is set to a high level and $\overline{CE2}$ is set to a low level the odd bytes (only) can be accessed through upper half of the data bus (D8 to D15). This mode is useful when handling the odd (upper) bytes in a sixteen bit interface system. Note that A0 is also ignored in this operating condition.

(4) If $\overline{CE1}$ and $\overline{CE2}$ are set to a high level, the card will be in standby mode where it consumes low power. The data bus is kept high impedance.

When $\overline{OE}$ is set to a low level data can be read from the card, depending on the address applied and the setting of $\overline{CE1}$ and $\overline{CE2}$ as mentioned above, except under combination (4).

When $\overline{OE}$ is set to a high level and $\overline{WE}$ is set to a high level the card is in an output disable mode and the data bus will be in a high impedance state regardless of the condition of $\overline{CE1}$ and $\overline{CE2}$.

Read/write mode

When a programming voltage ($V_{PP}H$) is applied to either or both of $V_{PP}1$ and $V_{PP}2$, read/write mode is enabled for the corresponding banks of memory IC's inside the card. $V_{PP}1$ enables the Even Byte bank and $V_{PP}2$ enables the Odd Byte bank.

By using the 4 combinations of $\overline{CE1}$ and $\overline{CE2}$ as descibed under Read only mode above the appropriate Data Out and Command/Data In bus selection can be made.

If $\overline{OE}$ is set to a high level and $\overline{WE}$ set to a low level, the control register will latch command data applied at the rising edge of the $\overline{WE}$ signal. Note that more than one bus cycle may be required to latch the command and/or the related data – please refer to the Command Definition table.

If $\overline{OE}$ is set to a low level and $\overline{WE}$ is set to a high level the card data can be read from the card depending on the condition of the control register.

After latching the command data, the card will go into programming, erasure or other operation mode. For details please refer to the Command Definition table, each individual command's definition and the programming and erasure algorithms.

Attribute memory

When $\overline{REG}$ is set to a low level attribute memory is selected.

The card then outputs FFh on the lower half of the data bus (D0 to D7) when the following conditions are applied :

(1) $\overline{CE1}$: low level, $\overline{CE2}$: high level, $\overline{OE}$: low level, $\overline{WE}$: high level, A0 : low level
(2) $\overline{CE1}$: low level, $\overline{CE2}$: low level, $\overline{OE}$: low level, $\overline{WE}$: high level.

Write protect mode

The card has a write protect switch on the opposite edge to the connector edge. When it is switched on, the card will be placed into a write protect mode, where data can be read from the card but it cannot be written to it. The WP output pin is set to a high level when the card is in write protect mode and V_{CC} is applied. When the card is not in write protect mode the WP output pin is set to a low level when V_{CC} is applied. By reading the state of the WP output the host system can easily check whether the card is in write protect mode or not.