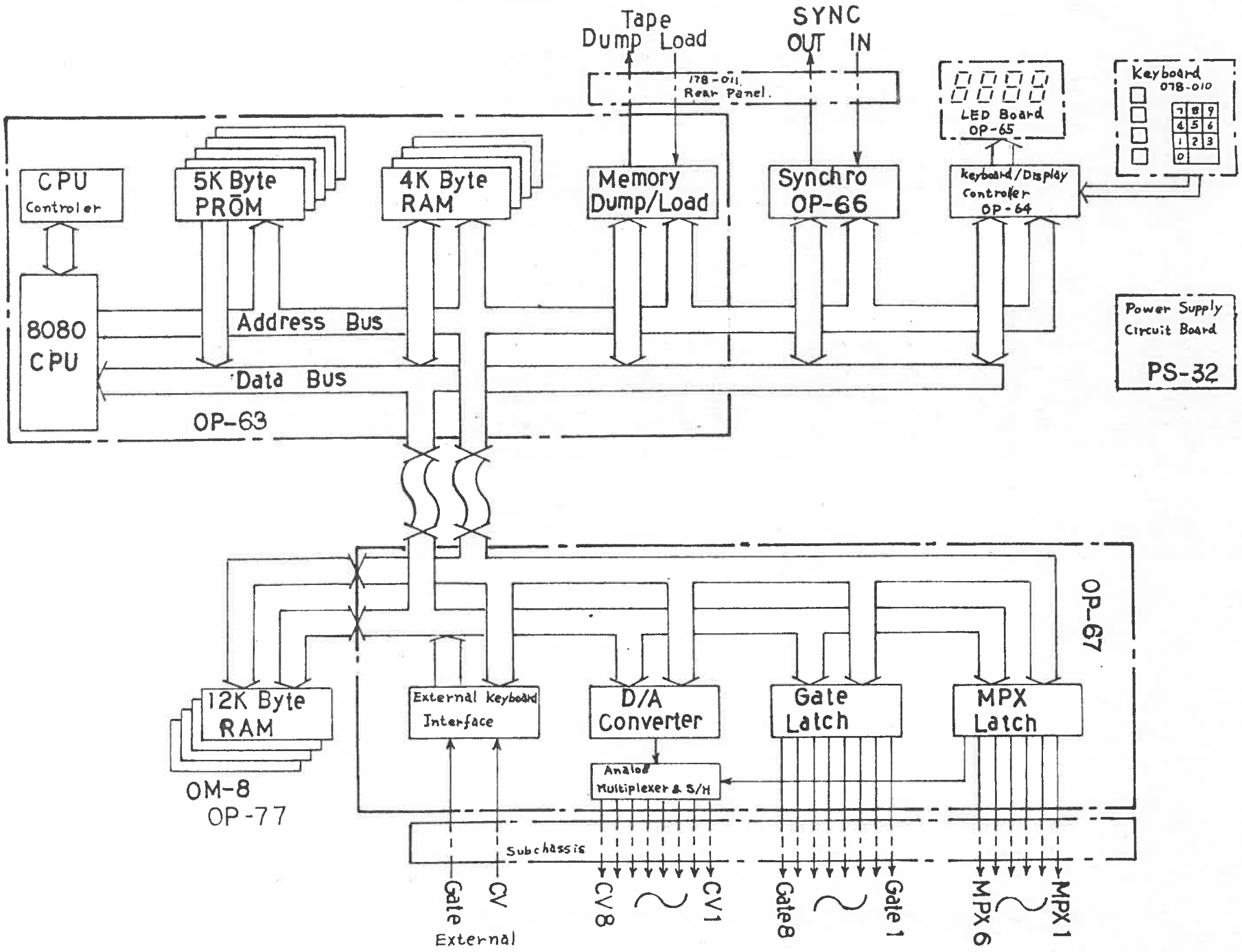
Micro Composer MC-8

CIRCUIT DESCRIPTION

CONTENTS

Roland

Printed in Japan Aug. 21, 1979



MC-8 Block Diagram

1. CIRCUIT LAYOUT (General Description)

As shown on the Block Diagram (Fig.1-1), MC-8 consists of 7 PCBs, not including 052-257, 325 as they are just for wiring purpose only:

CPU Board (OP-63), Display Board (OP-64), LED Board (OP-65), Timer Board (OP-66), Interface Board (OP-67), Optional Memory Board (OP-77)(has been an optional equipment Serial number up to 701002), and Power Supply Board (PS-32).

MC-8 contains a micro-computer. Its functional center is on the LSI 8080AFC or 8080A (IC157) referred to as the CPU (Central Processing Unit). The CPU, operating along with the program stored in its pROM, performs its function to control all operation of MC-8.

Briefly stated, the basic operations of MC-8 are as follows: (Please read through beforehand "MicroComposer Instruction Manual" of ROLAND as it would help for a better understanding.)

1. MC-8 stores the musical data (such as CV, Step Time, Gate Time) which are loaded in most case by manupulation of the ten-keys provided on the Front Panel.

2. When any one of the Panel Key is depressed, it is recognized through the Keyboard/Display Controller which then is sent to the RAM (Random Access Memory) for temporary storing. (See 2-3)

3. When the START key is depressed, the CPU starts to read out from the RAM the data stored for the music play and transfers them into the Interface ... the operation being under the control of the Tempo Clock from the Timer. (See 4-1)

4. In the Interface, the CV digital data is fed through the

D/A Converter to become analog voltage for CV Out. (See 5-3)

5. With data for Gate, MPX, they are latched by the Latches on the Interface so as to be output as GATE, MPX.
   (See 5-2)
6. The above operation 3 - 5 are to be repeated respectively to each channel of 1 - 8 until all musical data stored are exhausted.

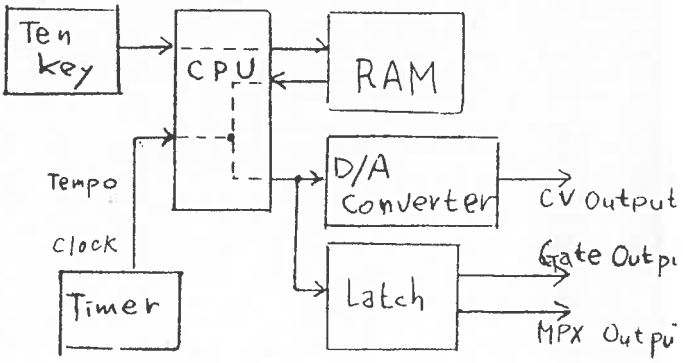


Fig 1-2

Now, we will discuss the operation of each PC Board on MC-8 a little further into detail.

2. CPU BOARD (OP-63)

As said before, the CPU Board (OP-63) is functionally the central part of MC-8 and, as CPU chip, employed here is the 8080 Type of μPD8080AFC or uPD8080A (IC157). Through the execution of the program stored in the Memory, the CPU performs its function to control all the operation of MC-8.

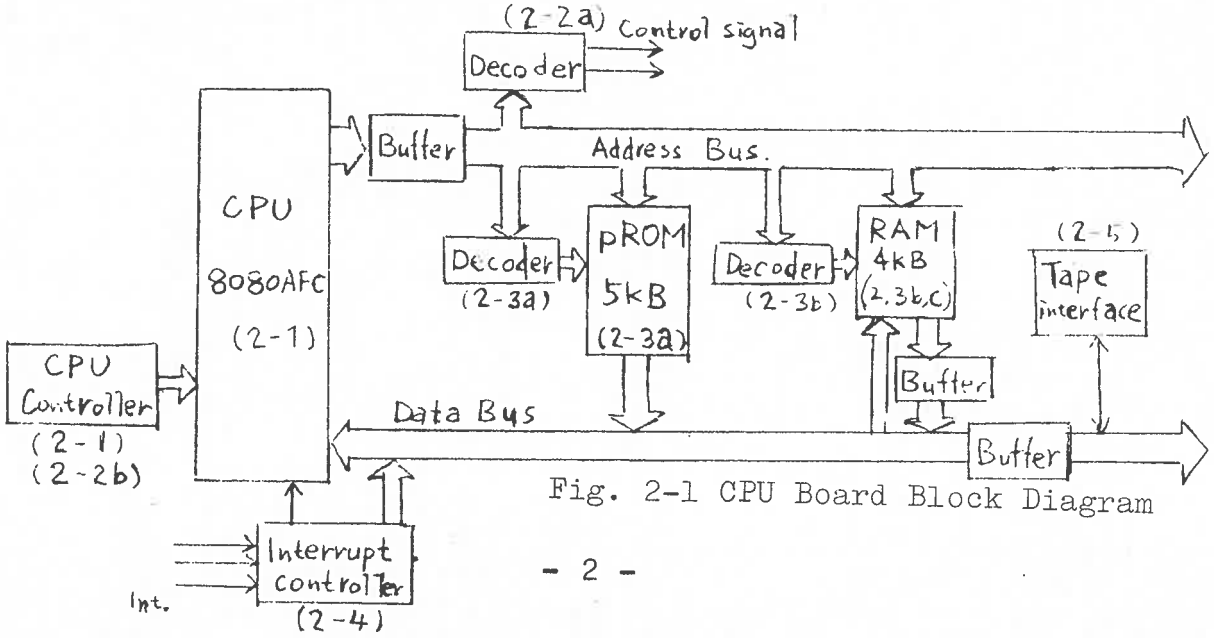


Fig. 2-1 CPU Board Block Diagram

2-1 CPU's Control Signal (Pin Description of CPU)

An example of how the control signals stand when the CPU is executing an instruction is illustrated in Fig.2-3. For each signal a brief explanation is given below.

