3888 Group

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

DESCRIPTION

The 3888 group is the 8-bit microcomputer based on the 740 family core technology.

The 3888 group is a protocol controller for data communication between processors in the multi-processor system and includes a 3-channel UART and a dual port RAM

For details on availability of microcomputers in the 3888 group, refer to the section on group expansion.

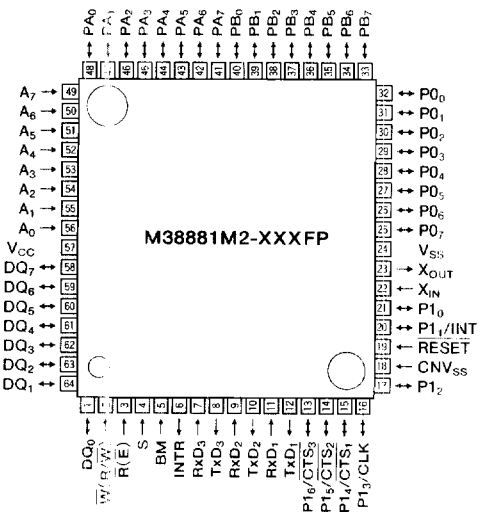
FEATURES

- Basic machine-language instructions 71
- The minimum instruction execution time ..... 0.32μs (at 12.5MHz oscillation frequency)
- Memory size
  - ROM ..... 8K bytes
  - RAM ..... 240 bytes
- Programmable I/O ports
  - (System data bus) 16
  - (Local data bus) 15
- Interrupts ..... 14 sources, 14 vectors
- Timers ..... 8 bit×4
- UART 3-channel
- Bus interface (RD/WR separate mode bus or 7700 mode bus)
- Dual port RAM ..... 216 bytes
- Communication registers
- Clock generating circuit ..... Internal feedback amplifier (connect to an external ceramic resonator or a quartz-crystal oscillator)
- Power source voltage ..... 4.5 to 5.5V
- Power dissipation 45mW
- Operating temperature range ..... −20 to +85℃

APPLICATIONS

Office equipment, etc.

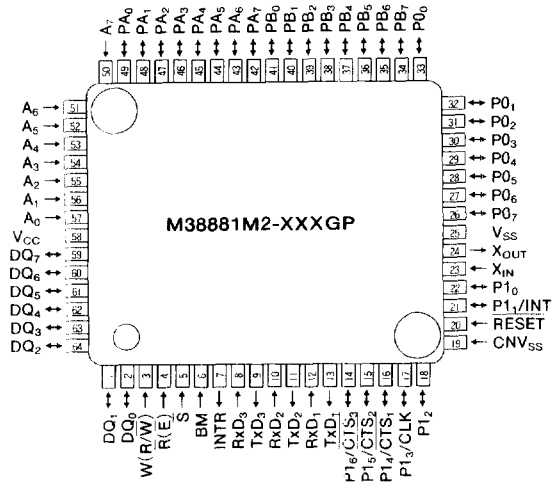
PIN CONFIGURATION (TOP VIEW)



Package type : 64P6N-A

64-pin plastic-molded QFP

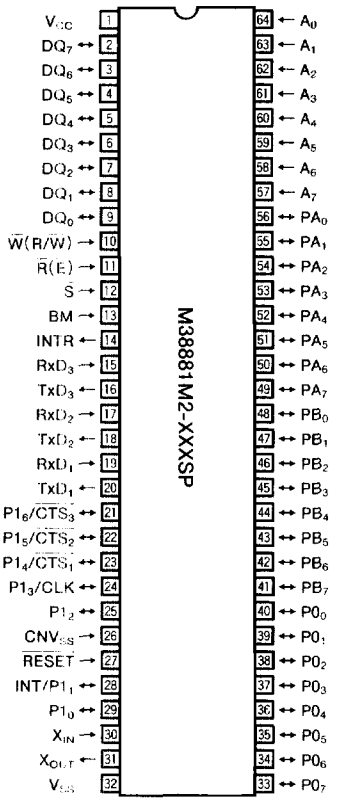
PIN CONFIGURATION (TOP VIEW)



Package type : 64P6S-A

64-pin plastic-molded QFP

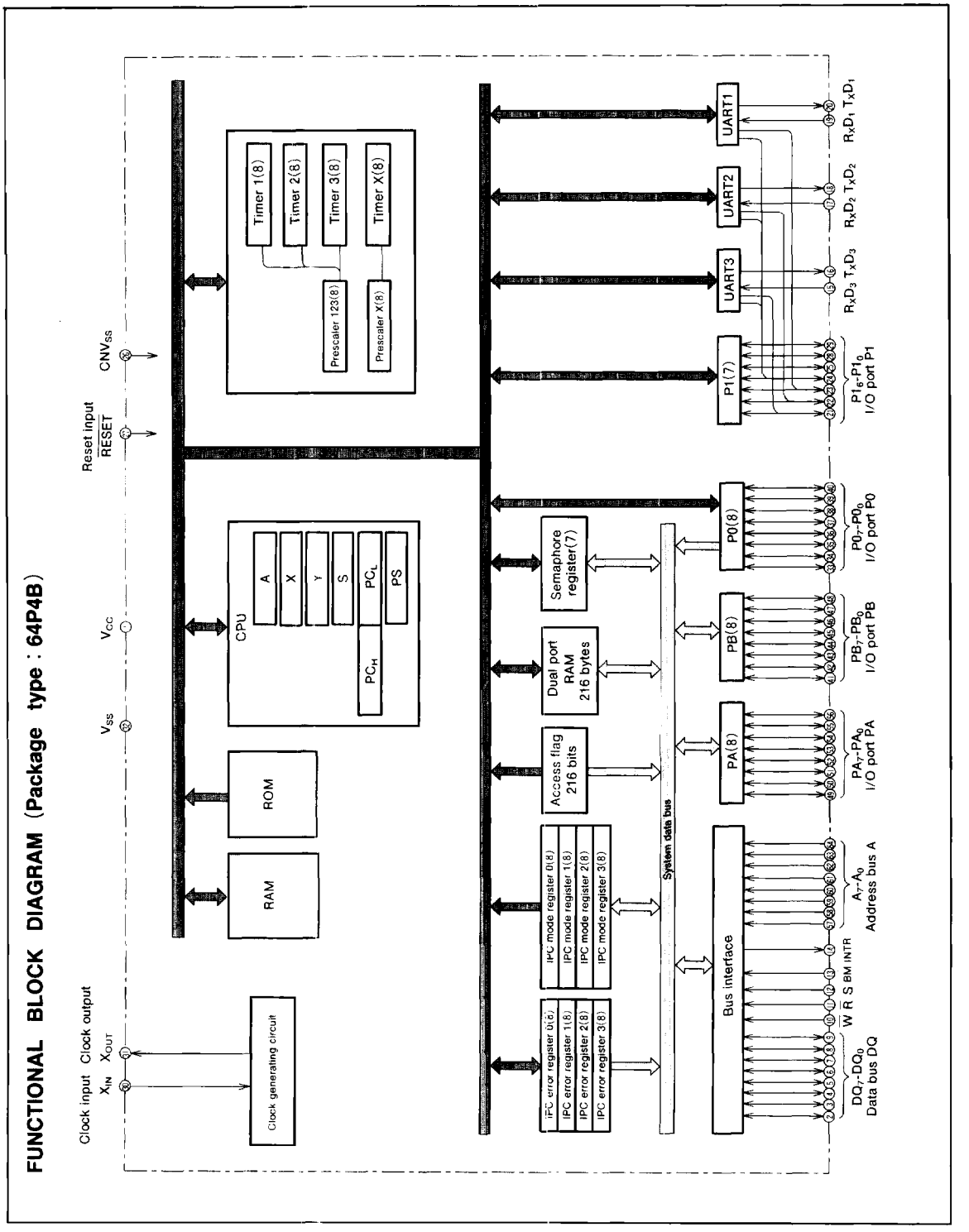
PIN CONFIGURATION (TOP VIEW)



Package type : 64P4B

64-pin shrink plastic-molded DIP

FUNCTIONAL BLOCK DIAGRAM (Package type : 64P4B)



MITSUBISHI ELECTRIC

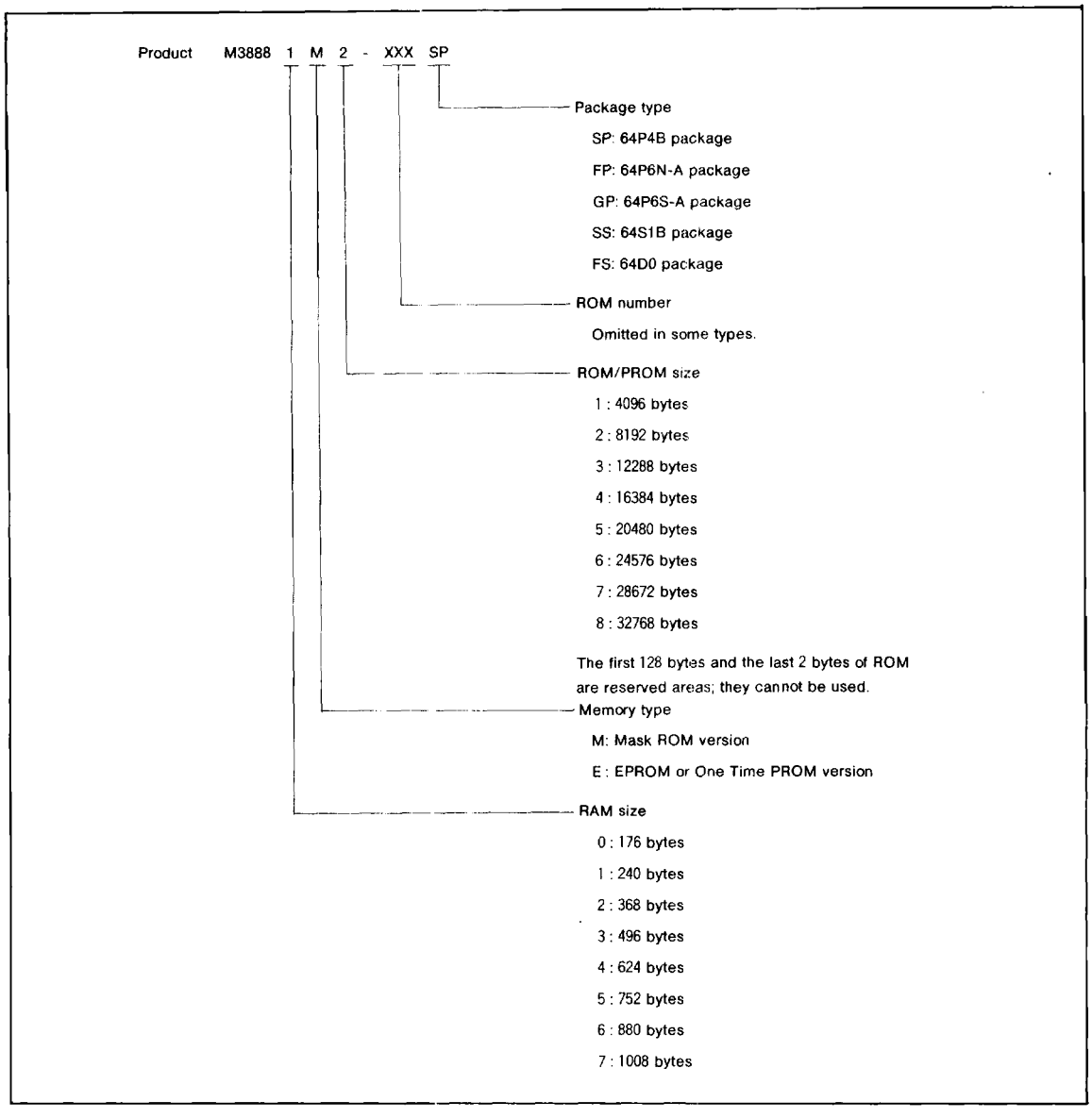
PIN DESCRIPTION

Pin	Name	Function	Alternate Function
V_{CC}, V_{SS}	Power source	• Apply voltage of 4.5 to 5.5V to V_{CC}, and 0V to V_{SS}.	
CNV_{SS}	CNV_{SS}	• Controls the operation mode of the chip. • This pin must be connected with V_{SS}.	
$\overline{RESET}$	Reset input	• Active "L"-type reset input pin	
X_{IN}	Clock input	• Input and output signals for the internal clock generating circuit. • Connect a ceramic resonator or a quartz-crystal oscillator between the X_{IN} and X_{OUT} pins to set the oscillation frequency. • When an external clock is used, connect the clock source to the X_{IN} pin and leave the X_{OUT} pin open.	
X_{OUT}	Clock output		
$P0_0$-$P0_7$	I/O port P0	• 8-bit I/O port connected to the local data bus • I/O direction register allows each pin to be individually programmed as either input or output. • The local bus port pull-up register allows each pin to be individually programmed to connect to a pull-up transistor or not. • This port is also connected to the system data bus. From the system data bus, the contents of each port can be read (it is used as an input port). • CMOS compatible input level • CMOS 3-state output	
$P1_0$	I/O port P1	• 7-bit I/O port connected to the local data bus as the same function as P0 • CMOS compatible input level • CMOS 3-state output	
$P1_1$/INT			External interrupt input pin
$P1_2$			
$P1_3$/CLK			UART1-3 function input pin
$P1_4/\overline{CTS_1}$			UART1 function input pin
$P1_5/\overline{CTS_2}$			UART2 function input pin
$P1_6/\overline{CTS_3}$			UART3 function input pin
PA_0-PA_7	I/O port PA	• 8-bit I/O ports connected to the system data bus with the same function as P0 • CMOS compatible input level • CMOS 3-state output	
PB_0-PB_7	I/O port PB		
A_0-A_7	Address input	• Input 8-bit system addresses. • TTL compatible input level	
DQ_0-DQ_7	Data input/output	• Input or output 8-bit system data. • TTL compatible input level	
$\overline{S}$	Chip select input	• When "L" is input to this pin, read and write of system data are enabled. • TTL compatible input level	
$\overline{R}$ ($\overline{E}$)	Read control input (Enable input)	• When "L" is input to this pin, the contents of the memory or register specified by A_0-A_7 are read from DQ_0-DQ_7. (In the 7700 mode, when "L" is input to this pin, read and write of system data are enabled.) • TTL compatible input level	
$\overline{W}$ (R/$\overline{W}$)	Write control input (Read/write input)	• When "L" is input to this pin, data input from DQ_0-DQ_7 is written into the memory or register specified by A_0-A_7. (In the 7700 mode, when "H" is input to this pin, read of system data is enabled,and when "L" is input to this pin, write of system data is enabled.) • TTL compatible input level	
BM	Bus mode control input	• The input level to this pin controls the functions of $\overline{R}$ and $\overline{W}$ • CMOS compatible input level	

PIN DESCRIPTION (CONTINUED)

Pin	Name	Function	Alternate Function
INTR	Interrupt request output	• When the contents of IPC error register are modified by the internal CPU, "H" is output. • When the register is read by the external host CPU, "L" is output.	
TxD_1-TxD_3	UART transmit output	• Output pin for UART transmit data	
RxD_1-RxD_3	UART receive input	• Input pin for UART receive data • CMOS compatible input level	

PART NUMBERING



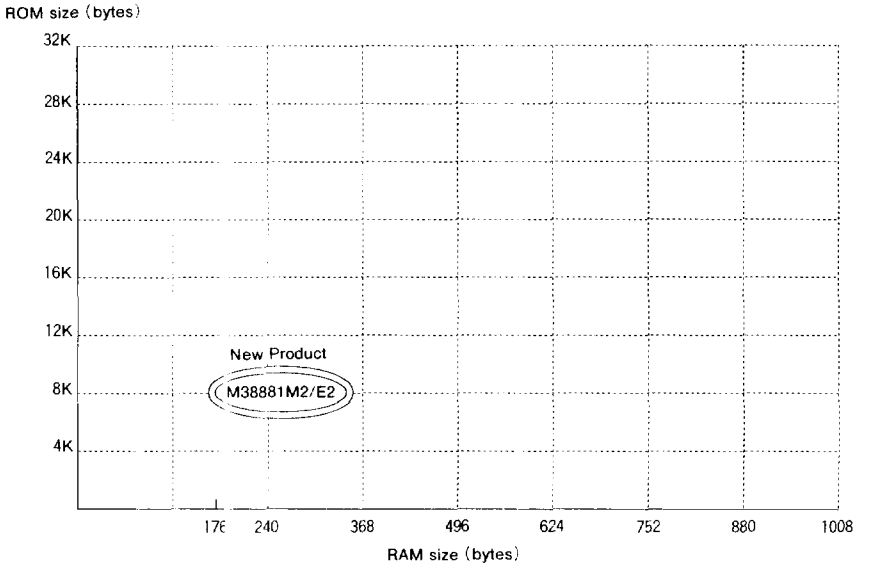
GROUP EXPANSION

Mitsubishi plans to expand the 3888 group as follows:

(1) Support for mask ROM, One Time PROM, and EPROM versions
ROM/PROM size ..... 8K bytes
RAM size ..... 240 bytes

(2) Packages
64P4B ..... Shrink plastic molded DIP
64P6N-A ..... 0.8mm-pitch plastic molded QFP
64P6S-A ..... 0.65mm-pitch plastic molded QFP
64S1B ..... Shrink ceramic DIP (EPROM version)
64D0 .....0.8mm-pitch ceramic LCC (EPROM version)

Memory Expansion Plan



Currently supported products are listed below.

