YAMAHA LSI

YM3802

MCS (MIDI Communication & Service Controller)

APPLICATION MANUAL

YAMAHA

YM3802アプリケーションマニュアル
CATALOG No. : LSI-2438022
1987.9

TABLE OF CONTENTS

1. INTRODUCTION

1.1 General

The YM3523 is an LSI device featuring an asynchronous serial communication interface, a frequency divider that acts a communication rate generator, an interface for the cassette tape recorder, transmit/receive data buffers, timers, counters and a parallel input/output port. With this LSI a part of the MIDI data processing can be performed by hardware.

The YM3523 LSI has two output pins and three counters that synchronize with the MIDI clock and the tape SYNC can be easily realized. The MIDI clock is generated by the MIDI clock timer, the tape SYNC signal or the clock message contained in the received serial data. Another way of the generation of the MIDI clock is a process with the host CPU control. This LSI has the priority transmission and reception capability of the MIDI system real-time message over other messages and also can support the processing of the system exclusive message.

Each of the MIDI counters can be utilized as general-purpose timer/counter.

1.2 Features

- Serial communication
  7- or 8-bits ..... Character,
  1- or 2-bits ..... Stop bit,
  1- or 4-bits ..... Parity bit,
  Start bit error detection,
  Automatic break detection and break character generation,
  Character length, Types of parity and stop bits and communication rate are selectable for transmission and reception separately.

- Cassette tape recorder interface
  One wave/two wave FSK modulation,
  Automatic follow-up function for polarity and phase,
  Transfer-rate check function,
  Used for ordinary serial communication or cassette tape recorder interface in transmission and reception (separately).

- Communication rate generater
  Communication rate generation of 75 ~ 19200 bps and 31.25K bps.
  Reception is performed at the internal clock rate that is 16 times the communication rate.
  Communication rates of DC through 125K bps can be obtained by selecting the frequency of a communication rate generating clock.

- General-purpose 8-bit parallel input/output port.

- General-purpose 14-bits timer.

- 128 bytes FIFO buffer for reception and 16 bytes FIFO buffer for transmission.

- MIDI support functions
  SYNC out, CLOCK out; output of a pulse signal synchronized to the system real-time message.
  Automatic transmitting function, priority transmitting function and priority receiving function (without involving the receiving FIFO buffer) of the system real-time message.
  8- and 15-bit counters for counting the interpolated, high-accuracy signal of the MIDI clock.
  Special 14-bit timer for determining MIDI clock generation timing. Detecting function of the MIDI clock from the received serial data.
  Tape SYNC function.
  Automatic output of the tape SYNC signal
  Active sense function
  ID code check function for the system exclusive message.

- Vector output function according to 8-level interrupt factors
- C-MOS, 3μ rule, 4MHz maximum system clock, 10mA power consumption
- Single 5V power supply, TTL level interface
- 40-pin plastic DIP package

1.3 Descriptions of terms

- MIDI

  Abbreviation for Musical Instrument Digital Interface, a communication standard for interconnecting the synthesizer, sequencer, rhythm machine and computer. Established in 1983 among the musical instrument industry.

  Operation and SYNC information on instruments, etc., can be transmitted real-time at a communication rate of 31.25K bps by the asynchronous serial communication using a 10-bit word comprising a start bit, eight data bits and a stop bit.

- Message

  A communication unit of data for MIDI. One message is composed of one character of status byte and data byte whose length is status byte-dependent. The MSB of the status byte is 1 and that of the data byte is 0.

- System real-time message

  A classification of the MIDI message. A general term indicating 8 types of status bytes from (F8)h to (FF)h with no data byte included. To provide a higher degree of real-time processing these messages can be transmitted by interrupting other message's data byte. The meanings of these messages are as follows:

(F8)h	Timing Clock:	SYNC clock. Indicates the note value of the ninety-sixth note (equivalent to 1/24th of the length of the quarter note). Hereafter called the MIDI clock.
(F9)h		Undecided
(FA)h	Start:	Start command for sequencer, etc. Starts from the opening of music.
(FB)h	Continue:	Continuous start command for sequencer, etc.
(FC)h	Stop:	Stop command for sequencer, etc.
(FD)h		Undecided
(FE)h	Active Sensing:	Message indicating the transmitter and transmission line are operating normally.
(FF)h	System Reset	

- System exclusive message

  One of the MIDI messages.

  Large data such as instrument tone parameters and nonstandard data are transferred with this message.

  Consists of status byte (F0)h and variable-length data byte. The first data byte is used as the ID code of a manufacturer. Thus the MIDI instrument manufacturer must have its ID code registered with the International MIDI standard committee. The format from the second data byte on can be decided by each manufacturer; a system exclusive message not in accordance with the manufacturer ID code can be neglected.

- Active sense message

  One of the MIDI system real-time messages. Its status byte is (FE)h. The MIDI specification requires that a transmitter transmit any message at least once per 300msec. If there is no message to be transmitted, this active sense message must be sent. When a receiver does not receive any message from a transmitter for 300msec, the transmitter or its transmission line is considered to be abnormal. In this case the sound generation is stopped and the receiver resumes its own operation.

  The receiver continues its own operation until it receives the first active sense message.

- FSK modulation

  FSK is an abbreviation for frequency-shift keying, a form of digital signal recording on a magnetic (cassette) tape in which the 1 and 0 are represented by two distinct frequencies.

  In the YM3523 the mark (H level = 1) and space (L level = 0) of the usual asynchronous serial signal are modulated with a frequency of two times the communication rate and a frequency equal to this rate, respectively. Demodulation function is also provided.

  With this type of modulation interface to the magnetic tape is obtained. The definition mentioned above is used throughout this manual.

- Tape SYNC

  Synchronizing multiple recording or automatic performance according to the SYNC signal recorded on a magnetic tape, etc.

- TCLK, TCLKM and TCLKF

  One-cycle time of the clock signal being input to terminals CLK, CLKM and CLKF, respectively.

2. SIMPLIFIED DESCRIPTION OF FUNCTIONS

2.1 Pin assignment

The pin configuration is shown in Fig. 2.1.

- $\overline{IC}$ ... Input

  The YM3523 is reset by the L level input signal to this terminal. The reset pulse width must be more than 32 clocks of the system clock to the CLK terminal (32 TCLK).

  The same reset operation can be made by the host CPU operation on the internal registers of YM-3523.

- CLK ... Input

  System clock input terminal

  The internal operation of the YM3523, input signal sampling and output signal changing are all performed in synchronization with the internal timing clock made by this system clock. The clock rate must be more than 32 times the communication rate used. In the case of an MIDI communication rate of 31.25K bps at least a 1MHz clock rate is necessary. The maximum rate is 4MHz.

- CLKI ... Output

  Internal timing clock output terminal.

- CLKM ... Input

  Input terminal for a clock that generates an MIDI communication rate of 31.25K bps (baud). Usually 1MHz or 0.5MHz is input, and 1/16 or 1/32 of this can be used as the communication rate.

  This clock signal is divided to obtain a count clock for the general-purpose timer and the MIDI clock timer and also for the output pulse width at the SYNC and CLICK terminals. A divide-by-two circuit is included so that the same setting time can be obtained, irrespective of the input signal to the CLKM terminal (1MHz or 0.5MHz).

- CLKF ... Input

  Input terminal for a clock that generates a communication rate of 75×2^n series.

  Usually 614.4KHz is input and communication rates from 1/8192nd of this frequency (75 bps) to 1/32nd of the same frequency (19200 bps) can be used. Can be used also for counting in the MIDI active sense function.

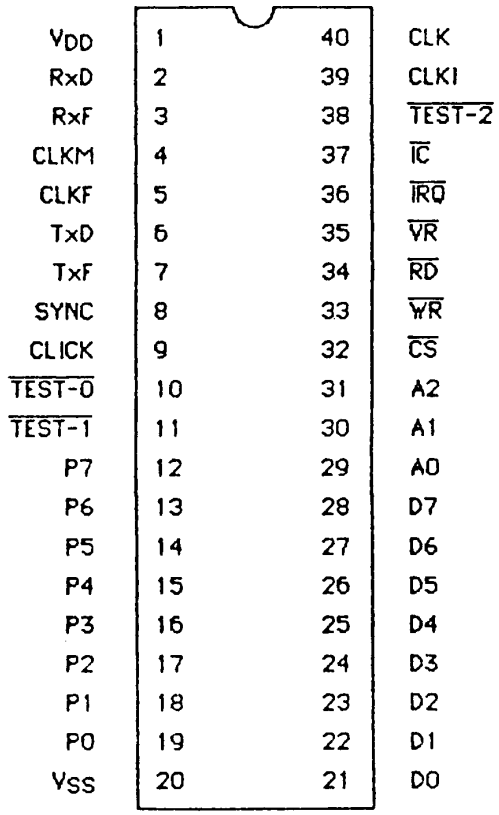


Figure 2.1 Pin assignment

- D0 ~ D7 ... Input/Output
  A0 ~ A2 ..... Input
  $\overline{CS}$ ..... Input
  $\overline{WR}$ ..... Input
  $\overline{RD}$ ..... Input
  $\overline{VR}$ ..... Input
  $\overline{IRQ}$ ..... Input
  These terminals are used to provide interface to the host CPU.

- $\overline{TEST\text{-}0}$ ~ $\overline{TEST\text{-}2}$ ... Input
  Terminals used for testing the LSI. Thus, no connections are generally made.

- RxD ... Input
  Serial data input terminal

- TxD ... Output
  Serial data output terminal

- RxF ... Input
  Audio signal input terminal from the magnetic tape such as a cassette tape. FSK-modulated serial data are input to this terminal at the TTL level.

- TxF ... Output
  Audio signal output terminal to the magnetic tape such as a cassette tape. FSK-modulated serial data are output from this terminal at the TTL level.

- SYNC ... Output
  A 2msec width pulse is output from this terminal in synchronization with the MIDI clock. Used as a SYNC signal for other hardware.

- CLICK ... Output
  A 2msec width pulse is output from this terminal in synchronization with the MIDI clock divided. Used when a metronome, etc., which is synchronized with the MIDI, is configured.

- P0 ~ P7 ... Input/output
  General-purpose I/O port with which the input/output direction of each bit can be separately set.

2.2 Internal block functions

The block diagram is given in Fig. 2.2.

- Transmitter

  Receives data from the FIFO-ITx or FIFO-Tx. Appropriate start, parity and stop bits are added to the received data, which is then converted to serial data. This serial data is sent direct to the TxD terminal, or FSK-modulated and sent to the TxF terminal.

  Communication rate, data length, type of parity bit and type of stop bit can each be set independent of the receiver, has the function of forcibly lowering the output to the break level (L level).

- FIFO-Tx

  16-byte transmitting data buffer. General serial data transmission is made by setting the data in this buffer. Because of the FIFO method the arbitrarily-set data is processed by the transmitter in the same order that it was set. An interrupt signal is generated when all the data stored in the buffer are sent to the transmitter.

  Whether or not the data set area is available in the buffer can be checked from the host CPU.

- FSK modulator

  The transmitter serial data output is FSK-modulated. Its conversion rate is dependent upon the communication rate.

  If the serial data is sent to the TxD terminal, the SYNC terminal output signal is FSK-modulated at a transfer rate of 1200 bps, which is then sent to the TxF terminal. The modulation execution and halt can be controlled from the host CPU.

- Receiver

  The serial data sent from the RxD terminal or the FSK demodulator is converted to the parallel data, and the parity and stop bits are checked. Usually, the parallel data and the error status are set in the FIFO-Rx.

  Communication rate, data length, type of parity bit and type of stop bit can be set independent of the transmitter.

- FIFO-Rx

  128-byte receiving data buffer. Usually, the data received by the receiver is set in the order it was received. This data can be read out anytime from the host CPU. The MIDI clock filter and the address hunter can prevent unnecessary data from being set in the FIFO-Rx. In addition to the 128 bytes for data storage the parity and stop bit check results are also stored for each data, therefore, this FIFO-Rx is in practice of the 10 bits x 128 words of storage buffer.

  An interrupt signal is generated after the receiver has received one-character data and set it in the empty buffer. The overrun error flag is set when new data is set in the buffer already full of data, permitting checking from the host CPU.

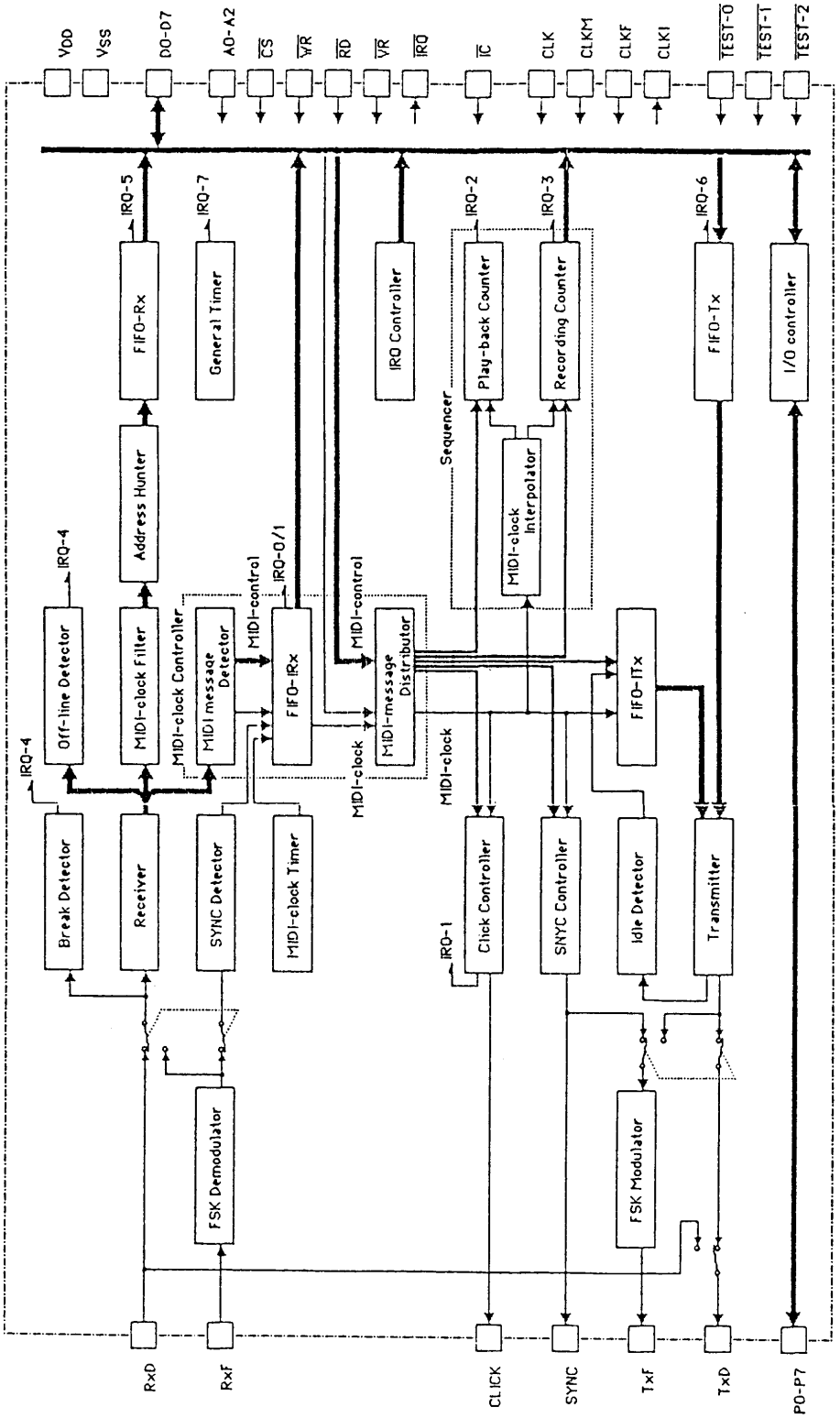


Figure 2.2 Block diagram

- FSK demodulator

  The FSK-modulated signal input to the RxF terminal is demodulated and sent to the receiver in the form of usual serial signal. In this case the conversion rate is in accordance with the receiver communication rate. The sampling timing of the serial signal that follows the input signal fluctuation with time is sent to the receiver where the serial-to-parallel conversion is performed at this timing.

  If the receiver handles the input signal at the RxD terminal this signal is demodulated at a transfer rate of 1200 bps and then sent to the SYNC detector. If the polarity accords, then the phase adjustment is performed at the time when the one wave-modulated data (L level) is demodulated. The positive or negative polarity is selectable.

  Also, automatic polarity follow-up can be performed by the polarity detector circuit. The baud check circuit is included and the result can be checked from the host CPU. Since the level of the input signal at the RxF terminal can be directly checked from the host CPU, it is possible that only the adjustments of baud, polarity and phase are conducted by the software, then the demodulator output is utilized.

- Break detector

  Break status detector circuit for the input signal to the receiver.

  If the L-level input continues during the transfer period of two characters that conform with the communication rate, data length, parity bit and stop bit set by the receiver, it is regarded as a break condition and thus an interrupt signal is issued.

- FIFO-ITx

  4-byte transmitting buffer used exclusively for the MIDI system real-time message.

  If data is set in the FIFO-ITx, the transmitter will transmit this data, irrespective of whether the FIFO-Tx data is present or not.

- Idle detector and off-line detector

  Functional blocks for the processing of the MIDI active sense.

  The idle detector measures the interval between the serial data transmissions by the transmitter, and if there is no transmission for a period of 80msec the MIDI active sense message is set in the FIFO-ITx.

  The off-line detector measures the interval between the receiver's serial data reception and the parallel data output. If there is no reception for a period of 300msec an interrupt signal is generated.

- MIDI clock filter

  With this filter the MIDI clock message (F8)h is prevented from being transferred to the FIFO-Rx. This can reduce the host CPU access to the receiving data buffer, if the reception of the clock message is used by the IRQ, etc.

- Address hunter

  When the MIDI system exclusive message is received, the manufacturer ID code or the manufacturer ID code plus one-byte ID code are checked against the ID code(s) pre-stored in the register. If there is no accord, this filter (address hunter) prevents the message from being transferred to the FIFO-Rx.

- SYNC detector and MIDI clock timer

  Internal MIDI clock generation sources.

  The MIDI clock timer is a 14-bit programmable interval timer. The same type of timer is used as the general-purpose timer (described later). The MIDI clock timer generates the MIDI clock timing signal instead of interrupt signals.

  The SYNC detector also generates the MIDI clock timing signal on the rising edge of the signal obtained by demodulating the tape SYNC signal that is input to the RxF terminal.

  In addition, the MIDI message detector (see later) generates the MIDI clock timing signal when it detects the reception of the MIDI clock message (F8)h by the receiver. The host CPU can also directly provide this timing signal.

According to this timing, concurrent operation of several functional blocks within the YM3523 can be made.

- SYNC controller

  Controls the SYNC signal output.

  A 2msec width pulse is sent to the SYNC terminal in accordance with the internal MIDI clock. Also, the FSK-modulated SYNC output signal is sent to the TxF terminal which is recorded on a magnetic tape for use as the SYNC signal for tape SYNC operation.

- CLICK counter

  7-bit programmable counter to count the internal MIDI clocks.

  This counter can be used as a general-purpose interval timer if the MIDI clock timer is employed as the MIDI clock generation source. When the values set in the register have been counted, a 2msec width pulse is sent to the CLICK terminal and at the same time an interrupt signal is issued. Count operation restarts by the automatic load of the counter with the preset value.

- Sequence–MIDI clock interpolator, Recording counter and Playback counter

  Functional blocks for time management based upon the internal MIDI clock.

  The MIDI clock interpolator generates count clocks at an interval equal to 1/nth of the MIDI clock generation interval. This provision is used to advantage when application requiring higher clock resolution is encountered.

  The recording counter is an 8-bit readable fixed counter which generates an interrupt when the count reaches zero. This counter is usable as a real-time clock with its carry operation processed by the host CPU software.

  The playback counter is a 15-bit programmable subtractive counter which counts the MIDI clock interpolator-generated count clocks and issues an interrupt signal when the count reaches zero or becomes negative.

  Although no automatic counter initialization is performed, the addition to the present count value is possible to cope with the host CPU's counter resetting delay.

  If the MIDI clock timer is used as the MIDI clock generation source, these counters can be utilized for general purposes.

- MIDI clock controller–MIDI message detector, FIFO-IRx and MIDI message distributor

  Control blocks for the internal MIDI clock.

  The MIDI message detector is a functional block that detects the reception of the MIDI system real-time message character by the receiver. When the reception of the clock message (F8)h is detected by the receiver this detector generates the internal MIDI clock timing signal. Similarly, when the reception of the system real-time messages (F9)h ~ (FD)h is detected the detected message contents are set in the FIFO-IRx.

  The FIFO-IRx is a 4-byte FIFO that processes the internal MIDI clock and keeps the process sequence of the system real-time data (messages) received by the receiver. When a selected MIDI clock timing signal (i.e., one of the MIDI clock timing signals of the MIDI

Sequencer
- MIDI clock interpolator
- Recording counter
- Playback counter

MIDI clock controller
- MIDI message detector
- FIFO-IRx
- MIDI message distributor

message detector. SYNC detector and MIDI clock timer) is generated, the (F8)h is set in the FIFO-IRx. Also, the data (F9)h ~ (FD)h sent from the MIDI message detector are set in this FIFO buffer. If the (F8)h is present at the exit of the FIFO it is automatically brought out as the MIDI clock and sent to the MIDI message distributor, causing an interrupt signal to be issued. In case other system real-time message exists at the FIFO exit another interrupt signal is generated, and data is not brought out until directed by the host CPU.

The data at the exit of the FIFO-IRx can be arbitrarily read out from the host CPU, irrespective of the operation of the FIFO, eliminating the need to wait for the data processing by the FIFO-Rx. The MIDI message distributor controls the functional blocks whose operation depends upon the internal MIDI clock.

According to the (F8)h signal sent from the FIFO-IRx or to the host CPU direction, the distributor sends the MIDI clock signal simultaneously to each functional block. Also, by the direction of the host CPU, the system real-time messages (F9)h ~ (FD)h are sent to the functional blocks either independently or in unison, controlling their operation (start, stop and initialization).

The functional blocks controlled by this distributor are the CLICK counter, SYNC controller and sequencer. The loading of data into the FIFO-ITx is also controlled by the distributor.

- General-purpose timer

  14-bit programmable interval timer (the same type as that for the MIDI clock timer)

  This timer counts a fixed 8 μsec clock. When the count of the values set in the register has been completed an interrupt signal is issued. The count restarts by the automatic load of the counter with the initial value.

- I/O controller

  Functional block that provides the input/output control on the terminals P0 ~ P7.

  Performs the setting of the I/O port input/output direction and of the output data. Both the input signal level at the input port terminal and the output signal level at the output port terminal can be read out.

- IRQ controller

  Controls the output level at the $\overline{\text{IRQ}}$ terminal, status register and vector register, according to the 10 interrupt requests within the LSI and to the contents of the interrupt control register.

  The status of interrupt requests is set by the bit map in the status register. The $\overline{\text{IRQ}}$ terminal output level goes low upon generation of an acknowledged interrupt request. From the interrupt requests acknowledged and being generated the vectors with high priority are set in the vector register. LSB of the vector is fixed at zero and the highest three bits are set by the host CPU.

2.3 Register layout

The register layout diagram is given in Fig. 2.3.

There are 38 registers built in to this LSI, as shown in this diagram. By way of example the register at the uppermost stage of the diagram is considered. R00 and IVR in this register are the register number and the register name, respectively. The register number's high-order digit and low-order digit are called the group number and the address number, respectively. A certain register can be accessed by first writing its group number into the R01 (RGR), then specifying its address number by the terminals A0 ~ A2. Repeated group number entry into the R01 is not needed when the registers of the same group are to be accessed. The registers with the address numbers 0 ~ 3 (R00 ~ R03) can always be accessed, irrespective of the group number.

Register functions are described in relevant chapters.

register	R/W	content	b7	b6	b5	b4	b3	b2	b1	b0
R00 IVR	R	IRQ vector	IRQ vector							
R01 RGR	W	system control	IC				register group number			
R02 ISR	R	IRQ status	IRQ status							
R03 ICR	W	IRQ clear request	IRQ clear request							
R04 IOR	W	IRQ vector offset request	IRQ vector offset							
R05 IMR	W	IRQ mode control					C/T	0/9	VE	VM
R06 IER	W	IRQ enable request	IRQ enable							
R14 DMR	W	MIDI realtime message control				ASE	MCE	CDE	MCDS	MCFS
R15 DCR	W	MIDI realtime message request	Tx	SYNC	CC	PC	RC	content of message		
R16 DSR	R	FIFO-iRx data	FIFO-iRx data							
R17 DNR	W	FIFO-iRx control								CLK
R24 RRR	W	Rx communication rate				RxD/F	Rx communication rate			
R25 RMR	W	Rx communication mode				RxCL	RxPE	RxPL	RxE/O	RxSL	RxST
R26 AMR	W	Address-hunter control	IDCL	maker ID code						
R27 ADR	W		BRDE	device ID code						
R34 RSR	R	FIFO-Rx status	RxRDY	RxOV	RxF	RxP	BRK	RxOL	AHBSY	RxBSY
R35 RCR	W	FIFO-Rx control	RxC	RxOVC		FLTE	BRKC	RxOLC	AHE	RxE
R36 RDR	R	FIFO-Rx data	FIFO-Rx data							
R44 TRR	W	Tx communication rate		TxRx	TxD/F	Tx communication rate				
R45 TMR	W	Tx communication mode			TxCL	TxPE	TxPL	TxE/O	TxSL	TxST
R54 TSR	R	FIFO-Tx status	TxEMP	TxBSY				TxIDL		TxBSY
R55 TCR	W	FIFO-Tx control	TxC				BLKE	TxIDLC		TxE
R56 TDR	W	FIFO-Tx data	FIFO-Tx data							
R64 FSR	R	FSK status	RxFS	SS	CSF	CFF			PS	PDF
R65 FCR	W	FSK control	ME			CFC	DE	APD	P/N	PDFC
R66 CCR	W	Click counter control							CLKM	OUTE
R67 CDR	W	Click counter load value	LD	7 bit data to load						
R74 SRR	R	Recording counter current value	8 bit data of current value							
R75 SCR	W	Sequencer control			ADD	CLR	4 bit interpolation rate			
R76 SPRL	W	Play-back counter value	lower 8 bit data to add							
R77 SPRH	W		higher 7 bit data to add							
R84 GTRL	W	General timer value	lower 8 bit data to load							
R85 GTRH	W		LD	higher 6 bit data to load						
R86 MTRL	W	MIDI-clock timer value	lower 8 bit data to load							
R87 MTRH	W		LD	higher 6 bit data to load						
R94 EDR	W	External I/O direction	direction of earch I/O port							
R95 EOR	W	External I/O output data	output request for I/O port							
R96 EIR	R	External I/O input data	pin level of earch I/O port							

A2,A1,A0 =	000	001	010	011	100	101	110	111
group-0	R00 IVR	R01 RGR	R02 ISR	R03 ICR	R04 IOR	R05 IMR	R06 IER	
1					R14 DMR	R15 DCR	R16 DSR	R17 DNR
2					R24 RRR	R25 RMR	R26 AMR	R27 ADR
3					R34 RSR	R35 RCR	R36 RDR	
4					R44 TRR	R45 TMR		
5					R54 TSR	R55 TCR	R56 TDR	
6					R64 FSR	R65 FCR	R66 CCR	R67 CDR
7					R74 SRR	R75 SCR	R76 SPRL	R77 SPRH
8					R84 GTRL	R85 GTRH	R86 MTRL	R87 MTRH
9					R94 EDR	R95 EOR	R96 EIR	

Figure 2.3 Register map

3. COMMUNICATION FUNCTIONS

3.1 Serial communication

3.1.1. Communication method

Fig. 3.1 illustrates the message format of the asynchronous serial communication employed in the YM3523. There are several modes of message format, depending upon the data bit length, the existence of parity bit, the parity bit length, the stop bit length and the type of stop bit.

The data bit length of seven or eight bits is treated as one character.