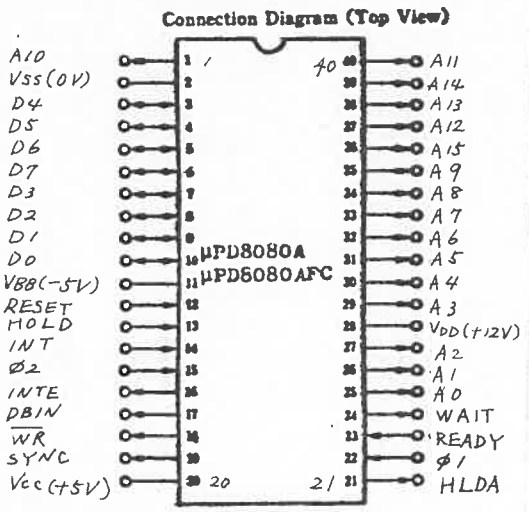


Fig. 2-2

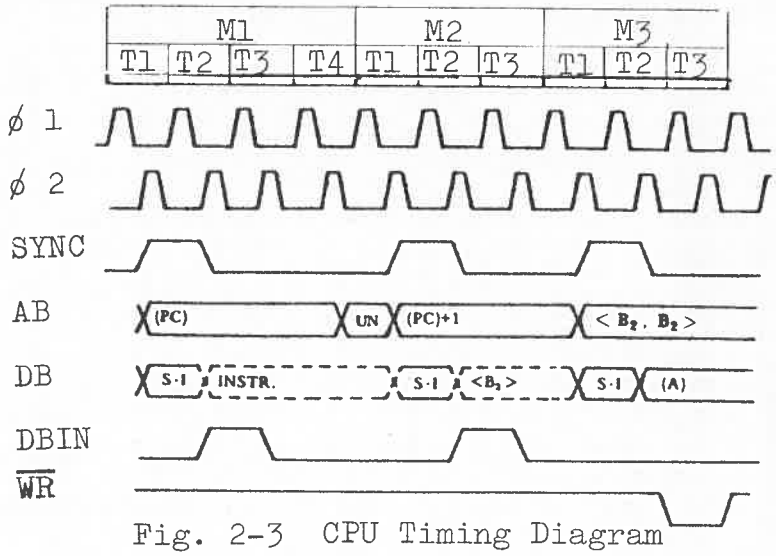


Fig. 2-3 CPU Timing Diagram

CLOCK, Ø1, Ø2:

All the CPU operations are based on the timing of 2 clocks, Ø1 and Ø2. They are both of the frequency 2 MHz, generated at the crystal controlled Clock Generator μPD8224 (IC 159) which is designed primarily for use with the 8080 family.

Address Bus (A_{15} - A_0):

Address is the name or number given to designate the memory location or the I/O devices (such as Tape Interface, Keyboard/Display Controller, D/A Converter, etc.) to which the CPU is to make access.

The CPU of 8080 family has 16 address lines (16 bits) through which to make access to the

binary	decimal	hexa-decimal	binary	decimal	hexa-decimal
0	0	0	1000	8	8
1	1	1	1001	9	9
10	2	2	1010	10	A
11	3	3	1011	11	B
100	4	4	1100	12	C
101	5	5	1101	13	D
110	6	6	1110	14	E
111	7	7	1111	15	F

Table 1 2-10-16 conversion

addresses of 0 - 65535 (2^{16} - 1). These 16 lines, or bits, are usually divided into groups of 4 bits each for convenience when we express their logical states in hexadecimal numbering system. For instance, when they are in binary number ..... 1001 0001 1011 0110, the corresponding hexadecimal number are ... $91B6_H$ (simpler and less vulnerable to error).
(Hereinafter, in denoting the hexadecimal, the suffix "H" will be added.)

On MC-8, the allocation of the memory addresses, expressed in hexadecimal, are as follows:

0000	-	13FF	pROM	(See 2-3a)
4000	-	4FFF	RAM on CPU Board	(See 2-3b,c)
5000	-	7FFF	RAM on OM-8	(See 6)
EOxx	-	EDxx	Various Control Signals	(See 2-2a, 5-1 6-2, etc.)

(xx = irrelevant)

Data Bus (D_7 - D_0):

This is signal lines through wich data and programed instruction are transferred in either direction between the CPU and those Memories and other peripheral circuits.

The data bus of the 8080 family is composed of 8 lines (8 bits). (A group of 8 bits is called "a byte". Bit and byte are the units of information amount. The byte is sometimes abbreviated to "B".)

The CPU, by reading out and executing the instructions pre-programmed into pROM, provides the whole operation of MC-8.

Such instruction in turn is stored each in the form of 1 - 3 bytes in binary integers. For the CPU to read them in, however, there are only 8 bits (= 1 byte) on the Data Bus. When, therefore, an instruction is of a multi-byte such as 3 bytes, it has to be divided into 3 of 1 byte each and be read one by one in time sequence.

For fetch and execution, each instruction is needed 1 - 5 machine cycles (M_1, M_2 ... M_5). What is here called the machine cycle is the basic timing of the CPU operation. It is subdivided into 3 - 5 states (T_1, T_2 ... T_5), and 1 state is equivalent to 1 clock cycle (0.5 μsec.).

We now explain some of the CPU operation following the machine cycle.

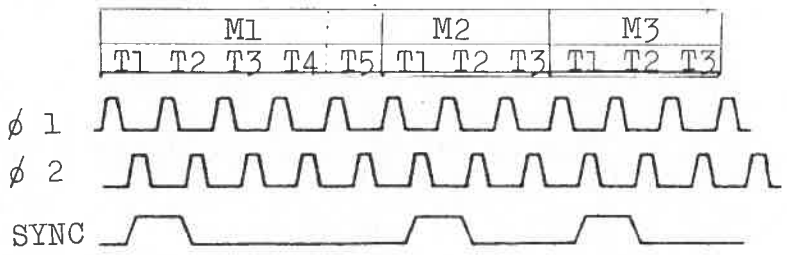


Fig. 2-4 Machine cycle- state and clock

1) CPU designates the address in the memory where the instruction to be executed is stored. By this, the content in the memory becomes ready to be read by the CPU. (Addressing)

2) The content of the memory addressed at T_1 as above is fetched into the CPU. (Fetching)

3) After decoding the memory content thus fetched, the CPU recognizes the meaning of the instruction and execute it as ordered. Here, the instruction content would be as such: to transfer the memory content into the CPU register (See 2-3a), or to decrement the

content of the register, etc. The CPU can execute such instructions of several tens different numbers.

SYNC (SYNCHRONIZING SIGNAL):

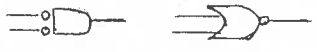
This is to indicate to the external circuits that it is the beginning of each machine cycle. In this case used for generation of STSTB (See 2-2c) in the Clock Generator 8224 (IC159) and for refreshing of OM-8 (See 6-3).

DBIN (DATA BUS IN):

This, when it is "H", indicates to the peripheral circuits that the data bus is in the input mode(CPU is in the mode of reading data or instruction).When the memory or the peripheral circuits are to offer data or instruction to the CPU, they must be placed and made valid on the Data Bus in synchronous with this DBIN.

$\overline{WR}$:

When in "L", it indicates that the Data Bus leading to the CPU is in the output mode(that is, the CPU is in the mode to write the data into memory or to transfer data to peripherals. When data are to be written into memory or peripherals from the CPU, they are output in the Data Bus in synchronous with this $\overline{WR}$. The memory or the peripheral circuits will then be able to take them in.

The vinculum or bar over the WR, as in $\overline{WR}$ means that it is of active low (when in "L", the logical expression is 1; when "H", to 0). On the logical gate symbol, it is indicated by a small circle placed before or after like ..... 