As of September 1994

Product	(P) ROM size (bytes) ROM size for User in ( )	RAM size (bytes)	Package	Remarks
M38881M2-XXXSP	8192 (8062)	240	64P4B	Mask ROM version
M38881E2-XXXSP				One Time PROM version
M38881E2SP				One Time PROM version (blank)
M38881E2SS			64S1B	EPROM version
M38881M2-XXXFP	8192 (8062)	240	64P6N-A	Mask ROM version
M38881E2-XXXFP				One Time PROM version
M38881E2FP				One Time PROM version (blank)
M38881E2FS			64D0	EPROM version
M38881M2-XXXGP	8192 (8062)	240	64P6S-A	Mask ROM version
M38881E2-XXXGP				One Time PROM version
M38881E2GP				One Time PROM version (blank)

MITSUBISHI ELECTRIC

FUNCTIONAL DESCRIPTION
CENTRAL PROCESSING UNIT (CPU)

The 3888 group uses the standard 740 family instruction set. Refer to the table of 740 family addressing modes and machine instructions or the SERIES 740 〈Software〉 User's Manual for details on the instruction set.

Machine-resident 740 family instructions are as follows:

The FST and SLW instruction cannot be used.

The STP, WIT, MUL, and DIV instruction can be used.

CPU Mode Register

The CPU mode register is allocated at address $003B_{16}$.

The CPU mode register contains the stack page selection bit and the processor mode bits.

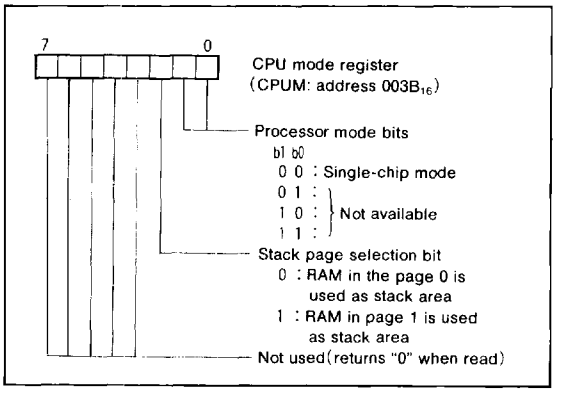


Fig. 1 Structure of CPU mode register

MEMORY

Special Function Register (SFR) Area

The Special Function Register area in the zero page contains control registers such as I/O ports and timers.

RAM

RAM is used for data storage and for stack area of subroutine calls and interrupts.

ROM

The first 128 bytes and the last 2 bytes of ROM are reserved for device testing and the rest is user area for storing programs.

Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

Zero Page

The 256 bytes from addresses 0000_{16} to $00FF_{16}$ are called the zero page area. The internal RAM and the special function registers (SFR) are allocated to this area.

The zero page addressing mode can be used to specify memory and register addresses in the zero page area. Access to this area with only 2 bytes is possible in the zero page addressing mode.

Special Page

The 256 bytes from addresses $FF00_{16}$ to $FFFF_{16}$ are called the special page area. The special page addressing mode can be used to specify memory addresses in the special page area. Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

RAM size (bytes)	Address $XXXX_{16}$
176	$00FF_{16}$
240	$013F_{16}$
368	$01BF_{16}$
496	$023F_{16}$
624	$02BF_{16}$
752	$033F_{16}$
880	$03BF_{16}$
1008	$043F_{16}$

ROM area

ROM size (bytes)	Address $YYYY_{16}$	Address $ZZZZ_{16}$
4096	$F000_{16}$	$F080_{16}$
8192	$E000_{16}$	$E080_{16}$
12288	$D000_{16}$	$D080_{16}$
16384	$C000_{16}$	$C080_{16}$
20480	$B000_{16}$	$B080_{16}$
24576	$A000_{16}$	$A080_{16}$
28672	9000_{16}	9080_{16}
32768	8000_{16}	8080_{16}

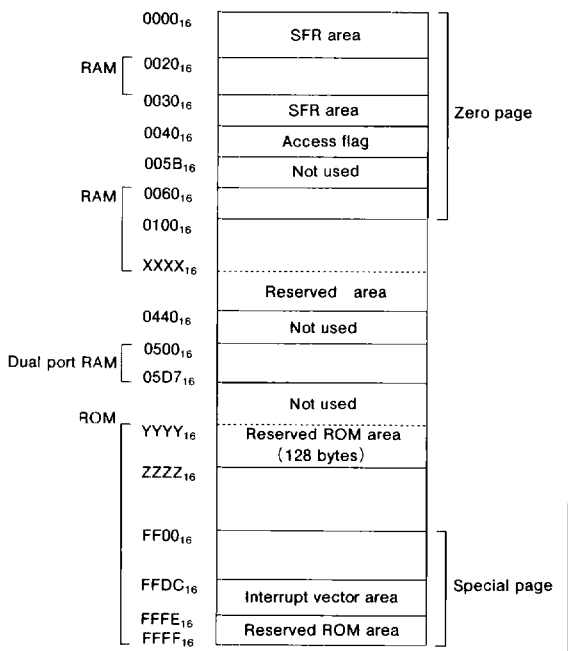


Fig. 2 Memory map diagram

0000_{16}	Port P0 (P0)
0001_{16}	Port P0 direction register (P0D)
0002_{16}	
0003_{16}	
0004_{16}	
0005_{16}	
0006_{16}	Port P1 (P1)
0007_{16}	Port P1 direction register (P1D)
0008_{16}	UART1 transmit/receive buffer register(U1TB/U1RB)
0009_{16}	UART1 status/control register 1(U1STS/U1CON1)
$000A_{16}$	UART1 control register 2(U1CON2)
$000B_{16}$	UART1 baud rate generator(U1BRG)
$000C_{16}$	UART2 transmit/receive buffer register(U2TB/U2RB)
$000D_{16}$	UART2 status/control register 1(U2STS/U2CON1)
$000E_{16}$	UART2 control register 2(U2CON2)
$000F_{16}$	UART2 baud rate generator(U2BRG)
0010_{16}	UART3 transmit/receive buffer register(U3TB/U3RB)
0011_{16}	UART3 status/control register 1 (U3STS/U3CON1)
0012_{16}	UART3 control register 2(U3CON2)
0013_{16}	UART3 baud rate genarator(U3BRG)
0014_{16}	IPC mode register 0(IPCM0)
0015_{16}	IPC mode register 1(IPCM1)
0016_{16}	IPC mode register 2(IPCM2)
0017_{16}	IPC mode register 3(IPCM3)
0018_{16}	IPC error register 0(IPCER0)
0019_{16}	IPC error register 1(IPCER1)
$001A_{16}$	IPC error register 2(IPCER2)
$001B_{16}$	IPC error register 3(IPCER3)
$001C_{16}$	Semaphore register(SEM)
$001D_{16}$	DPRAM direction register(DPRD)
$001E_{16}$	DPRAM map register(DPRM)
$001F_{16}$	Reserved area(Do not access)

0020_{16}	RAM
0021_{16}	RAM
0022_{16}	RAM
0023_{16}	RAM
0024_{16}	RAM
0025_{16}	RAM
0026_{16}	RAM
0027_{16}	RAM
0028_{16}	RAM
0029_{16}	RAM
$002A_{16}$	RAM
$002B_{16}$	RAM
$002C_{16}$	RAM
$002D_{16}$	RAM
$002E_{16}$	RAM
$002F_{16}$	RAM
0030_{16}	Prescaler 123(PRE123)
0031_{16}	Timer 1(T1)
0032_{16}	Timer 2(T2)
0033_{16}	Timer 3(T3)
0034_{16}	Prescaler X(PREX)
0035_{16}	Timer X(TX)
0036_{16}	
0037_{16}	
0038_{16}	
0039_{16}	Local bus port pull-up register(LPU)
$003A_{16}$	
$003B_{16}$	CPU mode register (CPUM)
$003C_{16}$	Interrupt request register 1 (IREQ1)
$003D_{16}$	Interrupt request register 2 (IREQ2)
$003E_{16}$	Interrupt control register 1 (ICON1)
$003F_{16}$	Interrupt control register 2 (ICON2)

Fig. 3 Memory map of special function register (SFR)

I/O PORTS

Port to be Accessed from System Bus or Local Bus

The port PA and port PB are connected to the system bus.
The port P1 is connected to the local bus. The port P0 is connected to the local bus and system bus.
When accessing the port P0 from the system bus, only data reading (a function as an input port) is enabled.
To access a port connected to the system bus, select a port-related register to be accessed by the system bus port control register (system bus address FF_{16}), then access the port through the system bus port data register (system bus address FE_{16}).

Direction Registers

The 3888 group has 31 programmable I/O pins arranged in 4 I/O ports (ports P0, P1, PA, and PB). The I/O ports have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, each pin can be set to be input or output. When "0" is written into the bit corresponding to a pin, that pin becomes an input pin. When "1" is written into that bit, that pin becomes an output pin.
If data is read from a pin which is set for output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating. However, when a pin of port P0 is read from the system bus side, the value of the pin is read regardless of the value of port P0 direction register.

Pull-up Control

Every port can be connected to a pull-up transistor when the corresponding pull-up set bit is "1".
Every I/O port is set to be an input port at reset, and then is disconnected from a pull-up transistor.

Pin	Name	Input/Output	I/O Format	Non-Port Function	Related SFRs	Diagram No
$P0_0$-$P0_7$	Port P0	Input/Output, individual bits	CMOS compatible input level, CMOS 3-state output		Local bus port pull-up register	(1)
$P1_0$, $P1_2$	Port P1					(2)
$P1_1$/INT				External interrupt input		(3)
$P1_3$/CLK				UART function input	UARTi status/control register 1	(4)
$P1_4$/CTS_1						(5)
$P1_5$/CTS_2					Local bus port pull-up register	(5)
$P1_6$/CTS_3						(5)
PA_0-PA_7	Port PA					(6)
PB_0-PB_7	Port PB					(6)

Note : Make sure that the input level at each pin is either 0V or V_{CC} during execution of the STP instruction. When an input level is at the intermediate potential, a current will flow from V_{CC} to V_{SS} through the input-stage gate.

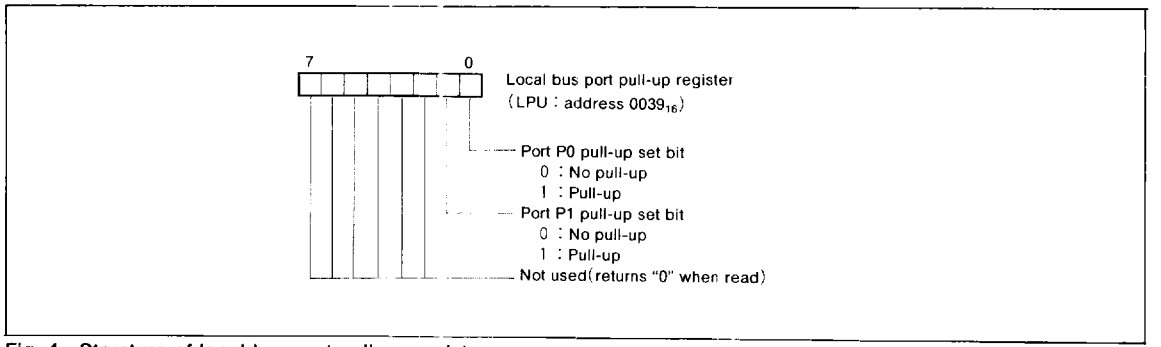


Fig. 4 Structure of local bus port pull-up register

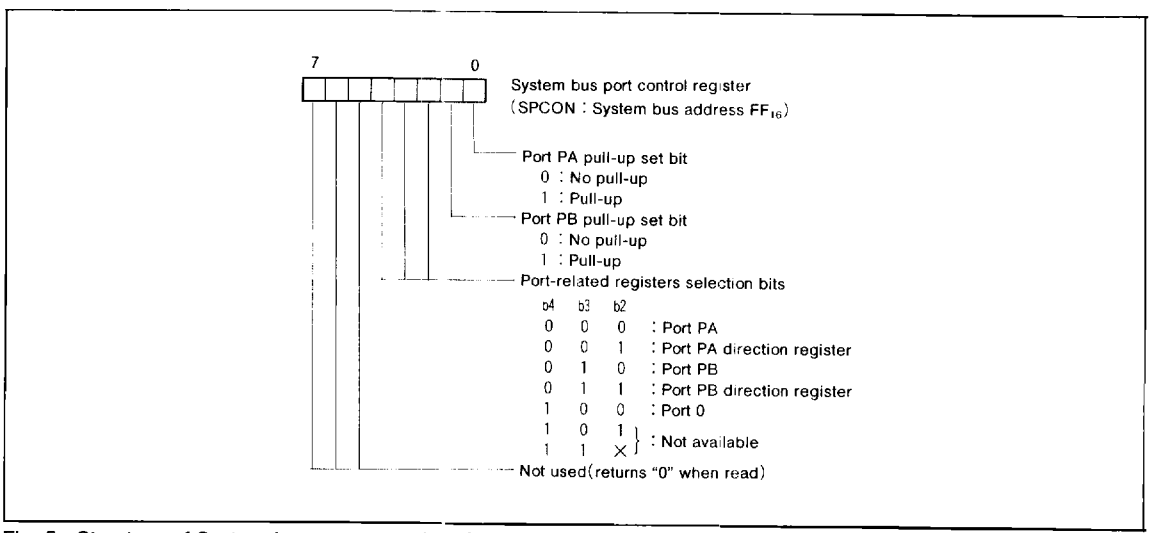
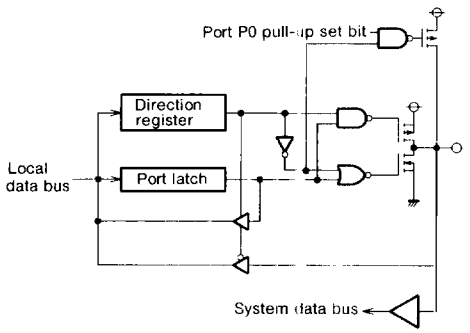
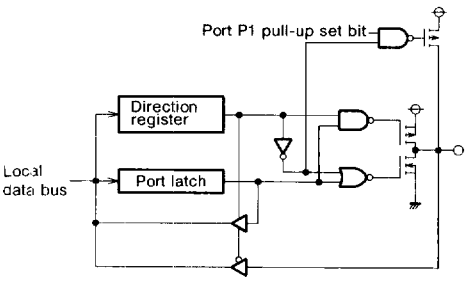