The parity bit modes are one-bit, four-bit and no parity, and odd or even parity can be used. In the four-bit parity mode, a pair of data bits correspond to one parity bit, as shown in Fig. 3.2.

The stop bit length are one or two bits, and two modes exist for the two bits length, i.e., one mode with two consecutive marks (H level) and the other mode with one mark and one space (L level). Fig. 3.3 indicates the stop bit modes.

The communication rates available are 1/16 or 1/32 of the input clock rate supplied to the CLKM terminal, or 1/32nd, 1/64th, ..., 1/8192nd of the input clock rate to the CLKF terminal. By inputting a clock of 1MHz or 0.5MHz to the CLKM terminal and MIDI communication rate of 31.25K bps can be obtained. And, if 614.4KHz is input to the CLKF terminal communication rates of from 75 bps to 19200 bps (75×2^n series) are available.

If the signal for the magnetic tape is handled by the internal FSK modulator and demodulator, the use of several CLKM communication rates (i.e., 1/16th and 1/32nd of the input) is inhibited.

3.1.2. Transmission procedure

When the transmit operation is acknowledged by the host CPU and data is set in the FIFO-Tx or FIFO-ITx, the transmitter adds the start bit, parity bit and stop bits to the data according to the transmit mode and transmits the data at the selected communication rate. When the data bit length is seven bits the MSB of the data set in the FIFO is ignored.

The transmitter output is sent direct to the TxD terminal or sent to the FSK modulator where the output is FSK-modulated per bit which is then routed to the TxF terminal.

3.1.3. Reception procedure

How the input signal to the RxF terminal is received is explained using Fig. 3.4. The receiver samples the input signal with the internal communication clock that is 16 times the communication rate (obtained by dividing the input clock at CLKM or CLKF terminal). If, under "waiting for a new message" condition, the receiver samples a space (L level) eight times after at least one mark (H level), it is recognized as a start bit. The eighth timing of the internal communication clock in this space condition is used as a midpoint of the start bit. Then the input sampling is performed by using every 16th clock of the internal communication clock as a midpoint for each bit, i.e., data bit, parity bit and stop bit. At the time when the last stop bit is sampled the receiver will go into the wait state for a next message.

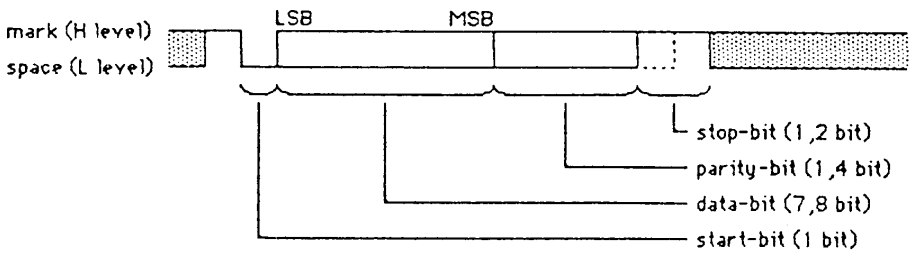


Figure 3.1 Communication signal format

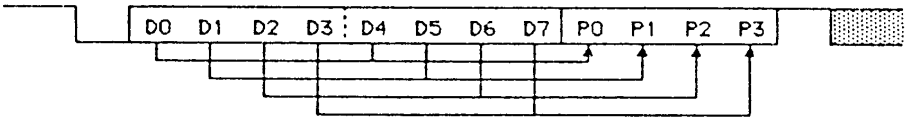


Figure 3.2 Data-bit, parity-bit relation in 4 bit parity mode

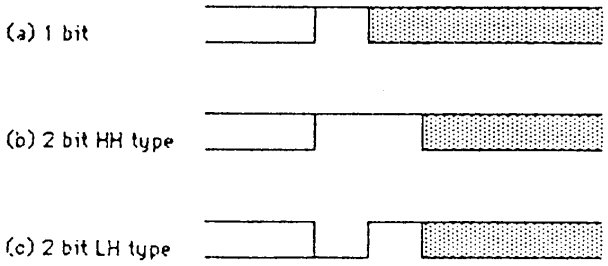


Figure 3.3 Stop-bit length and type

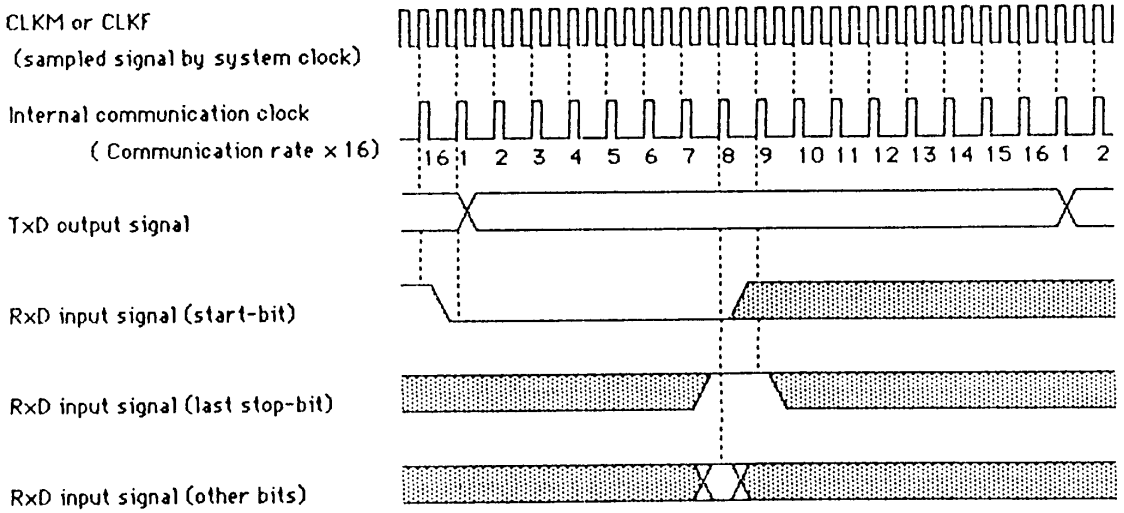


Figure 3.4 Clock & communication signal
(example in CLKM/32 or CLKF/32)

In the case of the reception of the FSK-modulated signal that is input to the RxF terminal, since the FSK demodulator generates a clock signal every time each bit is demodulated the per-bit sampling is performed based upon this clock. The start bit detection is made by the continuous reception of marks and spaces.

In either case, upon completion of the reception of one message 0 is added to the MSB in the case of the seven-bit length mode to form an eight-bit data signal. And, if any abnormal condition exists with the parity and stop bits the Parity Error and Framing Error signals are made active. These signals are loaded into the FIFO-Rx via the MIDI clock filter and address hunter and are accessible from the host CPU.

3.1.4. Operation of transmitter and receiver

The communication rate, communication mode (communication format) and wiring of the transmitter and receiver are set by means of registers R24, R25, R44 and R45. Register operations are prohibited when the transmitter and/or receiver are in operation. Fig. 3.5 shows the function of each register.

- TxRx (R44 – b6) – WRITE –
  If 1 is set to the TxRx, the input to the RxD terminal is unconditionally sent to the TxD terminal.

- TxD/F (R44 – b5), RxD/F (R24 – b5) – WRITE –
  If 0 is already set to the TxD/F, the transmitter serial output signal is sent to the TxD terminal as long as 1 is not set to the TxRx. At this time the serial input signal to the FSK modulator is the CLICK counter output signal to the CLICK terminal, and modulation is conducted at a rate of 1200bps (CLKF/512). If 1 is set to the TxD/F, then the transmitter serial output signal becomes the serial input signal to the FSK modulator, and modulation is performed at the transmitter's communication rate. As long as 1 is not set to the TxRx the output at the TxD terminal has the H level. If 0 is already set to the RxD/F, the input signal to the RxD terminal is used as the receiver input signal. At this time the FSK demodulator performs demodulation at a modulation rate of 1200 bps (CLKF/512), and its serial output signal is sent to the SYNC detector. If 1 is sent to the RxD/F, the FSK demodulator serial output signal becomes the receiver input, and the FSK demodulator conducts its operation at the communication rate of the receiver.

- R44 – b4 ~ b0, R24 – b4 ~ b0 – WRITE –
  According to the values set to these bits the communication rate of the transmitter and receiver is determined. By changing the input frequency to the CLKM and CLKF terminals the communication rate of the 110×2^n series, etc., can be realized. However, this alternation has an effect on the internal timers, SYNC and CLICK output pulse width and the active sense function.

R24 RRR	W	Rx communication rate	
b7 ↘ b6			
b5 RxD/F	select Receiver input connection	0 : RxD	1 : FSK demodulator
b4 ↘ b0	communication rate (in CLKM = 0.5 MHz CLKF = 614.4 KHz)	00xxx : CLKM/16 (31250 bps) --- inhibited while RxD/F = 1 01xxx : CLKM/32 (15625 bps) --- inhibited while RxD/F = 1 10xxx : CLKF/32 (19200 bps) 11000 : CLKF/64 (9600 bps) 11001 : CLKF/128 (4800 bps) 11010 : CLKF/256 (2400 bps) 11011 : CLKF/512 (1200 bps) 11100 : CLKF/1024 (600 bps) 11101 : CLKF/2048 (300 bps) 11110 : CLKF/4096 (150 bps) 11111 : CLKF/8192 (75 bps)	

R25 RMR	W	Rx communication mode	
b7 ↘ b6			
b5 RxCL	select data bit length	0 : 8 bit	1 : 7 bit
b4 RxPE	enable parity-bit check	0 : disable	1 : enable
b3 RxPL	select parity-bit length	0 : 1 bit	1 : 4 bit
b2 RxE/O	select parity-bit polarity	0 : even	1 : odd
b1 RxSL	select stop-bit length	0 : 1 bit	1 : 2 bit
b0 RxST	select stop-bit type in 2 bit length	0 : HH	1 : LH

Figure 3.5 Register functions for serial communication (1)

R44 TRR	W	Tx communication rate	
b7			
b6 TxRx	select TxD connection	0 : transmitter	1 : RxD
b5 TxD/F	select transmitter output connection	0 : TxD	1 : FSK modulator
b4 ~ b0	communication rate (in CLKM = 0.5 MHz CLKF = 614.4 KHz)	00xxx : CLKM/16 (31250 bps) --- inhibited while TxD/F = 1 01xxx : CLKM/32 (15625 bps) --- inhibited while TxD/F = 1 10xxx : CLKF/32 (19200 bps) 11000 : CLKF/64 (9600 bps) 11001 : CLKF/128 (4800 bps) 11010 : CLKF/256 (2400 bps) 11011 : CLKF/512 (1200 bps) 11100 : CLKF/1024 (600 bps) 11101 : CLKF/2048 (300 bps) 11110 : CLKF/4096 (150 bps) 11111 : CLKF/8192 (75 bps)	

R45 TMR	W	Tx communication mode	
b7 ~ b6			
b5 TxCL	select data-bit length	0 : 8 bit	1 : 7 bit
b4 TxPE	enable parity-bit generation	0 : disable	1 : enable
b3 TxPL	select parity-bit length	0 : 1 bit	1 : 4 bit
b2 TxE/O	select parity-bit polarity	0 : even	1 : odd
b1 TxSL	select stop-bit length	0 : 1 bit	1 : 2 bit
b0 TxST	select stop-bit type in 2 bit length	0 : HH	1 : LH

Figure 3.5 Register functions for serial communication (2)

- TxCL (R25 – b5), RxCL (R25 – b5) – WRITE –
  Determines the data bit length of the communication message.

- TxPE (R45 – b4), RxPE (R25 – b4) – WRITE –
  Determines whether parity is used or not.

- TxPL (R45 – b3), RxPL (R25 – b3) – WRITE –
  Determines the parity bit length.
  For the seven-bit data length one-bit parity is used, irrespective of the setting of this bit.

- TxE/O (R45 – b2), RxE/O (R25 – b2) – WRITE –
  Determines the parity bit polarity.

- TxSL (R45 – b1), RxSL (R25 – b1) – WRITE –
  Determines the stop bit length.

- TxST (R45 – b1), RxST (R25 – b1) – WRITE –
  Determines the type of stop bit. If, for a stop bit length of 2, 0 is set to this bit, 2 bits of mark (H level) is treated as stop bits; if 1 is set instead stop bits consists of one bit of space (L level) and one bit of mark. When a stop bit length is 1 setting of this bit is invalid.

3.1.5. Effects of the initial clear
Refer to sections 3.3 & 3.4.

3.2 Magnetic tape interface (FSK modulation)

3.2.1. Modulation method

Fig. 3.6 shows the serial data and the FSK-modulated signal used in the YM3523.

As can be seen from this figure, the FSK-modulated signal is produced by replacing marks (1s) and spaces (0s) in the serial data with two square waves and one square wave, respectively. When this signal is recorded and then reproduced by an audio cassette tape recorder, etc., distortions arise locally or wholly in the duty ratio and the communication rate; however, performing demodulation in synchronization with one- or two-wave signal can cope with these distortions.

3.2.2. Modulation procedure

When the modulation operation is acknowledged by the host CPU, the FSK modulator receives at the prescribed transfer rate the transmitter serial output signal or the SYNC output signal of the SYNC controller, modulates this signal and sends the modulated signal to the TxF terminal. In this case, the modulation rate for the transmitter serial output equals to the transmitter communication rate, while a modulation rate of 1200 bps is employed for the SYNC output signal. The CLKM communication rates of 1/16th and 1/32nd of the input frequency cannot be used.

3.2.3. Demodulation procedure

When the demodulation operation is acknowledged by the host CPU, the FSK demodulator demodulates the FSK-modulated signal which is input to the RxF terminal, and sends the serial signal to the receiver or the SYNC detector. If the serial data is to be sent to the receiver, the demodulation rate is determined by the communication rate set by the receiver. In this case, as described later, a clock signal is sent from the demodulator to the receiver each time one bit is demodulated, and the receiver samples the actual serial data using this clock. The CLKM communication rates (1/16th and 1/32nd of the input frequency) cannot be employed.

A demodulation rate of 1200 bps is used in the case of sending the serial data to the SYNC detector.

The operation of the FSK demodulator is described using Fig. 3.7. The internal communication clock – either 16 times the receiver communication rate or 16 times 1200 bps – is first produced. The input signal to the RxF terminal is read in synchronization with this clock, and its rising edge is detected. The first rising edge detected is regarded as the start timing for one bit, and the internal communication clock counter is set to 1. If the next rising edge is detected before the count of 13th clock, the serial output is set to the H level in synchronization with the 13th clock; if not detected it is set to the L level. The DSF (demodulated status flag) in the R64 (FSK status register) is set accordingly. At the same time, a clock signal is sent to the receiver to indicate the completion of one bit demodulation.

If more than one input signal rising edge is detected before the 13th clock, the CFF (carry first flag) in the R64 is set. Also, when the next input signal positive edge cannot be detected before the 21st clock, the CSF (carry slow flag) of the R64 is set. The 21st clock is regarded as the start timing for a temporary bit and the internal communication clock counter is set to 1. Then the serial output is set to the L level in synchronization witht he 13th clock, and at the same time a clock is sent to the receiver. At the 21st clock the counter is again set to 1. This cycle is repeated

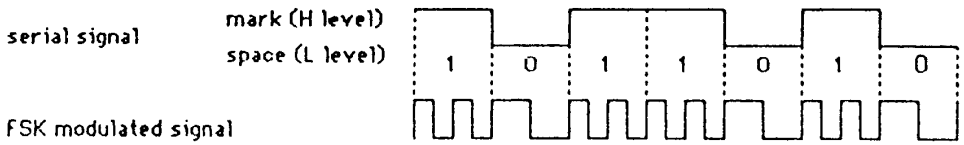


Figure 3.6 FSK modulation

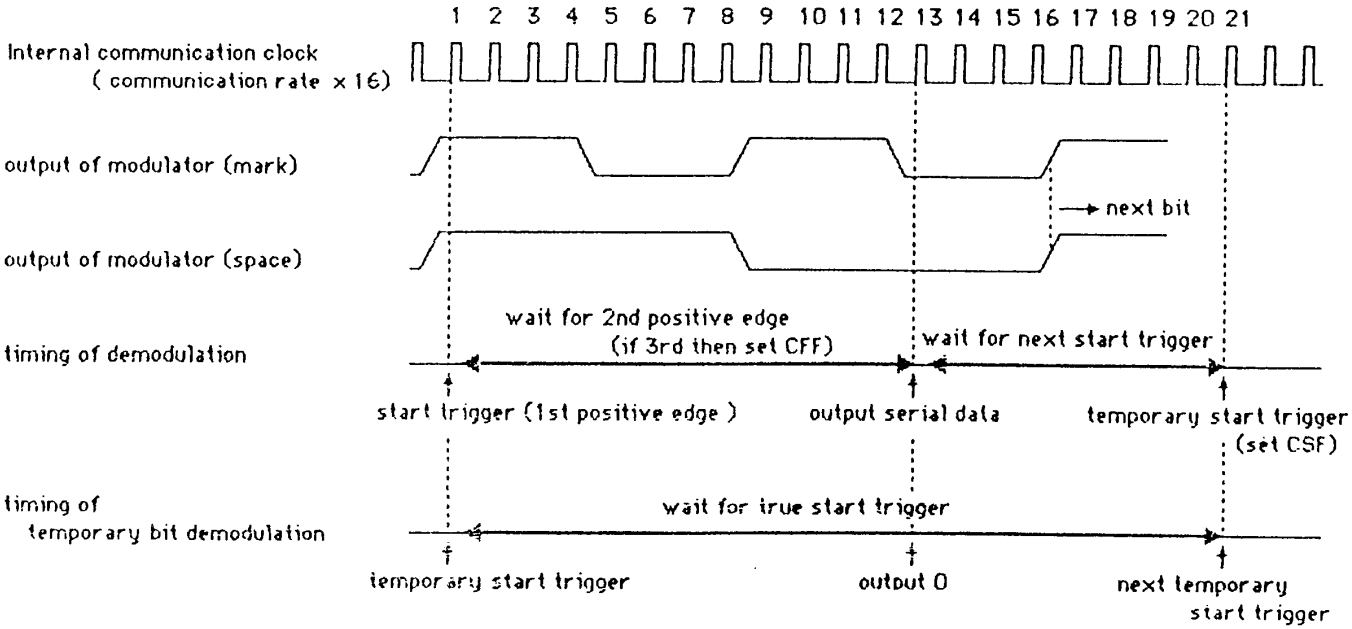


Figure 3.7 FSK demodulation

until the input signal positive edge is detected. When the positive edge is finally detected usual one bit process described above restarts. Although the serial output in the case of a temporary bit is treated as the L level within the demodulator, and the DSF of the R26 has the L level, the serial signal being sent to the receiver and the SYNC detector has the H level.

Fig. 3.8 illustrates the manner in which the phase alignment of the FSK-modulated signal is accomplished by the FSK demodulator. When demodulation starts from the serial signal mark (two wave-modulated signal = H level), it tends to be 180° out-of-phase, as shown in Fig. 3.8 (1) & (2). As is apparent from the figure, there is a possibility for one bit of space (one wave-modulated signal = L level) to be overlooked.

The polarity (H and L levels) inversion can be applied to the input signal from the RxF terminal to the demodulator. The positive or negative polarity can be specified from the host CPU, or according to the result from the internal polarity detection circuit the automatic polarity follow-up system functions.

The operation of this polarity detection circuit is described using Fig. 3.9. The input signal at the RxF terminal is loaded into the 12-bit shift register according to the internal communication clock. Then the data are extracted from the register in synchronization witht he 1st, 5th, 6th, 7th, 8th and 12th clocks (counting from the oldest data stage of the register). If the data synchronized with the 5th and 6th clocks only have different values from other data, the data before the 6th clock are considered mark (two wave modulation), while the data after the 7th clock are regarded as space (one wave modulation). If the space start position is known the direction of the input signal change at the start corresponds to the positive direction.

In Fig. 3.9 three types of input signals, each with a slightly different communication rate from the others, are given as an example. Polarity decision will be made for the input signal whose phase is opposite to that of the three input signals shown.

In the automatic polarity follow-up conducted according to the detected polarity the phase alignment is made in accordance with the timing of the mark-to-space transition detected during the polarity detection, and the internal communication clock counter is set to 1.

In the case of the automatic polarity decision the requirements for the input signal duty ratio are considerably strict. To prevent the dropout of bits in phase alignment it is desirable that the recorded data header portion have a mixture of marks and spaces and that use be made of the algorithm that the polarity is fixed after repeated decisions.

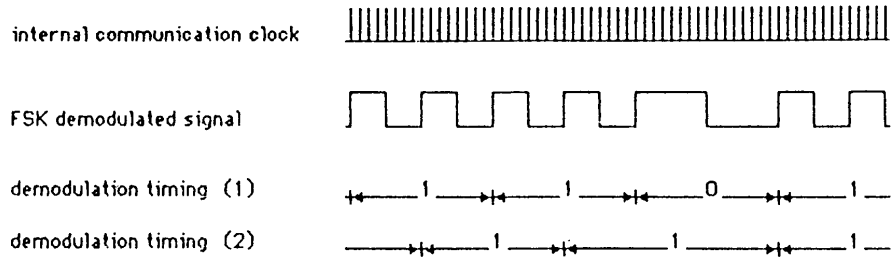


Figure 3.8 Phase tracing in FSK demodulatoin

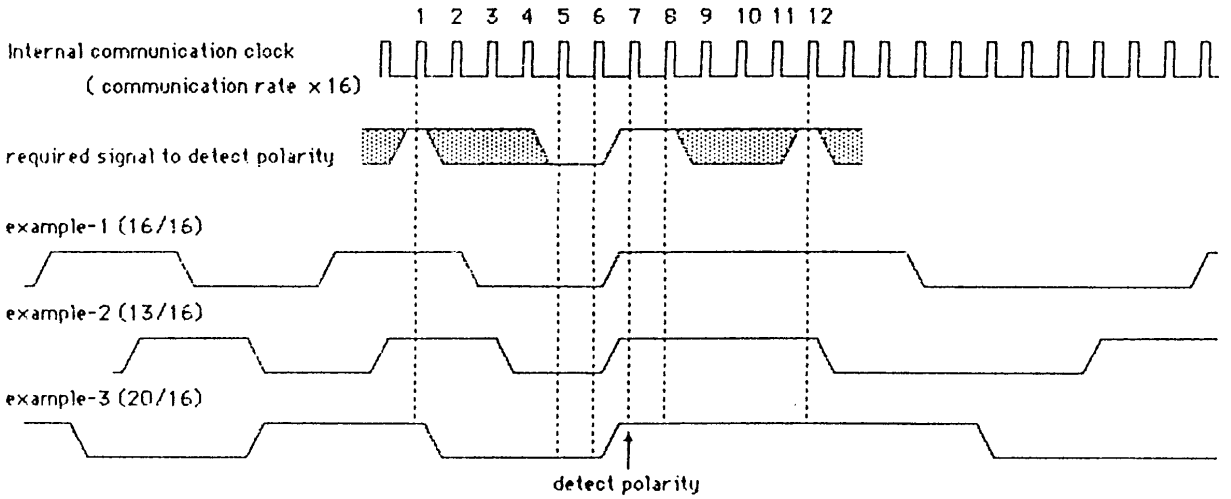


Figure 3.9 Polarity detection in FSK demodulation

3.2.4. Operation of the FSK modulator and demodulator

The operation of the FSK modulator and demodulator is controlled by the register R65. The status of the FSK demodulator is set in the register R64, and can be read out at any time. Fig. 3.10 shows the function of each register.

- RxFS (R64 – b7) – READ–
  Status flag that indicates the input level at the RxF terminal.

- SS (R64 – b6) – READ –
  Status flag that indicates the result of demodulation.
  A temporary bit (see section 3.2.3.), when demodulated, is treated as space (L level).

- CSF (R64 – b5) – READ –
  If the RxF input signal positive edge cannot be detected for a period equal to 21 clocks by the internal communication clock or 21/16th of a bit counting from the start of any one bit, 1 is set. This flag denotes that a signal with the communication rate slower than the prescribed demodulation rate may be input.

- CFF (R64 – b4) – READ –
  If more than one positive edge from the RxF input signal is detected during the period equal to 12 clocks by the internal communication clock or 12/16th of a bit counting from the start of any one bit, 1 is set. This flag indicates that a signal with the communication rate faster than the prescribed demodulation rate may be input.

- PS (R64 – b1) – READ –
  The detection result of the R x F input signal polarity is set in this status flag.

- PDF (R65 – b0) – READ –
  This flag is set when the polarity detection is made.

- ME (R65 – b7) – WRITE –
  If 1 is already written the FSK modulator operates; if 0 is written its operation stops.

- CFC (R65 – b4) – WRITE –
  If 1 is written 0 is set in the CSF and CFF.

- DE (R65 – b3) – WRITE –
  If 1 is already written the FSK demodulator operates; if 0 is written its operation stops.

R64 FSR	R	FSK status		
b7 RxFS	RxF pin status		0 : L level	1 : H level
b6 SS	demodulated serial signal status		0 : space (L)	1 : mark (H)
b5 CSF	carrier slow detected flag		0 : -	1 : carrier slow detected
b4 CFF	carrier fast detected flag		0 : -	1 : carrier fast detected
b3 ↓ b2				
b1 PS	polarity status		0 : positive	1 : negative
b0 PDF	polarity detected flag		0 : -	1 : polarity detected

R65 FCR	W	FSK control		
b7 ME	enable Modulator		0 : disable	1 : enable
b6 ↓ b5				
b4 CFC	clear carrier S/F detected flag		write 1 to clear flag	
b3 DE	enable Demodulator		0 : disable	1 : enable
b2 APD	disable auto polarity detector		0 : enable	1 : disable
b1 P/N	set polarity by manual		0 : positive	1 : negative
b0 PDFC	clear polarity detected flag		write 1 to clear flag	

Figure 3.10 Register functions for FSK Modulator/Demodulator

- APD (R65 – b2) – WRITE –
  If 0 is already written the automatic polarity follow-up function operates. If 1 is written, the input polarity is determined according to the subsequent P/N bit (see below) and demodulation is performed.

- P/N (R65 – b1) – WRITE –
  Sets the input polarity when the automatic polarity follow-up function is not used.

- PDFC (R65 – b0) – WRITE –
  If 1 is written 0 is set in the PDF.

3.2.5. Effects of the initial clear
Refer to section 3.3 & 3.4.

3.3 Functions related to transmitting

3.3.1 FIFO-ITx and FIFO-Tx

The FIFO-ITx is a 4-byte transmitting buffer intended exclusively for the MIDI system realtime messages (F8)h ~ (FE)h. For details, refer to section 4.1: "MIDI clock controller and related functions".

The FIFO-Tx is a general-purpose 16-byte transmitting buffer. If data is set in this buffer from the host CPU, the data will be sent to the transmitter in the set order. The flag showing the FIFO is not full and the flag showing the empty FIFO condition can be read out from the host CPU. If the FIFO-Tx becomes empty because of the read-out by the transmitter an interrupt signal will be sent to the IRQ controller.

The transmitter gives priority to the data (if present) of the FIFO-ITx; therefore, the data in this buffer will first be read out and transmitted.

3.3.2. Idle detector

Detects the long absence of transmit operation.

If there is no transmission for approximately 80msec (equal to the time of six counts of the signal obtained by dividing the input clock at the CLKF terminal by 2^{13} : if CLKF = 614.4KHz, then clock interval = 13.4msec, time = 66.7 – 80.1msec), the flag is set, and the active sense message (FE)h is set in the FIFO-ITx if already acknowledged by the host CPU. The same operation will be repeated every sixth counts. For the acknowledgment of the setting of the (FE)h in the FIFO-ITx, see section 4.1: "MIDI clock controller and related functions".

3.3.3. Transmitter and FSK modulator

Refer to section 3.1 for the transmitter, and section 3.2 for the FSK modulator.

An auxiliary function of the transmitter is the BREAK character transmit function. If instructed by the host CPU the transmitter goes to the L level and will maintain that level until released. In this case the internal transmit operation of the transmitter continues without interruption.

3.3.4. Register operation

The transmission-related functions are controlled by the registers R54, R55 and R56. Fig. 3.11 shows the function of each register.