READY:

This is to indicate to the CPU that the peripheral circuits are ready to handle data from the CPU. This signal takes

the timing from IC159 (μPD8224). (READY and WAIT will be explained in 2-2.)

WAIT:

The signal indicates that the CPU is in a WAIT mode.

INT (INTERRUPT REQUEST):

This is made to the CPU from the peripheral circuits that the CPU is required to interrupt the current instruction and jump to other program to serve others. (Detail in 2-4)

RESET:

When RESET is turned to "H", the content of the CPU register is cleared. After RESET is released, the program will start again at the 0 address in the Memory.

When the power is on, the $\overline{\text{RESIN}}$ (RESET IN) of 8224(IC159) is momentrily turned to "L" until C123 has been charged causing 8224 to send out RESET signal in appropriate timing.

Note that there would be some occasion such as that even after the power is switched on again immediately after it being turned off, the RESET could not be activated. This is due to that in C123 there still is remained electrical charge preventing $\overline{\text{RESIN}}$ to be turned to "L". In such occasion, the CPU does not function properly since it starts program in rondom address in the Memory.

If there is need for more details of the CPU and the signals around it, it is recommended to refer to:

8080 User's Manual (INTEL)
or, μCOM-80(F) User's Manual (NEC)

2-2 Control Signals for the CPU and its peripherals

a) Device Select Signals (Address Decoder)

The upper bits, A_{15} - A_8 are also used in making the control signal for the CPU to select peripheral circuits of Timer, Display, etc. (Device Select Signal) Responsible for the work are IC149 (74LS138) and the Gate ICs around it. The circuit of this address decoder and its functions are shown in Fig.2-5 and Table 2 below.

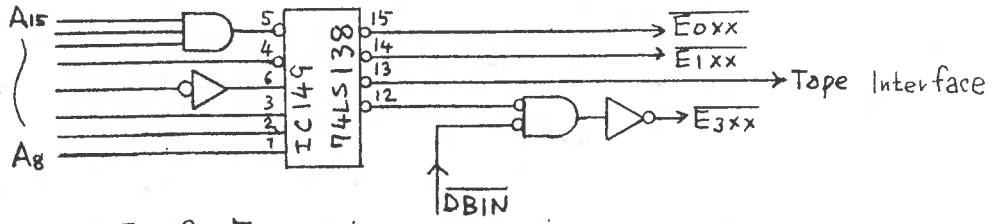


Fig 2-5 Address Decoder

A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8	A_{15}~A_8 Hexadecimal	IC149 out 15P 14P 13P 12P	usage (device selected)
H H H L L L L L	E0	(L) H H H	Timer
H H H L L L L H	E1	H (L) H H	keyboard/Display Controller
H H H L L L H L	E2	H H (L) H	Tape Interface
H H H L L L H H	E3	H H H (L)	Pnel Sw/Check (Timer Board)
the rest	—	H H H H	———

Table 2 Address Decoder Function Table

As seen on the Table, the output of 15P of IC149 is "L" only when the address is turned to "EOxx$_H$". Because of this, $\overline{\text{EOxx}}$ can be used as a control signal for the CHIP SELECT of the Timer Board IC505.

Just as the same, $\overline{\text{Elxx}}$ is used for the select signal of Display Board IC404 (See 3-1), $\overline{\text{E2xx}}$ for Tape Interface

select (See 2-5), and $\overline{E3xx}$ for Panel SW/Optional Check (See 4-4).

These circuits, except Tape Interface, are all positioned inside of the data bus buffer. For this, when the control signals ($\overline{EOxx}$, $\overline{E1xx}$, $\overline{E3xx}$) are turned to active low, IC151, 133, act as to inhibit the buffer from function.

b) READY & WAIT

When the CPU is to read pROM, the DBIN is turned to active for only 0.5 μsec (= 1-state in machine cycle). This means that the time allowed for the CPU to complete reading after its output of address is shorter than the access time allowed on the pROM, being unable to make proper reading.

Here, therefore, are used READY and WAIT to provide a wait time of 1-state (0.5 μsec.) between the output of the address and the CPU making actual reading, It is that the DBIN is being held active for an extended 1 μs.

With Keyboard/Display Controller μPD757 (IC404), the circuit is further slower in its reading and writing. Therefore, when the address is E1xx (when the CPU makes access to μPD757), a 3-state wait time is provided to have DBIN, $\overline{WR}$ held active for more extended 2 μs.

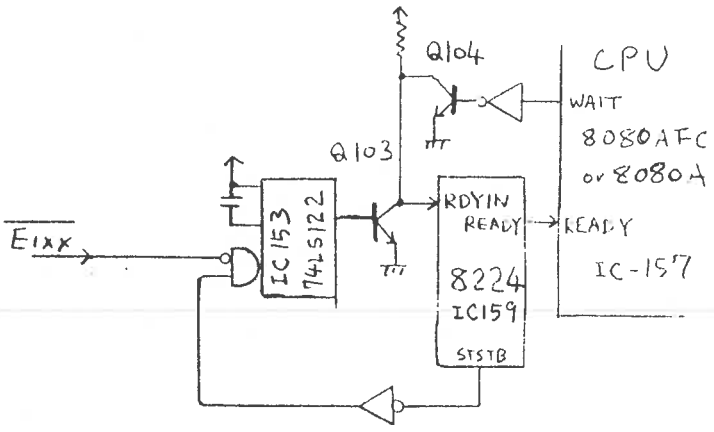


Fig. 2-6

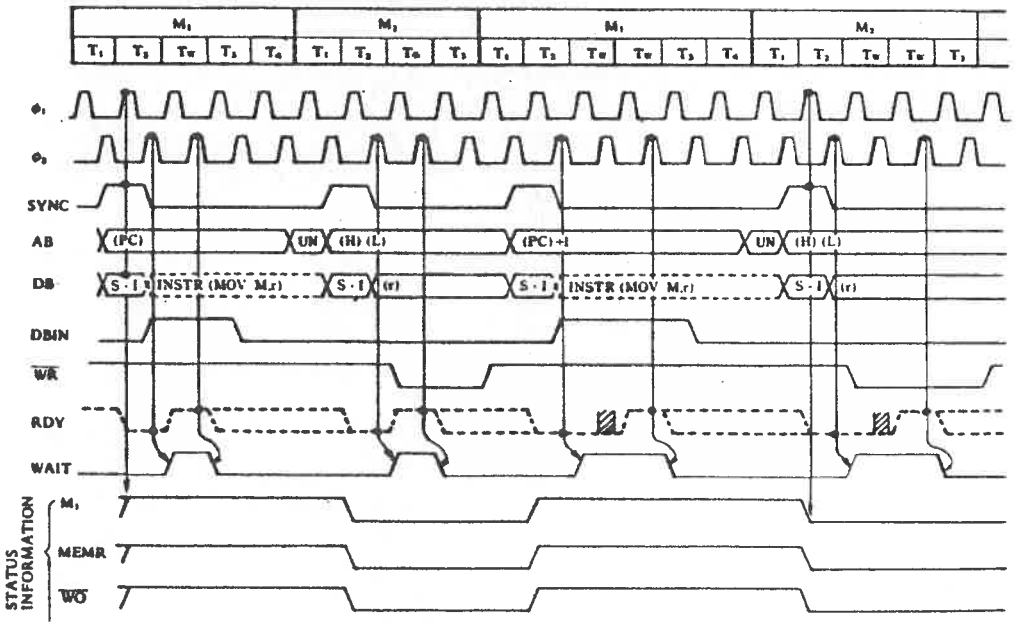


Fig 2-7 Timing Diagram of Wait Operation

Here now, let us explain a little more about the wait mode.

The CPU checks the READY input at T_2 (in machine cycle), and if it is in "L", the CPU enters into a wait mode, sending out the WAIT signal. During the time, DBIN or $\overline{WR}$ is to be kept on the state. When READY is then turned to "H", the CPU comes out of the wait mode. In the next step at T_3, it can proceed with reading (or, writing) of data.

With reading/writing from/to Memory, on output of WAIT from the CPU, RDYIN of IC159 (Clock Generator) is turned to "H" through IC158, Q104. On the following step, the WAIT comes to be released.

In the case for the Display/Keyboard to read/write, the RDYIN is held at "L" for about 1.5 μs through IC153 so as to hold the wait mode for 3-state.