Fig. 5 Structure of System bus port control register

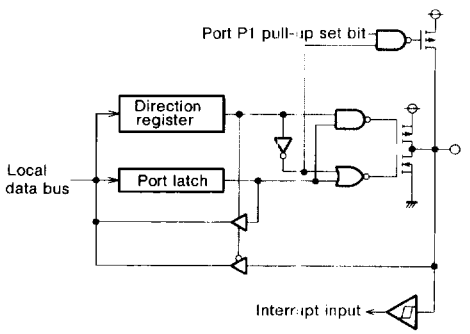
(1) Port P0



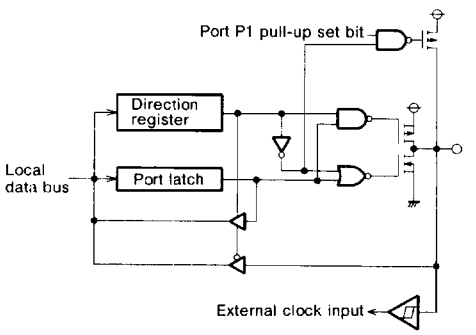
(2) Ports $P1_0$, $P1_2$



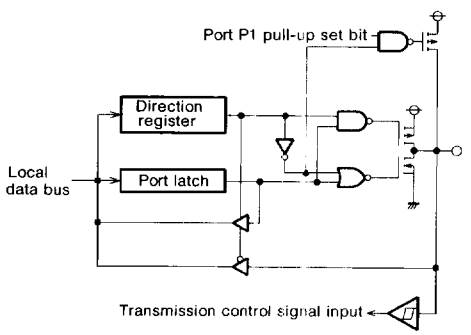
(3) Port $P1_1$



(4) Port $P1_3$



(5) Ports $P1_4$-$P1_6$



(6) Ports PA, PB

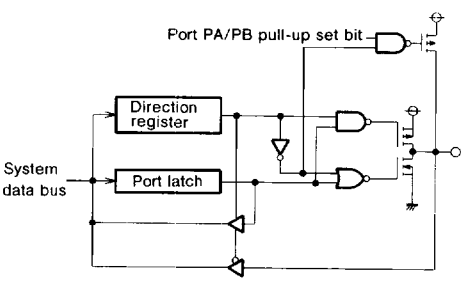


Fig. 6 Port block diagram

INTERRUPTS

A total of 14 sources can generate interrupts: 2 system bus, 1 external, 10 internal, and 1 software.

Interrupt Control

Each interrupt is controlled by an interrupt request bit, an interrupt enable bit, and the interrupt disable flag-except for the software interrupt set by the BRK instruction. An interrupt occurs when the corresponding interrupt request and enable bits are "1" and the interrupt disable flag is "0".
Interrupt enable bits can be set or cleared by software.
Interrupt request bits can be cleared by software, but cannot be set by software.
The I flag disables all interrupts except the BRK instruction interrupt.

Interrupt Operation

When an interrupt is honored, the contents of the program counter and processor status register are automatically stored into the stack. The interrupt disable flag is set to inhibit other interrupts from interfering. The corresponding interrupt request bit is cleared and the interrupt jump destination address is read from the vector table into the program counter.

Table 1. Interrupt vector addresses and priorities

Interrupt source	Priority	Vector address (Note 1)		Interrupt request generating conditions	Remarks
		High	Low		
Reset (Note 2)	1	$FFFD_{16}$	$FFFC_{16}$	At reset	Non-maskable
UART1 receive	2	$FFFB_{16}$	$FFFA_{16}$	At completion of UART1 reception	
UART2 receive	3	$FFF9_{16}$	$FFF8_{16}$	At completion of UART2 reception	
UART3 receive	4	$FFF7_{16}$	$FFF6_{16}$	At completion of UART3 reception	
IPC mode register 0	5	$FFF5_{16}$	$FFF4_{16}$	At writing into IPC mode register 0 from system data bus	
IPC error register 0	6	$FFF3_{16}$	$FFF2_{16}$	At reading IPC error register 0 from system data bus	
Timer X	7	$FFF1_{16}$	$FFF0_{16}$	At timer X underflow	
Timer 1	8	$FFEF_{16}$	$FFEE_{16}$	At timer 1 underflow	STP release timer underflow
Timer 2	9	$FFED_{16}$	$FFEC_{16}$	At timer 2 underflow	
Timer 3	10	$FFEB_{16}$	$FFEA_{16}$	At timer 3 underflow	
UART1 transmit	11	$FFE9_{16}$	$FFE8_{16}$	At UART1 transmit buffer emptying	
UART2 transmit	12	$FFE7_{16}$	$FFE6_{16}$	At UART2 transmit buffer emptying	
UART3 transmit	13	$FFE5_{16}$	$FFE4_{16}$	At UART3 transmit buffer emptying	
INT	14	$FFE3_{16}$	$FFE2_{16}$	At detection of rising edge of INT input	
BRK instruction	15	$FFDD_{16}$	$FFDC_{16}$	At BRK instruction execution	Non-maskable software interrupt

Note 1 : Vector addresses contain interrupt jump destination addresses.
2 : Reset function in the same way as an interrupt with the highest priority.

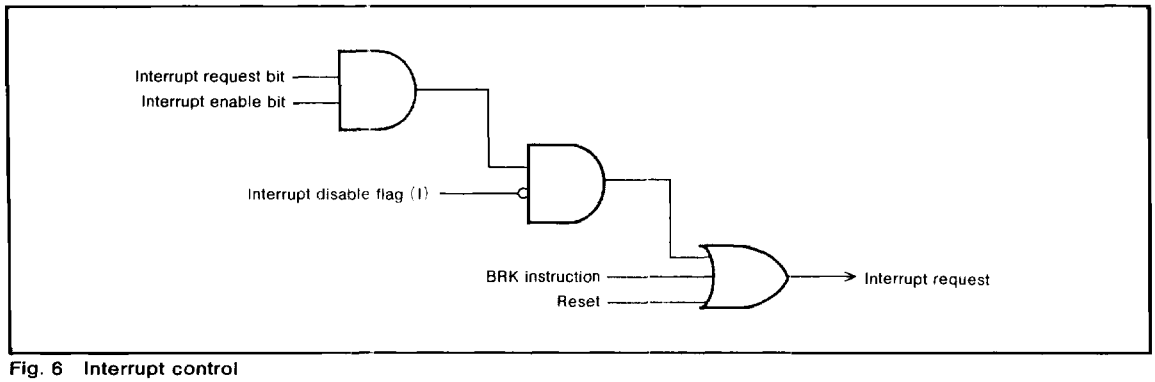


Fig. 6 Interrupt control

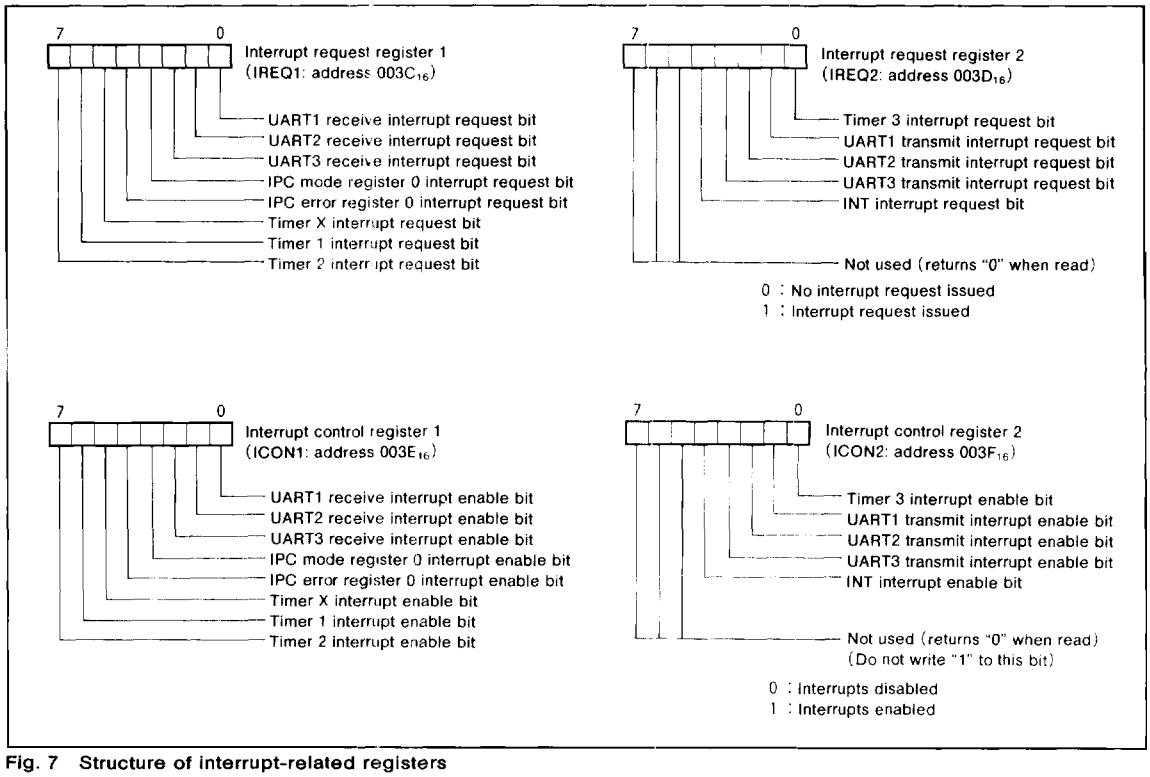


Fig. 7 Structure of interrupt-related registers

TIMERS

The 3888 group has 4 timers: timer X, timer 1, timer 2, and timer 3.

The timers count down. Once the contents of a timer counter reache "00_{16}", the next count pulse reloads the contents of the corresponding timer latch into the timer, and sets the corresponding interrupt request bit to "1".

The division ratio of each timer or prescaler is given by $1/(n+1)$, where n is the value set in the corresponding timer or prescaler latch.

Timer 1, Timer 2, and Timer 3

The count source of prescaler 123 is the oscillation frequency divided by 16. The output of prescaler 123 is counted by timer 2 and timer 3, and a timer underflow sets the corresponding interrupt request bit.

Timer X

The count source of prescaler X is the oscillation frequency divided by 16. The output of prescaler X is counted by timer X, and the timer X underflow sets the timer X interrupt request bit.

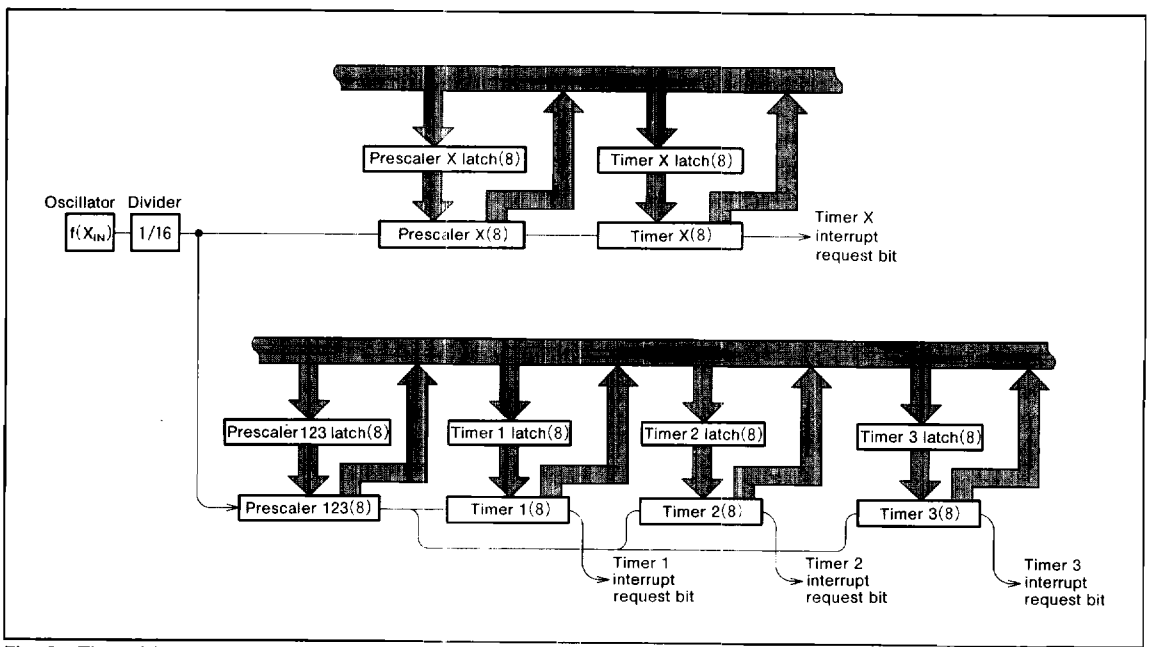


Fig. 8 Timer block diagram

UART

The 3888 group contains three channels of UART's (UARTi (i=1, 2, 3)). Functionally, they are all equal and can be separately operated. Each channel includes following:

- Transmit output pin TxDi
- Receive input pin RxDi
- Transmit control input pin $\overline{CTS}$i
- Receive/Transmit shift register
- Receive/Transmit buffer register
- UARTi control register 1
- UARTi control register 2
- UARTi status register
- Divider for baud rate generation

It also has a CLK pin (the input pin of the external clock for baud rate generation) which is shared by three channels. The $\overline{CTS}$i or CLK shares the same pin position with the port P1. An interrupt request can occur on each channel independently when the reception or transmission is completed. Because the differences between the channels are only pin numbers and internal addresses, the following description uses UART1 for reference.

Data Format

A data format for reception or transmission can be selected by setting the following bits:

- Parity enable bit (PARE)
- Parity selection bit (PARS)
- Character length selection bit (CHAS)
- Stop bit length selection bit (STPS)

Parity generation or detection is performed by hardware. When the character length is specified in 7 bits, "0" is read from bit 7 of the receive buffer register at receiving and the value of bit 7 in the transmit data is not transmitted.

When the stop bit length is specified in 2 bits, only the first stop bit is checked at receiving.

CRC

When transmitting or receiving multiple characters in a lump (block data), the CRC (Cyclic Redundancy Check) method can be used for error detection.

The generating polynomial is shown below:

$P(X) = X^8 + X^4 + X^3 + X^2 + 1$

The target data for CRC does not include the start bit, stop bit and parity bit.

To use the error detecting function of the CRC method, it is necessary to control the start and end of the block data by software.

The transmitter side transmits BBC (Block Check Character (Note 1)) after completion of the block data transmission.

The receiver side detects the completion of the block data reception by software, and detects whether or not any error exists, by the CRC remainder (Note 2) at that point of time.

Take the following items into consideration when using the CRC method.

- Specify data length in 8 bits.
- A parity bit can be appended.

Note 1 : Generated by the CRC coding circuit.

Note 2 : Generated by the CRC decoding circuit.