- TxEMP (R54 – b7) – READ –
  Status flag that indicates the FIFO-Tx is empty.

- TxRDY (R54 – b6) – READ –
  Status flag that indicates the FIFO-Tx is not full and hence data can be set.
  If this flag is 0, additional data, if sent to FIFO-Tx, will be lost.

R54 TSR	R	FIFO-Tx status	
b7 TxEMP	FIFO-Tx empty flag	0 : data exist	1 : empty
b6 TxRDY	FIFO-Tx ready flag	0 : FIFO-Tx full	1 : ready
b5 ↘ b3			
b2 TxIDL	Tx idle detected flag	0 : -	1 : Tx idle detected
b1			
b0 TxBSY	Transmitter busy flag	0 : idle	1 : busy

R55 TCR	W	FIFO-Tx control	
b7 TxC	clear FIFO-Tx & FIFO-ITx	write 1 to clear FIFO-Tx & FIFO-ITx	
b6 ↘ b4			
b3 BRKE	enable send break	0 : -	1 : set Tx output signal as L level
b2 TxIDLC	clear Tx idle flag	write 1 to clear flag	
b1			
b0 TxE	enable Transmitter	0 : disable	1 : enable

R56 TDR	W	FIFO-Tx data
b7 ↘ b0	set data to FIFO-Tx	8 bit data

Figure 3.11 Register functions for Transmitter

- TxIDL (R54 – b2) – READ –
  This flag is set each time non-transmit condition of approximately 80msec (when CLKF = 614.4KHz) is detected by the idle detector.

- TxBSY (R54 – b0) – READ –
  Status flag that indicates the transmitter is in transmit operation.

- TxC (R55 – b7) – WRITE –
  Clears the contents of the FIFO-ITx and FIFO-Tx. The IRQ is not generated.

- BRKE (R55 – b3) – WRITE –
  If 1 is written the transmitter serial output goes to the L level and stays there until 0 has been written. The transmitter internal operation continues during the L level period.

- TxLDLC (R55 – b2) – WRITE –
  Clears the TxIDL flag.

- TxE (R55 – b0) – WRITE –
  Enables the transmitter operation. When its operation is inhibited by writing 0 a character now under transmission will be transmitted completely. Read-out of a new character from the FIFO-Tx and FIFO-ITx is inhibited.

- R56 – b7 ~ b0 – WRITE –
  Data written into this register is set in the FIFO-Tx.

3.3.5. Note on IRQ

If the FIFO-Tx becomes empty because of the read-out by the transmitter an interrupt signal is sent to the IRQ controller. This signal is cleared by the operation on the IRQ controller or if the TxEMP flag becomes zero.

3.3.6. Effects of initial clear

The initial clear of the hardware and software will cause the following operations to be made:

- The TxD and TxF terminal outputs go to the H level and the L level, respectively.
- The operation of the transmitter and FSK modulator is stopped and initialized.
- Idle counter starts its count from zero.
- Zeros are written into the registers.
- All flags other than the status flags are cleared.

3.4 Functions related to receiving

3.4.1. FIFO-Rx

The FIFO-Rx is a receiving buffer of 10 bits x 128 words. Fig. 3.12 illustrates its configuration.

When the receiver receives one character of message the 8-bit data and the 2-bit error flag are set in the FIFO-Rx. If the data is read out by the host CPU the data in the FIFO-Rx is automatically forwarded. The error flag associated with the data to be read next can be read out by the status register RST (R34).

In the 4-bit length parity mode if any error is detected one word consisting of the 4-bit parity error flag 6 zeros is set in the FIFO-Rx following the 8-bit data plus 2-bit error flag.

The flag indicating that the FIFO has the data to be read out and the flag indicating that the FIFO under the full condition is provided with new data (i.e., overflow condition) can be read out by the host CPU. If the FIFO goes into the overflow state, the flag is set and at the same time the second oldest data (one word) is lost, the data that follows is forwarded and the latest data is set. Because of this, when the host CPU detects the framing error, parity error and overrun error, one word of data is abandoned, thereby guaranteeing the data thereafter.

If new data is set in the empty FIFO an interrupt signal is sent to the IRQ controller.

3.4.2. Break detector

Detector circuit for the BREAK condition.

When the receiver input signal has the L level for the period of two characters according to the reception mode and rate currently set, the flat is set and an interrupt signal is sent to the IRQ controller. The interrupt cannot be used at the same time with that from the off-line detector (see below).

3.4.3. Off-line detector

Detects the long absence of receive operation. If there is no parallel data output from the receiver for approximately 300msec (equal to the time of 24 counts of the signal obtained by dividing the input clock at the CLKF terminal by 2^{13} : if CLKF = 614.4KHz, then clock interval = 13.4msec, time = 308.2 – 321.6msec), the flag is set and an interrupt signal is sent to the IRQ controller. The interrupt cannot be used at the same time with that from the break detector (see above).

3.4.4. MIDI clock filter

Circuit to eliminate the MIDI clock message (F8)h from the received data. If this flag is enabled by the host CPU, the setting of data in the FIFO-Rx is inhibited only when the (F8)h is received without error.

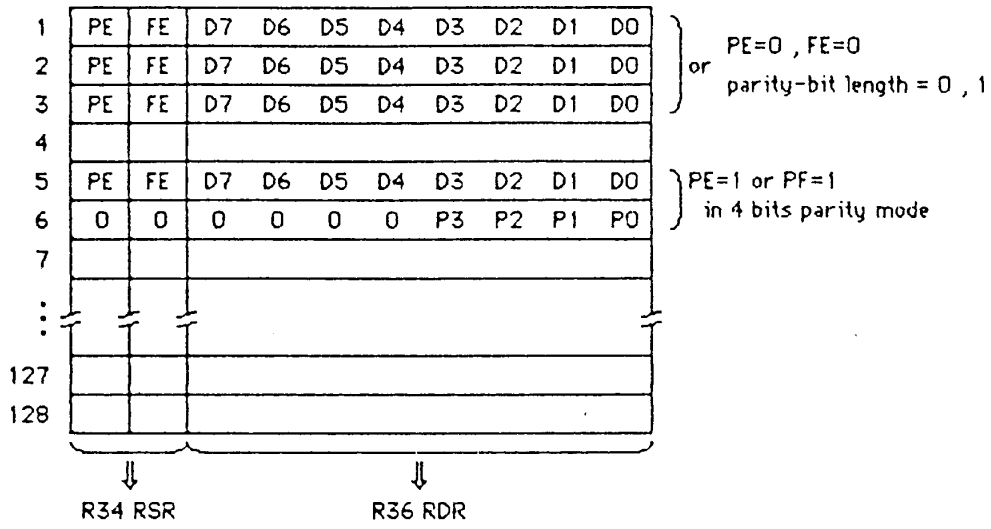


Figure 3.12 FIFO-Rx structure

3.4.5. Address hunter

Circuit to eliminate the special format data train referred to as the MIDI system exclusive message. If enabled, by the host CPU, the setting of the unnecessary system exclusive message data received without error into the FIFO-Rx is inhibited.

For details, see section 4.4 "Address hunter".

3.4.6. Receiver and FSK demodulator

See section 3.1 for the receiver, and section 3.2 for the FSK demodulator.

3.4.7. Register operations

Transmission-related functions are controlled by the registers R34 and R35. Refer to section 4.4 for address hunter controls.

- RxRDY (R34 – b7) – READ –
  Status flag that shows the FIFO-Rx has the data to be read out.

- RxOV (R34 – b6) – READ –
  Flag that shows the overflow has occurred in the FIFO-Rx.

- RxF (R34 – b5) – READ –
  Framing error flag for the next data to be read out.

- BRK (R34 – b3) – READ –
  Flag that indicates the BREAK condition has been detected.

R34 RSR	R	FIFO-Rx status		
b7 RxRDY	FIFO-Rx ready flag		0 : empty	1 : data ready
b6 RxOV	FIFO-Rx overflow detected flag		0 : -	1 : overflow detected
b5 RxF	framing error flag		0 : -	1 : framing error detected
b4 RxP	parity error flag		0 : -	1 : parity error detected
b3 BRK	break detected flag		0 : -	1 : break detected
b2 RxOL	off-line detected flag		0 : -	1 : off-line detected
b1 AHBSY	Address-hunter busy flag		0 : Address-hunter idle	1 : Address-hunter busy
b0 RxBSY	Receiver busy flag		0 : Receiver Idle	1 : Receiver busy

R35 RCR	W	FIFO-Rx control		
b7 RxC	clear FIFO-Rx		write 1 to clear FIFO-Rx	
b6 RxOVC	clear overflow detected flag		write 1 to clear flag	
b5				
b4 FLTE	enable MIDI-clock Filter		0 : disable	1 : enable
b3 BLKC	clear break detected flag		write 1 ti clear fiag	
b2 RxOLC	clear off-line detected flag		write 1 to clear flag.	
b1 AHE	enable Address-hunter		0 : disable	1 : enable
b0 RxE	enable Receiver		0 : disable	1 : enable

R36 RDR	R	FIFO-Rx data	
b7 ~ b0	get data from FIFO-Rx & increment FIFO-Rx		8 bit data

Figure 3.13 Register functions for Receiver

- RxOL (R34 – b2) – READ –
  Flag that indicates the off-line condition has been detected.

- AHBSY (R34 – b1) – READ –
  Status flag that shows the operating address hunter detects the MIDI system exclusive message and is processing it.

- RxBSY (R34 – b0) – READ –
  Status flag that shows the receiver receives the serial data and is converting this data to the parallel equivalent.

- RxC (R35 – b7) – WRITE –
  Clears the contents of the FIFO-Rx.

- RxOVC (R35 – b6) – WRITE –
  Clears the RxOL flag.

- FLTE (R35 – b4) – WRITE –
  Enables the operation of the MIDI clock filter.

- BLKC (R35 – b3) – WRITE –
  Clears the BLK flag.

- RxOLC (R35 – b2) – WRITE –
  Clears the RxOL flag.

- AHE (R35 – b1)
  Enables the operation of the address hunter.

- RxE (R35 – b0)
  Enables the operation of the receiver.

- R36 – b7 ~ b0
  Reads the first (oldest) data of the FIFO-Rx. The data in the FIFO is automatically forwarded when the read-out is performed.

3.4.8. Note on IRQ

If the empty FIFO-Rx receives the parallel data from the receiver, an interrupt signal is sent to the IRQ controller. This interrupt is cleared by operating upon the IRQ controller, or when the FIFO-Rx becomes empty by the read-out of data.

The interrupt signal from the break detector and the off-line detector to the IRQ controller reflects the contents of the BRK and RxOL flags. This signal is cleared by operating upon the IRQ controller, or cleared simultaneously if the BRK and RxOL flags are cleared. Conversely, the operation of the IRQ controller causes the BRK and RxOL flags to be cleared.

The simultaneous use of both IRQs from the host CPU is impossible.

3.4.9. Effects of the initial clear

The initial clear of the hardware and software will cause the following operations to be made:

- The operation of the receiver and FSK demodulator is stopped and both are initialized.
- The count of the break detector and off-line detector starts at zero.
- Registers are written with zeros.
- Flags other than status flags are cleared.

4. MIDI SERVICE FUNCTIONS

4.1 MIDI clock controller and related functions

4.1.1. Purpose of the MIDI clock controller

The MIDI system real-time message includes the clock message (MIDI clock) used for synchronized performance of a plurality of instruments (e.g., sequencer and rhythm machine), the start message, the stop message and the continue (continuous start) message. The MIDI clock is transmitted every 1/24th of a quarter note (ninety-sixth note) and each instrument measures the time based upon this clock for the performance.

The MIDI clock controller of the YM3523 is provided to attain the following purposes with respect to the messages: attain the following purposes with respect to the messages:

- Automatic control of the counters and SYNC output circuits synchronized with the reception of the clock message and the tape SYNC signal, or the clock message transmission.
- The speedup of the CPU response to the reception of the system real-time message.
- The priority transmission of the system real-time message and automatic transmission of some of the system real-time message.
- The generation of the internal MIDI clock triggered by the reception of the clock message, the tape SYNC signal, the internal timer, or by the host CPU operation, and subsequent IRQ generation.
- The concurrent control of the counters, the output circuit and the message transmission, all of which synchronized with the internal MIDI clock.
- The guarantee of the process sequence of the clock message and other system real-time messages when the reception of the clock message is used as a trigger for the internal MIDI clock.

4.1.2. SYNC detector

Functional module that detects the SYNC signal from the tape SYNC input signal and supplies this detected signal as one of the MIDI clock sources.

When zero is already set in the register R25–b5 (Receiver input controller) and the input signal to the RxD terminal is already fed to the receiver, this module operates using the serial output of the FSK demodulator as its input. If, during operation, the module detects the positive edge of its input level, the FIFO-IRx is informed of the generation of the MIDI clock. Since the FSK demodulator operates at a fixed demodulation rate of 1200 bps this positive edge detection is equivalent to the detection of the timing of the RxD terminal input signal variation from 1200 Hz to 2400Hz. If the transfer rate error is detected the serial output signal of the FSK demodulator is set to the H level; thus, at the instant that the input signal is disconnected from the RxD terminal there is the possibility of the MIDI clock generation.

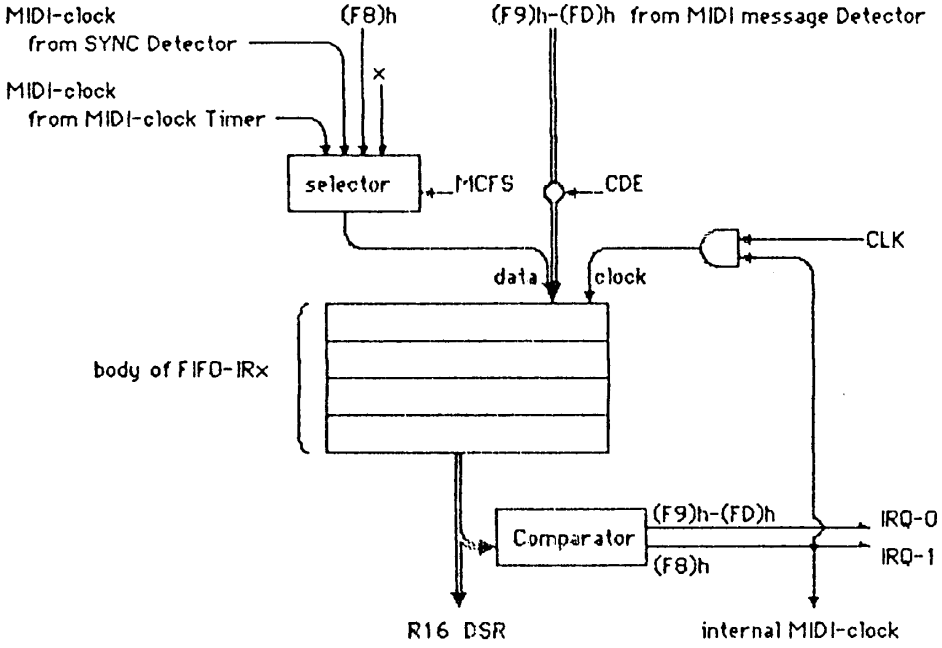


Figure 4.1 structure of FIFO-IRx

4.1.3. MIDI clock timer

An exclusive timer – one of the internal MIDI clock sources.

A pulse with an interval of eight μsec is counted by the 14-bit counter. This pulse signal is produced by dividing the CLKM by four or eight, depending upon the setting of the register R66–b1 (CLKM frequency select). Since the count is made only once every 16 TCLK, the 16 TCLK becomes the unit of count in case the 4 TCLKM or 8 TCLKM in longer than the 16 TCLK.

When the counter reaches a count of zero reloading of the count value is performed. With the timing of the loading or reloading of the count value the FIFO-IRx is informed of the MIDI clock generation.

If a value of less than or equal to one is loaded into the counter its operation is not guaranteed.

4.1.4. MIDI message detector

Functional block that monitors the reception data of the receiver and supplies the reception of the clock message (F8)h as one of the internal MIDI clock sources, and detects the reception of (F9)h ~ (FD)h in the system real-time messages (F9)h to (FF)h.

Message detection is performed only for the data received normally, that is, received without parity and framing errors, etc. The detection results will be sent to the FIFO-IRx.

4.1.5. FIFO-IRx

4-byte FIFO that is provided to guarantee the process sequence of the internal MIDI clock – especially that is generated on the basis of the reception of the clock message (F8)h by the receiver – and the system real-time messages (F9)h to (FD)h.

The clock message (F8)h is set in this FIFO by the MIDI clock generation signal from one of the following blocks (which is selected by the host CPU): the SYNC detector, the MIDI clock timer and the MIDI message detector. Also, if the MIDI message detector enabled by the host CPU detects the reception of the other system real-time messages (F9)h ~ (FD)h, the detected message is set in the FIFO. If new data is tried to be set in this FIFO which is already full of 4-byte data, the new data will be lost.

The oldest data stored in the FIFO can be read out by the host CPU. Unlike the usual FIFO, the read-out of the oldest data is performed without extracting it. Its extraction is made by the other operation of the host CPU. If the oldest data becomes the clock message (F8)h, it is automatically extracted, and the internal MIDI clock signal is sent to the MIDI message distributor. With this automatic process of the MIDI clock is done, and also of the MIDI clock that is processing other system real-time messages received is suspended.

When the FIFO's oldest data becomes the clock message (F8)h, an interrupt signal called the MIDI clock detect is sent to the IRQ controller; or when the oldest data becomes other system real-time messages (F9)h ~ (FD)h, an interrupt signal called the MIDI real-time message detect is sent to the IRQ controller.

4.1.6. MIDI message distributor

Distributor for the internal MIDI clock and other system real-time messages.

The counters and output circuits that receive the messages (FA)h, (FB)h and (FC)h perform the necessary operation, that is, start, stop and initialization, according to the meanings of the messages (start, stop and continue). The messages that are sent to the FIFO-ITx are transmitted by the transmitter.

According to the signal from the FIFO-IRx or the register operation by the host CPU (register operation is selectable from the host CPU), the internal MIDI clock is sent simultaneously to the FIFO-ITx, sequencer, CLICK counter and SYNC controller. Other real-time messages are sent to one or more designated functional blocks throught the register operation by the host CPU.

4.1.7. FIFO-ITx

4-byte transmitting buffer intended exclusively for the system real-time messages (F8)h ~ (FE)h.

The messages (F8)h ~ (FE)h sent from the MIDI message distributor or the idle detector (see section 3.3) are extracted and transmitted on a single byte basis while the transmitter is enabled.

The setting of the clock message (F8)h and active sense message (FE)h is enabled/disabled through the register operation by the host CPU. For the other system real-time messages (F9)h ~ (FD)h the FIFO-ITx only can be removed from designation by operating the MIDI message distributor. Therefore, the operation of the FIFO-ITx can be fully controlled from the host CPU.

4.1.8. Register operations

Registers R86 and R87 control the MIDI clock timer, and registers R14 through R17, control other MIDI clock controller-related functions.

Fig. 4.2 shows the function of each register.

- R87 – b5 ~ R86 – b0 – WRITE –

  Sets the load value for the MIDI clock timer. The MIDI clock timer "down-counts" the value set in the internal counter every unit of time. When a count of zero is reached the value set in this register is reloaded in the counter. At the time of reloading the MIDI clock generation signal is sent to the FIFO-IRx.

  For the unit of time, 1/4th or 1/8th of the CLKM, which is dependent upon the setting of the R66 – b1 (CLKM frequency select) and is 8 μsec for the preset CLKM, or 1/16th of the CLK will be selected and whichever is longer is taken.

  The value set in the register can settle as the reload value when data is written into R87. The reload value during the writing into R86 and R87 is equal to the value before R86 is rewritten. This prevents loading irregular values during the change of the reload value setting.

- LD (R87 – b7) – WRITE –

  If 1 is written into this bit reloading will be performed within 32 TCLK, regardless of the present count value of the MIDI clock timer. The MIDI clock generation signal is also sent to the FIFO-IRx.

* In the case of the repeated access to the registers R86 and R87 it is necessary that the same register be accessed with the interval equal to 16 TCLK plus 8 TCLKM.

- ASE (R14 – b5) – WRITE –

  Setting of the active sense message (FE)h from the idle detector in the FIFO-ITx is enabled/disabled.

- MCE (R14 – b4) – WRITE –

  Setting of the clock message (F8)h from the MIDI message distributor in the FIFO-ITx is enabled/disabled.

- CDE (R14 – b3) – WRITE –

  Setting of the system real-time messages (F9)h ~ (FD)h from the MIDI message detector in the FIFO-IRx is enabled/disabled.

- MCDS (R14 – b2) – WRITE –

  Determines whether the generation of the internal MIDI clock is performed by the internal MIDI clock from the FIFO-IRx or by the operation of the register R15 by the host CPU.

- MCFS (R14 – b1 ~ b0) – WRITE –

  Determines where the request comes from for setting the clock message (F8)h in the FIFO-IRx: the MIDI message detector, SYNC detector or MIDI clock timer.

- R15 – b7 ~ b3 – WRITE –

  Designates the destinations (blocks) to which messages (F9)h ~ (FD)h are to be sent. The clock message (F8)h is always sent to all blocks.

- R15 – b2 ~ b0 – WRITE –

  If this register is written, the message comprising its three bits plus five bits of 1s are sent to the blocks designated by b7 ~ b3. The MIDI clock (F8)h, however, is sent to all blocks. This MIDI clock (F8)h is sent with the timing of register write or with the timing of the internal MIDI clock sent from the FIFO-Rx, depending upon the condition of the MCDS (R14 – b2).

- R16 – b7 ~ b0 – READ –

  Reads the oldest data from the FIFO-IRx. The data read is not lost. (00)h has been set in this register when the FIFO is empty.

R86 MTRL	W	MIDI-clock Timer value (L)	
b7 ~ b0	lower 8 bit data		data (unit : 8 msec. IRQ occurs when data is loaded .)

R87 MTRH	W	MIDI-clock Timer value (H)	
b7 LD	imediate load request		write 1 to load value (data is loaded while $32T_{CLK}$ passed.)
b6			
b5 ~ b0	higher 6 bit data		data (unit : 8 x 256 msec.)

* count unit is ;
  - $4T_{CLKM}$ in R66-b1=0(CLKM=0.5MHz), $4T_{CLKM} > 16T_{CLK}$
  - $8T_{CLKM}$ in R66-b1=1(CLKM=1.0MHz), $8T_{CLKM} > 16T_{CLK}$
  - $16T_{CLK}$
* $16T_{CLK}+8T_{CLKM}$ is needed to access any register of R84~R87 continuously , except for accessing R84 then R85 , or R86 then R87 .
* (00 0000 0000 0000)b or (00 0000 0000 0001)b is inhibited .

Figure 4.2 (1) Register functions for MIDI-clock Timer

R14 DMR	W	MIDI real-time message control	
b7 ↘ b6			
b5 ASE	enable auto active-sence (FE)h output	0 : disable	1 : enable
b4 MCE	enable auto MIDI-clock (F8)h output	0 : disable	1 : enable
b3 CDE	enable MIDI-control (F9~FD)h detection	0 : disable	1 : enable
b2 MCDS	select MIDI-clock for distributor	0 : FIFO-IRx	1 : user
b1 MCFS ↘ b0	select MIDI-clock for FIFO-IRx	00 : (inhibited) 01 : MIDI message Detector 10 : SYNC Detector 11 : MIDI-clock Timer	

R15 DCR	W	MIDI real-time message request	
b7 Tx	target block of message		FIFO-ITx
b6 SYNC	write 1 to send message , but clock		SYNC Controller
b5 CC	(F8)h is always sent to all blocks .		Click Counter
b4 PC			Play-back Counter
b3 RC			Recording Counter
b2 ↘ b0	contents of message		000 : (F8)h - clock 001 : (F9)h 010 : (FA)h - start 011 : (FB)h - stop 100 : (FC)h - continue 101 : (FD)h 110 : (inhibited) 111 : (inhibited)

R16 DSR	R	FIFO-IRx data	
b7 ↘ b0	FIFO-IRx data		FIFO-IRx data (0 is set while empty)

R17 DNR	W	FIFO-IRx control	
b7 ↘ b1			
b0 CLK	FIFO-IRx increment clock		write 1 to increment FIFO-IRx

* $16T_{CLK}$ is needed to access any register of R14~R17 continuously

Figure 4.2 (2) Register functions for MIDI-clock Controller

- CLK (R7 – b0) – WRITE –

  If 1 is written into this bit an increment clock is sent to the FIFO-IRx, causing the oldest data to be lost.

* In the case of successive access to registers R14 through R17, inclusive, an interval of 16TCLK is necessary between each access.

4.1.9. Note on IRQ

If the receiver receives the system real-time messages (F9)h ~ (FD)h without parity and framing errors, this data is set in the FIFO-IRx via the MIDI message detector. If the FIFO-IRx has been empty the messages (F9)h ~ (FD)h can be regarded as its oldest data and hence the MIDI message detect interrupt is sent to the IRQ controller. This interrupt signal can be cleared only by the IRQ controller, independent of the increment of the FIFO-IRx. (If the IRQ is cleared the FIFO-IRx is not incremented and vice versa.) When, after the IRQ is cleared, the FIFO-IRx is incremented to cause its oldest data to become the (F9)h ~ (FD)h, an interrupt signal is sent to the IRQ controller.

According to the register setting by the host CPU, when the receiver receives the clock message (F8)h without error, the SYNC detector detects the SYNC signal of the tape SYNC input to the RxF terminal, or the MIDI clock timer detects the elapse of the prescribed time, the (F8)h is set in the FIFO-IRx. And, if the FIFO-IRx has been empty its oldest data can be considered the (F8)h and hence the MIDI clock detect interrupt signal is sent to the IRQ controller. At the same time the FIFO-IRx is incremented and becomes empty again. If the (F8)h is set in the FIFO-IRx under no empty condition the FIFO-IRx is incremented until the oldest data becomes the (F8)h and an interrupt signal is issued.

The simultaneous use of the MIDI clock detect interrupt and the CLICK counter interrupt is inhibited.

4.1.10. Effects of the initial

The initial clear of the hardware and software will cause the following operations to be made:

- Registers are written with zeros.
- FIFO-IRx is cleared.
- Interrupt signals are cleared.
- The value of the MIDI clock timer is not guaranteed.

4.2 Sequencer

4.2.1. Purpose of the sequencer

The two counters in the sequencer, the recording counter and the playback counter, are provided, as their names imply, to record and reproduce the MIDI information, etc. The record and reproduce (playback) here mean that the generation of a series of events is recorded with respect to timing and is reproduced without destroying the relative time relationship.

The two counters operate in synchronization with the internal MIDI clock; however, since the accuracy of the MIDI clock depends upon the ninety-sixth note, there is a tendency for the accuracy to become somewhat lower at very fast temps. To cope with this situation the MIDI clock interpolator is provided in which the count clock is generated at every 1/n-th point of the interval between any two successive internal MIDI clocks, thereby improving accuracy.

4.2.2. MIDI clock interpolator

The MIDI clock interpolator measures the generation interval of the internal MIDI clock originated from the MIDI message distributor, generates its count clock at 1/n-th of the measured interval and provides the recording and playback counters with the count clock generated. The integer(n) is referred to as the interpolation rate (Fig. 4.3 (a)).

The count clock generation interval is determined by counting the interval obtained by dividing the unit clock (CLK divided by 16), which equals to the interval between any MIDI clock and the next MIDI clock, by the interpolation rate. The resultant interval is further used until the next MIDI clock is reached. At the same time, the next count clock generation interval is measured. Therefore, if the MIDI clock generation interval varies or if the interpolation rate is changed, the count clock generation interval can follow this variation with a lag of one MIDI clock. In order to maintain the number of count clocks between the MIDI clocks the count clock generation higher than the interpolation rate is inhibited. Also, if the next MIDI clock generates before the interpolation rate is reached, a series of count clocks will generate to make up for a deficiency (Fig. 4.3 (b), (c)).