Thus, through the process, the DBIN, $\overline{WR}$ could be held active for an extended 1 μs for the most case or, if with read/write of the Display/Keyboard, for more longer time of 2 μs.

c) Status Strobe ($\overline{STSTB}$)

The CPU outputs the status signal onto the Data Bus at the time of T_1, Ø2. With this, the peripheral circuits should know of the current status of the CPU or what data be available on the Bus. Fig.2-7 is to show some example of these status signals, with M_1, MEMR, $\overline{WO}$ indicating that they are being latched. On MC-8, the only status signals being utilized are INTA (INTERRUPT ACKNOWLEDGE) (See 2-4) and M_1 (this is to indicate the first machine cycle of the instruction)(See 6-3). The pins used to indicate these two status words are D_6 and D_5 to the Data Bus.

To latch the status signal, the timing signal used is the $\overline{STSTB}$ to be output from μPD8224 (IC159). (STSTB is in fact the logical AND output from the inputs of SYNC and Ø1.)

After being latched by IC152, M_1 is then used for refreshing the dynamic RAM of OM-8 (See 2-3, 6-3), while INTA is for processing of the Interrupt (See 2-4).

As explained in b) above, when the CPU is to make access to Keyboard/Display, having the address output E1xx,

the Address Bus must be kept stable for the control of READY. This is accomplished by taking the logical AND between the STSTB and Elxx.

2-3 Memory

Memories used on MC-8 can be classified into the programmable read-only-memory (pROM) and random-access-memory (RAM). The former is the memory fabricated in a special method for use in read only. It is not for writing, and its stored data is non-volatile not to be lost with time even after the power is turned off. On the other hand, the latter RAM features as being free for both reading and writing. Its memory, however, becomes lost when the power is turned off. MC-8 employs both, with pROM for storing the instructions pre-programmed for the CPU to execute, while with RAM for storing musical data to be played and loaded during operation.

For RAM, there are further classifications into the static RAM and dynamic RAM. The static RAM is the one which stores the data on the on-chip flip-flop in the form of "1" and "0". The data once stored there can be kept for as long as the power is connected and if the data were not re-written. Memories on the dynamic RAM, on the other hand, are made to "1" or "0" according to the electrical charges be filled in the capacitor in a cell or not. Such charges will be lost by leak with time and along which the stored data, too. To avoid this, the memory cells must be re-charged periodically, and this operation is called the "Refresh".
(Refer to Section for OM-8)

a) pROM

μPD454D is the pROM chip employed on MC-8. It is the pROM of a 2048-bit memory (256-word x 8-bit = 256-byte). When 20 chips used together, they make the memory capacity of 5 kB ("k" equals 1,024, or 2^{10})

A chip of μPD454D is a 256 byte ROM. If it is to this one chip ROM alone, access can be made with 8 bit address lines (A_7 - A_0). However, as there are multiple number of the chips, another bit lines become necessary so as to make a selection of a particular chip at the time. For this, the upper 8 bits (A_{15}-A_8) are used. Fig.2-8, which includes IC108 - 110, 122, 134, is to show this decoder circuit, and Table 2 for the Function Table.

When the CPU is to read the stored memory of ROM, it makes address through A_{15} - A_0, with DBIN held at "H". For instance, the address expressed on these bit lines could be understood to be 0B2F in hexadecimal. With the former half of "0B" composed of the upper bits of A_{15} - A_8, the $\overline{CS}$ (CHIP SELECT) of IC131 is made to active low of "L" (See Table 2). Then, by the latter "2F" made by lower bits of A_7 - A_0, a stored program (1 byte) inside of the selected IC131 is reached. The program is then put on the Data Bus to be read by the CPU.

Address allocation of the pROM on MC-8 is as follows: 0000 - 00FF to IC144, 0100 - 01FF to IC143 ..... 1300 - 13FF to IC123. In all, they make the memory capacity of 5 kB, from 0000 to $13FF_H$.

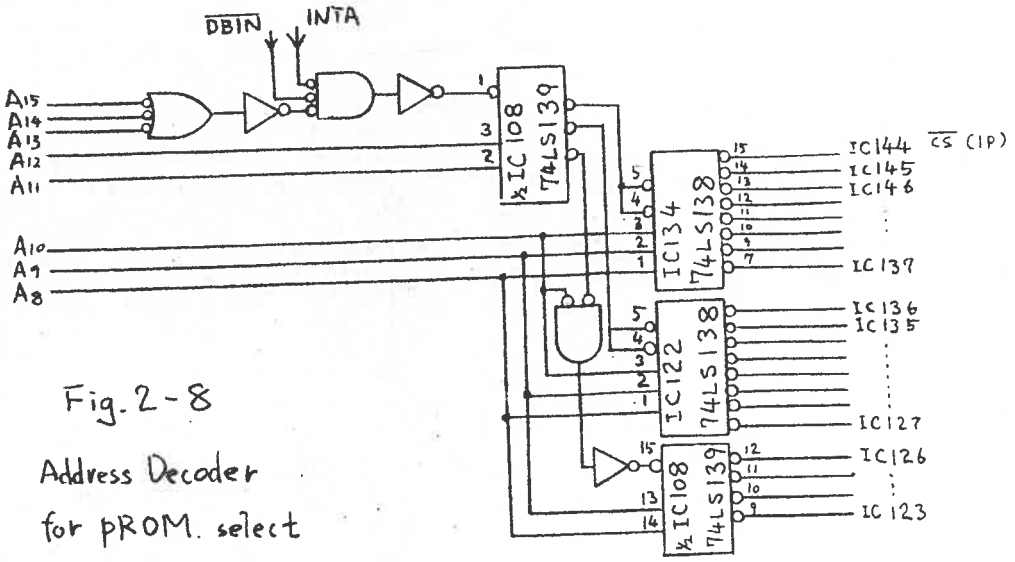


Fig. 2-8

Address Decoder for pROM. select

A15	A14	A13	A12	A11	A10	A9	A8	A15-A8 Hexa-decimal	DBIN	PROM $\overline{CS}$ pin
X	X	X	X	X	X	X	X	X	L	All PROMs' $\overline{CS}$: H
L	L	L	L	L	L	L	L	00	H	L: only IC144 pin 0
L	L	L	L	L	L	L	L	01	H	L: only IC143 pin 1
L	L	L	L	L	L	H	L	02	H	L: only IC142 pin 2
L	L	L	L	H	L	H	L	0A	H H H	L: only IC132 pin 10
L	L	L	H	L	L	H	H	13	H	L: only IC123 pin 19
more than 14									X	All PROMs' $\overline{CS}$: H

X= irrelevant

Table 2 PROM Selection

b) RAM

Eight chips of μPD410D of a 4096 word x 1 bit static RAM are implemented on the CPU Board. Eight chips of a 4096 x 1 bit in parallel use make the memory capacity of 4096 x 1 byte. Addresses 4000 - $4FFF_H$ are allotted to them.

The addressing method is that, while connecting A_{11} - A_0 to each RAM chip, the other A_{15} - A_{12} output "4_H" in hexadecimal. This 4_H, going through the decorder, causes

the $\overline{CS}$ on the chips to turn to "L" to open the gate onto the Data Bus. (See Fig.2-9, Table 3)

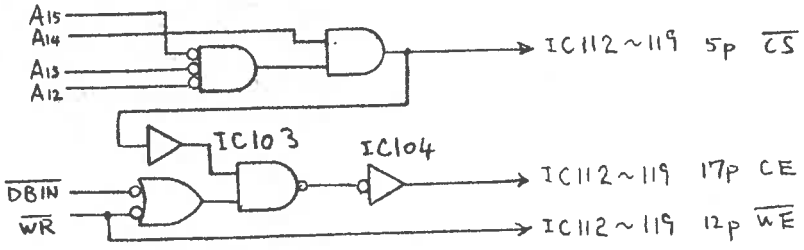


Fig 2-9. Decoder for RAM select.

CPU OUTPUT						RAM INPUT			
A15	A14	A13	A12	A15-A12 Hexa-decimal	DBIN	$\overline{WR}$	$\overline{CS}$	CE	$\overline{WE}$	
L	H	L	L	4	L	H	L	L	H	
					H	H	L	H	H	read
					L	L	L	H	L	write
others exclude above					X	X	H	X	X	

Table 3 RAM Selection

When this $\overline{CS}$ is held on "L", with DBIN on "H", the stored data addressed by A_{11} - A_0 are placed on output from IC112 - IC119 which, after passing through the buffer, come to the Data Bus D_7 - D_0 where the CPU reads them in.

When $\overline{CS}$ is "L", with $\overline{WR}$ on "L", the data placed on the Data Bus D_0 - D_7 are now those which the CPU is to write onto the RAM.

c) Working Memory

When the CPU is executing the instruction fetched from the pROM, it needs some RAM for storing temporarily the intermediate result. The memory used for such is called a working memory where the data stored only concerns to the instruction in current execution.