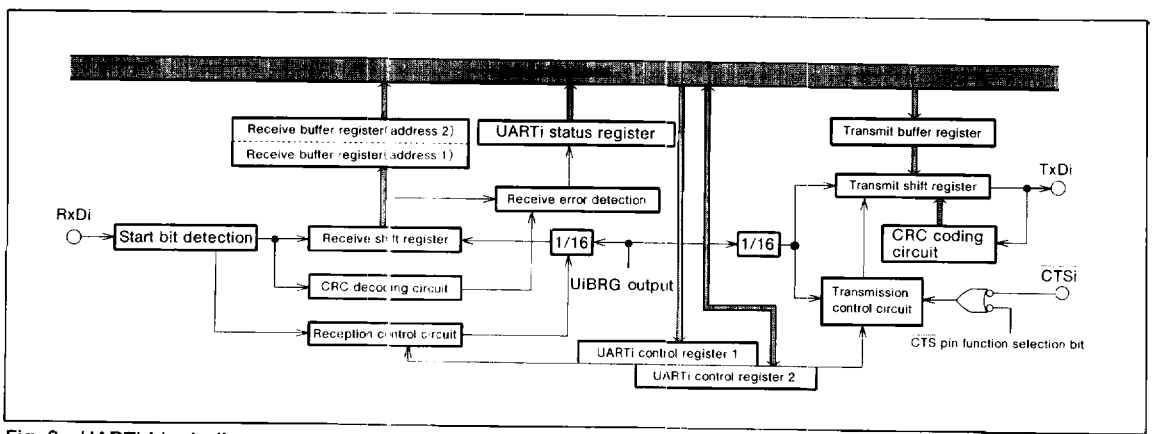


Fig. 9 UARTi block diagram

[Transmit operation]

When data (to be transmitted) is written into the transmit buffer register after writing "1" in the transmit initialize bit ($\overline{MR}$), the data to which the start bit (or parity bit depending on transmission condition setting) and the stop bit are appended is transferred to the transmit shift register. The transmit shift register begins shifting when it becomes transmit enable status (refer to Table 2) and transmits serial data from the TxDi pin.

In the transmit enable status, each time transmission of the stop bit of the serial data being transmitted has been completed, it is checked whether the next data has been written to the transmit buffer register. If next data is found written, transmission of that data begins. If next data is found not written, TxDi pin is held at "H" level until the next data is written, setting the transmitter empty flag (TEMP) to "1". When the transmit enable status is reset during transmission, the transmission is stopped after completing the transmission of the data already written into the transmit buffer register.

When the transmit initialize bit ($\overline{MR}$) is "0", transmitter side is in the initial status (Note 3). At this time, the transmitter ready flag (TxRDY) is "1" and the transmitter empty flag (TEMP) is "1", and the transmit shift register is at a stop.

Note 1 : When the transmitter ready flag (TxRDY) is "1", data can be written into the transmit buffer register. After data is written into the transmit buffer register, the transmitter ready flag (TxRDY) becomes "0".

Note 2 : Each time data is transferred from the transmit buffer register to the transmit shift register and the start bit is output from the TxDi pin, the transmitter ready flag (TxRDY) becomes "1" and the UARTi transmission interrupt request bit is set to "1". An interrupt is honored when the UARTi transmission interrupt enable bit is "1" and the interrupt disable flag (I) is "0".

Note 3 : When the transmit initialize bit ($\overline{MR}$) is cleared to "0" during transmission, the transmission breaks.

Table 2. Conditions of transmit enable status

TE	CTSE	$\overline{CTS}$ pin input level	Transmit enable status
1	0	X	enabled
	1	"H"	disabled
		"L"	enabled
0	X	X	disabled

TE : Transmit enable bit
CTSE : $\overline{CTS}$ pin function selection bit

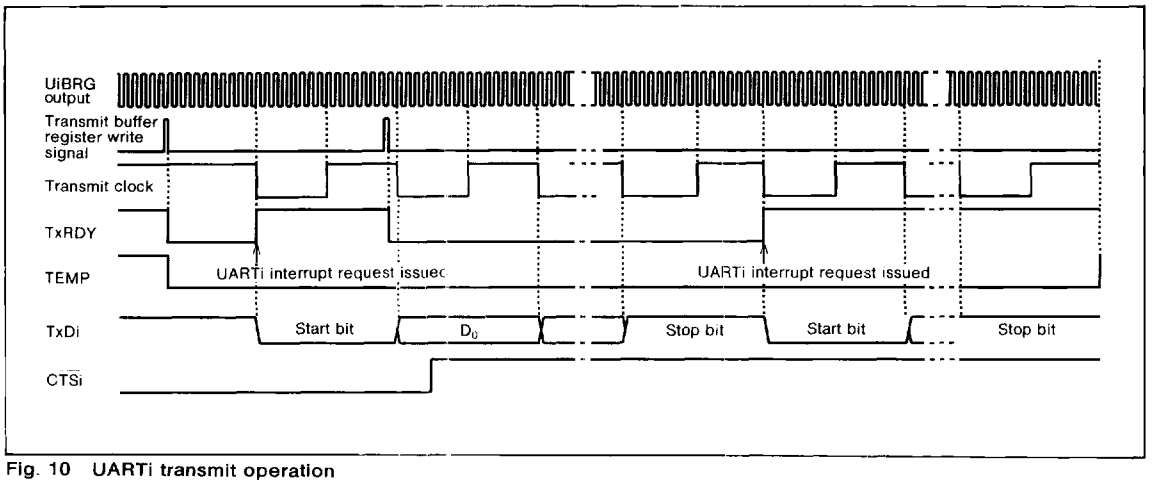


Fig. 10 UARTi transmit operation

[Transmit Operation Using CRC Method]

During transmission using the CRC method, the first block data is transmitted and then BCC is transmitted.

When the transmission of the block data is started, the CRC coding circuit (Note 1) is initialized by writing "0" in the transmit initialize bit (MR), and the first data of the block data is written into the transmit buffer register after the transmit initialize bit (MR) is reset to "1". However, if the previous transmitted data is BCC, the CRC coding circuit is automatically initialized.

The transmission of each data of block data is performed in the same way as ordinary transmission (without using the CRC method).

By writing "1" in the BCC transmission start bit (BST) after the last data of the block data is written into the transmit buffer register, the BCC is transmitted after the start bit (or parity bit depending on transmission condition setting) and stop bit are appended to the 8-bit CRC remainder that is held internally (Note 2).

At this time, it is not necessary to wait until the transmitter ready flag (TxRDY) is set to "1".

During transmission using the CRC method, when BCC has been transmitted, the transmitter empty flag (TEMP) becomes "1". The data to be transmitted following BCC is written into the transmit buffer register after the transmitter empty flag (TEMP) becomes "1".

If data is written into the transmit buffer register during the period from the writing to "1" in the BCC transmission start bit (BST) until the completion of BCC transmission, the written data is invalid and the transmission stops at the completion of BCC transmission.

The receiver side must initialize the CRC decoding circuit by software between the reception of BCC and the reception of next data. Therefore, the transmitter side must exert control to put a necessary interval between the transmission of BCC and the transmission of next data by software.

The CRC coding circuit is initialized by resetting.

Note 1 : The CRC coding circuit performs division with the generating polynomial for data (to be transmitted) as well as transmission and always holds its remainder in the internal register.

Note 2 : The BCC value to be transmitted cannot be read out.

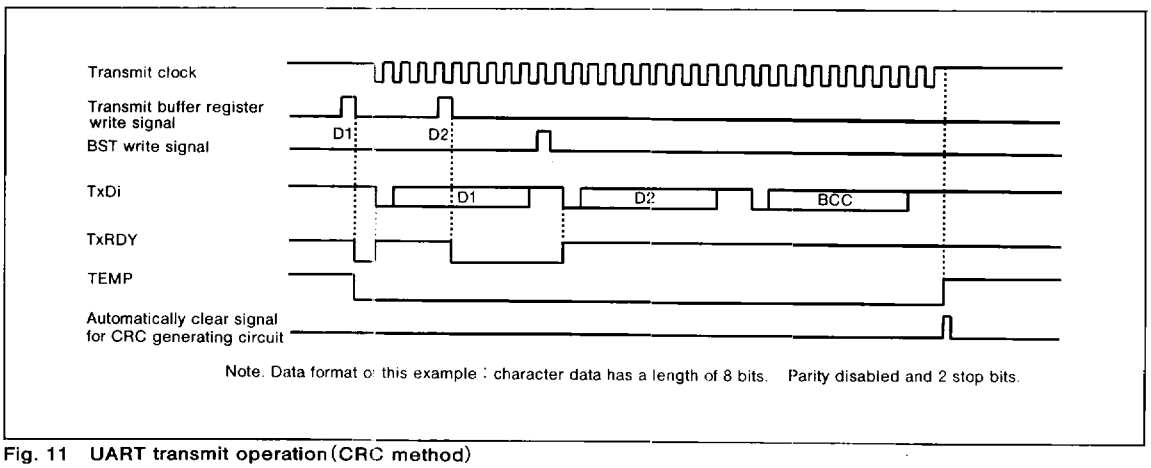


Fig. 11 UART transmit operation (CRC method)

[Receive Operation]

When "1" is written into the receive enable bit (RE), the 3888 group is put into the receive enable status (Note 1).
When a start bit is detected (Note 2), the data bits (and also the parity bit depending on reception condition setting) are taken into the receive shift register. When a stop bit is detected, the reception of 1-byte data is completed (Note 3), and then the data is transferred to the receive buffer register.
When the receive enable bit (RE) is cleared to "0", the 3888 group is put into the receive disable status. At this time, the receive buffer empty flag (RBE) is "1" (empty status), the receive buffer full flag (RBF) is "0", the receive shift register is at a stop, and start bit detection is stopped.

Receive Buffer Register

The receive buffer register is a FIFO type 2-byte (address 1 and address 2) register. A write/read operation is performed for the address specified by the write address pointer/read address pointer. (For resetting, the write address pointer/read address pointer points to address 1.)
Each time received data is written into the receive buffer register, the write address pointer is reversed. Accordingly, 2-byte data can be written. Each time the receive buffer register is read, the read address pointer is reversed. Accordingly, 2-byte data can be read out by reading the contents of the receive buffer register (UiRB) twice.

Table 3. Relationship between status flags and status of receive buffer register

RBE	RBF	Number of left data(before writing)in receive buffer register
1	0	0
0	0	1
0	1	2

(Unit : byte)

RBE : Receive buffer empty flag
RBF : Receive buffer full flag

Error during Data Reception

When an error occurs during data reception, the corresponding error flag is set to "1" but has no effect on the receive operation. Every error flag can be reset by writing "1" into the error reset bit (ERST).

- Overrun error (Flag OE)
  When the next data has been received while there remains 2-byte data to be read in the receive buffer register, an overrun error occurs. To initialize the receive buffer register, clear the overrun error flag (OE) to "0" and also write "1" into the receive buffer clear bit (RBC).
- Parity error (Flag PE)
  This error occurs when a parity error is detected in received data.
- Framing error (Flag FE)
  This error occurs when the first stop bit is detected as "L".

Interrupt Source Selection

When the receive interrupt source selection bit (RIS) is "0", the UARTi receive interrupt request bit is set to "1" each time data is put into the empty receive buffer register. Accordingly, the receive interrupt request can occur upon reception of each 1-byte data.
When the receive interrupt source selection bit (RIS) is "1", the UARTi receive interrupt request bit is set to "1" each time 2-byte data is arranged in the receive buffer register. Accordingly, the frequency of occurrence of receive interrupt can be reduced.

Note 1 : When no data is received, "H" is input to the RxDi pin.

Note 2 : When a falling edge is input to the RxDi pin, sampling is performed with a frequency clock that is 16 times as large as the baud rate. At this time, if an "L" input is detected 2 times continuously, the start bit is detected. Furthermore, sampling is performed 3 times, approximately in the middle of the "Start bit". (Refer to Figure 12.) At this time, if "L" is detected twice or more, data reception is started and data bits are taken into the receive shift register. If "L" is not detected twice or more, the processing is returned to start bit detection.

Note 3 : To judge the value of data bits, parity bit or stop bit, sampling is performed 3 times each in the middle of each bit period. When "L" or "H" is detected twice or more, each is judged as "0" or "1". (Refer to Figure 12.)

Note 4 : When "1" is written into the receive buffer clear bit, both of the write address pointer and the read address pointer are cleared to address 1.

[Receive Operation Using CRC Method]

At data reception using the CRC method, the completion of the block data and BCC reception is detected by software. Whether or not any error exists in the received data is detected by the contents of the CRC status flag (CRCSTS).

At a start of the block data reception, the CRC decoding cricuit (Note 1) is cleared by writing "1" into the receive buffer clear bit (RBC). At that time, the CRC status flag (CRCSTS) becomes "1" and the receive buffer register and receive status flags (RBE, RBF) are initialized.

When the CRC status flag (CRCSTS) (Note 3) is "0" after completion of BCC reception (Note 2) following the block data reception, this is regarded as no error. Otherwise, it is regarded as an error.

The CRC decoding circuit is cleared by resetting.

Note 1 : The CRC decoding circuit performs division on the received data with the same generating polynomial as that on the transmitter side, and transfers the result to the CRC status flag (CRCSTS) upon receipt of each stop bit.

Note 2 : Received BCC is held in the receive buffer register like other received data. Accordingly, discard BCC by software. (Perform a dummy read operation or write "1" into the receive buffer clear bit (RBC) to clear the receive buffer register.)

Note 3 : The CRC status flag is updated upon receipt of each stop bit. Accordingly read the CRC status flag in the period from receipt of BBC until receipt of the stop bit of the next data.

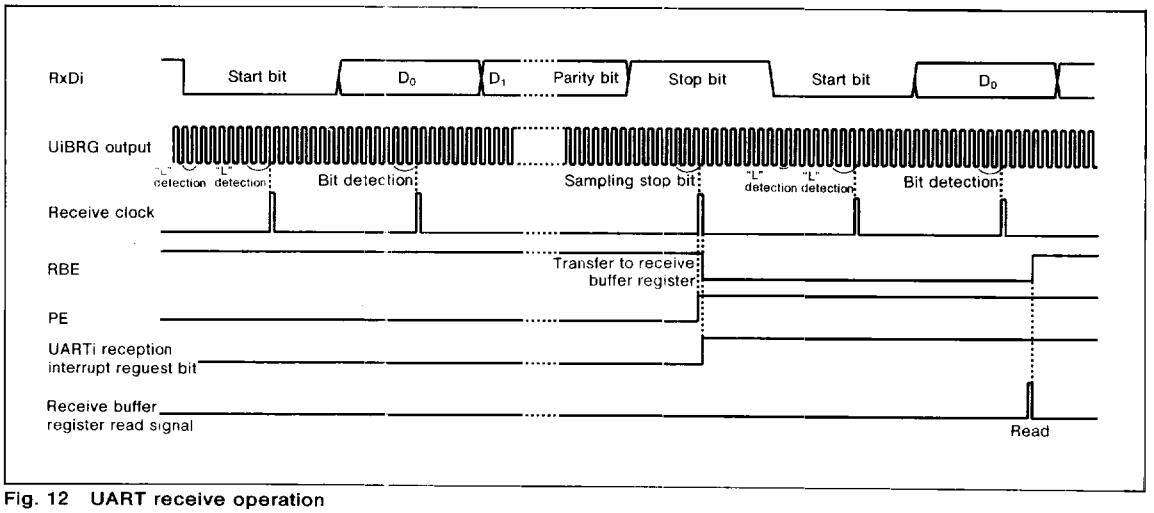


Fig. 12 UART receive operation

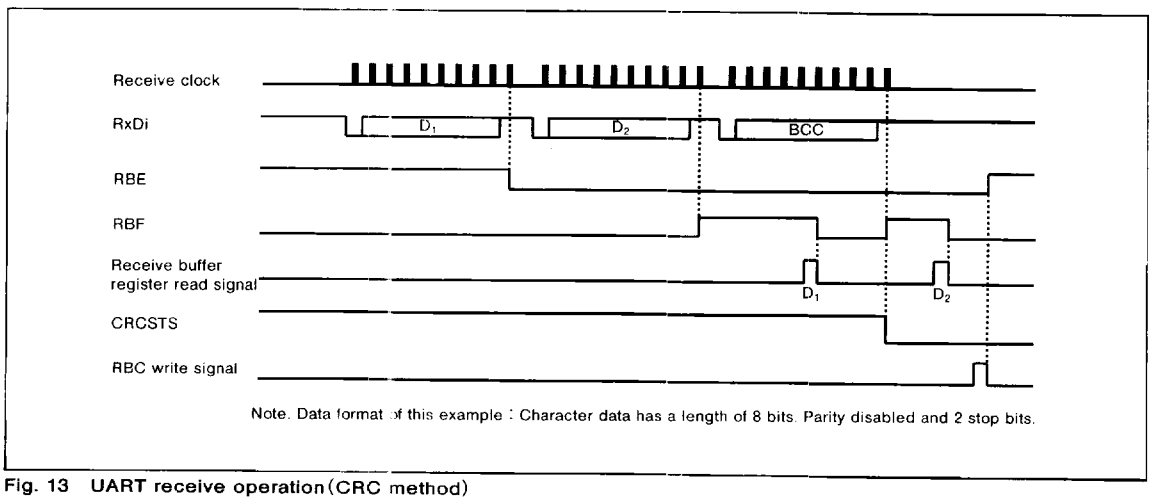


Fig. 13 UART receive operation (CRC method)

$\overline{\text{CTS}}$i Pin

This is an input pin for transmission control of the UARTi and also used as the port P1. For transmission control, set the CTSi pin function selection bit to "1" so that the CTSi pin may be used as a transmission control input. By inputting the "L" level to the CTSi pin on the transmitter side when the data receiver side is put into the receive enable status, the transmission from the transmitter side is enabled, so that data is output from the TxDi pin. To set the transmit enable status, the transmit enable bit must be "1".