The recording and playback counters start and stop their operations according to the start (FA)h, stop (FB)h and continue (FC)h messages. If, however, the start and stop of the count is simply performed using the count clock described above, some problem is encountered. At the second and third stages in Fig. 4.3 (d) are given the count clocks with interpolation rates of 1 and 3, based upon the above procedure. Under the generating condition of the MIDI clock and the control messages as shown at the first stage, the count clock indicated with a circle is to be count. Then, the measured times interpolated using two interpolation rates are 2 for 2nd stage, 7/3 for 3rd stage, respectively during the first start-stop period and 2 for 2nd stage, 4/3 for 3rd stage, respectively during the next continue-stop period. This implies that if instruments with different interpolation rates are synchronized for recording and reproduction, repeated start/stop operation could bring poor synchronization. To cope with this, the start and stop of the count clock generation is done on the basis of the MIDI clock unit in accordance with each control message. And a function is added in which if the control message of stop (FB)h is sent count clocks are generated in a lump that are to be generated before the next MIDI clock. With this technique the counted value is fixed at the interpolation rate times the MIDI clock, thereby maintaining synchronization (Fig. 4.3 (d) – fourth stage). The control messages of start, stop and continue can be independently sent to the recording and playback counters; thus, the MIDI clock interpolator provides its count clock output separately to each counter, permitting independent operation.

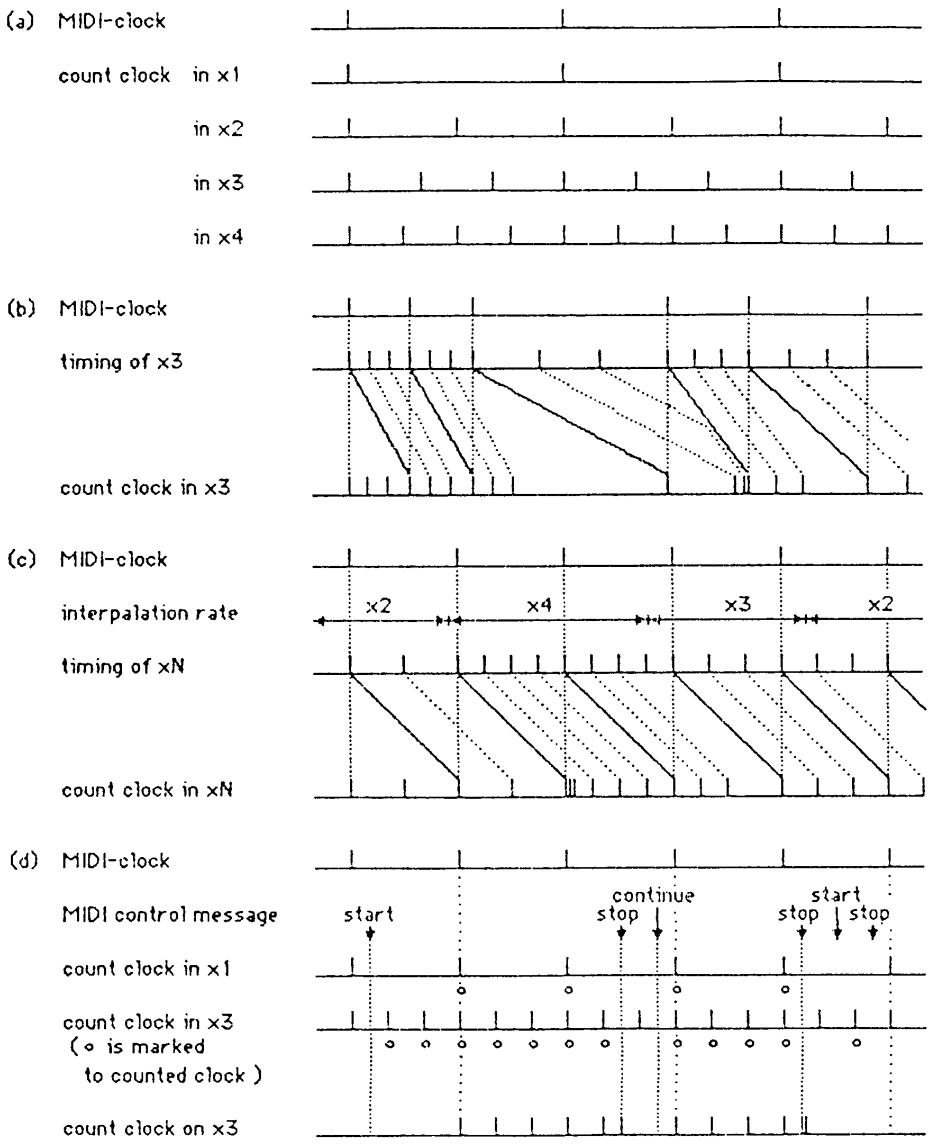


Figure 4.3 Interpolation of MIDI-clock

4.2.3. Recording counter.

8-bit read-only counter.

When the start message (FA)h is received the count value is set to 0, and the value increases one by one with the count clock from the MIDI clock interpolator. If the count clock is received at the count value of (FF)h the count value becomes 0, sending an interrupt signal to the IRQ controller.

This counter, as a real-time clock on the basis of the MIDI clock, is intended for time management such as for event recording, with its higher-order digits supplemented by software.

If the count carry is made by the IRQ and the count value is read out with the IRQ of the host CPU masked, the count value could become zero immediately after masking the IRQ. This raises the possibility of time recognition by pre-carry higher-order digits plus post-carry lower-order digits.

Also, the count value could become zero immediately after its read-out and the carry of the higher-order digits could be made before time recognition, raising the possibility of a pre-carry/lower order-post-carry/higher order combination. For this reason caution should be used for the host CPU process algorithm.

4.2.4. Playback counter

15-bit programmable subtracting-counter. A 16-bit (including one sign bit) arrangement internally. This counter has the function that the host CPU-designated value can be added to the count value. If the count value becomes zero or negative an interrupt signal is issued to the IRQ controller. If the start message (FA)h is received the count value is set to zero. Also, the register operation by the host CPU can cause this count value to become zero. If the count value zero-setting and addition are simultaneously performed, the zero-setting operation precedes the addition.

In the case of the reproduction of the generation of a series of events recorded on a relative time basis, an event is processed by the IRQ from this counter, then the time elapsed to the next event is added to the count value. The count proceeds with negative value; thus, even if the first event takes time in processing the time elapsed to the next event is not changed.

4.2.5. Register operations

Registers R74 through R77, inclusive, control each function of the sequencer. Fig. 4.4 shows the function of each register.

- R74 – b7 ~ b0 – READ –
  Reads the present count value of the recording counter.

- R76 – b7 ~ b0 and R77 – b6 ~ b0 – READ –
  The 15-bit value set in this register is added to the count value of the playback counter through the operation of ADD (R75 – b5).

- ADD (R75 – b5) – WRITE –
  If 1 is set to this bit the 15-bit value set in the R66 – R67 is added to the count value of the playback counter.

R74 SRR	R	Recording counter current value	
b7 ~ b0	current value		data

R75 SCR	W	Interpolator control	
b7 ~ b6			
b5 ADD	Play-back Counter addition request		write 1 to add
b4 CLR	Play-back Counter clear request		write 1 to clear
b3 ~ b0	interpolation rate		data --- (0000)b is inhibited

R76 SPRL	W	Play-back Counter value (H)	
b7 ~ b0	lower 8 bit data to add		data

R77 SPRH	W	Play-back Counter value (H)	
b7			
b6 ~ b0	higher 7 bit data to add		data

* 16T_{CLK} is needed to access any register of R74~R77 continuously , except for accessing R76 then R77 .

Figure 4.4 Register functions for Sequencer

- CLR (R75 – b4) – WRITE –
  If 1 is set to this bit 0 is set to the count value of the playback counter.

- R75 – 63 ~ b0 – WRITE –
  Sets the interpolation rate of the MIDI clock interpolator.

* In the case of the successive access to registers R74 through R77, inclusive, an interval of 16 TCLK is required between each access. This interval is not required when an access is first made to R76, then to R77.

4.2.6. Note on the IRQ

When the recording counter value changes from (FF)h to (00)h, an interrupt signal is sent to the IRQ controller. This signal is always cleared by the IRQ controller.

When the playback counter value is zero or negative an interrupt signal is sent to the IRQ controller. This signal is cleared by the IRQ controller only when the count value is positive.

4.2.7. Effects of the initial clear

The initial clear of the hardware and software will cause the following operations to be made:

- The values of the recording and playback counters become zero.
- Registers are written with zeros.
- The interpolation rate of the MIDI clock interpolator is not guaranteed.

4.3 SYNC controller and CLICK counter

4.3.1. SYNC controller

When the MIDI clock distributor generates the internal MIDI clock the output at the SYNC terminal is held at the H-level for 2 msec (equals to TCLKM x 2^{10} or TCLKM x 2^{11} depending upon the contents of the R66-b1: CLKM frequency select). If the internal MIDI clock is again generated at the prescribed time the output is set to the L-level after re-measuring the 2-msec period.

The operation of the SYNC controller is conducted from the receipt of the start (FA)h or continue (FC)h message from the MIDI clock distributor to the receipt of the stop (FB)h message.

This SYNC output signal is utilized as the input signal to the FSK modulator when the transmitter output is available at the TxD terminal. The TxF terminal can output the tape SYNC signal.

4.3.2. CLICK counter

7-bit internal MIDI clock counter provided to know the timing corresponding to the note value of a quarter note, etc., from the MIDI clock that indicates the note value of a ninety-sixth note. When this counter reaches a count of zero the count value is reloaded, an interrupt signal is sent to the IRQ controller and the output level at the CLICK terminal is held at the H-level for 2 msec (equals to TCLKM x 2^{10} or TCLKM x 2^{11} depending on the contents of the R66-b1: CLKM frequency select). If the count value again reaches zero when the output is still at the H-level, this output is set to the L-level after the remeasurement of 2 msec.

The counting operation is conducted from the receipt of the start (FA)h or continue (FC)h message from the MIDI clock distributor to the receipt of the stop (FB)h message. At the first MIDI clock after the receipt of the start (FA)h message the reloading of the count value, the output of the CLICK pulse and the generation of an interrupt signal. When the counter is loaded with a value by the host CPU operation the output of the CLICK pulse and the generation of an interrupt signal is performed. In this case the interrupt signal can be cleared after the elapse of 32 TCLKM from the writing of the value.

4.3.3. Register operations

Registers R66 and R67 control the operation of the SYNC controller and the CLICK counter. Fig. 4.5 shows the function of each register.

- CLKM (R66 – b1) – WRITE –

  Sets the frequency of the input clock at the CLKM pin. According to this setting, the count number of the CLKM, which determines the output pulse width at the SYNC and CLICK terminals, and the CLKM dividing ratio by which the count clock for the MIDI clock timer and general-purpose timer is produced are determined.

- OUTE (R66 – b0) – WRITE –

  The pulse output to the CLICK terminal is enabled/disabled.

- LD (R67 – b7) – WRITE –

  If 1 is written into this bit the simultaneously-written value b6 – b0 is loaded into the CLICK counter, providing the output of the CLICK pulse and the generation of an interrupt signal. The writing of 1 into this bit is inhibited when the value is 1.

- R67 – b6 ~ b0 – WRITE –

  Sets the count value for the CLICK counter load and reload. The operation is not guaranteed when 0 is written.

* Repeated access to register R67 requires an interval of 16 TCLK between each access.

R66 CCR	W	Click Counter control	
b7 ~ b2			
b1 CLKM	select CLKM frequency	0 : 0.5 MHz	1 : 1.0 MHz
b0 OUTE	enable CLICK output	0 : always L level	1 : click pulse enable

R67 CDR	W	Click Counter value	
b7 LD	imediate load request		write 1 to load imediately
b6 ~ b0	interval count data		7 bit data

* $16T_{CLK}$ is needed to access R67 continuously .

Figure 4.5 Register functions for SYNC Controller & Click Counter

4.4 Address hunter

4.4.1. System exclusive message

In the MIDI specification the system exclusive message for mass data transfer with unspecified formats is defined. Fig. 4.6 illustrates the message format. The (F0)h and the succeeding arbitrary-length data (00)h ~ (7F)h compose one system exclusive message. The system real-time messages (F8)h ~ (FF)h are sometime inserted into this message.

One byte next to the (F0)h is called the manufacturers ID code which is a registration code for a manufacturer using the MIDI standard. The MIDI-related equipment must accept only the system exclusive message using the ID code of a manufacturer who produced that equipment, and the mass data transfer using the manufacturer exclusive format can be made.

The manufacturer ID code is followed by a one-byte code tentatively called the device ID code which in the MIDI specification is intended for arbitrary data. If a certain manufacturer is producing several types of equipment in compliance with the MIDI. Each type is provided with an exclusive ID code. This device ID code is defined assuming the case in which only necessary or receivable information is to be processed.

4.4.2 Address hunter functions

When receiving the aforementioned system exclusive message the address hunter checks the manufacturer ID code and, if necessary, the device ID code, and prevents unnecessary data from being loaded into the FIFO-Rx for the purpose of reducing the host CPU processing.

Usually, the data received by the receiver is set in the FIFO-Rx via the MIDI clock filter and the address hunter. If the address hunter is enabled, and the (F0)h data is sent from the receiver to the address hunter, the BUSY flag is first set and this data is evacuated without being set in the FIFO-Rx. Next, when the data (00)h ~ (7F)h is received this is also evacuated (not set in the FIFO-Rx) as it is the manufacturer ID code. In case the operation mode requires the device ID code checking additional one-byte data (99)h ~ (7F)h is also evacuated. Upon completion of the evacuation of the necessary ID code(s) comparison is started between the necessary ID code and the registered one (pre-stored in the register by the host CPU).

If accorded, the evacuated (F0)h and ID code(s) are set in the FIFO-Rx. This completes internally the operation of the address hunter and all data thereafter are set in the FIFO-Rx. The BUSY flag still remains set.

If not accorded, the evacuated (F0)h and the ID code(s) and following data received while BUSY flag is set are abandoned without being set in the FIFO-Rx.

If the data (80)h ~ (F7)h is received when the BUSY flag is being set, the flag is cleared and the data received is set in the FIFO-Rx. However, if the data is (F0)h the process is repeated by first setting the BUSY flag.

The data (F8)h ~ (FF)h is regarded as the insertion of the system real-time message and always set to the FIFO-Rx. The data with the generation of parity and framing errors are all set in the FIFO-Rx and are not subject to the ID check and the start/stop check for the system exclusive message. These data received during the period from the normal (F0)h reception to the ID code check replace the evacuated data in order in the FIFO-Rx.

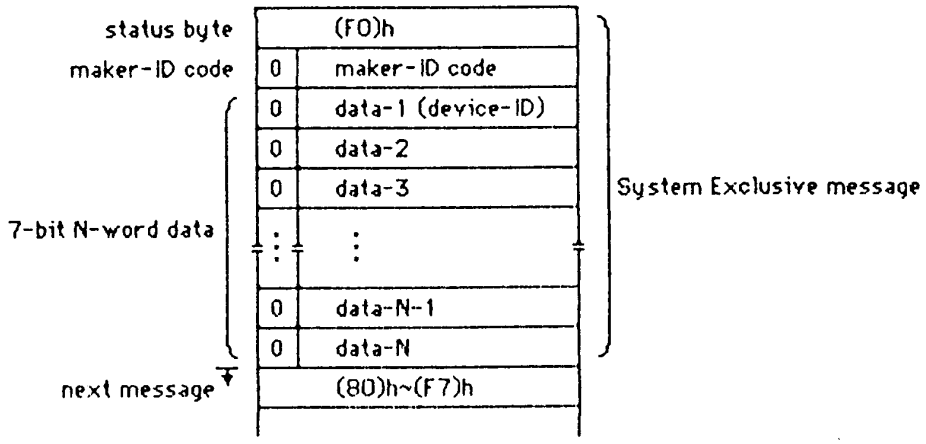


Figure 4.6 Format of System Exclusive message

4.4.3. Register operations

Registers R26 and R27 set the operation mode of the address hunter. Fig. 4.7 shows the function of each register. Since registers R34 and R35 also have the operation control bit for the address hunter, and hence described again.

- IDCL (R27 – b7) – WRITE –
  Determines whether the device ID code is to be checked.

- R26 – b6 ~ b0 – WRITE –
  Registers the manufacturer ID code.

- BRDE (R27 – b7) – WRITE –
  If 1 is set to this bit the (7F)h can be regarded as the device ID code in addition to the registered ID code, and the data that follows is set in the FIFO-Rx.

- R27 – b6 ~ b0 – WRITE –
  Registers the device ID code.

- AHBSY (R34 – b1) – READ –
  The BUSY flag for the address hunter.

- AHE (R35 – b1) – WRITE –
  The operation of the address hunter is enabled/disabled.

R26 AMR	W	Address-hunter control (1)	
b7 IDCL	select ID code length		0 : maker-ID only 1 : maker-ID & device-ID
b6 ~ b0	define maker-ID code		7 bit maker-ID code

R27 ADR	W	Address-hunter control (2)	
b7 BRDE	enable broadcast		0 : disable 1 : enable
b6 ~ b0	define device-ID code		7 bit device-ID code

Figure 4.7 Register Functions for Address-hunter

4.4.4. Effects of the initial clear

The initial clear of the hardware and software will cause the following operations to be made:

- Registers are written with zeros
- BUSY flag is reset.

4.5 Utilization for other than MIDI services

The functional blocks provided for the MIDI service can be utilized for systems not related to the MIDI, such as given below.

- Addition of a general-purpose timer

  The MIDI clock detect interrupt can simply be used as a timer interrupt by specifying the MIDI clock timer as the internal MIDI clock source.

- Long interval pulse output circuit

  A 2-msec width pulse output is available at a fixed time interval at the SYNC and CLICK terminals by specifying the MIDI clock timer as the internal MIDI clock source and sending the start message (FA)h to the SYNC controller and the CLICK counter.

  Also, using the mode in which the internal MIDI clock is generated by the host CPU operation a pulse output circuits is realized which automatically provides 2-msec width pulses by a single access.

- Use of the address hunter

  The address hunter could be used for specifications other than the MIDI by altering the format.

- Real-time clock

  The real-time clock can readily be implemented by operating the recording counter, with the MIDI clock timer used as the internal clock source.

- High-speed timer

  Using the playback counter makes possible to a certain extent the processing of short interval which software cannot cope with.

5. OTHER FUNCTIONS

5.1 General-purpose timer

5.1.1. Function

Independent 14-bit timer

A pulse signal with an interval of 8msec, which depending upon the R66–b1 (CLKM frequency select) is 1/4-th or 1/8-th of the CLKM, is counted with a 14-bit counter. Since the count is made only once every 16 TCLK, if 16 TCLK is longer than 4 TCLKM or 8 TCLKM the count unit is 16 TCLK.

When the counter reaches zero the reloading of the count value is performed and an interrupt signal is sent to the IRQ controller. With only the count value loading there is no generation of an interrupt signal.

If a value 0 or 1 is loaded into the counter its operation is not guaranteed.

5.1.2. Register operations

Registers R84 and R85 control the general-purpose timer. Fig. 5.1 shows the function of each register.

- R85–b5 ~ R84–b0 – WRITE –

  Sets the counter load value. With the standard CLKM input 8μsec is the unit.

- LD (R85–b7) – WRITE –

  If 1 is set to this bit loading of the counter is made regardless of the present count value. No interrupt signal is generated.

* For the registers R84 and R85, if repeated access to the same register is necessary the interval between each access should be 16 TCLK plus 8 TCLKM.

5.1.3. Effects of the initial clear

The initial clear of the hardware and software will cause the following operation to be made:

- Registers are written with zeros; thus, the generation timing of an interrupt signal is not guaranteed.

R84 GTRL	W	General Timer value (L)	
b7 ~ b0	lower 8 bit data		data (unit : 8 msec. IRQ occurs when count becomes 0 .)

R85 GTRH	W	General Timer value (H)	
b7 LD	imediate load request		write 1 to load value
b6			
b5 ~ b0	higher 6 bit data		data (unit : 8 x 256 msec.)

R86 MTRL	W	MIDI-clock Timer value (L)	
b7 ~ b0	lower 8 bit data		data (unit : 8 msec. IRQ occurs when data is loaded .)

R87 MTRH	W	MIDI-clock Timer value (H)	
b7 LD	imediate load request		write 1 to load value (data is loaded while $32T_{CLK}$ passed.)
b6			
b5 ~ b0	higher 6 bit data		data (unit : 8 x 256 msec.)

* count unit is ;
  - $4T_{CLKM}$ in R66-b1=0(CLKM=0.5MHz), $4T_{CLKM}$>$16T_{CLK}$
  - $8T_{CLKM}$ in R66-b1=1(CLKM=1.0MHz), $8T_{CLKM}$>$16T_{CLK}$
  - $16T_{CLK}$
* $16T_{CLK}+8T_{CLKM}$ is needed to access any register of R84~R87 continuously , excepi for accessing R84 then R85 , or R86 then R87 .
* (00 0000 0000 0000)b or (00 0000 0000 0001)b is inhibited .

Figure 5.1 Register functions for General Timer

5.2 I/O Controller

5.2.1. Functions

Controls the input/output of the eight I/O ports P0 ~ P7.

The input and output direction of each I/O port can be independently set. The port set as the input has a high impedance state; while the port set as the output has the H- or L-level, depending upon the output data set in the register.

The level at each I/O port terminal can be read into from the host CPU.

5.2.2. Register operations

Registers R94 through R96, control the I/O controller. Fig. 5.2 shows the function of each register.

- R94–b7 ~ b0 – WRITE –
  Sets the input and output direction of each I/O port.

- R95–b7 ~ b0 – WRITE –
  Output data for each I/O port is written.

- R96–b7 ~ b0 – READ –
  Level at each I/O port terminal is read out.

5.2.2. Effects of the initial clear

The initial clear of the hardware and software will cause the following operation to be made:

- Registers are written with zeros; thus, all I/O ports become high-impedance input ports.

R94 EDR	W	External I/O direction	
b7 ↓ b0	direction of earch I/O port	0 : input	1 : output

R95 EOR	W	External I/O output data	
b7 ↓ b0	output request of earch I/O port	0 : L level	1 : H level

R96 EIR	R	External I/O input data	
b7 ↓ b0	pin level of earch I/O port	0 : L level	1 : H level

Figure 5.2 Register functions for External I/O Controller

5.3 IRQ Controller

5.3.1. Functions

Receives ten kinds of interrupt signals from each functional block and controls the IRQ terminal output, the IRQ vector output and the release of interrupt signals.

The offline detector and the break detector have alternative interrupt signals; the interrupt signal from the CLICK counter and the MIDI clock detect interrupt signal are also alternative. These alternative interrupts and other interrupts are processed in the form of eight-level interrupts. The setting condition at each interrupt level, the clearing condition other than the operation of R03 and the effect of the R03 operation are given in Fig. 5.3.

IRQ level	Setting condition	Clear condition	Clear influence
IRQ-7 General Timer	When the timer reaches a count of zero.	———	———
IRQ-6 FIFO-Tx empty	When the FIFO-Tx becomes empty through the data extraction by the transmitter.	When the FIFO-Tx is load with data.	———
IRQ-5 FIFO-Rx ready	When the empty FIFO-Rx is loaded with data.	When the FIFO-Rx becomes empty.	———
IRQ-4 (1) Off-line detect	When the reception is not made for 300msec.	When 1 is written into RxOLC (R35-b2).	RxOL (R34-b2) is cleared.

* In addition, each IRQ is cleared by initial clear and by writing a value of 1 into the corresponding bit of R03.

Figure 5.3(a) Setting conditions, clearing conditions, clearing influence of each IRQ.

IRQ level	Setting condition	Clear condition	Clear influence
IRQ-4 (2) Break detect	When the receiver serial input is at the L-level for two-character period.	When 1 is written into the BRKC (R35-b3).	BRK (R34-b3) is cleared.
IRQ-3 Recording Counter	When the count value of the recording counter is zero.	———	———
IRQ-2 Play-back Counter	When the count value of the playback counter is zero or negative.	———	———
IRQ-1 (1) Click Counter	When its count value reaches zero in the case of being selected as the factor of the IRQ-1.	———	———
IRQ-1 (2) MIDI-clock detect	When the oldest data of the FIFO-IRx becomes the (F8)h in the case of being selected as the factor of the IRQ-1.	———	———
IRQ-0 MIDI real-time message detect	When the oldest data of the FIFO-IRx becomes the (F9)h ~ (FD)h.	———	———

* In addition, each IRQ is cleared by initial clear and by writing a value of 1 into the corresponding bit of R03.

Figure 5.3(b) Setting conditions, clearing conditions, clearing influence of each IRQ.

5.3.2. Register operations

The IRQ controller is controlled by registers R00, R02, R03, R04, R05 and R06. Fig. 5.4 shows the function of each register.

- R00 – WRITE –

  IRQ vectors are already set which will be sent to D7 ~ D0 during the vector read. Like the other registers its contents can be read out by the host CPU. Thus, if the host CPU has no IRQ vector processing function the use of this register can eliminate the need for polling in the IRQ processing.

  Of the interrupt signals enabled and active at present, the highest priority interrupt signal determines b4 ~ b1. If the enabled interrupt signals are not active, (1000)b is set. The vector offset values set in the R04 determine b7 ~ b5, and b0 is always zero.

- R02 – READ –

  The state of the interrupt signal at each level is already set.

- R03 – WRITE –

  Clears the interrupt signal at each level independently.

  The interrupt signals from the offline detector and the break detector, which are at the same level, are both cleared by this register. However, the interrupt signal from the CLICK counter and the MIDI clock detect interrupt signal from the FIFO-Rx, which are also at the same level, are subject to selection, i.e., only the presently selected signal is cleared. If the other interrupt is selected the status changes.

- R04–b7 ~ b5 – WRITE –

  Specifies the values b7 ~ b5 of the IRQ vectors.

- C/T (R05–b3) – WRITE –

  Selects between the interrupt signal from the CLICK counter and the MIDI clock detect interrupt signal from the FIFO-IRx. Since two flip-flops are used to retain the status of the interrupt signal separately, change of selection may bring status change.

- O/B (R05–b2) – WRITE –

  Selects between the interrupt signal from the break detector and that from the offline detector. Since a signal flip-flop is used to retain the interrupt signal status, status remains unchanged if change of selection is made.

- VE (R05–b1) – WRITE –

  The IRQ vector output is enabled/disabled.

- VM (R05–b0) – WRITE –

  If 0 is already written this register satisfies the IRQ vector output request only if its own IRQ output is active; if 1 is already written the register always provide the output of the IRQ vector.

- R06–b7 ~ b0 – WRITE –

  The internal interrupt signal at each level is enabled/disabled. The IRQ output becomes active only when the enabled-level interrupt signal is issued. The RIQ vector is set according to the enabled interrupt signal which is active at present.