Although the CPU itself has some of its own working register, its capacity is limited. On MC-8, therefore, further working memory locations are provided at the addresses 4000 - $41BF_H$. Memory location for musical data, therefore, becomes to be the next onward from $41C0_H$.

2-4 Interrupt Control

The peripheral circiuts can access to the CPU for requirement that the CPU interrupts the program currently executing to jump to another specific program. This is called the interrupt request or interrupt.

a) Classification of Interrupt

There are three kinds of interrupts with MC-8:

INT 1: This is the interrupt request produced at the time of Tempo Clock going positive. (Tempo Clock is a clock produced at the Timer Board, with its 1 count step time equal to 1 cycle.)

At INT 1, the CPU makes CV, Gate outputs varied as may be required. (Further explanation on 4-1)

INT 2: This is made by the Interface Board. It is for the CPU to read the input data as may be varied

in the External Input Signal (CV, Gate) when operated under the External Input mode. (See 5-4a)

INT 3: This is sent from the Timer Board to inform the CPU of the time passage of every 25 ms. With every INT 3, the CPU send out the CV data to D/A converter. (See 5-3) It also rewrites the Time Display in every 0.1 sec. (for 4 counts of INT 3).

Although there is still another INT 4 remained possible on the circuit, this is not put to any practical use.

b) Handling of Interrupt Request

When any of INT 1 - 3 is sent out, it is known to the CPU through its pin INT (INTERRUPT REQUEST). The CPU accepts the request as soon as it completes the current instruction and honor the interrupt request by sending out INTA (INTERRUPT ACKNOWLEDGE). By this, the data which tells the CPU which INT is generated is placed on the Data Bus. The CPU reconizes the INT by reading this, and jumps to the program as pre-programed for the INT.

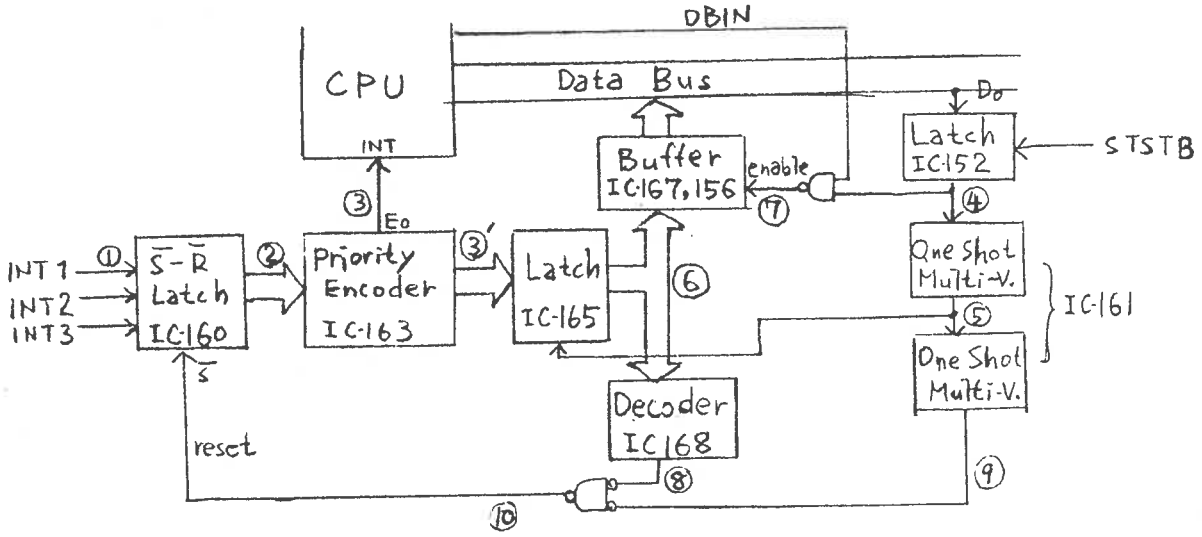


Fig 2-10 Block Diagram of Interrupt Control

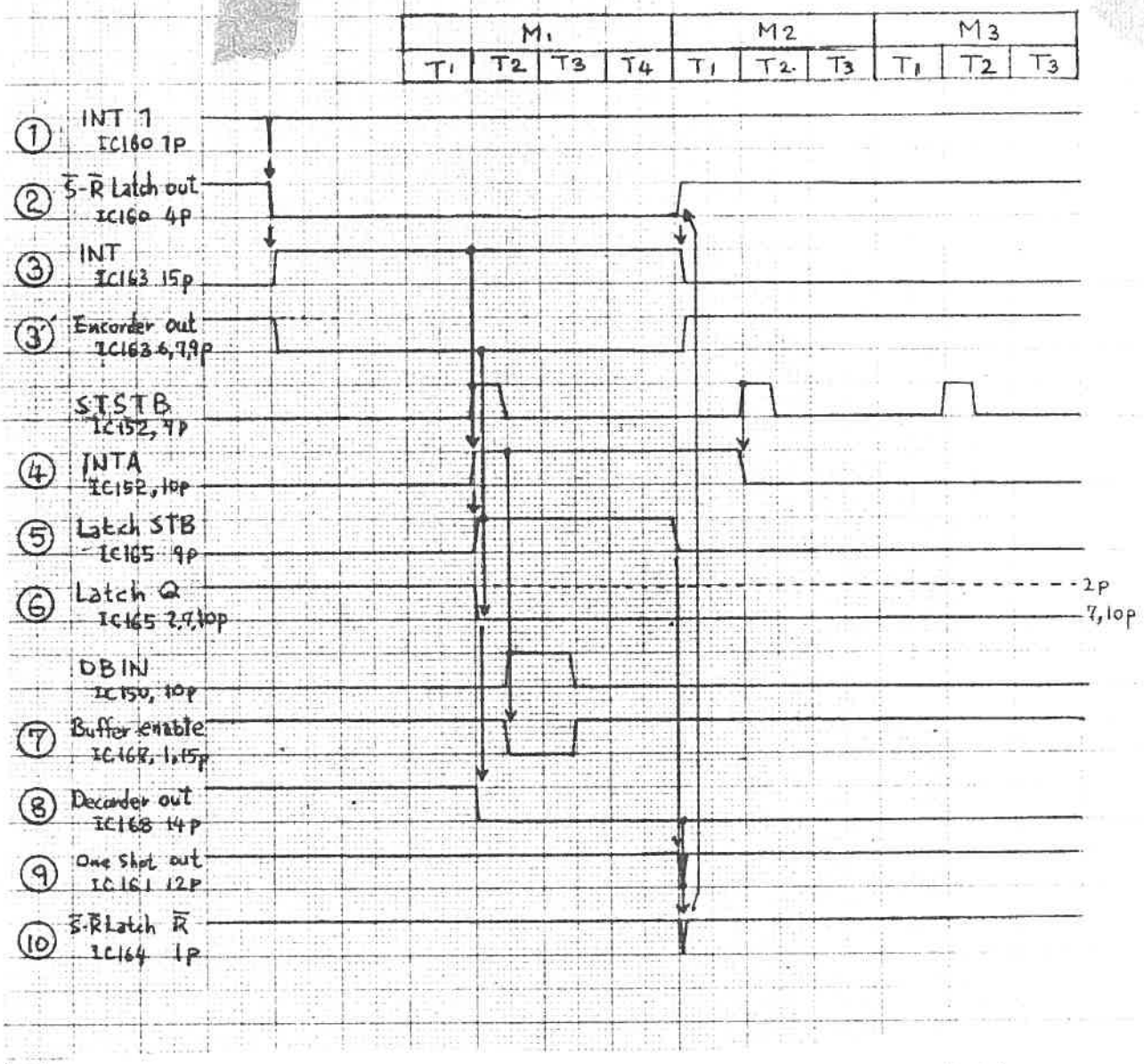


Fig. 2-11 Timing Chart (an example INT-1)

We will explain a little more about how the Interrupt requests will be handled taking some practical example as follows: Please read through with referring to the block diagram (Fig.2-10) and Timing Chart (Fig.2-11).

Firstly, a pulse signal out of INT 1-3 enters the pin $\overline{S}$ of the $\overline{S}$-$\overline{R}$ Latch (IC160). (1) Here, we assume that the IC160 is in a state being "reset". (When the power is turned on, all S inputs of IC160 are momentarily turned to "L" through IC148, 164 to achieve this reset.)

The signal is latched on IC160 (2), and when coming out, drives the pin E_o of the Encoder (IC163, 74148) to turn to "H" (3), to send out INT (INTERRUPT REQUEST) to the CPU.