UARTi Divider for Baud Rate Generation (UiBRG)

This is an 8-bit programmable divider to generate baud rate for transmission or reception.

When the set value is N (0 to 255), the division ratio becomes 1/(N+1). A count source is selected from the following by the division ratio selection bit of BRG (ERi) and the BRG clock selection bit (EXi).

- X_{IN} clock
- X_{IN} clock divided by 32
- External clock input from the CLK pin

When X_{IN} is selected as a count source, do not set "00_{16}" in the UiBRG. Set 4.0MHz (duty : 40-60%) or less as an input clock frequency when selecting the external clock. The transfer rate of data (to be transmitted or received) is 1/16 of the UiBRG output. Before writing data into the divider for baud rate generation, clear the receive enable bit and the transmit initialize bit to "0".

Table 4. Baud rate

EX	BR	Baud rate (bps)
0	0	$\frac{f(X_{IN})}{16\times(N+1)}$
0	1	$\frac{f(X_{IN})}{512\times(N+1)}$
1	X	$\frac{f(CLK)}{16\times(N+1)}$

EX : BRG clock selection bit
BR : Division ratio selection bit of BRG

UARTi Transmit Buffer Register (UiTB)

The data written in the transmit buffer register is transferred to the transmit shift register when the transmit shift register becomes empty, and is transmitted out from there. When the transmit shift register is empty during transmission, the data to be transmitted next can be written into the transmit buffer register.

UARTi Receive Buffer Register (UiRB)

The receive buffer register is a FIFO type 2-byte register. Reading can be held up to the time when 3-byte data has been received. This register can also read received data in the order of reception at any timing. However, when an overrun error occurs, already received data is lost. In this case, initialize the receive butter register.

UARTi Control Register 1 (UiCON1)

Only writing is enabled for the UARTi control register 1. Use the LDM or STA instruction to set a value in the UARTi control register. (If the CLB or SEB instruction is used, values other than the specified bit become undefined.)

UARTi Control Register 2 (UiCON2)

Both reading and writing are enabled for the UARTi control register 2. However, when reading the following bits, "0" is always read :

- BCC transmission start bit (BST)
- Receive buffer clear bit (RBC)
- Error flag reset selection bit (ERST)

UARTi Status Register (UiSTS)

Only reading is enabled for the UARTi status register.

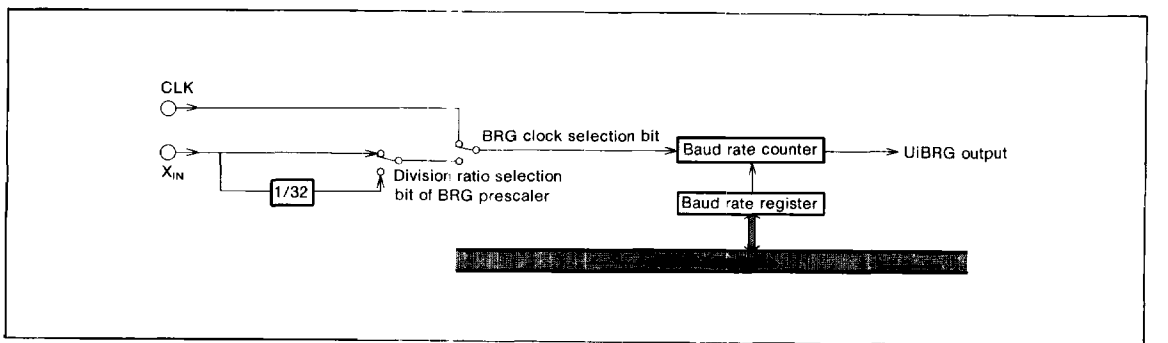


Fig. 14 BRG block diagram

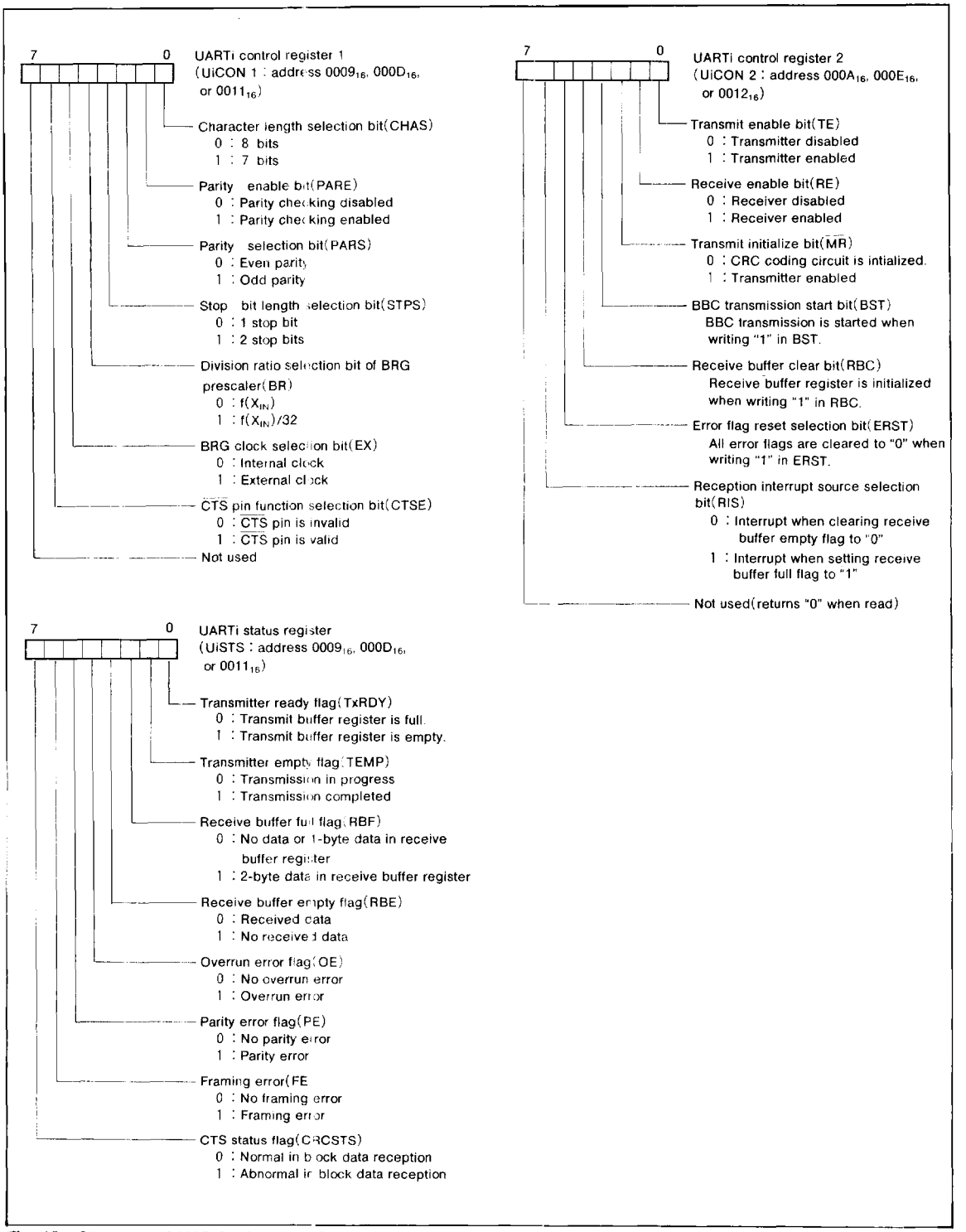


Fig. 15 Structure of UART-related registers

Bus Interface

The host CPU can access the dual port RAM and registers arranged in the address space of the system bus through the bus interface. The bus interface is provided with address pins A_0 to A_7, data pins DQ_0 to DQ_7, and 5 control pins $\overline{S}$, $\overline{R}$, $\overline{W}$, BM and INTR. These pins, except the BM pin, can be directly connected to the TTL (BM pin : The input levels are CMOS compatible). When "L" is input to the BM pin, the bus interface enters the RD/WR signal separate connection mode. When "H" is input to the BM pin, the bus interface enters the MITSUBISHI 16-bit microcomputer 7700 series connection mode.

- RD/WR signal separate connection mode
  - $\overline{W}$ Memory write pulse
  - R Memory read pulse
- 7700 connection mode
  - R/$\overline{W}$ Data bus direction specification
  - $\overline{E}$ Bus enable pulse

Fig. 17 shows the bus interface block diagram.

When "L" is input to the $\overline{S}$ pin, writing/reading from the system bus is enabled.

When "H" is input to the $\overline{S}$ pin, both writing and reading from the external are disabled. At this time, the output of the data pins DQ_0 to DQ_7 is floating.

Addressing is performed by the address pins A_0 to A_7.

When $\overline{W}$ pin is set to "L" in the RD/WR signal separate connection mode, the contents of the data pins D_0 to D_7 are written at the specified address. When the $\overline{R}$ pin is set to "L", the contents at the specified address are output to the data pins DQ_0 to DQ_7.

When the R/$\overline{W}$ pin is set to "L" and the $\overline{E}$ pin is set to "L" in the 7700 connection mode, the contents of data pins DQ_0 to DQ_7 are written at the specified address.

When the R/$\overline{W}$ pin is set to "H" and the $\overline{E}$ pin is set to "L", the contents of the specified address are output to the data pins DQ_0 to DQ_7.

Fig. 16 shows the arrangement of the DPRAM and registers corresponding to the system bus.

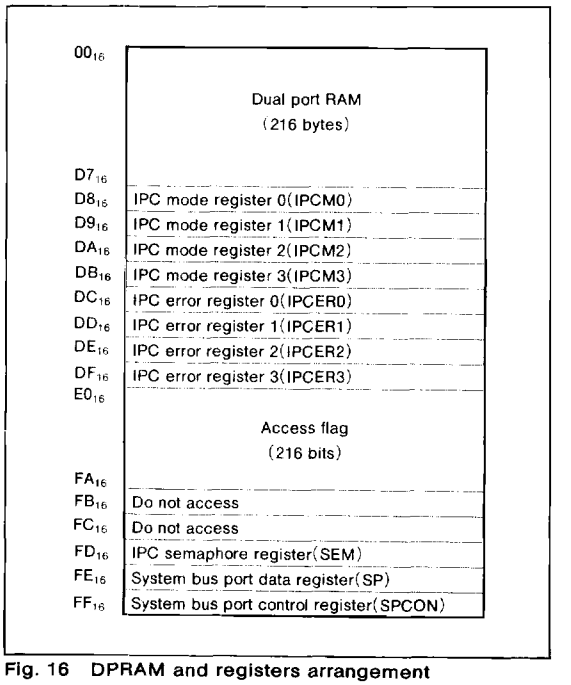


Fig. 16 DPRAM and registers arrangement (corresponding to the system bus)

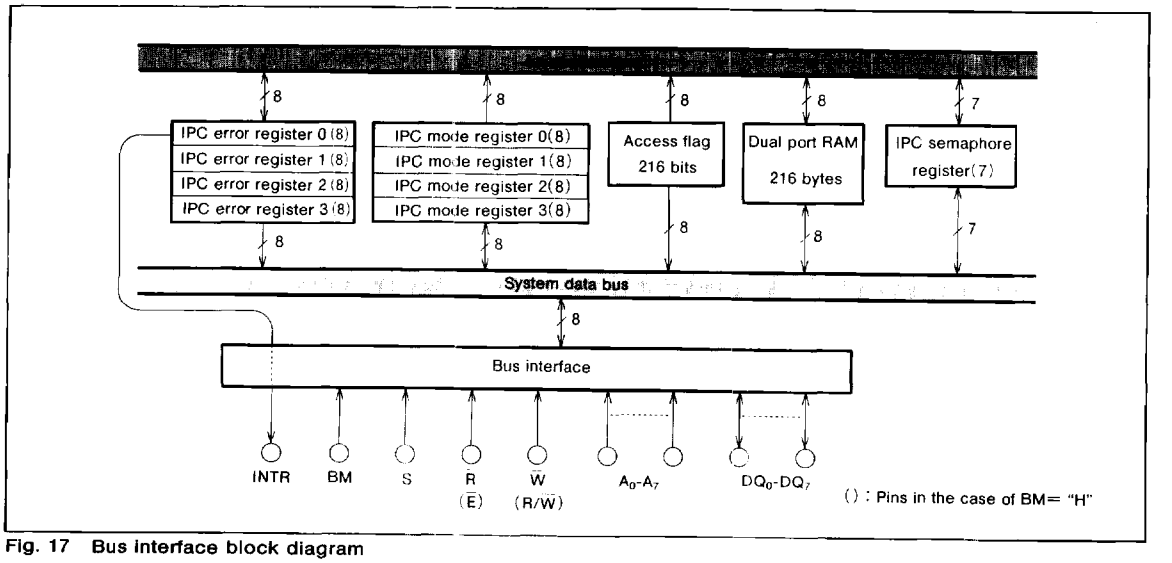


Fig. 17 Bus interface block diagram

Dual port RAM

The dual port RAM has a total capacity of 216 bytes and permits reading and writing at any timing from both buses (system data bus and local data bus). Dual port RAM is arranged at addresses 0500_{16} to $05D7_{16}$ for access from the local data bus, and at addresses 00_{16} to $D7_{16}$ for access from the system data bus. Fig. 18 shows the dual port RAM arrangement in the zero page area.

When access occurs simultaneously from both buses, the operation shown in Table 5 is performed by the arbitration circuit. When the dual port RAM is read or written from the system data bus, the effective addresses are system bus addresses at the falling time of R or W ($\overline{E}$ in the 7700 mode).

It is also possible to arrange a part of the dual port RAM at 0060_{16} to $007F_{16}$ in the zero page area on the local data bus side in units of 32-byte blocks. At this time, the zero page RAM and dual port RAM are overlapped at 0060_{16} to $007F_{16}$. However, it is possible to read or write specified blocks of the dual port RAM, while reading or writing of the zero page RAM is inhibited. Blocks to be arranged are specified by using the DPRAM map register.