R00 IVR	R	IRQ vector	
b7 ~ b5	IRQ vector offset		same to R04-b7~b5
b4 ~ b1	IRQ vector appairs according to the active, highest priority , and enabled IRQ . priority : IRQ-7 > IRQ-6 > ...		0000 : IRQ-0 MIDI real-time message detected (F9~FD) 0001 : IRQ-1 Click Counter / MIDI-clock detected 0010 : IRQ-2 Play-back counter 0011 : IRQ-3 Recording counter 0100 : IRQ-4 Off-line detected / break detected 0101 : IRQ-5 FIFO-Rx ready 0110 : IRQ-6 FIFO-Tx empty 0111 : IRQ-7 General Timer 1000 : caused by external requests (other than MCS itself)
b0			always 0

R02 ISR	R	IRQ status	
b7	IRQ status 0 : inactive 1 : active		IRQ-7 General Timer
b6			IRQ-6 FIFO-Tx empty
b5			IRQ-5 FIFO-Rx ready
b4			IRQ-4 Off-line detected / Break detected
b3			IRQ-3 Recording Counter
b2			IRQ-2 Play-back Counter
b1			IRQ-1 Click Counter / MIDI-clock detected
b0			IRQ-0 MIDI real-time message detected (F9~FD)

R03 ICR	W	IRQ clear request	
b7	IRQ clear request write 1 to clear earch IRQ		IRQ-7 General Timer
b6			IRQ-6 FIFO-Tx empty
b5			IRQ-5 FIFO-Rx ready
b4			IRQ-4 Off-line detected / Break detected
b3			IRQ-3 Recording Counter
b2			IRQ-2 Play-back Counter
b1			IRQ-1 Click Counter / MIDI-clock detected
b0			IRQ-0 MIDI real-time message detected (F9~FD)

Figure 5.4 Register functions for IRQ Controller (1)

R04 IOR	W	IRQ vector offset request	
b7 ~ b5	IRQ vector offset		IRQ vector offset for R00-b7~b5
b4 ~ b0			

R05 IMR	W	IRQ mode control	
b7 ~ b4			
b3 C/T	select IRQ-1 source	0 : Click Counter	1 : MIDI-clock detected
b2 O/B	select IRQ-4 source	0 : Off-line detected	1 : Break detected
b1 VE	enable vector output	0 : disable	1 : enable
b0 VM	select vector output timing	0 : while own $\overline{IRQ}$ pin active	1 : always

R06 IER	W	IRQ enable request	
b7	IRQ enable request 0 : IRQ-n disable 1 : IRQ-n enable to activate IRQ pin and to set IRQ vector .		IRQ-7 General Timer
b6			IRQ-6 FIFO-Tx empty
b5			IRQ-5 FIFO-Rx ready
b4			IRQ-4 Off-line detected / Break detected
b3			IRQ-3 Recording Counter
b2			IRQ-2 Play-back Counter
b1			IRQ-1 Click Counter / MIDI-clock detected
b0			IRQ-0 MIDI real-time message detected (F9~FD)

Figure 5.4 Register functions for IRQ Controller (2)

5.4 System control

5.4.1. Initial clear

The YM3523 initial clear (reset) is started by bringing the input at the $\overline{IC}$ terminal to the L-level or by writing a value of 1 to the bit 7 of the register R01. Because the initial clear operation is completed after a maximum of 32 TCLK it is necessary that returning the $\overline{IC}$ terminal input to the L-level or writing a value of 0 to the R01–b7 be made after that period.

The contents of the initial clear operation are included in the description of each functional block. They are summarized in the following:

- Registers other than the R01 are written with zeros.
- Registers that clear flags and IRQs are written with ones.
- The MIDI message distributor sends the stop message (FB)h to each functional block.
- The count values of counters and timers become zeros.
- The reload values of counters and timers become zeros and their operations are not guaranteed.
- Transmitter and receiver stop their receive/transmit operation and are in the operation inhibit state.
- FSK-modulator and -demodulators stop modulation and demodulation and are in the operation inhibit state.
- The timing counters of the offline, break and idle detectors are initialized.

5.4.2. Register group number designation

The registers of the YM3523 are each designated by their address numbers and group numbers. For register access the address numbers are designated by the input level at the terminals A2 ~ A0, but the group numbers must be pre-written into the R01–b3 ~ b0.

The group numbers, once written, are valid until they are re-written. The registers with the same group number and the registers R00 ~ R03 can be successively accessed without the group number re-designation. (Registers R00 ~ R03 can be accessed with only their address numbers, regardless of group numbers.)

5.4.3. Register operations

Fig. 5.5 shows the function of the register R01.

- IC (R01–b7) – WRITE –
  The YM3523 is initial cleared by first writing a value of 1, and after the elapse of 32 TCLK, writing a value of 0.

- R01–b3 ~ b0 – WRITE –
  Designates the register group numbers.

R01 RGR	W	system control	
b7 IC	initial clear request		write 1 , wait $32T_{CLK}$, write 0 to reset system
b6 ↓ b4			
b3 ↓ b0	register-group number		register-group number

Figure 5 5 Register functions for the system control

6. INTERFACES

6.1 Clock interface

6.1.1. CLK: System clock

As the system clock a clock frequency of not more than 4MHz should be provided. Since only the minimum setup times for the H and L levels and the maximum transfer time between the two levels are required, an exact 50% duty factor is not required. A CPU with a 2MHz clock rate can be used with the YM3523 by utilizing the CPU's 4MHz signal generated within the circuit for producing the exact duty factor of the CPU clock.

6.1.2. CLKM and CLKF: Communication rate generating clock

These signals are sampled on the rising edge of the system clock (CLK). Thus, if the frequency exactly 1/2-nd of the CLK is employed it must be synchronized with the CLK. If an asynchronous signal is provided to the CLK the setup time for the H and L levels should be selected to exceed one cycle of the CLK to ensure sampling.

6.2 CPU interface

6.2.1. Register read/write

Register read/write is performed by means of terminals D0 ~ D7, A0 ~ A2, $\overline{CS}$, $\overline{WR}$ and $\overline{RD}$.

D0 ~ D7	:	Three-state data bus
A0 ~ A2	:	Designates the register address number
$\overline{CS}$	:	Chip select signal to the YM3523
$\overline{RD}$	:	Read request signal
$\overline{WR}$	:	Write request signal

In the interior of the YM3523 $\overline{CS}$ is OR'ed with $\overline{RD}$ or $\overline{WR}$, and $\overline{CS \cdot RD}$ and $\overline{CS \cdot WR}$ signals activate its operation. Thus, the timing of $\overline{CS}$ and $\overline{RD}$ or of $\overline{CS}$ and $\overline{WR}$ can be freely decided. Since A0 ~ A2 must be setup before $\overline{CS \cdot RD}$ or $\overline{CS \cdot WR}$, the timing is important if a CPU with the control lines such as R/W is used.

The read/write operation is performed in accordance with the system clock CLK of the YM3523; thus, the WAIT circuit, etc., must be used to ensure timing when the CPU clock and the CLK are asynchronous.

6.2.2. $\overline{IRQ}$

The $\overline{IRQ}$ pin has an open drain output and the wried OR circuit can be configured using a few peripheral LSIs.

When the $\overline{VR}$ pin is made active, the contents of the register R00 (as an IRQ vector) are output to D0 ~ D7. According to the setting of the internal registers (R05, R06), they are output to D0 ~ D7 either regardless of $\overline{IRQ}$ status, or while only $\overline{IRQ}$ terminal is active. Orals according to the setting of the internal registers, the D0 ~ D7 may remain at the high impedance state.

6.2.3. $\overline{IC}$

The YM3523 is initialized by holding the input to the $\overline{IC}$ pin at the L level for 32 TCLK or more.

6.3 Transmit/receive interface

Since terminals TxD, TxF, RxD and RxF are set for interface at the TTL level, an appropriate circuit is required to provide interface to the level of the MIDI, RS232C and audio line.

ELECTRICAL CHARACTERISTICS

Absolute maximum ratings

Power supply voltage	− 0.3 − + 7.0	V
Input voltage	− 0.3 − V DD + 0.5	V
Operating ambient temperature	− 25 − 85	°C
Storage temperature	− 50 − 125	°C

Recommended usage conditions

Item	Symbol	Min	Standard	Max	Unit
Power supply voltage	V DD	4.75	5.0	5.25	V
Operating ambient temperature	T op	0		70	℃

Electrical characteristics

(A) D C characteristics

(Input terminal)

Item	Symbol & Condition	Min	Standard	Max	Unit
Low level input voltage	V IL	−0.3	−	0.8	V
High level output voltage	V IH	2.0	−	U DD+0.5	V
Input leakage current	I LI V I = 0～5 V (Except for the pins with pull-up registers)	−	−	10	μA
Pull-up register	R u ($\overline{TEST0}$ ～$\overline{TEST2}$, $\overline{IC}$)	100	−	1000	KΩ

(Output terminal)

Item	Symbol & Condition	Min	Standard	Max	Unit
Low level output voltage	V OL I OL = 2 mA (P0 ～P7, D0～D7, $\overline{IRQ}$)	V SS	−	0.4	V
Low level output voltage	V OL I OL = 1 mA (Output terminals other than shown above)	V SS	−	0.4	V
High level output voltage	V OH I OH = − 1 mA (Except for $\overline{IRQ}$)	4.0	−	V DD	V
Output leakage current	I OL V O = 0～5 V	−	−	10	μA

(Power supply terminal)

Item	Symbol	Condition	Min	Standard	Max	Unit
Power supply current	I DP	V DD = 5 V	–	6	10	mA

(Capacitance)

Item	Symbol	Condition	Min	Standard	Max	Unit
Input capacitance	C I	f = 1 MHz			10	p F
Output load capacitance	C L 1	P0～P7, D0 ～D7, $\overline{IRQ}$			100	p F
	C L 2	Output terminals other than shown above			50	p F

(B) AC characteristics

(CLK input)

Symbol	Item	Min	Max	Unit
T CC	CLK Cycle time	250	2500	n sec
T HC	CLK H level setup time	100	–	n sec
T LC	CLK L level setup time	100	–	n sec
T RC	CLK Build up time	–	20	n sec
T FC	CLK Release time	–	20	n sec

(CLK input and output pins) (CLICK, SYNC, TxF, TxD, $\overline{IRQ}$)

Symbol	Item	Min	Max	Unit
T DCC	CLK-CLKI Delay time	–	50	n sec
T DOC	CLK-output change over delay time	–	100	n sec

(Register write)

Symbol	Item	MIN	MAX	Unit
T SAW	Address setup time	20	–	n sec
T HAW	Address hold time	20	–	n sec
T WW	CS · WR Pulse width	100	–	n sec
T SDW	Data setup time	50	–	n sec
T HDW	Data hold time	20	–	n sec

(Register read-out)

Symbol	Item	MIN	MAX	Unit
T SAR	Address setup time	20	–	n sec
T HAR	Address hold time	20	–	n sec
T SRC	CS · WR Setup time*	0	–	n sec
T HRC	CS · WR Hold time*	150	–	n sec
T ADC	Data access time*	–	150	n sec
T HDR	Data hold time	5	–	n sec

* : Time to the CLK

(IRQ vector read-out)

Symbol	Item	MIN	MAX	Unit
T SVC	$\overline{VR}$ Setup time*	0	–	n sec
T HVC	$\overline{VR}$ Hold time*	150	–	n sec
T ADC	Data access time*	–	150	n sec
T HDV	Data hold time	5	–	n sec

* : Time corresponding to the CLK

(I/O port output)

Symbol	Item	MIN	MAX	Unit
T SCW	Clock setup time	70	–	n sec
T SWC	$\overline{CS} \cdot \overline{WR}$ Setup time	0	–	n sec
T APC	I/O port hold time	100	–	n sec

(I/O port input)

Symbol	Item	MIN	MAX	Unit
T SCR	Clock setup time	70	–	n sec
T SRC	$\overline{CS} \cdot \overline{RD}$ setup time	0	–	n sec
T SPC	I/O port setup time	0	–	n sec
T HPC	I/O port hold time	100	–	n sec

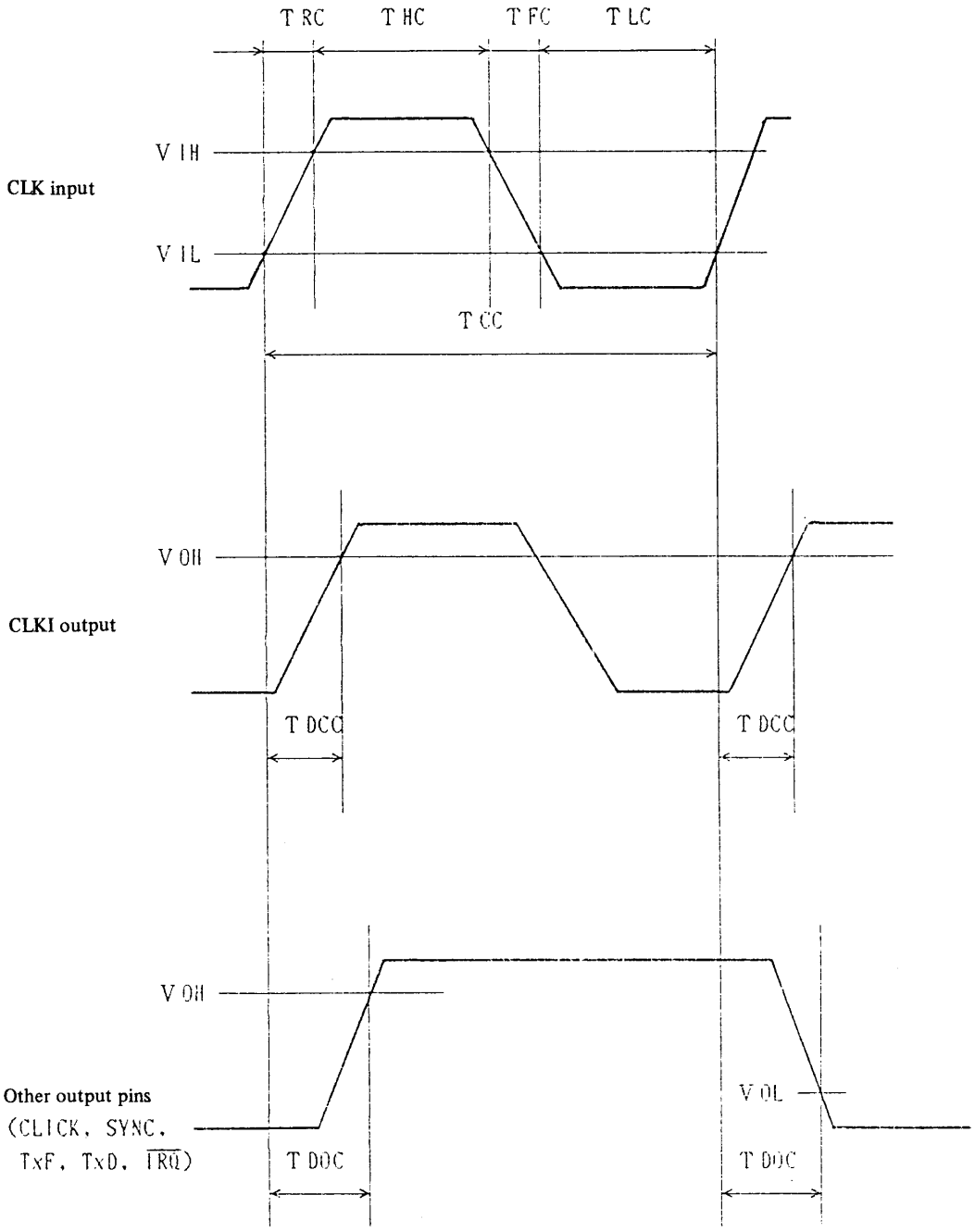
Terminal input (CLKM, CLKF, R x D, R x F, $\overline{IC}$)

Symbol	Item	MIN	MAX	Unit
T SIC	Input signal setup time	0	–	n sec
T HIC	Input signal hold time	70	–	n sec

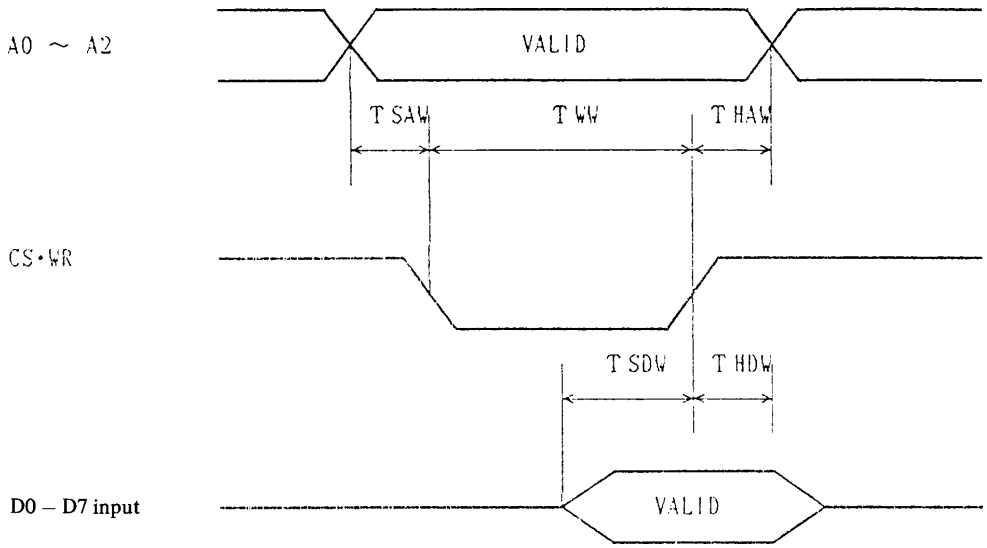
(Note 1) : Perfect operation of the $\overline{IC}$ requires sampling on the L level 32 times.

(Note 2) : At least one sampling is made if the setup period for the L-or H-level is T CLK or more.

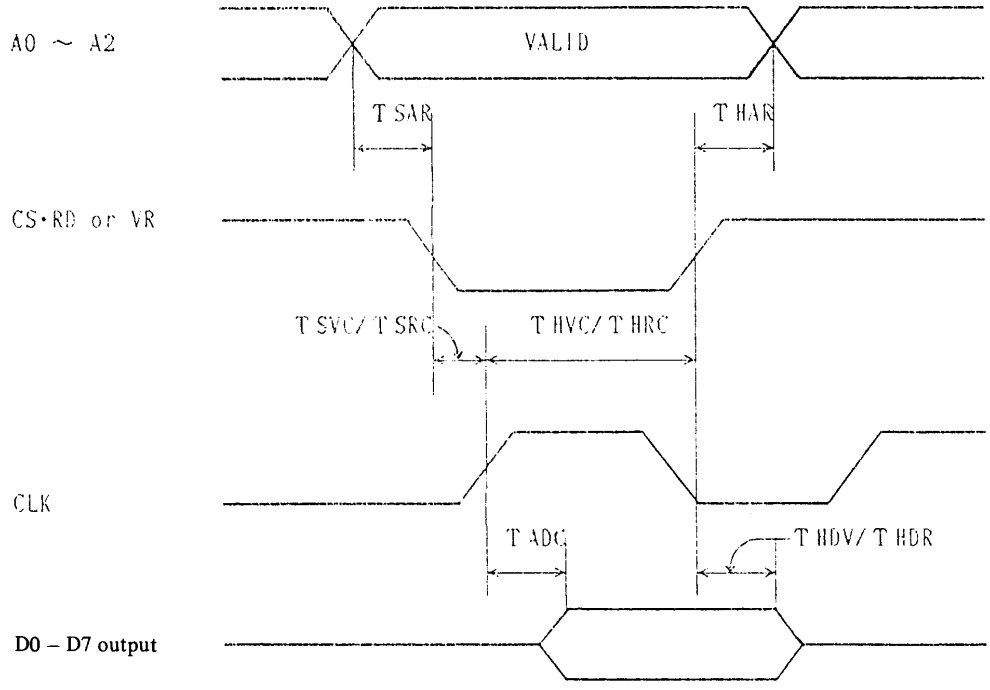
○ CLK input, CLKI output and other output pins



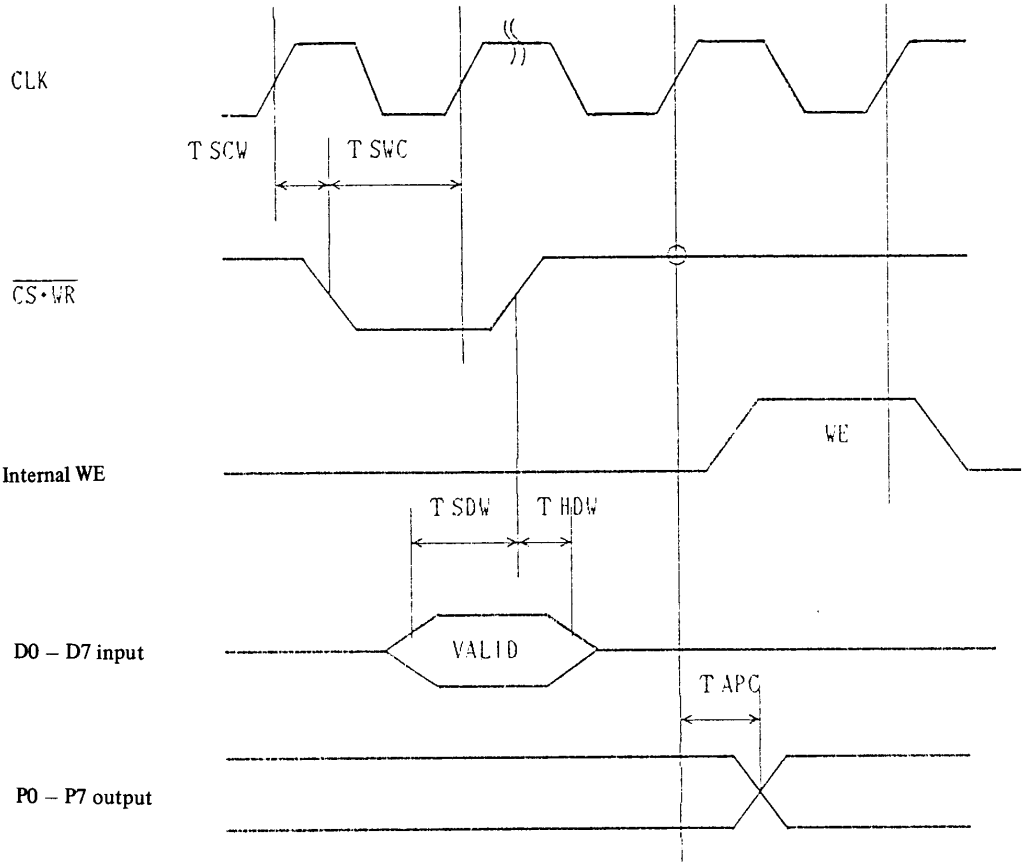
○ Register write



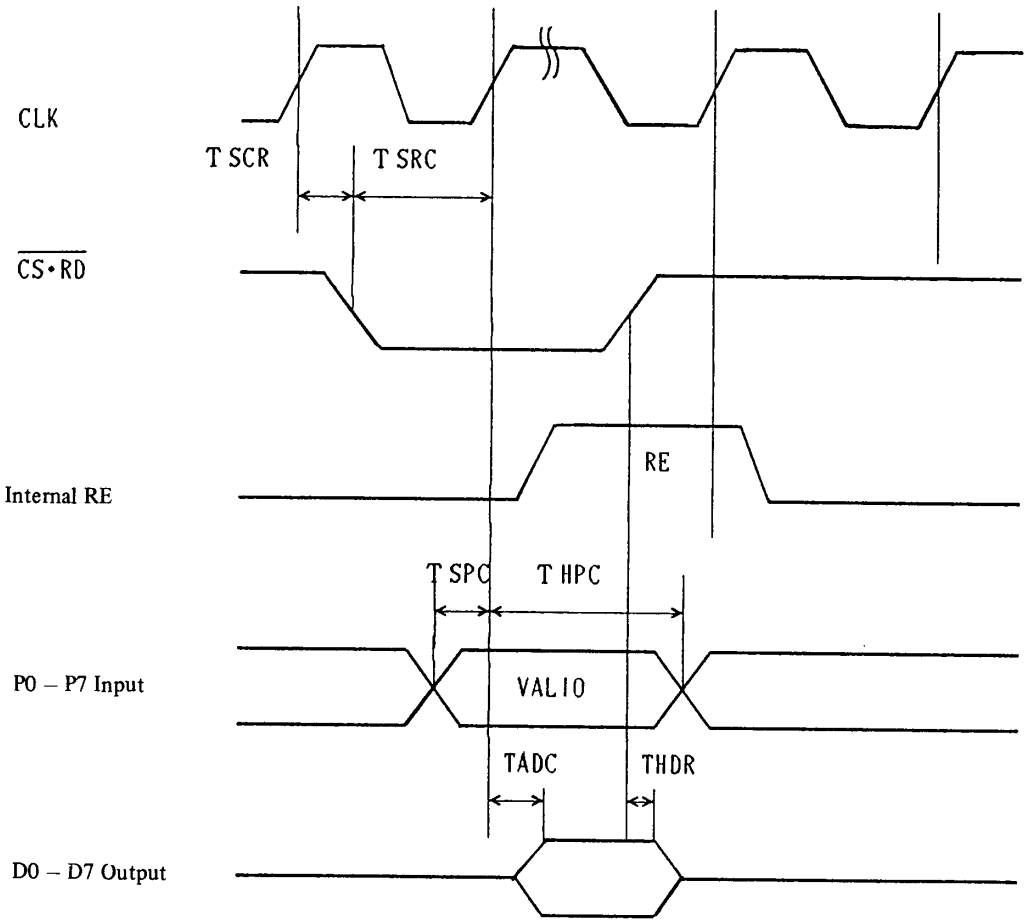
○ Register, vector read-out



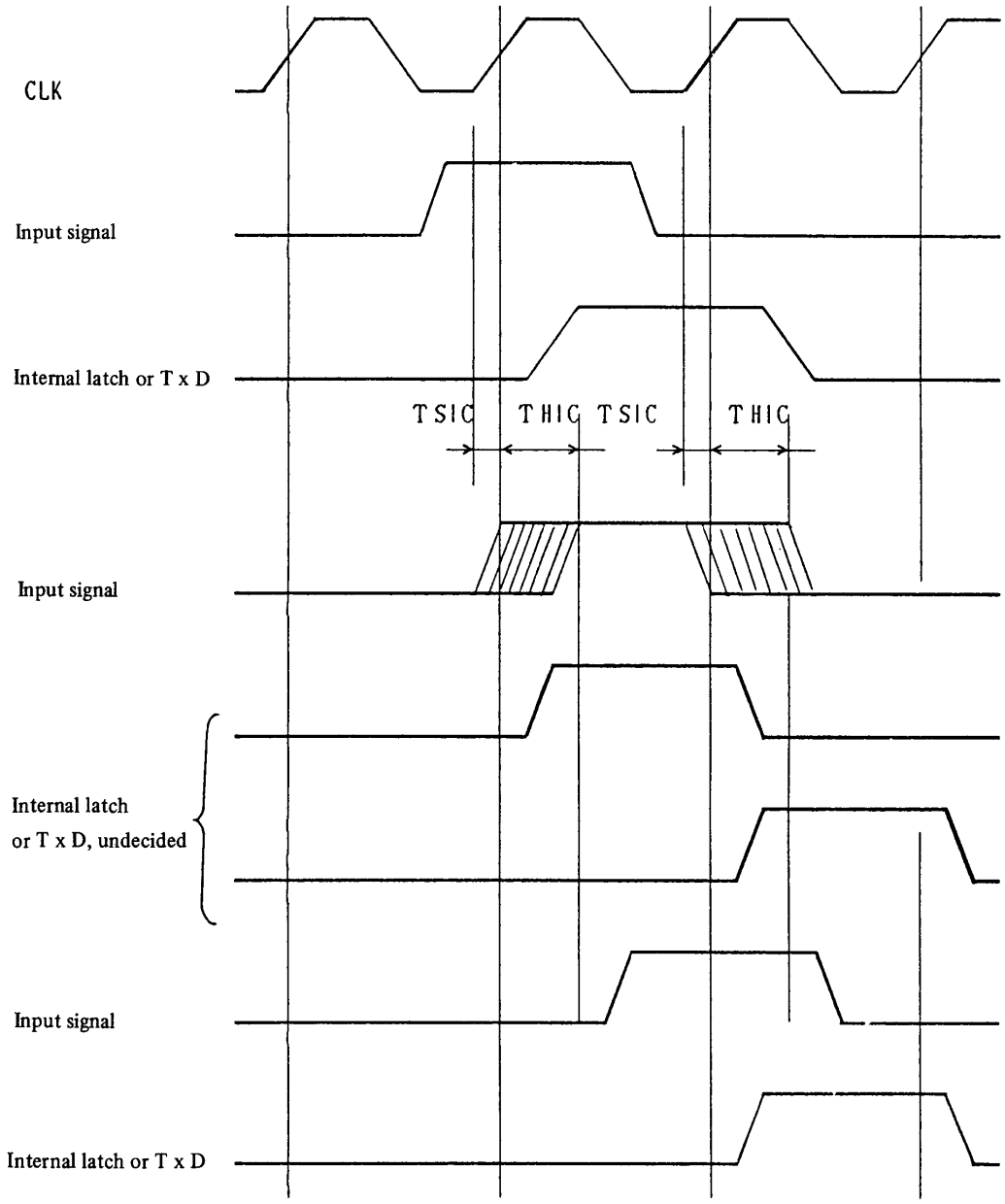
○ I/O port output



○ I/O port input

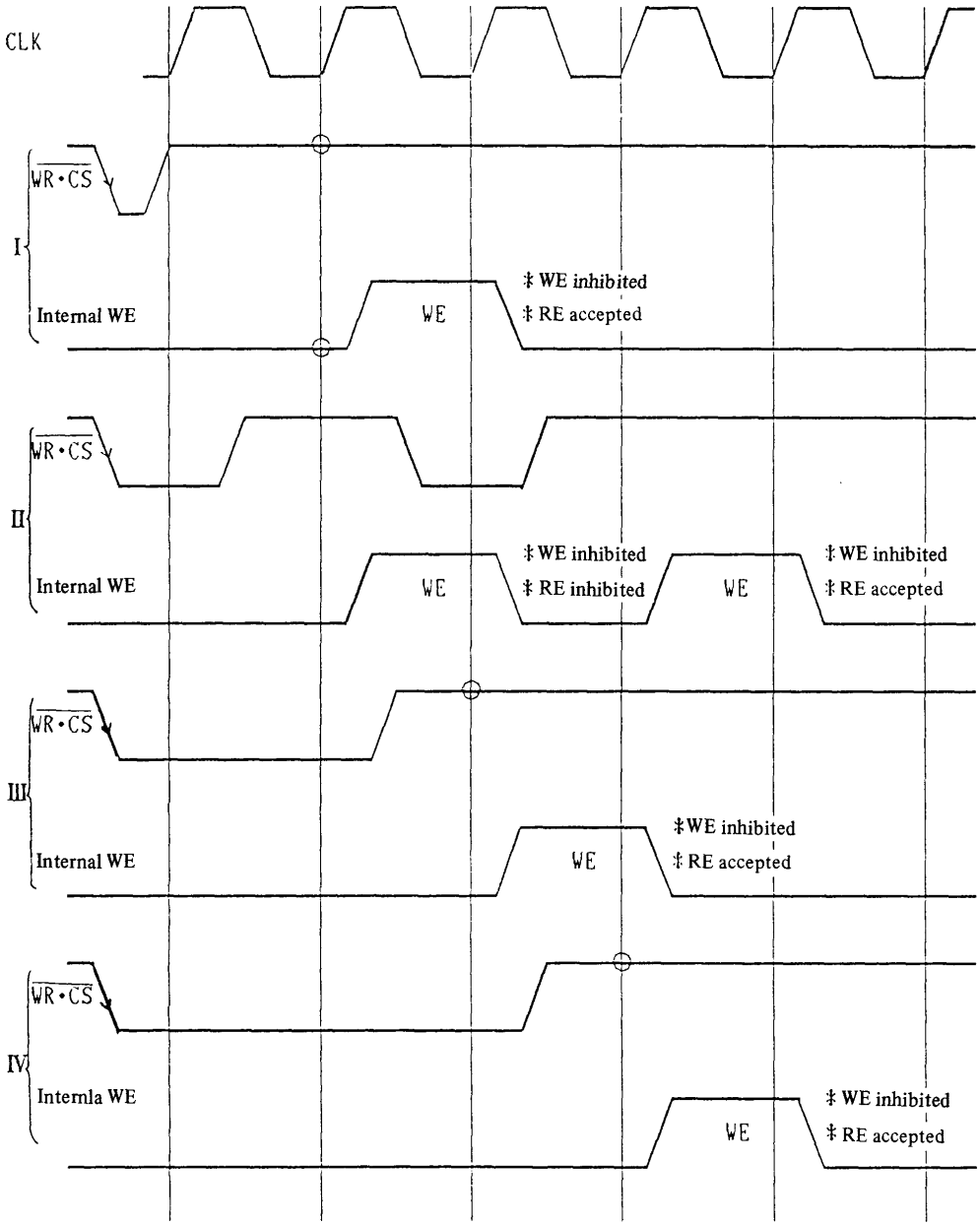


○ Internal input (CLKM, CLKF, R x D, R x F, IC)