INTA (Interrupt Acknowledge)

The CPU, on receipt of INT, outputs INTA to D_0. This is latched on IC152 (74LS175) by the signal of STSTB (4). This INTA (4) is used to trigger the One Shot Multivibrator (IC161, 74LS123) to produce a pulse of width about 1.4 μsec (5). The pulse causes the output of the Encorder (3) to be latched on IC165 (74LS175). (6)

This latched signal contains the information decoded to show what Interrupt Request is among INT 1 - 3. The CPU reads this when it is put on the Data Bus. After INT is defined, the CPU can proceed with the prearranged program for the INT requested.

During the time when INTA being latched, the Data Bus Buffer shall be kept enable, while ROM gates should be made inhibit (thus allowing the CPU to read the Interrupt placed on the Data Bus).

The output of the Latch (6) is fed to the Decoder (IC168, 74LS138), too. Decoded output is then used to be the same pulse again with (2) . This signal, combined with the pulse (9) produced at the trailing edge of the pulse (5) , drives the IC160 (S-R Latch) to be reset.

The above mentioned IC163 is a priority encoder. If there would be two Interrupts at the same time or a new one comes during one execution, the one to be executed first will be selected on the priority basis of INT 1 > INT 2 > INT 3, as shown on Table 4.

INT			IC160 OUTPUT			IC163 OUTPUT		
1	2	3	pin4	pin 7	pin 9	pin 9	pin 7	pin 6
H	H	H	H	H	H	H	H	H
⎍	X	X	L	X	X	H	L	L
H	⎍	X	H	L	X	L	H	L
H	H	⎍	H	H	L	H	H	L

X = irrelevant

⎍ = interrupt pulse

Table 4 INT priority

Latch "reset" signal (10) causes only reset of the INT that has been completed of its execution. Therefore, if another INT is still latched on IC160, the CPU can fetch this INT(un-reset)as soon as the current execution of the Interrupt is completed.

2-5 Tape Interface

On MC-8 all the data concerning musical notes are placed in RAM where, however, the data will be lost when the power is turned off. For longer storage therefore some other way is necessary and this is accomplished by sending the data through FSK (Frequency Shift Keying) Modulation for recording onto an ordinary audio tape.

When [DUMP] is depressed:

The CPU, reading out the RAM data (which is in parallel 8-bit construction), outputs them from D_7 after de-multiplexing into time sequence. The signal of the logical NAND output of $\overline{E2xx_H}$ (of Tape Interface Control Signal) and $\overline{WR}$ causes this D_7 output latched. The data then are to be FSK modulated by the VCO on IC146.

The FSK signals here are of the frequencies of 2100 Hz (at H) and 1300 Hz (on L). To have the frequency on the proper range, adjustment can be made by VR101 and VR102. Also, since the outputs of IC146 are in square wave form, they are reshaped through the LPF consisting of CR prior to their output from DUMP JACK.

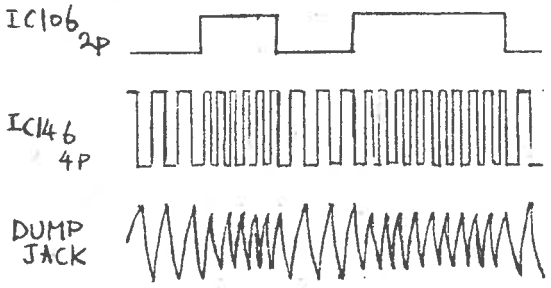


Fig. 2-12 FSK Modulation

When [LOAD] is depressed:

FSK signals coming from LOAD JACK are fed through the BPF (Q101,Q102) to FSK Demodulator consiting of PLL (Phase Locked Loop),IC147,IC170 where they are converted into digital and interfaced to TTL compatible level signals. (Here,VR102 functions for adjusting the frequency of free running.)

They are sent thereafter through the buffer. The buffer is enabled by the logical AND between $E2xx_H$ and DBIN. The data are output from D_0 and read by the CPU. (The process is entirely the reverse of the DUMP.)

When [VERIFY] is depressed:

The CPU compares successively the signals demodulated after coming in from the LOAD JACK for externals with those contained in the stored data in the RAM. When any difference between them is found, the CPU sends out error signals onto the Display.

3. DISPLAY BOARD (OP-64)

Display Board (OP-64) is one of the peripheral devices around the CPU. It is composed of the LSI μPD757 (IC 404), the Keyboard Display Controller, as its center. This LSI μPD757 performs the function that it puts on the Display the data sent from the CPU and, also, detects and informs the CPU of what key is depressed on the Keyboard panel.

3-1 μPD757

The Keyboard Display Controller is capable of scanning the 4 x 12, 48 key matrix and of dynamic 12-figure on-the-panel displays each consisting of 7-segment LED.

Its pin functions are explained below:

$D_3 - D_0$:

These are pins to the Data Bus for communicating data to/from the CPU.
The bits on the Bus are lead to D3-DO on the CPU.

The read/write timing of μPD757 lags behind associated

address by 1μs-1.5μs, while on the other hand, the CPU outputs the data at the same time with addressing. Therefore, a WAIT has to be applied onto the CPU to get the both timing be matched. (This is achieved inside of the CPU itself. See 2-2, READY & WAIT)

$\overline{CS}$ (CHIP SELECT):

The signal is obtained by taking the logical NAND output of $Elxx_H$ of the device select signal (See 2-2, I/O Device Select Signal) and WR or DBIN. In other words, when the CPU is to exchange data between μPD757, it can be done by making access to Elxx.

D/I (Data/Instruction):

The CPU designates signal to be placed on the DataBus through A_0 whether it should be the data for the Display or the data detected of the Keyboard key depressed, or it should be the control instruction to μPD757.

$\overline{RESET}$:

The signal makes the reset of the registers inside of μPD757 (contained of the data to be displayed, etc.) which is done at the time of the power being turned on. (The same time with the reset of the CPU)

Ø (Clock):

The clock that drives μPD757 onto function is of the frequency 1 MHz maximum. This is obtained from Ø2 (TTL level) generated at the CPU's Clock Generater (IC159, 8224) through frequency dividing into 1/2 by the F/F circuit.

L :

S_a - S_g :

T_0 - T_{11}:

KR_0 - KR_3:

(To be explained on 3-2, 3-3)

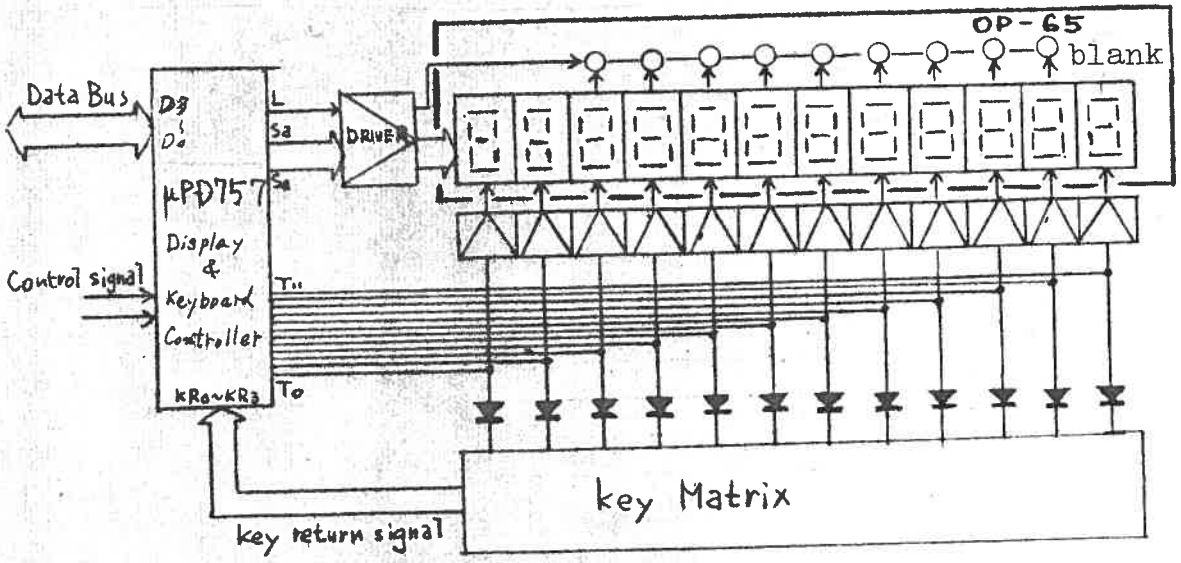


Fig 3-1 BLOCK DIAGRAM of OP64 and OP65

3-2 Display Functions

Keyboard Display Controller μPD757 (IC404) outputs signals for the 7-segment display LEDs. The signals are output from T_0 - T_{11} for digit and from S_a - S_1 for segment selection. In other words, positions of the figure are selected through T_0 - T_{11}, while the segments necessary to display the figure are nominated by S_a - S_1. Fig. 3-2 shows the method of the 7-segment displays of figures and characters from the original shape.