The contents of DPRAM map register becomes "07_{16}" at reset.

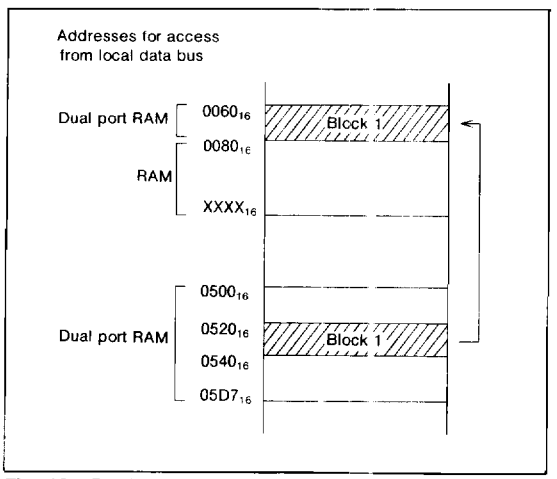


Fig. 18 Dual port RAM arrangement in zero page area

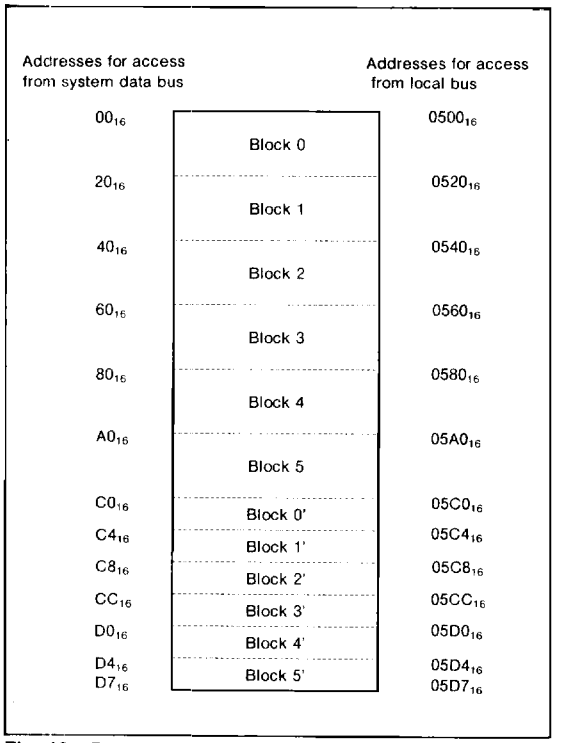


Fig. 19 Dual port RAM memory map

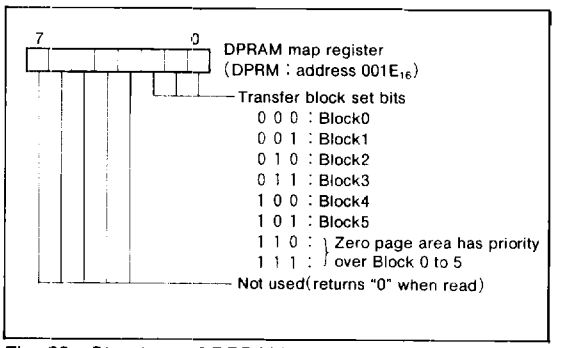


Fig. 20 Structure of DPRAM map register

Table 5. Data after accessing the same address from system and local data buses simultaneously

Bus operation		Data	
System data bus	Local data bus	System data bus side	Local data bus side
Read	Read	Correct data	Correct data
Read	Write	Correct data precedent/subsequent to writing	Data written
Write	Read	Data written	Correct data precedent/subsequent to writing
Write	Write	Data written into dual port RAM later	

MITSUBISHI ELECTRIC

Access Flag

This flag aims at arbitrating dual port RAM access. A 1-bit access flag is assigned to 1 byte of the dual port RAM.

All flags comprise 216 bits (27 bytes). Fig. 21 shows the correspondence between dual port RAM addresses and access flags, as viewed from the system data bus side.

The access flags can also be read from both system data bus and local data bus. However, writing is disabled.

When write access is made to the corresponding dual port RAM from the system data bus or local data bus, each bit is set to "1". When read access is made from the bus specified by the direction register (explained next), each bit is cleared to "0".

The access flag is cleared to "0" by resetting.

DPRAM Direction Register (DPRD)

This register specifies one of both buses from which each bit of the access flag is to be cleared by reading. As shown in Fig. 19 dual port RAM memory map, 1 bit of the direction register is associated with 1 block, and this register consists of a total of 6 bits.

When the corresponding DPRAM direction register is "0", each access flag is cleared by reading from the system data bus. When this register is "1", each access flag is cleared by reading from the local data bus. When writing from either of these buses, the access flag is set.

The DPRAM direction register is cleared by resetting.

Local bus address	System bus address	7	6	5	4	3	2	1	0
0040_{16}	$E0_{16}$	07	06	05	04	03	02	01	00
0041_{16}	$E1_{16}$	0F	0E	0D	0C	0B	0A	09	08
0042_{16}	$E2_{16}$	17	16	15	14	13	12	11	10
0043_{16}	$E3_{16}$	1F	1E	1D	1C	1B	1A	19	18
0044_{16}	$E4_{16}$	27	26	25	24	23	22	21	20
0045_{16}	$E5_{16}$	2F	2E	2D	2C	2B	2A	29	28
0046_{16}	$E6_{16}$	37	36	35	34	33	32	31	30
0047_{16}	$E7_{16}$	3F	3E	3D	3C	3B	3A	39	38
0048_{16}	$E8_{16}$	47	46	45	44	43	42	41	40
0049_{16}	$E9_{16}$	4F	4E	4D	4C	4B	4A	49	48
$004A_{16}$	EA_{16}	57	56	55	54	53	52	51	50
$004B_{16}$	EB_{16}	5F	5E	5D	5C	5B	5A	59	58
$004C_{16}$	EC_{16}	67	66	65	64	63	62	61	60
$004D_{16}$	ED_{16}	6F	6E	6D	6C	6B	6A	69	68
$004E_{16}$	EE_{16}	77	76	75	74	73	72	71	70
$004F_{16}$	EF_{16}	7F	7E	7D	7C	7B	7A	79	78
0050_{16}	$F0_{16}$	87	86	85	84	83	82	81	80
0051_{16}	$F1_{16}$	8F	8E	8D	8C	8B	8A	89	88
0052_{16}	$F2_{16}$	97	96	95	94	93	92	91	90
0053_{16}	$F3_{16}$	9F	9E	9D	9C	9B	9A	99	98
0054_{16}	$F4_{16}$	A7	A6	A5	A4	A3	A2	A1	A0
0055_{16}	$F5_{16}$	AF	AE	AD	AC	AB	AA	A9	A8
0056_{16}	$F6_{16}$	B7	B6	B5	B4	B3	B2	B1	B0
0057_{16}	$F7_{16}$	BF	BE	BD	BC	BB	BA	B9	B8
0058_{16}	$F8_{16}$	C7	C6	C5	C4	C3	C2	C1	C0
0059_{16}	$F9_{16}$	CF	CE	CD	CC	CB	CA	C9	C8
$005A_{16}$	FA_{16}	D7	D6	D5	D4	D3	D2	D1	D0

Fig. 21 Correspondence between dual port RAM addresses and access flags

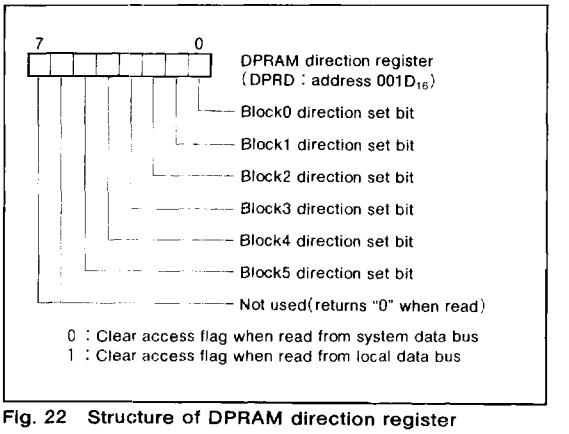


Fig. 22 Structure of DPRAM direction register (viewed from system data bus side)

Semaphore Register (semaphore flag : SEM)

This register is for handshaking of the host CPU and consists of a 6-bit block semaphore flag and a ready flag.

The block semaphore flag can be read and written from both system data bus and local data bus. Reading can be performed in units of byte but writing can be performed only in units of bit.

As shown in Table 6, a bit position to be written is specified in the 3 low-order bits of data, and value 1/0 to be written is specified in bit 7.

The ready flag can be read and written from the local data bus and can be read only from the system bus. Reading/writing is performed in the same way as the block semaphore flag.

When data is written into the IPC mode register 0 from the system data bus, it is cleared to "0".

All the bits of the semaphore register are cleared to "0" by resetting.

IPC Mode Register (IPCMi (i==0, 1, 2, 3))

This register consists of 4 bytes, being an 8-bit register that can be freely set by the user.

This register can be read and written from the system data bus, but can be read only from the local data bus.

This register is used to set a mode of the UART and others from the host CPU through the system data bus.

When data is written into the IPC mode register 0 from the system data bus, the ready flag of the semaphore register is cleared to "0" and also an IPC mode register 0 interrupt request occurs.

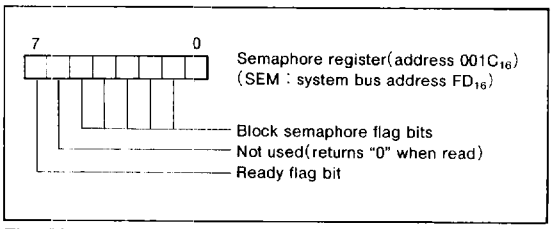


Fig. 23 Structure of semaphore register

Table 6. Example of writing to semaphore register

	1-byte data to be written into SEM	
Bit position for writing	To clear each bit to "0"	To set each bit to "1"
Bit 0	00_{16}	80_{16}
Bit 1	01_{16}	81_{16}
Bit 2	02_{16}	82_{16}
Bit 3	03_{16}	83_{16}
Bit 4	04_{16}	84_{16}
Bit 5	05_{16}	85_{16}
Bit 7 (Ready flag)	07_{16}	87_{16}

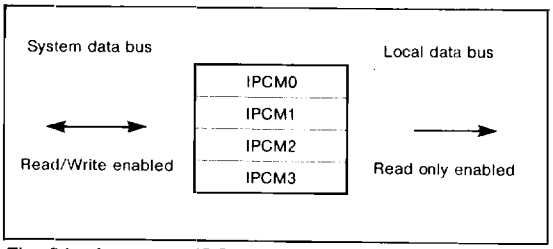


Fig. 24 Access to IPC mode register

IPC Error Register(IPCERi (i=0, 1, 2, 3))

This register consists of 4 bytes, being an 8-bit register that can be freely set by the user.

The register can be read only from the system data bus, but can be read and written from the local data bus.

When access is made by reading the contents of IPC error registers 0-3 from the system data bus, only the bits from which "1" is read are cleared to "0" by hardware. When reading from the local data bus, no value change occurs. All the bits of the IPC error registers 0-3 are cleared to "0".

When data is written into the IPC error register 0 from the local data bus, the INTR pin becomes "H". When the IPC error register 0 is read from the system data bus, the INTR pin becomes "L" and also an IPC error register 0 interrupt request occurs. The INTR pin becomes "L" by resetting.

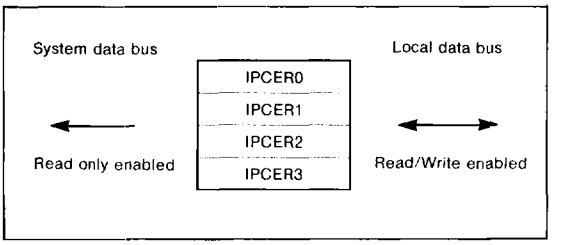


Fig. 25 Access to IPC error register

Caution

The IPC error register is designed on the assumption that access is not made simultaneously from the local data bus and system data bus. Accordingly, in reading this register from the system data bus while writing is performed into the IPC error register from the local data bus, take the following into consideration.

For writing from the local data bus, use the SEB or CLB instruction. If another instruction such as the LDM instruction is used, the value read from the system data bus is undefined. The value read from the system data bus is the data precedent to writing from the local data bus or the data subsequent to writing. In any case, a correct value is read. However, judge by software whether it is the data precedent to writing or the data subsequent to writing.

When access is made by a read modify write instruction such as the SEB instruction from the local data bus and the same register is read from the system data bus between the read cycle and the write cycle, the bit being "1" is cleared to "0" regardless of the value written from the local data bus. The reason is that the 3888 CPU holds the clear signal of hardware until the next instruction is fetched. The value read from the system data bus is one precedent to writing in the write cycle of the local data bus.

RESET CIRCUIT

To reset the 3888 group, $\overline{\text{RESET}}$ pin should be held at a "L" level for 2μs or more. Then $\overline{\text{RESET}}$ pin is returned to an "H" level (the power source voltage should be between 4.5V and 5.5V, and X_{IN} oscillation width is stable), reset is released. Internal operation begins after 8 to 13 X_{IN} clock cycles are completed. After the reset is completed, the program starts from the address contained in address $FFFD_{16}$ (high-order) and address $FFFC_{16}$ (low-order).

Make sure that the reset input voltage is no more than 0.9V for a power source voltage of 4.5V.