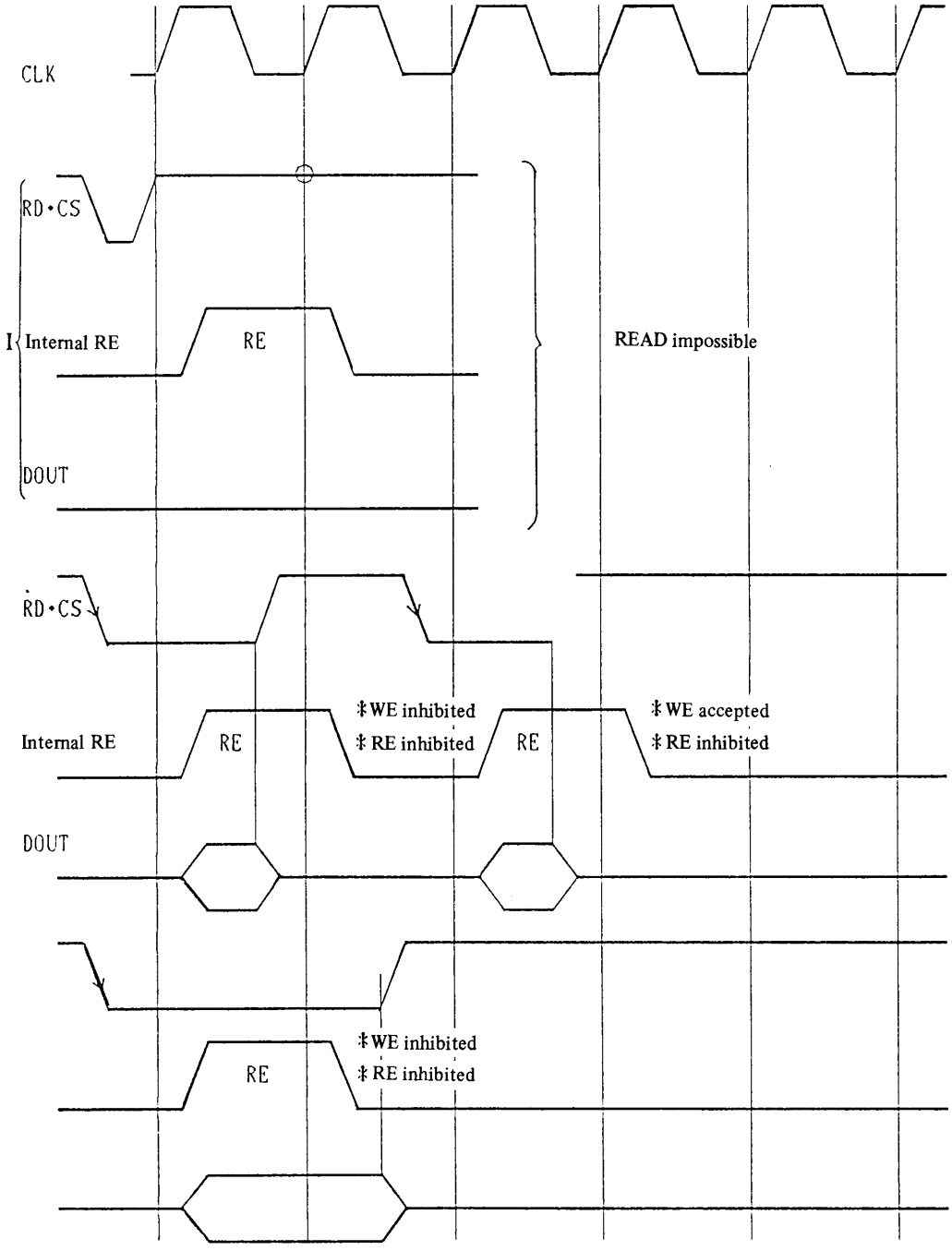
○ REGISTER WRITE operation

1. Relationship of $\overline{WR \cdot CS}$ and internal WE
2. Continuous WE inhibited
3. Continuous WE, RE

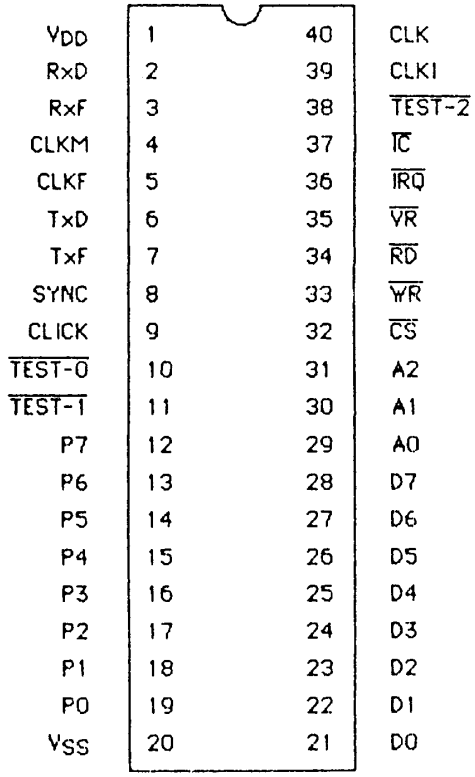


○ REGISTER READ operation

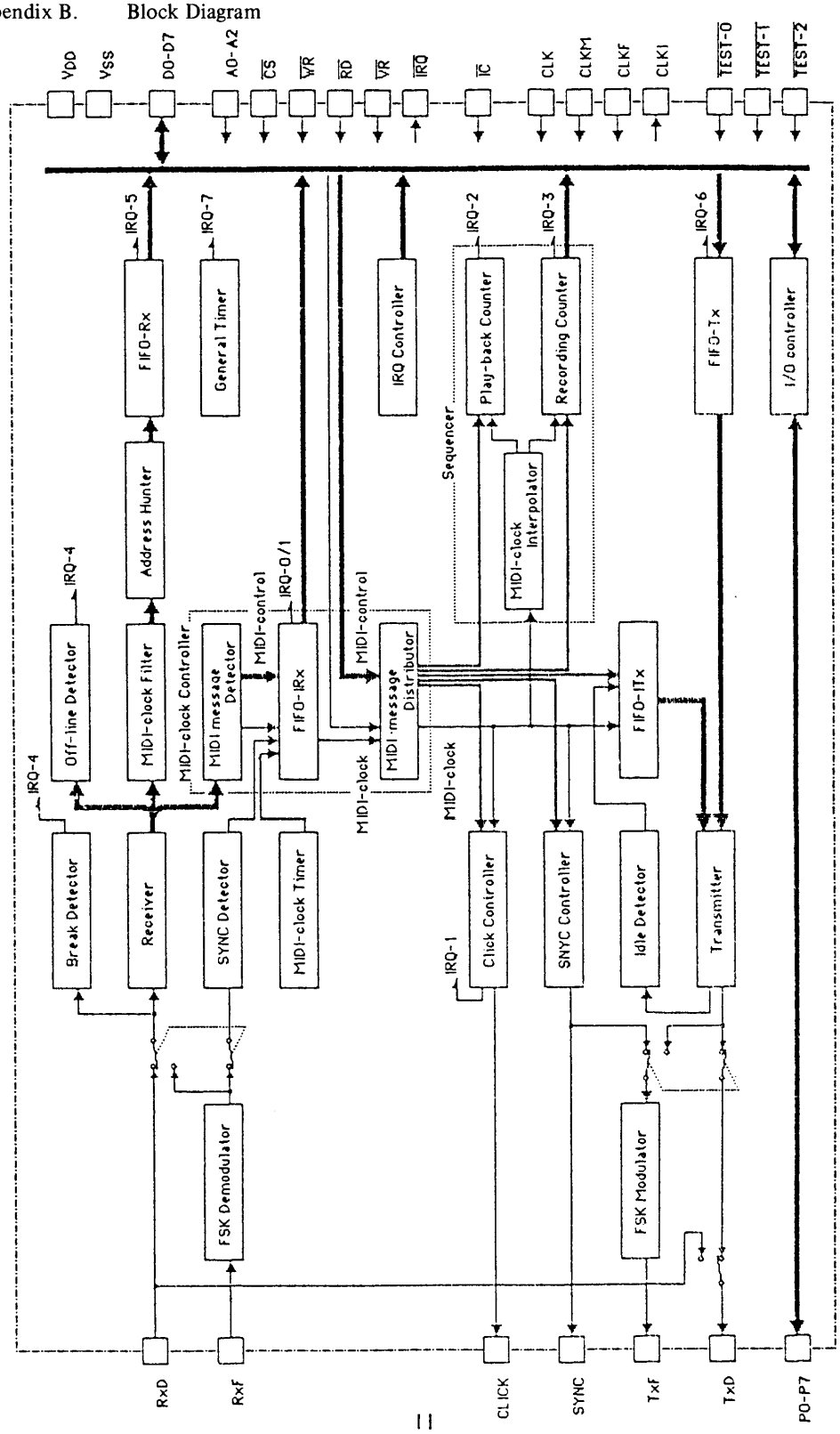
1. Relationship of $\overline{RD \cdot CS}$ and internal SE
2. Continuous RE inhibited
3. Continuous RE



Appendix A. Pin Configuration



Appendix B. Block Diagram



Appendix C. Register Map

register	R/W	content	b7	b6	b5	b4	b3	b2	b1	b0
R00 IVR	R	IRQ vector	IRQ vector							
R01 RGR	W	system control	IC				register group number			
R02 ISR	R	IRQ status	IRQ status							
R03 ICR	W	IRQ clear request	IRQ clear request							
R04 IDR	W	IRQ vector offset request	IRQ vector offset							
R05 IMR	W	IRQ mode control					C/T	O/B	VE	VM
R06 IER	W	IRQ enable request	IRQ enable							
R14 DMR	W	MIDI realtime message control			ASE	MCE	CDE	MCDS	MCFS	
R15 DCR	W	MIDI realtime message request	Tx	SYNC	CC	PC	RC	content of message		
R16 DSR	R	FIFO-IRx data	FIFO-IRx data							
R17 DNR	W	FIFO-IRx control								CLK
R24 RRR	W	Rx communication rate			RxD/F	Rx communication rate				
R25 RMR	W	Rx communication mode			RxCL	RxPE	RxPL	RxE/O	RxSL	RxST
R26 AMR	W	Address-hunter control	IDCL	maker ID code						
R27 ADR	W		BRDE	device ID code						
R34 RSR	R	FIFO-Rx status	RxRDY	RxOV	RxF	RxP	BRK	RxOL	AHBSY	RxBSY
R35 RCR	W	FIFO-Rx control	RxC	RxOVC		FLTE	BRKC	RxOLC	AHE	RxE
R36 RDR	R	FIFO-Rx data	FIFO-Rx data							
R44 TRR	W	Tx communication rate		TxRx	TxD/F	Tx communication rate				
R45 TMR	W	Tx communication mode			TxCL	TxPE	TxPL	TxE/O	TxSL	TxST
R54 TSR	R	FIFO-Tx status	TxEMP	TxRDY				TxIDL		TxBSY
R55 TCR	W	FIFO-Tx control	TxC				BLKE	TxIDLC		TxE
R56 TDR	W	FIFO-Tx data	FIFO-Tx data							
R64 FSR	R	FSK status	RxFS	SS	CSF	CFF			PS	PDF
R65 FCR	W	FSK control	ME			CFC	DE	APD	P/N	PDFC
R66 CCR	W	Click counter control							CLKM	OUTE
R67 CDR	W	Click counter load value	LD	7 bit data to load						
R74 SRR	R	Recording counter current value	8 bit data of current value							
R75 SCR	W	Sequencer control			ADD	CLR	4 bit interpolation rate			
R76 SPRL	W	Play-back counter value	lower 8 bit data to add							
R77 SPRH	W			higher 7 bit data to add						
R84 GTRL	W	General timer value	lower 8 bit data to load							
R85 GTRH	W		LD		higher 6 bit data to load					
R86 MTRL	W	MIDI-clock timer value	lower 8 bit data to load							
R87 MTRH	W		LD		higher 6 bit data to load					
R94 EDR	W	External I/O direction	direction of earch I/O port							
R95 EOR	W	External I/O output data	output request for I/O port							
R96 EIR	R	External I/O input data	pin level of earch I/O port							

A2,A1,A0 = 000	001	010	011	100	101	110	111
group-0	R00 IVR	R01 RGR	R02 ISR	R03 ICR	R04 IDR	R05 IMR	R06 IER	
1					R14 DMR	R15 DCR	R16 DSR	R17 DNR
2					R24 RRR	R25 RMR	R26 AMR	R27 ADR
3					R34 RSR	R35 RCR	R36 RDR	
4					R44 TRR	R45 TMR		
5					R54 TSR	R55 TCR	R56 TDR	
6					R64 FSR	R65 FCR	R66 CCR	R67 CDR
7					R74 SRR	R75 SCR	R76 SPRL	R77 SPRH
8					R84 GTRL	R85 GTRH	R86 MTRL	R87 MTRH
9					R94 EDR	R95 EOR	R96 EIR	

Appendix D. Table of Register Functions

R00 IVR	R	IRQ vector
b7 ~ b5	IRQ vector offset	same to R04-b7~b5
b4 ~ b1	IRQ vector appairs according to the active, highest priority , and enabled IRQ . priority : IRQ-7 > IRQ-6 > ...	0000 : IRQ-0 MIDI real-time message detected (F9~FD) 0001 : IRQ-1 Click Counter / MIDI-clock detected 0010 : IRQ-2 Play-back counter 0011 : IRQ-3 Recording counter 0100 : IRQ-4 Off-line detected / break detected 0101 : IRQ-5 FIFO-Rx ready 0110 : IRQ-6 FIFO-Tx empty 0111 : IRQ-7 General Timer 1000 : caused by external requests (other than MCS itself)
b0		always 0

R01 RGR	W	system control
b7 IC	initial clear request	write 1 , wait $32T_{CLK}$, write 0 to reset system
b6 ~ b4		
b3 ~ b0	register-group number	register-group number

R02 ISR	R	IRQ status
b7	IRQ status 0 : inactive 1 : active	IRQ-7 General Timer
b6		IRQ-6 FIFO-Tx empty
b5		IRQ-5 FIFO-Rx ready
b4		IRQ-4 Off-line detected / Break detected
b3		IRQ-3 Recording Counter
b2		IRQ-2 Play-back Counter
b1		IRQ-1 Click Counter / MIDI-clock detected
b0		IRQ-0 MIDI real-time message detected (F9~FD)

R03 ICR	W	IRQ clear request
b7	IRQ clear request write 1 to clear earch IRQ	IRQ-7 General Timer
b6		IRQ-6 FIFO-Tx empty
b5		IRQ-5 FIFO-Rx ready
b4		IRQ-4 Off-line detected / Break detected
b3		IRQ-3 Recording Counter
b2		IRQ-2 Play-back Counter
b1		IRQ-1 Click Counter / MIDI-clock detected
b0		IRQ-0 MIDI real-time message detected (F9~FD)

R04 IOR	W	IRQ vector offset request
b7 ~ b5	IRQ vector offset	IRQ vector offset for R00-b7~b5
b4 ~ b0		

R05 IMR	W	IRQ mode control		
b7 ~ b4				
b3 C/T	select IRQ-1 source	0 : Click Counter	1 : MIDI-clock detected
b2 O/B	select IRQ-4 source	0 : Off-line detected	1 : Break detected
b1 VE	enable vector output	0 : disable	1 : enable
b0 VM	select vector output timing	0 : while own $\overline{IRQ}$ pin active	1 : always

R06 IER	W	IRQ enable request
b7	IRQ enable request 0 : IRQ-n disable 1 : IRQ-n enable to activate IRQ pin and to set IRQ vector .	IRQ-7 General Timer
b6		IRQ-6 FIFO-Tx empty
b5		IRQ-5 FIFO-Rx ready
b4		IRQ-4 Off-line detected / Break detected
b3		IRQ-3 Recording Counter
b2		IRQ-2 Play-back Counter
b1		IRQ-1 Click Counter / MIDI-clock detected
b0		IRQ-0 MIDI real-time message detected (F9~FD)

R14 DMR	W	MIDI real-time message control	
b7 ~ b6			
b5 ASE	enable auto active-sence (FE)h output	0 : disable	1 : enable
b4 MCE	enable auto MIDI-clock (F8)h output	0 : disable	1 : enable
b3 CDE	enable MIDI-control (F9~FD)h detection	0 : disable	1 : enable
b2 MCDS	select MIDI-clock for distributor	0 : FIFO-IRx	1 : user
b1 ~ b0 MCFS	select MIDI-clock for FIFO-IRx	00 : (inhibited) 01 : MIDI message Detector 10 : SYNC Detector 11 : MIDI-clock Timer	

R15 DCR	W	MIDI real-time message request	
b7 Tx	terget block of message		FIFO-ITx
b6 SYNC	write 1 to send message , but clock		SYNC Controller
b5 CC	(F8)h is always sent to all blocks .		Click Counter
b4 PC			Play-back Counter
b3 RC			Recording Counter
b2 ~ b0	contents of message		000 : (F8)h - clock 001 : (F9)h 010 : (FA)h - start 011 : (FB)h - stop 100 : (FC)h - continue 101 : (FD)h 110 : (inhibited) 111 : (inhibited)

R16 DSR	R	FIFO-IRx data	
b7 ~ b0	FIFO-IRx data		FIFO-IRx data (0 is set while empty)

R17 DNR	W	FIFO-IRx control	
b7 ~ b1			
b0 CLK	FIFO-IRx increment clock		write 1 to increment FIFO-IRx

* $16T_{CLK}$ is needed to access any register of R14~R17 continuously .

R24 RRR	W	Rx communication rate	
b7 ~ b6			
b5 RxD/F	select Receiver input connection	0 : RxD	1 : FSK demodulator
b4 ~ b0	communication rate (in CLKM = 0.5 MHz CLKF = 614.4 KHz)	00xxx : CLKM/16 (31250 bps) --- inhibited while RxD/F = 1 01xxx : CLKM/32 (15625 bps) --- inhibited while RxD/F = 1 10xxx : CLKF/32 (19200 bps) 11000 : CLKF/64 (9600 bps) 11001 : CLKF/128 (4800 bps) 11010 : CLKF/256 (2400 bps) 11011 : CLKF/512 (1200 bps) 11100 : CLKF/1024 (600 bps) 11101 : CLKF/2048 (300 bps) 11110 : CLKF/4096 (150 bps) 11111 : CLKF/8192 (75 bps)	

R25 RMR	W	Rx communication mode	
b7 ~ b6			
b5 RxCL	select data bit length	0 : 8 bit	1 : 7 bit
b4 RxPE	enable parity-bit check	0 : disable	1 : enable
b3 RxPL	select parity-bit length	0 : 1 bit	1 : 4 bit
b2 RxE/O	select parity-bit polarity	0 : even	1 : odd
b1 RxSL	select stop-bit length	0 : 1 bit	1 : 2 bit
b0 RxST	select stop-bit type in 2 bit length	0 : HH	1 : LH

R26 AMR	W	Address-hunter control (1)	
b7 IDCL	select ID code length	0 : maker-ID only	1 : maker-ID & device-ID
b6 ↓ b0	define maker-ID code	7 bit maker-ID code	

R27 ADR	W	Address-hunter control (2)	
b7 BRDE	enable broadcast	0 : disable	1 : enable
b6 ↓ b0	define device-ID code	7 bit device-ID code	

R34 RSR	R	FIFO-Rx status		
b7	RxRDY	FIFO-Rx ready flag	0 : empty	1 : data ready
b6	RxOV	FIFO-Rx overflow detected flag	0 : -	1 : overflow detected
b5	RxF	framing error flag	0 : -	1 : framing error detected
b4	RxP	parity error flag	0 : -	1 : parity error detected
b3	BRK	break detected flag	0 : -	1 : break detected
b2	RxOL	off-line detected flag	0 : -	1 : off-line detected
b1	AHBSY	Address-hunter busy flag	0 : Address-hunter idle	1 : Address-hunter busy
b0	RxBSY	Receiver busy flag	0 : Receiver Idle	1 : Receiver busy

R35 RCR	W	FIFO-Rx control	
b7	RxC	clear FIFO-Rx	write 1 to clear FIFO-Rx
b6	RxOVC	clear overflow detected flag	write 1 to clear flag
b5			
b4	FLTE	enable MIDI-clock Filter	0 : disable 1 : enable
b3	BLKC	clear break detected flag	write 1 ti clear flag
b2	RxOLC	clear off-line detected flag	write 1 to clear flag.
b1	AHE	enable Address-hunter	0 : disable 1 : enable
b0	RxE	enable Receiver	0 : disable 1 : enable

R36 RDR	R	FIFO-Rx data
b7 ↓ b0	get data from FIFO-Rx & increment FIFO-Rx	8 bit data

R44 TRR	W	Tx communication rate	
b7			
b6	TxRx	select TxD connection	0 : transmitter 1 : RxD
b5	TxD/F	select transmitter output connection	0 : TxD 1 : FSK modulator
b4 ↓ b0	communication rate (in CLKM = 0.5 MHz CLKF = 614.4 KHz)	00xxx : CLKM/16 (31250 bps) --- inhibited while TxD/F = 1 01xxx : CLKM/32 (15625 bps) --- inhibited while TxD/F = 1 10xxx : CLKF/32 (19200 bps) 11000 : CLKF/64 (9600 bps) 11001 : CLKF/128 (4800 bps) 11010 : CLKF/256 (2400 bps) 11011 : CLKF/512 (1200 bps) 11100 : CLKF/1024 (600 bps) 11101 : CLKF/2048 (300 bps) 11110 : CLKF/4096 (150 bps) 11111 : CLKF/8192 (75 bps)	

R45 TMR	W	Tx communication mode	
b7 ↓ b6			
b5	TxCL	select data-bit length	0 : 8 bit 1 : 7 bit
b4	TxPE	enable parity-bit generation	0 : disable 1 : enable
b3	TxPL	select parity-bit length	0 : 1 bit 1 : 4 bit
b2	TxE/O	select parity-bit polarity	0 : even 1 : odd
b1	TxSL	select stop-bit length	0 : 1 bit 1 : 2 bit
b0	TxST	select stop-bit type in 2 bit length	0 : HH 1 : LH

R54 TSR	R	FIFO-Tx status	
b7 TxEMP	FIFO-Tx empty flag	0 : data exist	1 : empty
b6 TxRDY	FIFO-Tx ready flag	0 : FIFO-Tx full	1 : ready
b5 ~ b3			
b2 TxIDL	Tx idle detected flag	0 : -	1 : Tx idle detected
b1			
b0 TxBSY	Transmitter busy flag	0 : idle	1 : busy

R55 TCR	W	FIFO-Tx control	
b7 TxC	clear FIFO-Tx & FIFO-ITx	write 1 to clear FIFO-Tx & FIFO-ITx	
b6 ~ b4			
b3 BRKE	enable send break	0 : -	1 : set Tx output signal as L level
b2 TxIDLC	clear Tx idle flag	write 1 to clear flag	
b1			
b0 TxE	enable Transmitter	0 : disable	1 : enable

R56 TDR	W	FIFO-Tx data
b7 ~ b0	set data to FIFO-Tx	8 bit data

R64 FSR	R	FSK status	
b7 RxFS	RxF pin status	0 : L level	1 : H level
b6 SS	demodulated serial signal status	0 : space (L)	1 : mark (H)
b5 CSF	carrier slow detected flag	0 : -	1 : carrier slow detected
b4 CFF	carrier fast detected flag	0 : -	1 : carrier fast detected
b3 ~ b2			
b1 PS	polarity status	0 : positive	1 : negative
b0 PDF	polarity detected flag	0 : -	1 : polarity detected

R65 FCR	W	FSK control	
b7 ME	enable Modulator	0 : disable	1 : enable
b6 ~ b5			
b4 CFC	clear carrier S/F detected flag	write 1 to clear flag	
b3 DE	enable Demodulator	0 : disable	1 : enable
b2 APD	disable auto polarity detector	0 : enable	1 : disable
b1 P/N	set polarity by manual	0 : positive	1 : negative
b0 PDFC	clear polarity detected flag	write 1 to clear flag	

R66 CCR	W	Click Counter control	
b7 ~ b2			
b1 CLKM	select CLKM frequency	0 : 0.5 MHz	1 : 1.0 MHz
b0 OUTE	enable CLICK output	0 : always L level	1 : click pulse enable

R67 CDR	W	Click Counter value	
b7 LD	imediate load request	write 1 to load imediately	
b6 ~ b0	interval count data	7 bit data	

* $16T_{CLK}$ is needed to access R67 continuously .