(Among those possible displays, MC-8 uses 0 to 9, "-" and blank.)

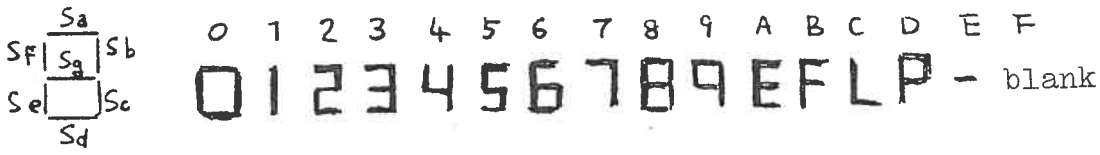


Fig 3-2

There are also the lamp data output signal "L" which enables 12 pieces of LED to be used as a pilot through T0-T11. (MC-8 uses 7 circuits.)

As seen from Fig.3-3, S_a - S_1 and L are the outputs to send out in synchronous with T_0 - T_{11}. In order to avoid overlaps of each, they are momentarily disabled by being put to "L" for a duration of 7 clocks on each transient time of digits.

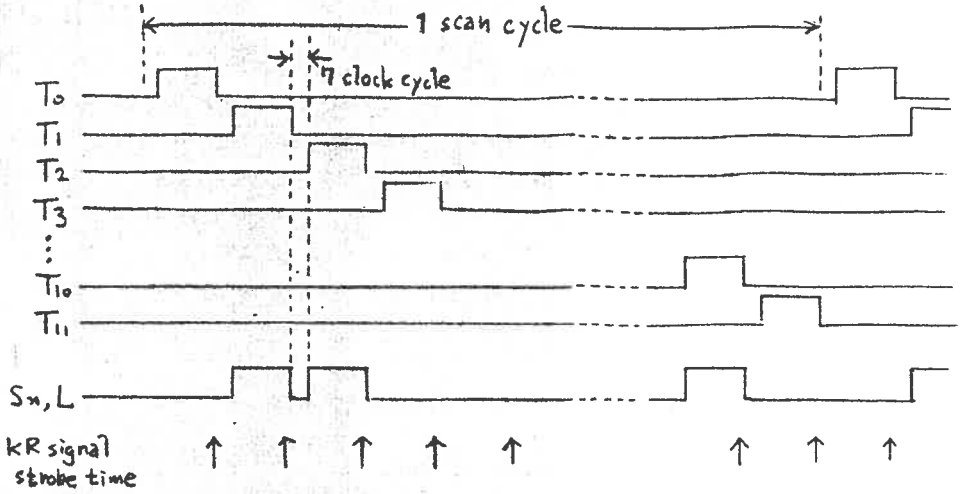


Fig 3-3 Timing Diagram of μPD757

3-3 Keyboard Reading

The keys on the Panel are constituting a matrix (4 x 12). The digit signals from T_0 - T_{11} (used for Display, too) are fed into the key matrix, and the signals from the matrix (Key Return Signal) are input through KR_0 - KR_3.(pins 5-8) Each digit signal pulse is then sampled at their last clock time to be read by the μPD757. (See Fig.3-3, KR Signal Strobe Time)

START, STOP are provided with the Remote Jack, so that an on-off switching from off-position is made possible just as the same with the key on the Panel.
Memory Protect Key is made to the same function to duplicate the STOP key.

4. TIMER BOARD (OP-66)

Timer Board (OP-66) consists of IC 8253 of Programmable Interval Timer, FSK Modulator/Demodulator Circuit for SYNC signal, and VCO for Tempo Adjustment. IC 8253 is organized as 3 independent 16 bit counters (Counter No.0 No.1 and No.2) and all operation are software controlled.

Functions of Timer Board are mostly as follows:

1. To run MC-8 at a speed as determined by Time Base and Tempo,
2. To inform the CPU of the lapse of the time every 25ms.
3. To operate SYNC function ,

4. To function Optional Check, (see 4-4)

5. Switching of CYCLE, SYNC.

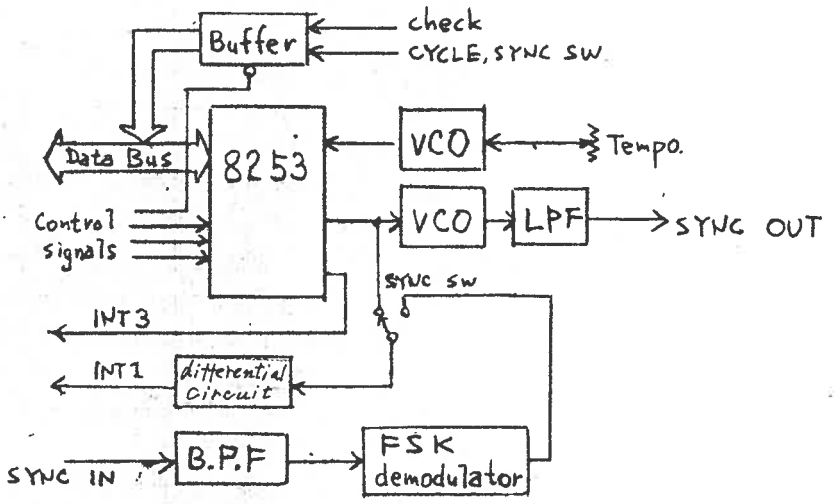


Fig 4-1 Block Diagram of Timer Board

4-1 Tempo Control

Tempo Clock, which provides the base for the Step time and Gate time of musical data, is firstly generated at the VCO of IC503 (MC14046) in a frequency about 140 KHz of square wave form. (VCO control voltage is fed from the TEMPO VR.) This frequency is then divided by the Counter No.0 of 8253 with an appropriate value determined by a given Time base, Tempo.

When the mode is SYNC-off, INT 1 pulse is generated at each leading edge of the Tempo Clock signal. (SYNC-on will be explained later in 4-2.)

And, with this INT 1, the CPU starts counting of Step Time and Gate Time and, where necessary, produces Gate, CV outputs. The process is as follows:

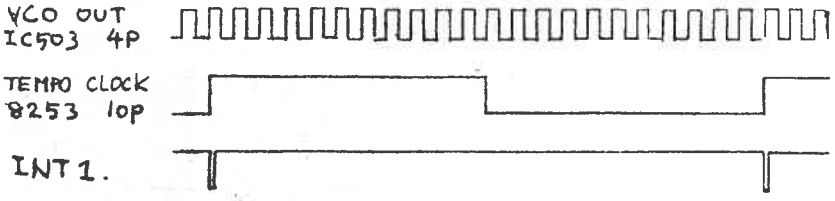
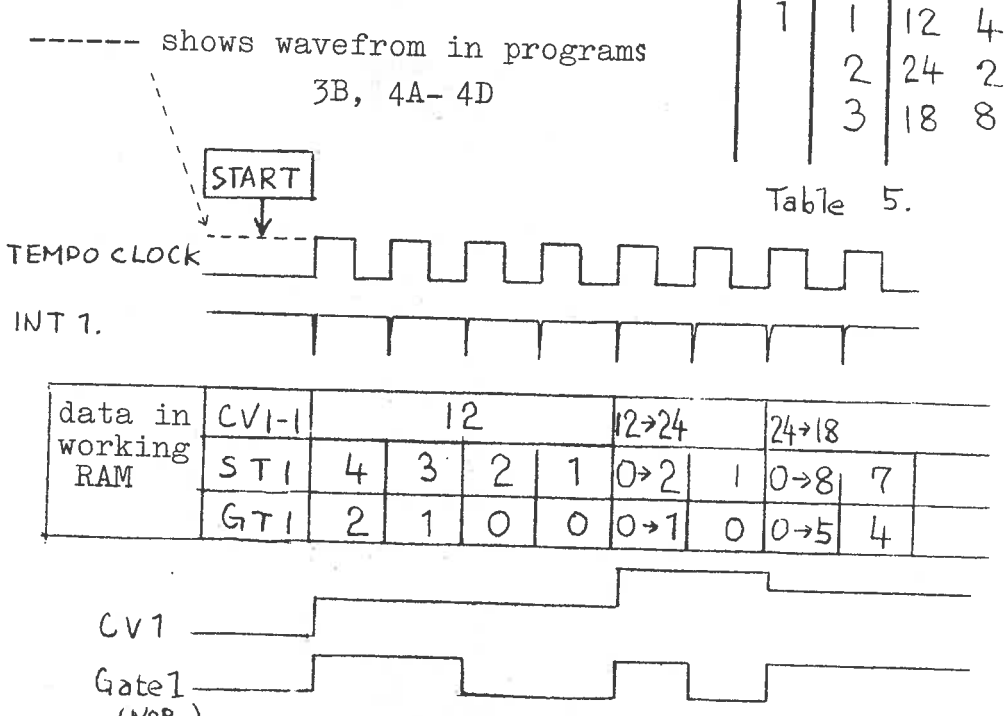


Fig 4-2 Tempo Control (when SYNC "off")

Let's assume that the program stored are such as shown in Table 5. With these, we are loading and going to play back. In the explanation to follow, Step time 1 will be written as ST 1, and Gate time as GT 1. Please read referring to Fig. 4-3.