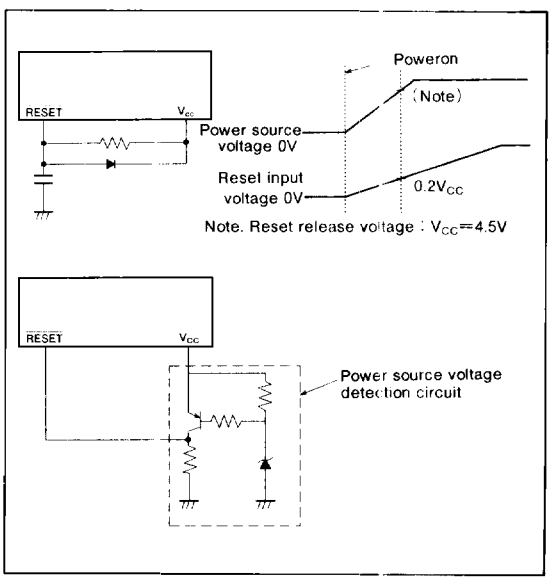


Fig. 26 Reset circuit

	Registers connected to local data bus	Local data bus addresses	Register contents
(1)	Port P0 direction register	$(0001_{16})\cdots$	00_{16}
(2)	Port P1 direction register	$(0007_{16})\cdots$	– 0 0 0 0 0 0 0
(3)	UART1 status register	$(0009_{16})\cdots$	1 0 0 0 1 0 1 1
(4)	UART1 control register 1	$(0009_{16})\cdots$	– 0 0 0 0 0 0 0
(5)	UART1 control register 2	$(000A_{16})\cdots$	00_{16}
(6)	UART2 status register	$(000D_{16})\cdots$	1 0 0 0 1 0 1 1
(7)	UART2 control register 1	$(000D_{16})\cdots$	– 0 0 0 0 0 0 0
(8)	UART2 control register 2	$(000E_{16})\cdots$	00_{16}
(9)	UART3 status register	$(0011_{16})\cdots$	1 0 0 0 1 0 1 1
(10)	UART3 control register 1	$(0011_{16})\cdots$	– 0 0 0 0 0 0 0
(11)	UART3 control register 2	$(0012_{16})\cdots$	00_{16}
(12)	IPC error register 0	$(0018_{16})\cdots$	00_{16}
(13)	IPC error register 1	$(0019_{16})\cdots$	00_{16}
(14)	IPC error register 2	$(001A_{16})\cdots$	00_{16}
(15)	IPC error register 3	$(001B_{16})\cdots$	00_{16}
(16)	Semaphore register	$(001C_{16})\cdots$	0 – 0 0 0 0 0 0
(17)	DPRAM direction register	$(001D_{16})\cdots$	– – 0 0 0 0 0 0
(18)	DPRAM map register	$(001E_{16})\cdots$	– – – – – 1 1 1
(19)	Prescaler 123	$(0030_{16})\cdots$	FF_{16}
(20)	Timer 1	$(0031_{16})\cdots$	01_{16}
(21)	Timer 2	$(0032_{16})\cdots$	FF_{16}
(22)	Timer 3	$(0033_{16})\cdots$	FF_{16}
(23)	Prescaler X	$(0034_{16})\cdots$	FF_{16}
(24)	Timer X	$(0035_{16})\cdots$	FF_{16}
(25)	Local bus port pull-up register	$(0039_{16})\cdots$	– – – – – – 0 0
(26)	CPU mode register	$(003B_{16})\cdots$	– – – – – 0 * 0
(27)	Interrupt request register 1	$(003C_{16})\cdots$	00_{16}
(28)	Interrupt request register 2	$(003D_{16})\cdots$	00_{16}
(29)	Interrupt control register 1	$(003E_{16})\cdots$	00_{16}
(30)	Interrupt control register 2	$(003F_{16})\cdots$	00_{16}
(31)	Access flag	$(0040_{16}\text{-}005A_{16})\cdots$	00_{16}
(32)	Processor status register	$(PS)\cdots$	X X X X X 1 X X
(33)	Program counter	$(PC_H)\cdots$	Contents of address $FFFD_{16}$
		$(PC_L)\cdots$	Contents of address $FFFC_{16}$
	Registers connected to system data bus	System bus addresses	Register contents
(34)	IPC error register 0	$(DC_{16})\cdots$	00_{16}
(35)	IPC error register 1	$(DD_{16})\cdots$	00_{16}
(36)	IPC error register 2	$(DE_{16})\cdots$	00_{16}
(37)	IPC error register 3	$(DF_{16})\cdots$	00_{16}
(38)	Access flag	$(E0_{16}\text{-}FA_{16})\cdots$	00_{16}
(39)	Semaphore register	$(FD_{16})\cdots$	0 – 0 0 0 0 0 0
(40)	Port PA direction register	$(FE_{16})\cdots$	00_{16}
(41)	Port PB direction register	$(FE_{16})\cdots$	00_{16}
(42)	System bus port control register	$(FF_{16})\cdots$	– – 0 0 0 0 0 0

Note. – : Not used
X : Undefined
* : The initial value of this bit is determined by the level at the CNV_{SS} pin. In the 3888 group, the CNV_{SS} pin is connected to V_{SS}. Thus this bit is set to "0" after a reset.
The contents of all other registers and RAM are undefined after a reset, so they must be initialized by software.

Fig. 27 Internal status of microcomputer after reset

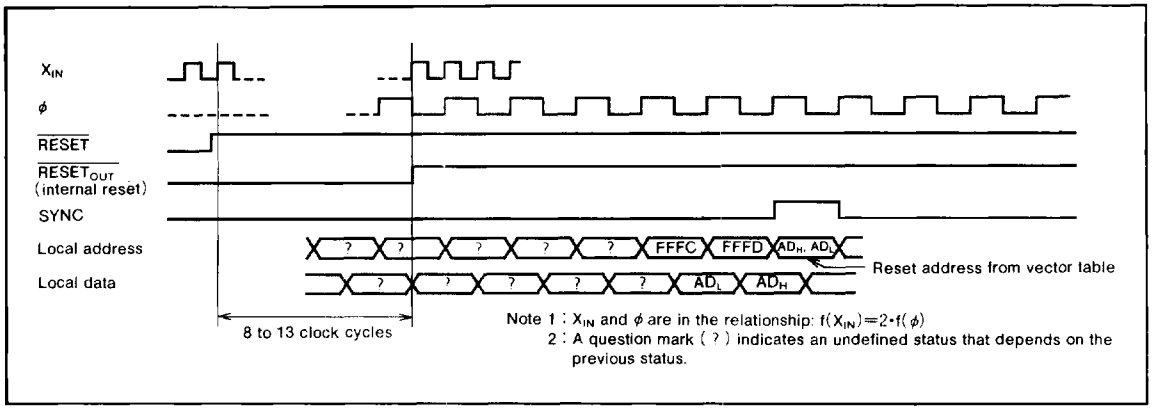


Fig. 28 Timing of reset

CLOCK GENERATING CIRCUIT

An oscillation circuit can be created by connecting a resonator between X_{IN} and X_{OUT}. Use the circuit constants in accordance with the resonator manufacture's recommended value. When using an external clock signal, input the clock signal to the X_{IN} pin and leave the X_{OUT} pin open.

Oscillation Control

(1) Stop Mode

When the STP instruction is executed, oscillation stops with the internal clock ϕ at "H". Timer 1 is set to "01_{16}" and prescaler 123 is set to "FF_{16}".

Oscillation restarts when an external interrupt is honored, but the internal clock ϕ remains at "H" until timer 1 underflows.

This allows time for the clock circuit oscillation to stabilize. If oscillation is restarted by a reset, no wait time is generated, so keep the RESET pin at "L" level until oscillation has stabilized.

(2) Wait Mode

When the WIT instruction is executed, the internal clock ϕ stops at a "H" level, but the oscillator itself does not stop. The internal clock restarts if a reset occurs or when an interrupt is honored.

Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

To ensure that interrupts will be honored, to release the STP or WIT state, interrupt enable bits must be set to "1" before the STP or WIT instruction is executed.

When the STP status is released, timer 1 interrupt request occurs simulaneously with the corresponding interrupt request occurrence, so set the timer 1 interrupt enable bit to "0" before the STP instruction is executed.

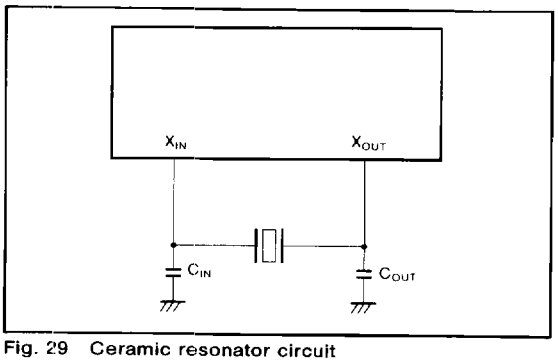


Fig. 29 Ceramic resonator circuit

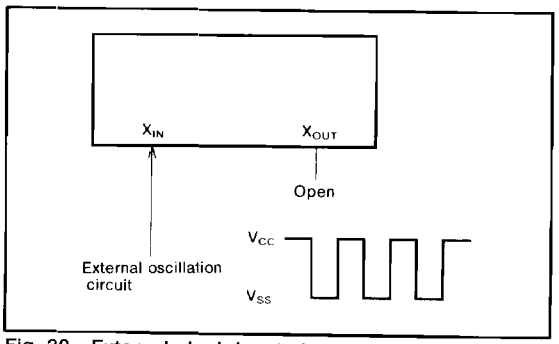


Fig. 30 External clock input circuit

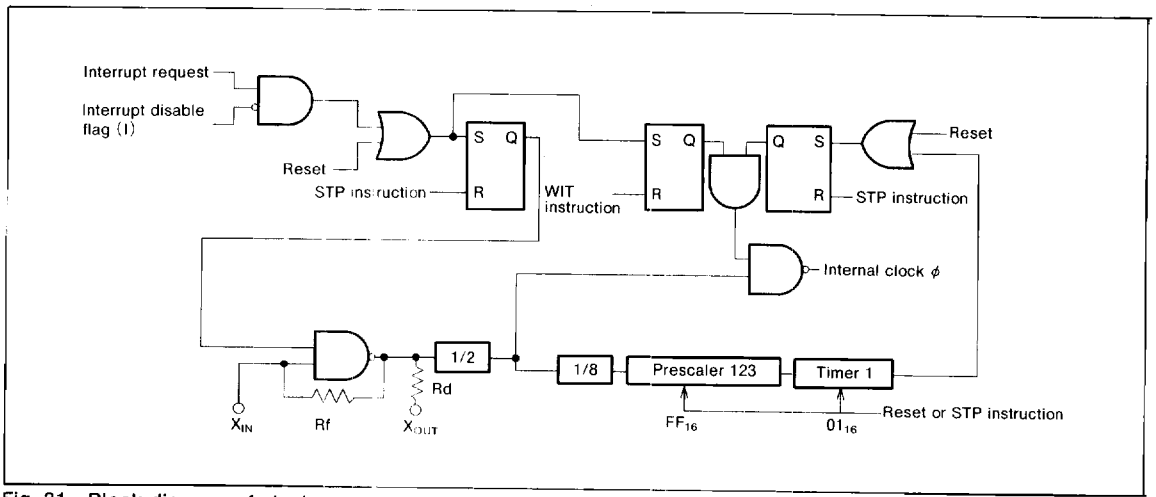


Fig. 31 Block diagram of clock generating circuit

MITSUBISHI ELECTRIC

NOTES ON PROGRAMMING

Processor Status Register

The contents of the processor status register (PS) after a reset are undefined, except for the interrupt disable flag (I) which is "1". After a reset, initialize flags which affect program execution.

In particular, it is essential to initialize the index X mode (T) and the decimal operation mode (D) flag because of their effect on calculations.

Interrupts

The contents of the interrupt request bits are not modified immediately after they have been written. After writing to an interrupt request register, execute at least one instruction before executing a BBC or BBS instruction.

Decimal Calculations

To calculate in decimal notation, set the decimal mode flag (D) to "1", then execute an ADC or SBC instruction. Only the ADC and SBC instruction yield proper decimal results. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.

In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flag are invalid.

The carry flag can be used to indicate whether a carry or borrow has occurred. Initialize the carry flag before each calculation. Clear the carry flag before an ADC and set the flag before an SBC.

Timers

If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is $1/(n+1)$.

Multiplication and Division Instructions

The index X mode (T) and the decimal mode (D) flag do not affect the MUL and DIV instruction.

The execution of these instructions does not modify the contents of the processor status register.

Ports

The contents of the port direction registers cannot be read.

The following cannot be used :

- the data transfer instruction (LDA, etc.)
- the operation instruction when the index X mode flag (T) is "1".
- the addressing mode which uses the value of a direction register as an index
- the bit-test instruction (BBC or BBS, etc.) to a direction register
- the read-modify-write instruction (ROR, CLB, or SEB, etc.) to a direction register

Use instructions such as LDM and STA, etc., to set the port direction registers.

Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock ϕ by the number of cycles needed to execute an instruction.

The number of cycles required to execute an instruction is shown in the list of machine instructions.

The frequency of the internal clock ϕ is half of the X_{IN} frequency.

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

1. Mask ROM Order Confirmation Form
2. Mark Specification Form
3. Data to be written to ROM, in EPROM form (three identical copies)

PROM PROGRAMMING METHOD

The built-in PROM of the blank One Time PROM version and built-in EPROM version can be read or programmed with a general-purpose PROM programmer using a special programming adapter.

Set the address of PROM programmer in the user ROM area.

Package	Name of Programming Adapter
64P4B, 64S1B	PCA4738S-64A
64P6N-A	PCA4738F-64A
64P6S-A	PCA4738G-64
64D0	PCA4738L-64A

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 32 is recommended to verify programming.

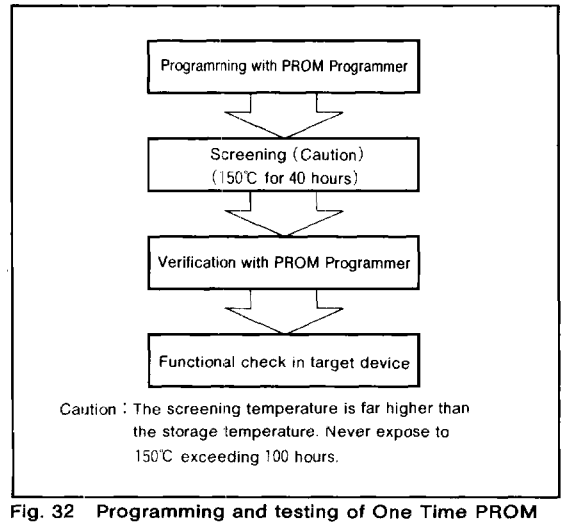


Fig. 32 Programming and testing of One Time PROM version

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages are based on V_{SS}. Output transistors are cut off.	−0.3 to 7.0	V
V_I	Input voltage $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7, A_0-A_7, DQ_0-DQ_7, R, W, S, BM, RxD_1-RxD_3		−0.3 to V_{CC}+0.3	V
V_I	Input voltage RESET, X_{IN}		−0.3 to V_{CC}+0.3	V
V_I	Input voltage CNV_{SS}		−0.3 to 13	V
V_O	Output voltage $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7, DQ_0-DQ_7, TxD_1-TxD_3, X_{OUT}, INTR		−0.3 to V_{CC}+0.3	V
Pd	Power dissipation	DIP (64P4B)	500(1200)(Note)	mW
		QFP (64P6N-A)	300(750)(Note)	mW
		QFP (64P6S-A)	280(700)(Note)	mW
Topr	Operating temperature		−20 to 85	°C
Tstg	Storage temperature		−40 to 125	°C

Note : Parenthesized values are valid when Ta=25°C. $Pd=V_{CC}\times I_{CC}+\Sigma[(-I_{OH})\times(V_{CC}-V_{OH})]+\Sigma(I_{OL}\times V_{OL})$.

RECOMMENDED OPERATING CONDITIONS (V_{CC} = 4.5 to 5.5V, T_a = −20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits Min.	Limits Typ.	Limits Max.	Unit
V_{CC}	Power source voltage	4.5	5.0	5.5	V
V_{SS}	Power source voltage		0		V
V_{IH}	"H" input voltage $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7, RxD_1-RxD_3, BM	0.8V_{CC}		V_{CC}	V
V_{IH}	"H" input voltage A_0-A_7, DQ_0-DQ_7, R, W, S	2.0		V_{CC}	V
V_{IH}	"H" input voltage RESET, X_{IN}, CNV_{SS}	0.8V_{CC}		V_{CC}	V
V_{IL}	"L" input voltage $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7, RxD_1-RxD_3, BM	0		0.2V_{CC}	V
V_{IL}	"L" input voltage A_0-A_7, DQ_0-DQ_7, R, W, S	0		0.8	V
V_{IL}	"L" input voltage RESET, CNV_{SS}	0		0.2V_{CC}	V
V_{IL}	"L" input voltage X_{IN}	0		0.2V_{CC}	V
$\Sigma I_{OH(peak)}$	"H" total peak output current $P0_0$-$P0_7$, PA_0-PA_7, PB_0-PB_7 (Note 1)			160	mA
$\Sigma I_{OH(peak)}$	"H" total peak output current $P1_0$-$P1_6$, TxD_1-TxD_3, INTR, DQ_0-DQ_7			80	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current $P0_0$-$P0_7$, PA_0-PA_7, PB_0-PB_7 (Note 1)			160	mA
$\Sigma I_{OL(peak)}$	"L" total peak output current $P1_0$-$P1_6$, TxD_1-TxD_3, INTR, DQ_0-DQ_7			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current $P0_0$-$P0_7$, PA_0-PA_7, PB_0-PB_7 (Note 1)			80	mA
$\Sigma I_{OH(avg)}$	"H" total average output current $P1_0$-$P1_6$, TxD_1-TxD_3, INTR, DQ_0-DQ_7			40	mA
$\Sigma I_{OL(avg)}$	"L" total average output current $P0_0$-$P0_7$, PA_0-PA_7, PB_0-PB_7 (Note 1)			80	mA
$\Sigma I_{OL(avg)}$	"L" total average output current $P1_0$-$P1_6$, TxD_1-TxD_3, INTR, DQ_0-DQ_7			40	mA
$I_{OH(peak)}$	"H" peak output current $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7 (Note 2)			−10	mA
$I_{OH(peak)}$	"H" peak output current TxD_1-TxD_3, INTR, DQ_0-DQ_7			−2	mA
$I_{OL(peak)}$	"L" peak output current $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7 (Note 2)			10	mA
$I_{OL(peak)}$	"L" peak output current TxD_1-TxD_3, INTR, DQ_0-DQ_7			3.2	mA
$I_{OH(avg)}$	"H" average output current $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7 (Note 3)			−5	mA
$I_{OH(avg)}$	"H" average output current TxD_1-TxD_3, INTR, DQ_0-DQ_7			−1	mA
$I_{OL(avg)}$	"L" average output current $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7 (Note 3)			5	mA
$I_{OL(avg)}$	"L" average output current TxD_1-TxD_3, INTR, DQ_0-DQ_7			1.6	mA
$f(X_{IN})$	Internal clock oscillation frequency			12.5	MHz

Note 1. The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100ms. The total peak current is the peak value of all the currents.
2. The peak output current is the peak current flowing in each port.
3. The average output current $I_{OL(avg)}$, $I_{OH(avg)}$ in an average value measured over 100ms.