R74 SRR	R	Recording counter current value	
b7 ~ b0	current value		data

R75 SCR	W	Interpolator control	
b7 ~ b6			
b5 ADD	Play-back Counter addition request		write 1 to add
b4 CLR	Play-back Counter clear request		write 1 to clear
b3 ~ b0	interpolation rate		data --- (0000)b is inhibited

R76 SPRL	W	Play-back Counter value (H)	
b7 ~ b0	lower 8 bit data to add		data

R77 SPRH	W	Play-back Counter value (H)	
b7			
b6 ~ b0	higher 7 bit data to add		data

* $16T_{CLK}$ is needed to access any register of R74~R77 continuously , except for accessing R76 then R77 .

R84 GTRL	W	General Timer value (L)	
b7 ~ b0	lower 8 bit data		data (unit : 8 msec. IRQ occurs when count becomes 0 .)

R85 GTRH	W	General Timer value (H)	
b7 LD	imediate load request		write 1 to load value
b6			
b5 ~ b0	higher 6 bit data		data (unit : 8 x 256 msec.)

R86 MTRL	W	MIDI-clock Timer value (L)	
b7 ~ b0	lower 8 bit data		data (unit : 8 msec. IRQ occurs when data is loaded .)

R87 MTRH	W	MIDI-clock Timer value (H)	
b7 LD	imediate load request		write 1 to load value (data is loaded while $32T_{CLK}$ passed.)
b6			
b5 ~ b0	higher 6 bit data		data (unit : 8 x 256 msec.)

* count unit is ;
  - $4T_{CLKM}$ in R66-b1=0(CLKM=0.5MHz), $4T_{CLKM}>16T_{CLK}$
  - $8T_{CLKM}$ in R66-b1=1(CLKM=1.0MHz), $8T_{CLKM}>16T_{CLK}$
  - $16T_{CLK}$
* $16T_{CLK}+8T_{CLKM}$ is needed to access any register of R84~R87 continuously , except for accessing R84 then R85 , or R86 then R87 .
* (00 0000 0000 0000)b or (00 0000 0000 0001)b is inhibited .

R94 EDR	W	External I/O direction		
b7 ~ b0	direction of earch I/O port		0 : input	1 : output

R95 EOR	W	External I/O output data		
b7 ~ b0	output request of earch I/O port		0 : L level	1 : H level

R96 EIR	R	External I/O input data		
b7 ~ b0	pin level of earch I/O port		0 : L level	1 : H level

Appendix E. Table of IRQ Setting/Cleaning Conditions

IRQ level	Setting condition	Clear condition	Clear influence
IRQ-7 General Timer	When the timer reaches a count of zero.	———	———
IRQ-6 FIFO-Tx empty	When the FIFO-Tx becomes empty through the data extraction by the transmitter.	When the FIFO-Tx is load with data.	———
IRQ-5 FIFO-Rx ready	When the empty FIFO-Rx is loaded with data.	When the FIFO-Rx becomes empty.	———
IRQ-4 (1) Off-line detect	When the reception is not made for 300msec.	When 1 is written into RxOLC (R35-b2).	RxOL (R34-b2) is cleared.
IRQ-4 (2) Break detect	When the receiver serial input is at the L-level for two-character period.	When 1 is written into the BRKC (R35-b3).	BRK (R34-b3) is cleared.
IRQ-3 Recording Counter	When the count value of the recording counter is zero.	———	———
IRQ-2 Play-back Counter	When the count value of the playback counter is zero or negative.	———	———

* In addition, each IRQ is cleared by initial clear and by writing a value of 1 into the corresponding bit of R03.

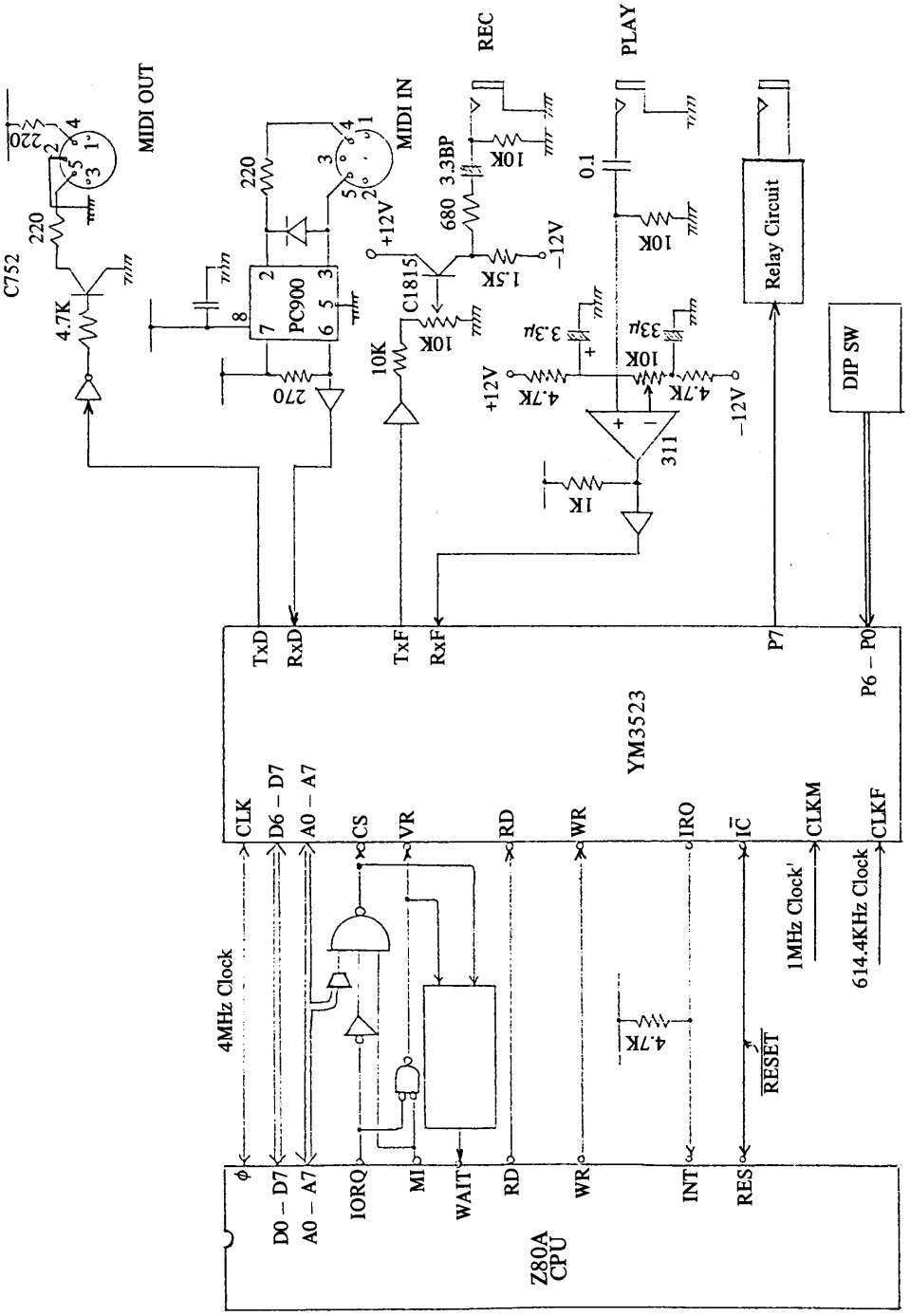
Setting conditions, clearing conditions, clearing influence of each IRQ.

IRQ level	Setting condition	Clear condition	Clear influence
IRQ-1 (1) Click Counter	When its count value reaches zero in the case of being selected as the factor of the IRQ-1.	———	———
IRQ-1 (2) MIDI-clock detect	When the oldest data of the FIFO-IRx becomes the (F8)h in the case of being selected as the factor of the IRQ-1.	———	———
IRQ-0 MIDI real-time message detect	When the oldest data of the FIFO-IRx becomes the (F9)h ~ (FD)h.	———	———

* In addition, each IRQ is cleared by initial clear and by writing a value of 1 into the corresponding bit of R03.

Setting conditions, clearing conditions, clearing influence of each IRQ.

Appendix F. Circuit Example(s)