ELECTRICAL CHARACTERISTICS (V_{CC} = 4.5 to 5.5V, V_{SS} = 0V, T_a = −20 to 85℃, unless otherwise noted)

Symbol	Parameter	Test conditions		Limits Min.	Limits Typ.	Limits Max.	Unit
V_{OH}	"H" output voltage $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7	I_{OH}=−10mA		V_{CC}−2			V
V_{OH}	"H" output voltage INTR, TxD_1-TxD_3	I_{OH}=−1mA		0.8V_{CC}			V
V_{OH}	"H" output voltage DQ_0-DQ_7	I_{OH}=−1mA		V_{CC}−1			V
V_{OL}	"L" output voltage $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7	I_{OL}=10mA				2	V
V_{OL}	"L" output voltage DQ_0-DQ_7, INTR, TxD_1-TxD_3	I_{OL}=1.6mA				0.4	V
V_{T+}-V_{T-}	Hysteresis RxD_1-RxD_3, CLK, CTS_1-CTS_3				0.5		V
V_{T+}-V_{T-}	Hysteresis RESET				0.5		V
I_{IH}	"H" input current A_0-A_7, R, W, S, RxD_1-RxD_3	V_I=V_{CC}				5	μA
I_{IH}	"H" input current $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7, DQ_0-DQ	V_I=V_{CC} (Pins are floating, Pull-up transistors are disconnected)				5	μA
I_{IH}	"H" input current RESET, CNV_{SS}	V_I=V_{CC}				5	μA
I_{IH}	"H" input current X_{IN}	V_I=V_{CC}			4		μA
I_{IL}	"L" input current A_0-A_7, R, W, S, RxD_1-RxD_3	V_I=V_{SS}				−5	μA
I_{IL}	"L" input current $P0_0$-$P0_7$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7, DQ_0-DQ_7	V_I=V_{SS} (Pins are floating, Pull-up transistors are disconnected)				−5	μA
I_{IL}	"L" input current RESET, CNV_{SS}	V_I=V_{SS}				−5	μA
I_{IL}	"L" input current X_{IN}	V_I=V_{SS}			−4		μA
I_{IL}	"L" input current $P0_0$-$P0_6$, $P1_0$-$P1_6$, PA_0-PA_7, PB_0-PB_7	V_I=V_{SS} (Pull-up transistors are connected)			−0.2		mA
V_{RAM}	RAM hold voltage	With clock stopped		2		5.5	V
I_{CC}	Power source current (Note)	f(X_{IN})=12.5MHz			9	16	mA
		When WIT instruction is executed with f(X_{IN})=12.5MHz			2	4	mA
		When STP instruction is executed with oscillation stopped	T_a=25℃		0.1	1	μA
			T_a=85℃			10	μA
I_{CC}	Power source current (increased by accessing from system data bus)	Access ferquency of system data bus (CL=100pF)	100kHz		0.4	0.8	mA
			1MHz		2	4	mA
			5MHz		9	18	mA

Note : The input level of each input pin is either V_{CC} or V_{SS}. Access to the dual port RAM from the system data bus is not performed.
The current flowing from the input or output pin in the pull-up status is not incuded.

TIMING REQUIREMENTS (V_{CC} = 4.5 to 5.5V, V_{SS} = 0V, T_a = −20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{W(RESET)}$	Reset input "L" pulse width	2			μs
$t_{C(X_{IN})}$	External clock input cycle time	80			ns
$t_{WH(X_{IN})}$	External clock input "H" pulse width	30			ns
$t_{WL(X_{IN})}$	External clock input "L" pulse width	30			ns
$t_{WH(INT)}$	INT input "H" pulse width	80			ns
$t_{WL(INT)}$	INT input "L" pulse width	80			ns
$t_{C(CLK)}$	UART external clock input cycle time	250			ns
$t_{WH(CLK)}$	UART external clock input "H" pulse width	100			ns
$t_{WL(CLK)}$	UART external clock input "L" pulse width	100			ns
$t_{r(CLK)}$	UART external clock input rising time			25	ns
$t_{f(CLK)}$	UART external clock input falling time			25	ns

TIMING REQUIREMENTS FOR SYSTEM DATA BUS IN RD/WR SEPARATE MODE

(V_{CC} = 4.5 to 5.5V, V_{SS} = 0V, T_a = −20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{SU(A-R)}$	Address set-up time	10			ns
$t_{SU(A-W)}$	Address set-up time	10			ns
$t_{SU(S-R)}$	Chip select set us time	0			ns
$t_{SU(S-W)}$	Chip select set us time	0			ns
$t_{h(R-A)}$	Address hold time	0			ns
$t_{h(W-A)}$	Address hold time	0			ns
$t_{h(R-S)}$	Chip select hold time	0			ns
$t_{h(W-S)}$	Chip select hold time	0			ns
$t_{SU(D-W)}$	Data input set up time	20			ns
$t_{h(W-D)}$	Data input hold time	0			ns
$t_{W(R)}$	Read pulse width	50			ns
$t_{W(W)}$	Write pulse width	75			ns
$t_{SU(R-R)}$	Read-Read interval	40			ns
$t_{SU(R-W)}$	Read-Write interval	40			ns
$t_{SU(W-R)}$	Write-Read interval	90			ns
$t_{SU(W-W)}$	Write-Write interval	90			ns

TIMING REQUIREMENTS FOR SYSTEM DATA BUS IN 7700 MODE

(V_{CC} = 4.5 to 5.5V, V_{SS} = 0V, T_a = −20 to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{SU(A-E)}$	Address set up time	10			ns
$t_{SU(RW-E)}$	RW set up time	10			ns
$t_{SU(S-E)}$	Chip select set up time	0			ns
$t_{h(E-A)}$	Address hold time	0			ns
$t_{h(E-RW)}$	RW hold time	10			ns
$t_{h(E-S)}$	Chip select hold time	0			ns
$t_{SU(D-E)}$	Data input set up time	20			ns
$t_{h(E-D)}$	Data input hold time	0			ns
$t_{W(E)}$	Enable pulse width	75			ns
$t_{SU(E-E)}$	Enable pulse interval after read	40			ns
$t_{SU(E-E)}$	Enable pulse interval after write	90			ns

SWITCHING CHARACTERISTICS FOR SYSTEM DATA BUS IN RD/WR SEPARATE MODE

(V_{CC} = 4.5 to 5.5V, V_{SS} = 0V, T_a = −20 to 85℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{d(S-Q)}$	Chip select access time			55	ns
$t_{d(R-Q)}$	Read access time			50	ns
$t_{u(R-Q)}$	Data valid time after read	0			ns
$t_{en(R-Q)}$	Output enable time after read	5			ns
$t_{dis(R-Q)}$	Output disable time after read	0		15	ns

SWITCHING CHARACTERISTICS FOR SYSTEM DATA BUS IN 7700 MODE

(V_{CC} = 4.5 to 5.5V, V_{SS} = 0V, T_a = −20 to 85℃, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
$t_{d(S-Q)}$	Chip select access time			55	ns
$t_{d(E-Q)}$	Read access time			50	ns
$t_{u(E-Q)}$	Data valid time after read	0			ns
$t_{en(E-Q)}$	Output enable time after read	5			ns
$t_{dis(E-Q)}$	Output disable time after read	0		15	ns

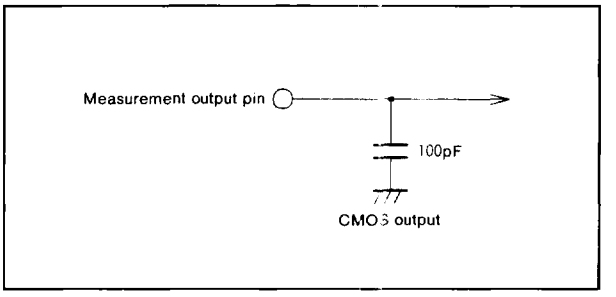
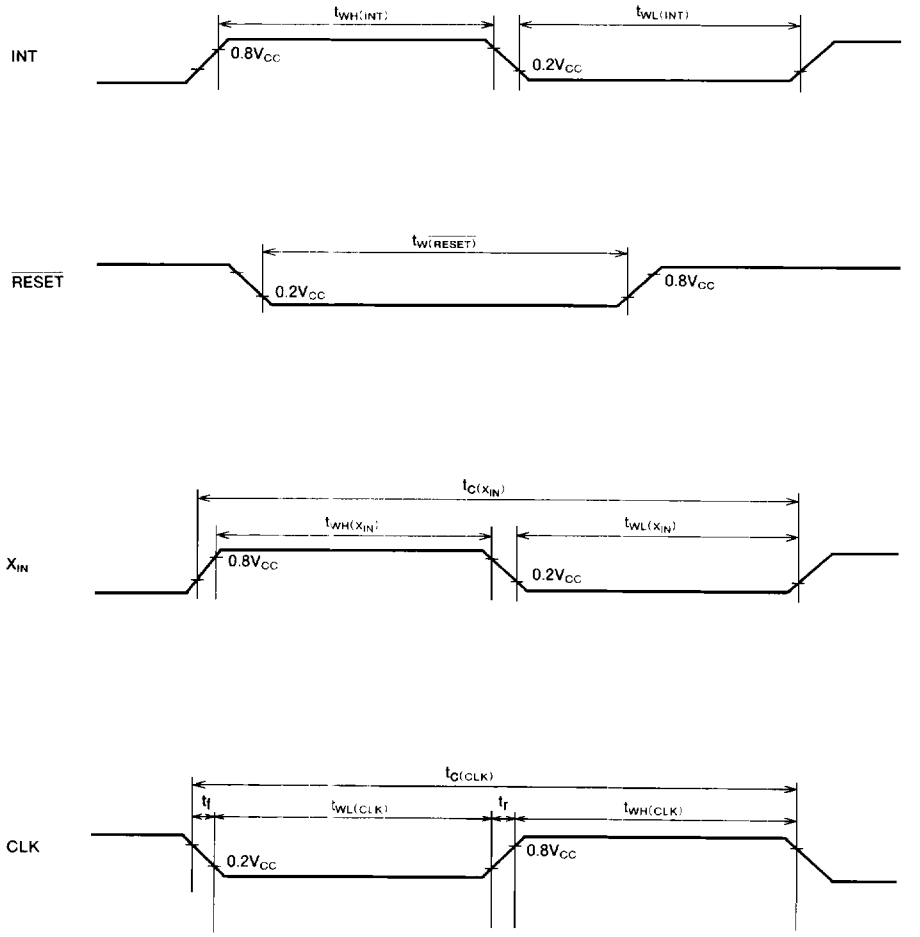


Fig. 33 Circuit for measuring output switching characteristics

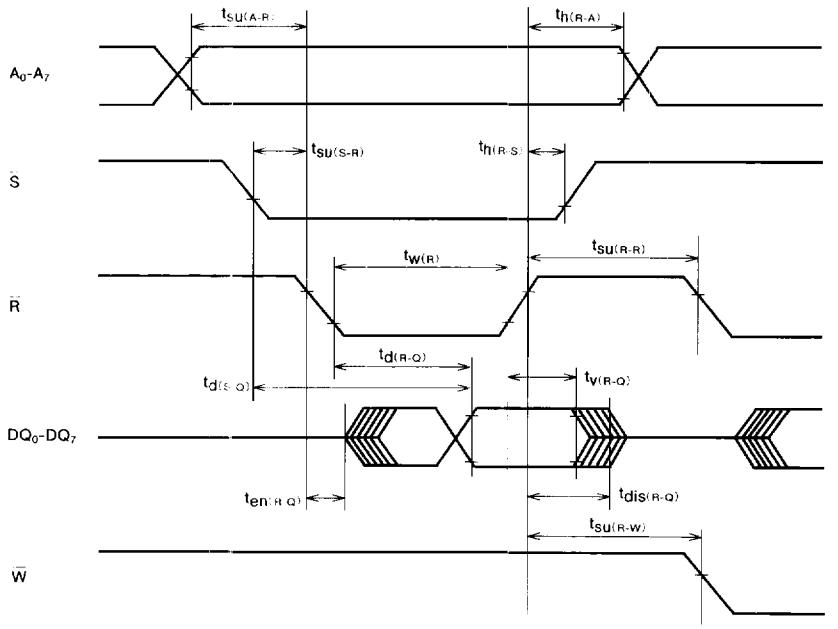
TIMING DIAGRAM

(1) Timing diagram

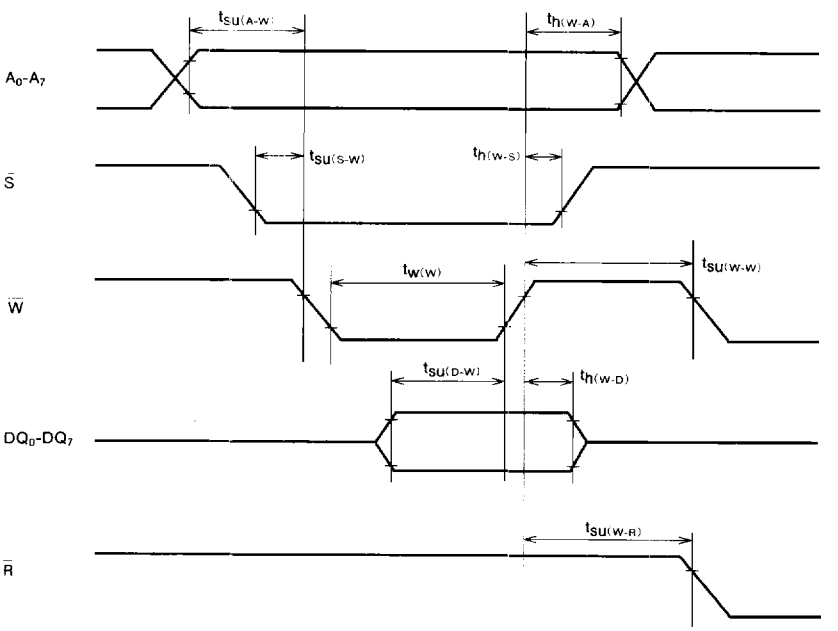


(2) Timing diagram in RD/WR separate mode

Read Cycle



Write Cycle



MITSUBISHI ELECTRIC

(3) Timing diagram in 7700 mode