Circuit Example using YM3523

```
MAIN       z80.i8080 ass.vr-1.2 23-DEC-84  17:57     PAGE  1

                        ;
                        ;-------------------------------------------------------------------------------
                        ;
                        ;       MCS (YM-3523) driver
                        ;
                        ;                                                       1984    (C) YAMAHA
                        ;
                        ;-------------------------------------------------------------------------------
                        ;
4000                    .=      4000h           ; program address
                        ;
      0F00              WORKAD  =       0F00h           ; working area address
                        ;
                        .radix  10
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       MCS interface
                        ;-------------------------------------------------------------------------------
                        ;
                        ;         - hook definition -
                        ;
                        ;       MVHIRQ  : IRQ from outside of MCS
                        ;                 This hook address is called by IRQ handler when IRQ of MCS is
                        ;                inactive .
                        ;
                        ;       MVHGTM  : timing of general timer
                        ;                 This hook address is called in earch interval time of general
                        ;                timer .
                        ;
                        ;       MVHMTM  : timing of MIDI-clock / click counter
                        ;                 This hook address is called in earch timing of MIDI-clock ,
                        ;                or of click counter . Trigger of hook call is defined by
                        ;                suroutine M.HOKS , so user must remenber it .
                        ;
                        ;       MVHPBC  : timing of play-back counter
                        ;                 This hook address is called when play-back counter becomes
                        ;                zero or minus value .  User must set next interval value in
                        ;                returnning parametor .
                        ;                 Returnning parametor is ;
                        ;                        <DE>    : next interval value
                        ;
                        ;       MVHO.B  : off-line/break detect
                        ;                 This hook address is called when off-line or break is detected .
                        ;                Trigger of hook call is defined by subroutine M.HOKS , so user
                        ;                must remenber it .
                        ;
                        ;       MVHSRM  : MIDI real-time message detect
                        ;                 This hook address is called when some of MIDI system real-time
                        ;                message ( F9h - FDh ) is received . MIDI-clock waits the return
                        ;                of this hook .
                        ;                 Calling parametor is ;
                        ;                        <D>     : message value.
                        ;
                        ;       MVHRCV  : receive buffer full
                        ;                 This hook address is called when receive buffer overflows .
```

```
                        ;               You must read every received data by M.RSRX before return .
                        ;
                        ;-------------------------------------------------------------------------------
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       label definition
                        ;-------------------------------------------------------------------------------
                        ;
                        ;       *** I/O address ***
                        ;
0000                    MIREG0  = 0             ; MCS register  R00     (IRQ vector register)
0001                    MIREG1  = 1             ;               R01     (system control register)
0002                    MIREG2  = 2             ;               R02     (IRQ status register)
0003                    MIREG3  = 3             ;               R03     (IRQ clear register)
0004                    MIREG4  = 4             ;               R04,R14, ...
0005                    MIREG5  = 5             ;               R05,R15, ...
0006                    MIREG6  = 6             ;               R06,R16, ...
0007                    MIREG7  = 7             ;               R17,R27, ...
                        ;
                        ;       *** working area ***
                        ;
0F00                    MVHOOK  = WORKAD        ; hook address table (14 byte)
0F00                    MVHIRQ  = MVHOOK        ;       IRQ from outside of mcs
0F02                    MVHGTM  = MVHOOK+2      ;       timing of general timer
0F04                    MVHMTM  = MVHOOK+4      ;       timing of MIDI-clock timer / click counter
0F06                    MVHPBC  = MVHOOK+6      ;       timing of play-back counter
0F08                    MVHO.B  = MVHOOK+8      ;       off-line/break detect
0F0A                    MVHSRM  = MVHOOK+10     ;       MIDI real-time message detect
0F0C                    MVHRCV  = MVHOOK+12     ;       receive buffer overflow
                        ;
0F0E                    MVRXBF  = MVHOOK+14     ; receive data buffer (256 byte)
100E                    MVSTBF  = MVRXBF+256    ;         status buffer (256 byte)
110E                    MVRXBI  = MVSTBF+256    ; buffer pointer for input
110F                    MVRXBO  = MVRXBI+1      ;                 for output
                        ;
1110                    MVRCLK  = MVRXBO+1      ; real-time clock counter (2 byte)
1112                    MVACFG  = MVRCLK+2      ; b7 : auto control enable flag by MIDI message
1113                    MVTXCN  = MVACFG+1      ; Tx control word save area
1114                    MVRXCN  = MVTXCN+1      ; Rx control word save area
                        ;
1115                    MVWEND  = MVRXCN+1      ; end of working area
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.INIT  : reset & set-up
                        ;                 This subroutine resets MCS and MCS driver module , set-up MCS
                        ;               driver module , and set I-register of host CPU (Z80) .
                        ;                 There is no calling parametor .
                        ;-------------------------------------------------------------------------------
                        ;
                        M.INIT:
4000  ED  57                    ld      a,i             ; save IFF status
4002  F5                        push    af
4003  F3                        di                      ; disable IRQ
```

```
                        ;
4004  3E 80               ld      a,80h           ; reset MCS
4006  D3 01               out     (MIREG1),a      ;    set IC (initial clear) bit of MCS
4008  00                  nop                     ;    wait 32Tclk
4009  00                  nop
400A  00                  nop
400B  00                  nop
400C  AF                  xor     a               ;    clear IC bit of MCS
400D  D3 01               out     (MIREG1),a
                        ;
400F  21 00 0F            ld      hl,MVHOOK       ; clear working area
4012  01 15 02            ld      bc,MVWEND-MVHOOK
4015  AF                  xor     a
4016  77           MLIN10: ld      (hl),a
4017  23                  inc     hl
4018  0B                  dec     bc
4019  10 FB               djnz    MLIN10
401B  11 00 0F            ld      de,MVHOOK       ; initialize hook address table
401E  21 77 40            ld      hl,MDIN12
4021  01 0C 00            ld      bc,12
4024  ED B0               ldir
4026  3E 41               ld      a,01000001b     ; initialize Rx & Tx control word
4028  32 14 11            ld      (MVRXCN),a
402B  32 13 11            ld      (MVTXCN),a
                        ;
402E  21 A0 40            ld      hl,MDIRQT       ; set IRQ vector table
4031  7C                  ld      a,h             ;    to I-register
4032  ED 47               ld      i,a
4034  3E 00               ld      a,0             ;    to R04 of MCS
4036  D3 01               out     (MIREG1),a
4038  7D                  ld      a,l
4039  D3 04               out     (MIREG4),a
403B  3E 01               ld      a,1             ; enable all IRQ by R16
403D  D3 01               out     (MIREG1),a
403F  3E FF               ld      a,11111111b
4041  D3 06               out     (MIREG6),a
4043  3E 06               ld      a,6             ; enable click counter & select 0.5MHz for CLKM
4045  D3 01               out     (MIREG1),a
4047  3E 01               ld      a,00000001b
4049  D3 06               out     (MIREG6),a
                        ;
404B  11 E8 03            ld      de,1000         ; set initial value
404E  CD 84 41            call    M.GTVL          ;       to general timer interval (8 msec)
4051  11 D0 07            ld      de,2000         ;       to MIDI-clock timer interval (16 msec)
4054  CD 96 41            call    M.MCVL
4057  16 18               ld      d,24            ;       to clock counter interval (MIDI-clock/24)
4059  CD A8 41            call    M.CCVL
405C  16 8C               ld      d,10001100b     ; start FSK modulator/demodulator
405E  CD 45 42            call    M.FSKS
4061  11 0E 0F            ld      de,MVRXBF       ; start Rx & Tx
4064  CD 1B 41            call    M.CMOD
4067  11 00 00            ld      de,0            ; initialize IRQ vector mode
406A  CD C3 40            call    M.HOKS
406D  16 00               ld      d,0             ; initialize control mode of real-time messages
406F  CD E8 40            call    M.CTLS
                        ;
4072  F1                  pop     af              ; return IFF status
```

```
                                ret     po
                                ei
                                ret                     ; return from subroutine
                        ;
                        ;       --- initial data of hook address table --
4076 C9                 MDIN10:  ret
4077 76 40 76 40        MDIN12: .word   MDIN10,MDIN10
407B 76 40 76 40                .word   MDIN10,MDIN10
407F 76 40 76 40                .word   MDIN10,MDIN10
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------
                        ;       M.IRQI  : entrance of IRQ
                        ;                 This is an entance of IRQ procedure while interrupt mode of
                        ;                 host CPU (Z80) is mode 1 .  In mode 2 , IRQ procedure is
                        ;                 automaticaly called according to IRQ procedures table .
                        ;                 Output of IRQ vector is not allowed for any peripheral circuit
                        ;                 but for MCS .  I-register of host CPU is set automatically by
                        ;                 M.INIT but IFF gets no effect by this module , so user must set
                        ;                 interrupt mode by himself .
                        ;                  There is no calling parametor .
                        ;-------------------------------------------------------------------------
                        ;
                        M.IRQI:
4083 F3                         di                      ; disable interrupt
                        ;
4084 E5                         push    hl              ; save registers
4085 F5                         push    af
                        ;
4086 DB 00                      in      a,(MIREGO)      ; read IRQ vector of MCS
                        ;
4088 21 A0 40                   ld      hl,MDIRQT       ; calculate procedure address
408B 6F                         ld      l,a
                        ;
408C F1                         pop     af              ; return registers
408D E3                         ex      (sp),hl         ;    save procedure address of terget IRQ
                        ;
408E C9                         ret                     ; jump to procedure of terget IRQ
                        ;
                        ;       --- IRQ vector table ---
                        ;
40A0                            .= .+31/32*32   ; set location counter as lower 5 bit is 0 .
                        MDIRQT:
40A0 57 42                      .word   MSIRQ0          ; address of IRQ-0 manager
40A2 91 42                      .word   MSIRQ1          ; address of IRQ-1 manager
40A4 A1 42                      .word   MSIRQ2          ; address of IRQ-2 manager
40A6 C2 42                      .word   MSIRQ3          ; address of IRQ-3 manager
40A8 D1 42                      .word   MSIRQ4          ; address of IRQ-4 manager
40AA E1 42                      .word   MSIRQ5          ; address of IRQ-5 manager
40AC 1D 43                      .word   MSIRQ6          ; address of IRQ-6 manager
40AE 25 43                      .word   MSIRQ7          ; address of IRQ-7 manager
40B0 35 43                      .word   MSIRQX          ; address of external IRQ manager
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------
```

```
                        ;       M.HOOK  : set hook address table
                        ;                 This subroutine defines earch hook address .
                        ;                 Calling parametors are ;
                        ;                         <DE>    : address table of calling parametor block
                        ;                         (DE)    :       IRQ from outside of MCS
                        ;                         (DE)+2  :       timing of general timer
                        ;                         (DE)+4  :       timing of MIDI-clock / click counter
                        ;                         (DE)+6  :       timing of play-back counter
                        ;                         (DE)+8  :       off-line/break detect
                        ;                         (DE)+10 :       MIDI real-time message detect
                        ;                         (DE)+12 :       receive buffer full
                        ;-------------------------------------------------------------------------------
                        ;
                        M.HOOK:
40B2  ED  57                    ld      a,i             ; save IFF status
40B4  F5                        push    af
40B5  F3                        di                      ; disable IRQ
                        ;
40B6  EB                        ex      de,hl           ; transfer new hook address
40B7  11  00  0F                ld      de,MVHOOK       ;       to hook address table
40BA  01  0C  00                ld      bc,12
40BD  ED  B0                    ldir
                        ;
40BF  F1                        pop     af              ; return IFF status
40C0  E0                        ret     po
40C1  FB                        ei
40C2  C9                        ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.HOKS  : select hook source
                        ;                 This subroutine selects hook source of off-line/break
                        ;                detection and MIDI-clock timing / click counter .
                        ;                 Calling parametors are ;
                        ;                         <D> = 0 : off-line , others : break.
                        ;                         <E> = 0 : click counter , others : MIDI-clock timing.
                        ;-------------------------------------------------------------------------------
                        ;
                        M.HOKS:
40C3  ED  57                    ld      a,i             ; save IFF status
40C5  F5                        push    af
40C6  F3                        di                      ; disable IRQ
                        ;
40C7  0E  03                    ld      c,00000011b     ; enable IRQ vector & set vector mode
                        ;
40C9  7A                        ld      a,d             ; set off-line/break select bit
40CA  B7                        or      a
40CB  28  04                    jr      z,MLHS10
40CD  79                        ld      a,c
40CE  F6  04                    or      00000100b
40D0  4F                        ld      c,a
                        MLHS10:
40D1  7B                        ld      a,e             ; set MIDI-clock/click select bit
40D2  B7                        or      a
40D3  28  04                    jr      z,MLHS20
40D5  79                        ld      a,c
```

```
                        40D6  F6  08                or      00001000b
                        40D8  4F                    ld      c,a
                                              MLHS20:
                        40D9  3E  00                ld      a,0             ; set data to register R05
                        40DB  D3  01                out     (MIREG1),a
                        40DD  79                    ld      a,c
                        40DE  D3  05                out     (MIREG5),a
                                              ;
                        40E0  3E  12                ld      a,00010010b     ; clear break/off-line & MIDI-clock/click IRQ
                        40E2  D3  03                out     (MIREG3),a
                                              ;
                        40E4  F1                    pop     af              ; return IFF status
                        40E5  E0                    ret     po
                        40E6  FB                    ei
                        40E7  C9                    ret                     ; return from subroutine
                                              ;
                                              ;       .page
                                              ;
                                              ;-------------------------------------------------------------------------------
                                              ;       M.CTLS  : select control source
                                              ;                 This subroutine selects control source of function blocks
                                              ;                controllable by MIDI system real-time messages .
                                              ;                 Calling parametor is ;
                                              ;                        <D> = 0------- : other control is sent by user,
                                              ;                              1------- :                  by driver,
                                              ;                              --0----- : active sence is ignored,
                                              ;                              --1----- :              is sent automatically,
                                              ;                              -----0-- : MIDI-clock is sent automatially,
                                              ;                              -----1-- :                  by user,
                                              ;                              ------01 : MIDI-clock comes from receiver,
                                              ;                              ------10 :                  sync detector,
                                              ;                              ------11 :                  midi-clock timer.
                                              ;-------------------------------------------------------------------------------
                                              ;
                                              M.CTLS:
                        40E8  ED  57                ld      a,i             ; save IFF status
                        40EA  F5                    push    af
                        40EB  F3                    di                      ; disable IRQ
                                              ;
                        40EC  7A                    ld      a,d             ; set auto controal enable flag
                        40ED  32  12  11            ld      (MVACFG),a
                                              ;
                        40F0  E6  3F                and     00111111b       ; set register R14
                        40F2  F6  18                or      00011000b       ;    enable auto F8h transfer
                        40F4  47                    ld      b,a             ;         & control message detection
                        40F5  3E  01                ld      a,1
                        40F7  D3  01                out     (MIREG1),a
                        40F9  78                    ld      a,b
                        40FA  D3  04                out     (MIREG4),a
                                              ;
                        40FC  F1                    pop     af              ; return IFF status
                        40FD  E0                    ret     po
                        40FE  FB                    ei
                        40FF  C9                    ret                     ; return from subroutine
                                              ;
                                              ;       .page
                                              ;
```

```
                        ;-------------------------------------------------------------------------------
                        ;       M.CNTL  : send control message
                        ;                 This subroutine sends control message by user , while user
                        ;                control is selected by subroutine M.CTLS , to earch function
                        ;                blocks controllable by MIDI system real-time messages .
                        ;                 Calling parametor is ;
                        ;                        <D> = *------- : write 1 to send message to FIFO-ITx,
                        ;                              -*------ :                 to SYNC controller,
                        ;                              --*----- :                 to click counter,
                        ;                              ---*---- :                 to play-back counter,
                        ;                              ----*--- :                 to recording counter,
                        ;                              -----000 : send F8h (MIDI-clock)  to all,
                        ;                              -----001 : send F9h               to terget,
                        ;                              -----010 : send FAh (start)       to terget,
                        ;                              -----011 : send FBh (stop)        to terget,
                        ;                              -----100 : send FCh (continue)    to terget,
                        ;                              -----101 : send FDh               to terget,
                        ;-------------------------------------------------------------------------------
                        ;
                        M.CNTL:
4100  ED  57                    ld      a,i             ; save IFF status
4102  F5                        push    af
4103  F3                        di                      ; disable IRQ
                        ;
4104  3E  01                    ld      a,1             ; send control by setting register R15
4106  D3  01                    out     (MIREG1),a
4108  7A                        ld      a,d
4109  D3  05                    out     (MIREG5),a
                        ;
410B  E6  0F                    and     00001111b       ; check message received by recording counter
410D  FE  0A                    cp      00001010b       ;  if 'start' then do ...
410F  20  06                    jr      nz,MLCN10
4111  21  00  00                ld      hl,0            ;    clear real-time clock value
4114  22  10  11                ld      (MVRCLK),hl
                        MLCN10:
4117  F1                        pop     af              ; return IFF status
4118  E0                        ret     po
4119  FB                        ei
411A  C9                        ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.CMOD  : set-up cmmunication mode
                        ;                 This subroutine sets communication rate & mode of transmitter
                        ;                & receiver , selects TxD OR TxF , RxD OR RxF as communication
                        ;                port , and enables function of address hunter and MIDI-clock
                        ;                filter .
                        ;                 Calling parametors are :
                        ;                        <DE>    : address of calling parametor block.
                        ;                        (DE)    :       transfer rate & connection.
                        ;                        (DE)+1  :                 mode.
                        ;                        (DE)+2  :       receiver rate & connection.
                        ;                        (DE)+3  :                 mode.
                        ;                            above prametors format is same to MCS registers
                        ;                        (DE)+4  ---*---- : 1 then enable MIDI-clock filter.
                        ;                                ------*- :              address hunter.
```

```
                        ;-------------------------------------------------------------------------------
                        ;
                        M.CMOD:
4116 ED 57                      ld      a,i             ; save IFF status
4118 F5                         push    af
4119 F3                         di                      ; disable IRQ
                        ;
411A 3E 05                      ld      a,5             ; disable Tx
411C D3 01                      out     (MIREG1),a
411E 3A 13 11                   ld      a,(MVTXCN)
4121 E6 FE                      and     11111110b
4123 D3 05                      out     (MIREG5),a
4125 DB 04              MLCM10: in      a,(MIREG4)      ; wait until Tx not busy
4127 E6 01                      and     1
4129 20 FA                      jr      nz,MLCM10
                        ;
412B 3E 04                      ld      a,4             ; set Tx rate to R44
412D D3 01                      out     (MIREG1),a
412F 1A                         ld      a,(de)
4130 D3 04                      out     (MIREG4),a
                        ;
4132 13                         inc     de              ; set Tx mode to R45
4133 1A                         ld      a,(de)
4134 D3 05                      out     (MIREG5),a
                        ;
4136 3E 03                      ld      a,3             ; disable Rx
4138 D3 01                      out     (MIREG1),a
413A 3A 14 11                   ld      a,(MVRXCN)
413D E6 FE                      and     11111110b
413F D3 05                      out     (MIREG5),a
4141 DB 04              MLCM20: in      a,(MIREG4)      ; wait until Rx not busy
4143 E6 01                      and     1
4145 20 FA                      jr      nz,MLCM20
                        ;
4147 13                         inc     de              ; set Rx rate to R24
4148 3E 02                      ld      a,2
414A D3 01                      out     (MIREG1),a
414C 1A                         ld      a,(de)
414D D3 04                      out     (MIREG4),a
                        ;
414F 13                         inc     de              ; set Rx mode to R25
4150 1A                         ld      a,(de)
4151 D3 05                      out     (MIREG5),a
                        ;
4153 13                         inc     de              ; set Rx control register R35
4154 3E 03                      ld      a,3
4156 D3 01                      out     (MIREG1),a
4158 1A                         ld      a,(de)
4159 F6 01                      or      00000001b       ;    enable receiver
415B D3 05                      out     (MIREG5),a
415D 32 14 11                   ld      (MVRXCN),a      ; save Rx control word
                        ;
4160 3E 05                      ld      a,5             ; set Tx control register R55
4162 D3 01                      out     (MIREG1),a
4164 3A 13 11                   ld      a,(MVTXCN)
4167 D3 05                      out     (MIREG5),a
                        ;
```

```
                                pop     af              ; return IFF status
                                ret     po
                                ei
                                ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.AHNT  : set address-hunter
                        ;                 This subroutine sets mode , maker-ID code , and device-ID code
                        ;               to address hunter .  Action of address hunter is enabled or
                        ;               disabled by M.CMND .
                        ;                Calling parametors are ;
                        ;                       <E> *------- : ID length.
                        ;                                       0------- : hunt is done by maker-ID code only,
                        ;                                       1------- :                 by both ID code.
                        ;                                   -******* : maker-ID code.
                        ;                               <D> *------- : broad casting enable .
                        ;                                       0------- : device-ID code is checked with only
                        ;                                                  entryed one,
                        ;                                       1------- : device-ID code is checked with both
                        ;                                                  entryed one and 7Fh.
                        ;                                  -******* : device-ID code.
                        ;-------------------------------------------------------------------------------
                        ;
                        M.AHNT:
                                ld      a,i             ; save IFF status
                                push    af
                                di                      ; disable IRQ
                        ;
                                ld      a,2             ; set R26 & R27
                                out     (MIREG1),a
                                ld      a,e
                                out     (MIREG6),a
                                ld      a,d
                                out     (MIREG7),a
                        ;
                                pop     af              ; return IFF status
                                ret     po
                                ei
                                ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.GTVL  : set general timer interval
                        ;                 This subroutine set interval time of general timer.
                        ;                 Calling parametors are ;
                        ;                       <D>  *-------   : load request.
                        ;                       <DE> --******* *******
                        ;                                       : interval value.
                        ;-------------------------------------------------------------------------------
                        ;
                        M.GTVL:
                                ld      a,i             ; save IFF status
                                push    af
                                di                      ; disable IRQ
```

```
                        ;
4188  3E 08             ld      a,8             ; set R84 & R85
418A  D3 01             out     (MIREG1),a
418C  7B                ld      a,e
418D  D3 04             out     (MIREG4),a
418F  7A                ld      a,d
4190  D3 05             out     (MIREG5),a
                        ;
4192  F1                pop     af              ; return IFF status
4193  E0                ret     po
4194  FB                ei
4195  C9                ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.MCVL  : set MIDI-clock timer interval
                        ;                 This subroutine set interval time of MIDI-clock timer.
                        ;                 Calling parametors are ;
                        ;                         <D>  *--------   : load request.
                        ;                         <DE> --******* *******
                        ;                                          : interval value.
                        ;-------------------------------------------------------------------------------
                        ;
                        M.MCVL:
4196  ED 57             ld      a,i             ; save IFF status
4198  F5                push    af
4199  F3                di                      ; disable IRQ
                        ;
419A  3E 08             ld      a,8             ; set R86 & R87
419C  D3 01             out     (MIREG1),a
419E  7B                ld      a,e
419F  D3 06             out     (MIREG6),a
41A1  7A                ld      a,d
41A2  D3 07             out     (MIREG7),a
                        ;
41A4  F1                pop     af              ; return IFF status
41A5  E0                ret     po
41A6  FB                ei
41A7  C9                ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.CCVL  : set click counter interval
                        ;                 This subroutine set interval count (devision rate) of click
                        ;                counters.
                        ;                 Calling parametor is ;
                        ;                         <D>  *--------   : load request.
                        ;                              --******   : interval value.
                        ;-------------------------------------------------------------------------------
                        ;
                        M.CCVL:
41A8  ED 57             ld      a,i             ; save IFF status
41AA  F5                push    af
41AB  F3                di                      ; disable IRQ
                        ;
```

```
 571  41AC  3E 06              ld      a,6             ; set R67
 572  41AE  D3 01              out     (MIREG1),a
 573  41B0  7A                 ld      a,d
 574  41B1  D3 07              out     (MIREG7),a
 575                      ;
 576  41B3  F1                 pop     af              ; return IFF status
 577  41B4  E0                 ret     po
 578  41B5  FB                 ei
 579  41B6  C9                 ret                     ; return from subroutine
 580                      ;
 581                      ;       .page
 582                      ;
 583                      ;-------------------------------------------------------------------------------
 584                      ;       M.RCLK  : read real-time clock
 585                      ;                 This subroutine returns the value of real-time clock using
 586                      ;                recording counter . The value becomes 0 when recording counter
 587                      ;                receives 'start' message .
 588                      ;                 There is no calling parametor . returning parametors are ;
 589                      ;                       <DE>    : 2 byte value incremented by earch MIDI-clock.
 590                      ;-------------------------------------------------------------------------------
 591                      ;
 592                      M.RCLK:
 593  41B7  ED 57              ld      a,i             ; save IFF status
 594  41B9  F5                 push    af
 595  41BA  F3                 di                      ; disable IRQ
 596                      ;
 597  41BB  ED 5B 10 11        ld      de,(MVRCLK)     ; load higher 8 bit into <D>
 598  41BF  3E 07              ld      a,7             ; read lower 8 bit from recording counter R74
 599  41C1  D3 01              out     (MIREG1),a
 600  41C3  DB 04              in      a,(MIREG4)
 601  41C5  5F                 ld      e,a             ;    into <E>
 602  41C6  32 10 11           ld      (MVRCLK),a      ; set count area
 603                      ;
 604  41C9  87                 add     a,a             ; check count value
 605  41CA  38 07              jr      c,MLRC10        ;  if count is small then do ...
 606  41CC  DB 02              in      a,(MIREG2)      ;    check recent overflow
 607  41CE  E6 08              and     00001000b       ;     if overflow has happened recently
 608  41D0  28 01              jr      z,MLRC10        ;             after interruot has been disabled
 609  41D2  14                 inc     d               ;       then increment higher 8 bit value
 610                      MLRC10:
 611  41D3  F1                 pop     af              ; return IFF status
 612  41D4  E0                 ret     po
 613  41D5  FB                 ei
 614  41D6  C9                 ret                     ; return from subroutine
 615                      ;
 616                      ;       .page
 617                      ;
 618                      ;-------------------------------------------------------------------------------
 619                      ;       M.RSTX  : transfer sirial data
 620                      ;                 This subroutine sends data by transmitter according to the
 621                      ;                rate , mode , and connection set by M.CMOD .
 622                      ;                 Calling parametor is ;
 623                      ;                       <D>     : data to send.
 624                      ;                 Returning parametors are ;
 625                      ;                       <E>     = 0 : transfere complete,
 626                      ;                                 others : transmitter is overflow.
 627                      ;                       <D>     : calling parametor is remained.
```

```
                        ;-------------------------------------------------------------------------------
                        ;
                        M.RSTX:
41D7  ED  57                    ld      a,i             ; save IFF status
41D9  F5                        push    af
41DA  F3                        di                      ; disable IRQ
                        ;
41DB  1E  01                    ld      e,1             ; set overflow-flag of returning parametor
41DD  3E  05                    ld      a,5             ; check Tx status R54
41DF  D3  01                    out     (MIREG1),a
41E1  DB  04                    in      a,(MIREG4)
41E3  E6  40                    and     01000000b       ;  if ready then do ...
41E5  28  04                    jr      z,MLTX10
41E7  1D                        dec     e               ;    clear overflow-flag of returning parametor
41E8  7A                        ld      a,d             ;    set data to FIFO-Tx
41E9  D3  06                    out     (MIREG6),a
                        MLTX10:
41EB  F1                        pop     af              ; return IFF status
41EC  E0                        ret     po
41ED  FB                        ei
41EE  C9                        ret                     ; return from subroutine
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       M.RSRX  : receive sirial data
                        ;                 This subroutine receives data by receiver accordig to the
                        ;                 rate , mode , connection , and mode of address hunter and
                        ;                 MIDI-clock filter set by M.CMOD .
                        ;                 There is no calling parametor . Returnning parametors are ;
                        ;                         <D>     : received data.
                        ;                         <E>     : status from Rx status register of MCS.
                        ;-------------------------------------------------------------------------------
                        ;
                        M.RSRX:
41EF  ED  57                    ld      a,i             ; save IFF status
41F1  F5                        push    af
41F2  F3                        di                      ; disable IRQ
                        ;
41F3  11  0E  11                ld      de,MVRXBI       ; load buffer pointers (<D>:=output,<E>:=input)
41F6  7A                        ld      a,d             ; check both pointer
41F7  BB                        cp      e
41F8  20  0C                    jr      nz,MLRX10       ;  if same (no data in buffer) then do ...
41FA  3E  03                    ld      a,3             ;    read status & data from MCS directly
41FC  D3  01                    out     (MIREG1),a      ;                 from R34 & R36
41FE  DB  04                    in      a,(MIREG4)
4200  5F                        ld      e,a
4201  DB  06                    in      a,(MIREG6)
4203  57                        ld      d,a
4204  18  0E                    jr      MLRX90
                        MLRX10:
4206  4A                        ld      c,d             ;  if differ (any data in buffer) then do ...
4207  06  00                    ld      b,0             ;    read status & data from software buffer
4209  21  0E  0F                ld      hl,MVRXBF
420C  09                        add     hl,bc
420D  5E                        ld      d,(hl)
420E  24                        inc     h
```

```
                                        ld      e,(hl)
                                        inc     a               ;    increment output pointer
4211  32  0F  11                        ld      (MVRXBO),a
                                MLRX90:
4214  F1                               pop     af              ; return IFF status
4215  E0                               ret     po
4216  FB                               ei
4217  C9                               ret                     ; return from subroutine
                                ;
                                ;       .page
                                ;
                                ;-------------------------------------------------------------------------------
                                ;       M.IODR  : set direction of I/O port
                                ;                 This subroutine sets direction of external I/O port .
                                ;                 Calling parametors are ;
                                ;                         <D>     : direction of earch port. Output level is
                                ;                                   high-inpedance when 0 is writen , and is
                                ;                                   set by 'write data' when 1 is writen .
                                ;-------------------------------------------------------------------------------
                                ;
                                M.IODR:
4218  ED  57                           ld      a,i             ; save IFF status
421A  F5                               push    af
421B  F3                               di                      ; disable IRQ
                                ;
421C  3E  09                           ld      a,9             ; set register R94
421E  D3  01                           out     (MIREG1),a
4220  7A                               ld      a,d
4221  D3  04                           out     (MIREG4),a
                                ;
4223  F1                               pop     af              ; return IFF status
4224  E0                               ret     po
4225  FB                               ei
4226  C9                               ret                     ; return from subroutine
                                ;
                                ;       .page
                                ;
                                ;-------------------------------------------------------------------------------
                                ;       M.IOWR  : write data to I/O port
                                ;                 This subroutine sets output level of external I/O port whose
                                ;                direction is set 'output'.
                                ;                 Calling parametors are ;
                                ;                         <D>     : write data.
                                ;-------------------------------------------------------------------------------
                                ;
                                M.IOWR:
4227  ED  57                           ld      a,i             ; save IFF status
4229  F5                               push    af
422A  F3                               di                      ; disable IRQ
                                ;
422B  3E  09                           ld      a,9             ; set R95
422D  D3  01                           out     (MIREG1),a
422F  7A                               ld      a,d
4230  D3  05                           out     (MIREG5),a
                                ;
4232  F1                               pop     af              ; return IFF status
4233  E0                               ret     po
```

```
 742  4234 FB                    ei
 743  4235 C9                    ret                     ; return from subroutine
 744                        ;
 745                        ;       .page
 746                        ;
 747                        ;-------------------------------------------------------------------------------
 748                        ;       M.IORD  : read data from I/O port
 749                        ;                 This subroutine reads pin level of all external I/O port.
 750                        ;                 There is no calling parametor . Returnning parametors are ;
 751                        ;                       <D>     : pin level of earch port.
 752                        ;-------------------------------------------------------------------------------
 753                        ;
 754                        M.IORD:
 755  4236 ED 57                 ld      a,i             ; save IFF status
 756  4238 F5                    push    af
 757  4239 F3                    di                      ; disable IRQ
 758                        ;
 759  423A 3E 09                 ld      a,9             ; read data from R96
 760  423C D3 01                 out     (MIREG1),a
 761  423E DB 06                 in      a,(MIREG6)
 762  4240 57                    ld      d,a
 763                        ;
 764  4241 F1                    pop     af              ; return IFF status
 765  4242 E0                    ret     po
 766  4243 FB                    ei
 767  4244 C9                    ret                     ; return from subroutine
 768                        ;
 769                        ;       .page
 770                        ;
 771                        ;-------------------------------------------------------------------------------
 772                        ;       M.FSKS : read FKS modulator/demodulator status
 773                        ;                 This subroutine write control word to FSK control register
 774                        ;                and reads FSK status register of MCS .
 775                        ;                 Calling parametor is ;
 776                        ;                       <D>     : control word for R65
 777                        ;                 Returnning parametor is ;
 778                        ;                       <D>     : status word from R64
 779                        ;-------------------------------------------------------------------------------
 780                        ;
 781                        M.FSKS:
 782  4245 ED 57                 ld      a,i             ; save IFF status
 783  4247 F5                    push    af
 784  4248 F3                    di                      ; disable IRQ
 785                        ;
 786  4249 3E 06                 ld      a,6             ; write control word to R65
 787  424B D3 01                 out     (MIREG1),a
 788  424D 7A                    ld      a,d
 789  424E D3 05                 out     (MIREG5),a
 790                        ;
 791  4250 DB 04                 in      a,(MIREG4)      ; read status word from R64
 792  4252 57                    ld      d,a
 793                        ;
 794  4253 F1                    pop     af              ; return IFF status
 795  4254 E0                    ret     po
 796  4255 FB                    ei
 797  4256 C9                    ret                     ; return from subroutine
 798                        ;
```

```
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       MSIRQ0 : IRQ-0 manager for MIDI real-time message detection
                        ;-------------------------------------------------------------------------------
                        ;
                        MSIR00:
4257  F5                        push    af              ; save registers
4258  C5                        push    bc
4259  D5                        push    de
425A  E5                        push    hl
                        ;
425B  3E  01                    ld      a,00000001b     ; clear IRQ
425D  D3  03                    out     (MIREG3),a
                        ;
425F  3E  01                    ld      a,1             ; get message content from MCS R16
4261  D3  01                    out     (MIREG1),a
4263  DB  06                    in      a,(MIREG6)
4265  57                        ld      d,a
                        ;
4266  3A  12  11                ld      a,(MVACFG)      ; check auto control enable flag
4269  87                        add     a,a
426A  30  11                    jr      nc,MSI010       ;  if enabled then do ...
426C  7A                        ld      a,d             ;    send message to all terget block by R15
426D  F6  F8                    or      11111000b
426F  D3  05                    out     (MIREG5),a
4271  3E  01                    ld      a,1             ;    increment FIFO-IRx by R17
4273  D3  07                    out     (MIREG7),a
4275  21  8B  42                ld      hl,MSI090       ;    call hook address
4278  E5                        push    hl
4279  2A  0A  0F                ld      hl,(MVHSRM)
427C  E9                        jp      (hl)
                        MSI010:
427D  21  85  42                ld      hl,MSI020       ;  if disabled then do ...
4280  E5                        push    hl              ;    call hook address
4281  2A  0A  0F                ld      hl,(MVHSRM)
4284  E9                        jp      (hl)
4285  3E  01                    ld      a,1             ;    increment FIFO-IRx by R17
4287  D3  01                    out     (MIREG1),a
4289  D3  07                    out     (MIREG7),a
                        MSI090:
428B  E1                        pop     hl              ; return registers
428C  D1                        pop     de
428D  C1                        pop     bc
428E  F1                        pop     af
428F  FB                        ei                      ; return from interrupt
4290  C9                        ret
                        ;
                        ;       .page
                        ;
                        ;-------------------------------------------------------------------------------
                        ;       MSIRQ1 : IRQ-1 manager for click/MIDI-clock
                        ;-------------------------------------------------------------------------------
                        ;
                        MSIRQ1:
4291  F5                        push    af              ; save registers
4292  C5                        push    bc
```

```
                        push    de
                        push    hl
                ;
                        ld      a,00000010b     ; clear IRQ
                        out     (MIREG3),a
                ;
                        ld      hl,MSI090       ; call hook and return from interrupt
                        push    hl
                        ld      hl,(MVHMTM)
                        jp      (hl)
                ;
                ;       .page
                ;
                ;-----------------------------------------------------------------------------
                ;       MSIRQ2 : IRQ-2 manager for play-back counter
                ;-----------------------------------------------------------------------------
                ;
                MSIRQ2:
                        push    af              ; save registers
                        push    bc
                        push    de
                        push    hl
                ;
                        ld      hl,MSI210       ; call hook
                        push    hl              ;       that will return next interval in <DE>
                        ld      hl,(MVHPBC)
                        jp      (hl)
                MSI210:
                        ld      a,7             ; set next interval value in R76,R77
                        out     (MIREG1),a
                        ld      a,e
                        out     (MIREG6),a
                        ld      a,d
                        out     (MIREG7),a
                        ld      a,00100000b+2   ; add next interval to current count value
                        out     (MIREG5),a      ;  ( interpolation rate is 2 )
                ;
                        ld      a,00000100b     ; clear IRQ
                        out     (MIREG3),a
                ;
                        jp      MSI090          ; return from interrupt
                ;
                ;       .page
                ;
                ;-----------------------------------------------------------------------------
                ;       MSIRQ3 : IRQ-3 manager for recording counter
                ;-----------------------------------------------------------------------------
                ;
                MSIRQ3:
                        push    af              ; save register
                ;
                        ld      a,00001000b     ; clear IRQ
                        out     (MIREG3),a
                ;
                        ld      a,(MVRCLK+1)    ; increment higher 8 bit of real-time counter
                        inc     a
                        ld      (MVRCLK+1),a
```

```
                        ;
42CE F1                 pop     af              ; return register
42CF FB                 ei                      ; return from interrupt
42D0 C9                 ret
                        ;
                        ;       .page
                        ;
                        ;-----------------------------------------------------------------------------
                        ;       MSIRQ4 : IRQ-4 manager for off-line/break
                        ;-----------------------------------------------------------------------------
                        ;
                        MSIRQ4:
42D1 F5                 push    af              ; save registers
42D2 C5                 push    bc
42D3 D5                 push    de
42D4 E5                 push    hl
                        ;
42D5 3E 10              ld      a,00010000b     ; clear IRQ
42D7 D3 03              out     (MIREG3),a
                        ;
42D9 21 88 42           ld      hl,MSI090       ; call hook and return from interrupt
42DC E5                 push    hl
42DD 2A 08 0F           ld      hl,(MVHO.B)
42E0 E9                 jp      (hl)
                        ;
                        ;       .page
                        ;
                        ;-----------------------------------------------------------------------------
                        ;       MSIRQ5 : IRQ-5 manager for Rx
                        ;-----------------------------------------------------------------------------
                        ;
                        MSIRQ5:
42E1 F5                 push    af              ; save registers
42E2 C5                 push    bc
42E3 D5                 push    de
42E4 E5                 push    hl
                        ;
42E5 3E 20              ld      a,00100000b     ; clear IRQ
42E7 D3 03              out     (MIREG3),a
                        ;
42E9 01 0E 11           ld      bc,MVRXBI       ; load buffer pointers (<B>:=output,<C>:=input)
42EC 16 00              ld      d,0             ; set buffer input offset into <DE>
42EE 59                 ld      e,c
                        ;
42EF 3E 03              ld      a,3             ; set 3 into register group number
42F1 D3 01              out     (MIREG1),a
42F3 DB 04              MSI510: in      a,(MIREG4)      ; repeat until FIFO-Rx empty , do ...
42F5 B7                 or      a               ;    read Rx status from R34
42F6 F2 1A 43           jp      p,MSI530
42F9 4F                 ld      c,a             ;    save status in <C>
42FA 1C                 inc     e               ;    get next input point
42FB 7B                 ld      a,e
42FC B8                 cp      b               ;    compare with output pointer
42FD 20 08              jr      nz,MSI520       ;     if same (buffer full) then do ...
42FF 21 88 42           ld      hl,MSI090       ;       call hook of buffer-overflow
4302 E5                 push    hl
4303 2A 0C 0F           ld      hl,(MVHRCV)
```

```
 970  4306 E9                        jp      (hl)            ;       exit & return from interrupt
 971  4307 21 0E 10          MSI520: ld      hl,MVSTBF       ;   set status in status buffer
 972  430A 09                        add     hl,bc
 973  430B 71                        ld      (hl),c
 974  430C DB 06                     in      a,(MIREG6)      ;   read received data from R36
 975  430E 21 0E 0F                  ld      hl,MVRXBF       ;   set data in data buffer
 976  4311 09                        add     hl,bc
 977  4312 77                        ld      (hl),a
 978  4313 7B                        ld      a,e             ;   set new input pointer
 979  4314 32 0E 11                  ld      (MVRXBI),a
 980  4317 C3 F3 42                  jp      MSI510          ;   end repeat
 981                         MSI530:
 982  431A C3 9B 42                  jp      MSI090          ; return from interrupt
 983                         ;
 984                         ;
 985                         ;       .page
 986                         ;
 987                         ;-------------------------------------------------------------------------------
 988                         ;       MSIRQ6 : IRQ-6 manager for Tx
 989                         ;-------------------------------------------------------------------------------
 990                         ;
 991                         MSIRQ6:
 992  431D F5                        push    af              ; save register
 993                         ;
 994  431E 3E 40                     ld      a,01000000b     ; clear IRQ
 995  4320 D3 03                     out     (MIREG3),a
 996                         ;
 997  4322 F1                        pop     af              ; return register
 998  4323 FB                        ei                      ; return from interrupt
 999  4324 C9                        ret
1000                         ;
1001                         ;       .page
1002                         ;
1003                         ;-------------------------------------------------------------------------------
1004                         ;       MSIRQ7 : IRQ-7 manager for general timer
1005                         ;-------------------------------------------------------------------------------
1006                         ;
1007                         MSIRQ7:
1008  4325 F5                        push    af              ; save registers
1009  4326 C5                        push    bc
1010  4327 D5                        push    de
1011  4328 E5                        push    hl
1012                         ;
1013  4329 3E 80                     ld      a,10000000b     ; clear IRQ
1014  432B D3 03                     out     (MIREG3),a
1015                         ;
1016  432D 21 9B 42                  ld      hl,MSI090       ; call hook and return from interrupt
1017  4330 E5                        push    hl
1018  4331 2A 02 0F                  ld      hl,(MVHGTM)
1019  4334 E9                        jp      (hl)
1020                         ;
1021                         ;       .page
1022                         ;
1023                         ;-------------------------------------------------------------------------------
1024                         ;       MSIRQX : IRQ-X manager for external IRQ
1025                         ;-------------------------------------------------------------------------------
1026                         ;
```

```
1027                                MSIRQX:
1028    4335  F5                            push    af              ; save registers
1029    4336  C5                            push    bc
1030    4337  D5                            push    de
1031    4338  E5                            push    hl
1032                                ;
1033    4339  21  8B  42                      ld      hl,MSI090       ; call hook and return from interrupt
1034    433C  E5                            push    hl
1035    433D  2A  00  0F                      ld      hl,(MVHIRQ)
1036    4340  E9                            jp      (hl)
1037                                ;
1038                                         .end
```

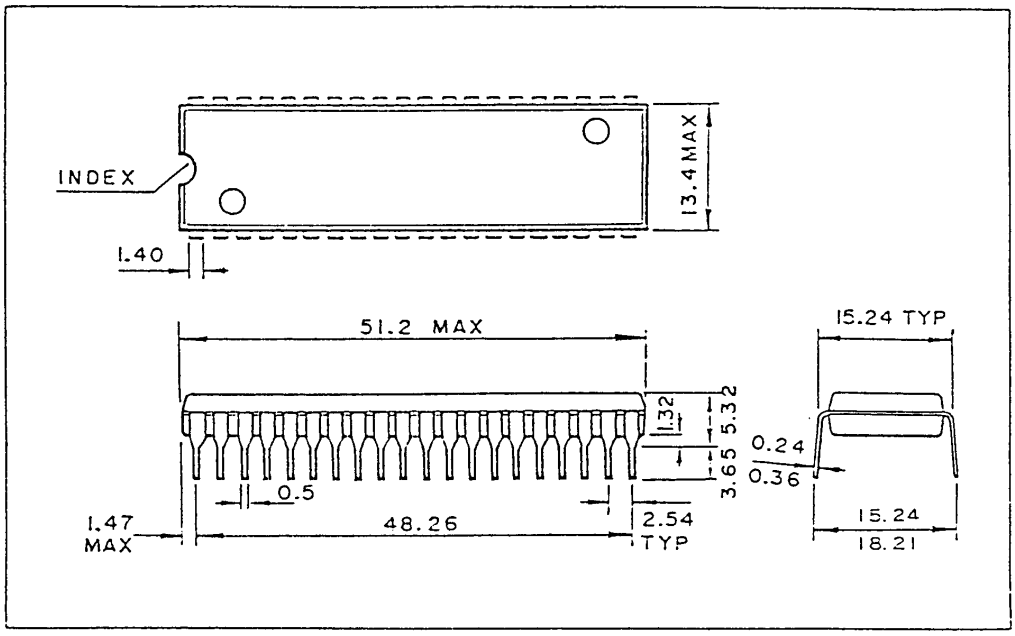

*****SYMBOL TABLE*****

```
MDIN10   4076    MDIN12   4077    MDIRCT   40A0    MIREG0 = 0000    MIREG1 = 0001    MIREG2 = 0002    MIREG3 = 0003
MIREG4 = 0004    MIREG5 = 0005    MIREG6 = 0006    MIREG7 = 0007    MLCM10   412A    MLCM20   4146    MLCN10   4117
MLHS10   40D1    MLHS20   40D9    MLIN10   4016    MLRC10   41D3    MLRX10   4206    MLRX90   4214    MLTX10   41E8
MSIRQX   4335    MSIRQ0   4257    MSIRQ1   4291    MSIRQ2   42A1    MSIRQ3   42C2    MSIRQ4   4201    MSIRQ5   42E1
MSIRQ6   431D    MSIRQ7   4325    MSI010   427D    MSI020   4295    MSI090   4283    MSI210   42AD    MSI510   42F3
MSI520   4307    MSI530   431A    MVACFG = 1112    MVHGTM = 0F02    MVHIRQ = 0F00    MVHMTM = 0F04    MVHOOK = 0F00
MVHO.B = 0F03    MVHPBC = 0F06    MVHRCV = 0F0C    MVHSRM = 0F0A    MVRCLK = 1110    MVRXBF = 0F0E    MVRXBI = 110E
MVRXBO = 110F    MVRXCN = 1114    MVSTBF = 100E    MVTXCN = 1113    MVWEND = 1115    M.AHNT   4172    M.CCVL   4148
M.CMCD   4118    M.CNTL   4100    M.CTLS   40E8    M.FSKS   4245    M.GTVL   4194    M.HOKS   40C3    M.HOOK   4082
M.INIT   4000    M.IOOR   4218    M.IORD   4236    M.IOWR   4227    M.IRQI   4083    M.MCVL   4196    M.RCLK   4187
M.RSRX   41EF    M.RSTX   41D7    WORKAD = 0F00
```

ERRORS DETECTED 0

XXX

Appendix H. Package Outline



The specifications of this product are subject to improvement changes without prior notice.

AGENCY

—YAMAHA CORPORATION—

Address inquiries to:
Electronic System Division

- **Toyooka Factory** 203, Matsunokijima, Toyooka-mura, Iwata-gun, Shizuoka-ken, 438-01 Electronic Equipment business section Tel. 053962-3125
- **Tokyo Office** 3-4, Surugadai Kanda, Chiyoda-ku, Tokyo, 104 Ryumeikan Bldg. 4F Tel. 03-255-4481
- **Osaka Office** 1-16, Shin-ashiyashimo, Suita-shi, Osaka, 565 Tel. 06-877-7731
- **U.S.A.** YAMAHA International Corp. 6600 Orangethorp Ave. Buena Park California 90620

0.3K-0915 Printed in Japan 87.1