		CV1	ST1	GT1
1	1	12	4	2
	2	24	2	1
	3	18	8	5

Table 5.



data in working RAM	CV1-1	12				12→24		24→18	
	ST1	4	3	2	1	0→2	1	0→8	7
	GT1	2	1	0	0	0→1	0	0→5	4

Fig.4-3 Output Timing Diagram

1. Musical data as in Table 5 are loaded together with appropriate Time base and Tempo. They are all stored in RAM.

2. When START Key is depressed, TEMPO Clock begins to appear. At the same time, INT 1 is output, too.

3. With INT 1, the CPU transfers the data (CV1-1, ST 1, GT 1 ... in Measure 1, Step 1.) onto the working RAM (See 2-3c). Then, sending out the CV value from there, the CPU switches the Gate 1 to on if the GT-1 is not 0.

4. All outputs are held there until the next INT 1 is fed.

5. And, with the next INT 1, the CPU decrements ST 1, GT 1 placed on the working RAM by 1.

6. If the number of GT 1 comes to 0, the Gate 1 is turned to off.

7. If ST 1 also comes to 0, the CPU is now to rewrite the working memory with the next data of CV1-1, GT 1, ST 1. The output of CV, Gate from the CPU are too to correspond to the new musical data.

8. By repeating these 4 - 7, MC-8 is kept to run.

Note: Shown on Fig.4-3 is the Timing Diagram which are applicable to the units on production now. The former pROM of program 3B, 4A-4D had a little difference (units with Ser. Nos. 600000 - 790500). It is shown on Fig.4-3 by dotted line.

4-2 SYNC

MC-8 is capable of input/output of Tempo Clock through FSK Mudulation, a very useful feature when making multi-channel recording. (See "MicroComposer Instruction Manual - Synchronous Recording")

Tempo Clock produced at 8253 is FSK modulated at VCO of IC506 (MC14046), which, passing through an LPF of CR, is output from SYNC OUT Jack. (For FSK, see 2-5) The FSK frequency is the same with that of Tape Interface, to be of H = 2100 Hz, and L = 1300 Hz.

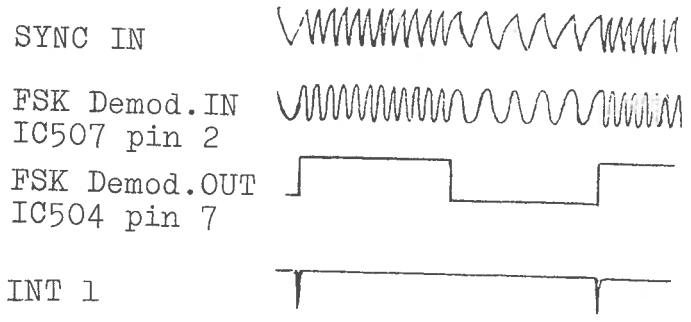


Fig.4-4 SYNC IN timing diagram
- SYNC "on" -

With SYNC IN, it passes through the BPF consisting of Q501 Q502. At IC507 (LM565) IC504 (LM311), it is FSK demodulated. When the mode is with this SYNC on, the above SYNC IN de-modulated square wave signal is used as the Time Clock in the place of that from the Counter No.0 of 8253. (During the time, Tempo VR should be inhibited.)

Time Clock, SYNC signal timing are made to be a little different between pROM of 3B, 4A-4D on the previous production and of 4E, 4F on production now. (a small circuit change is also accompanied.) In Fig.4-4, the timing of both of them, regarding Tempo Clock - SYNC, are shown.

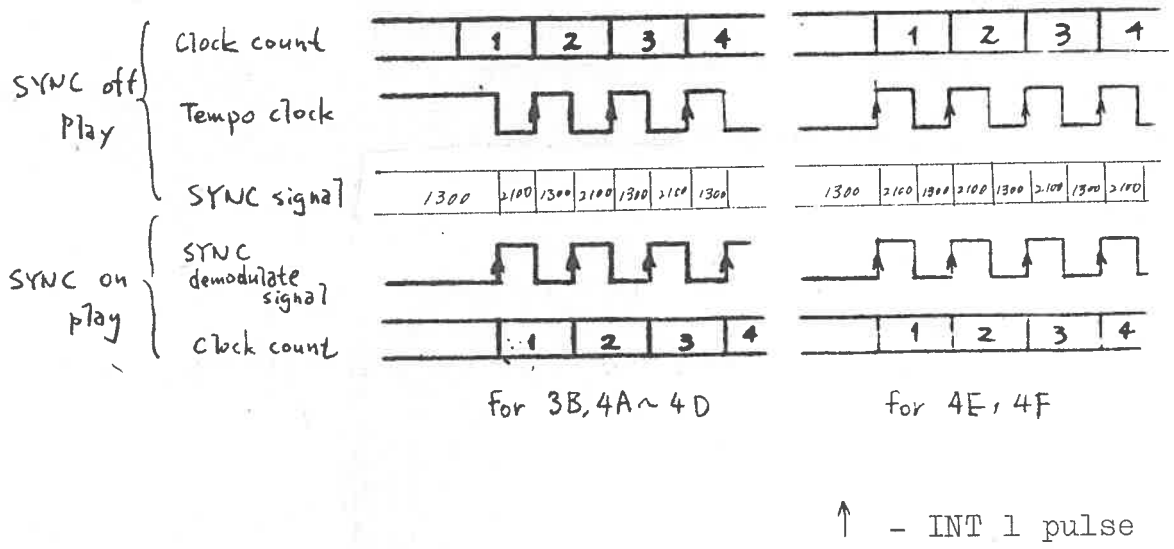


Fig. 4-5 Tempo clock, CYNC Timing chart

4-3 Time Count (INT 3)

Count No.1 of 8253 sends out INT 3 on every 25 ms, counting 50,000 clocks on ∅2 (2 MHz) produced at the Clock Generator 8224 of the CPU Board. This INT 3 is processed at the CPU for showing on Display the time passage of every o.1 sec. Also, on its every 25 ms, CV data are to be sent to the Interface.

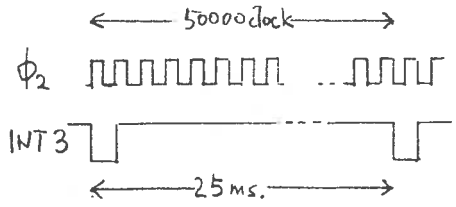


Fig. 4-6 Clock Counter

4-4 Check Terminals/Switches of SYNC, CYCLE

a) Check Terminals

To check, 1. ROMs for misprogramed, 2. reading of CPU

from PROMs for imperfection, there are two means:
1. connect to the data bus the output from the external PROM which contains program for checking purpose only.
2. store checking program into resident RAM from the audio tape through Tape Interface.

In either case, CPU can check the resident program by executing the check program as follow.

1. Using PROM: when power is on with Check A connected to ground, CPU run program from address 8000H.
2. Using tape-RAM: when power is on with Check B to ground, CPU runs program from address 41C0H.

This procedure can also be applied when checking program in the expansion RAMs.

Checking results are indicated through displays on the control panel. (refer to separate issue for how to read displays.)

b) Switches of SYNC, CYCLE

CYCLE mode can be indicated to the CPU by turning D_7 to "L". SYNC is known to the CPU by turning D_0 to "L".

In the MC-8, two kinds of signals are used as a TEMPO Clock according to the modes: output from SYNC demodulation in SYNC-on mode; Counter 0 output(IC8253) in SYNC-off mode. Selection between two outputs are also made by SYNC Switch.

Timing for the CPU to read in the data under the mode of Check A, B, CYCLE, SYNC is taken with $E300_H$.

5. INTERFACE BOARD (OP-67)

Functions of the Interface Board are as follows:

1. to output Gate 1 - 8 and MPX 1 - 6,

2. to conduct D/A conversion and output CV 1 - 8,

3. to control portamento effect on CV 1.

4. when in EXT INPUT mode, to conduct A/D conversion of CV, Gate inputs to send them into the CPU.

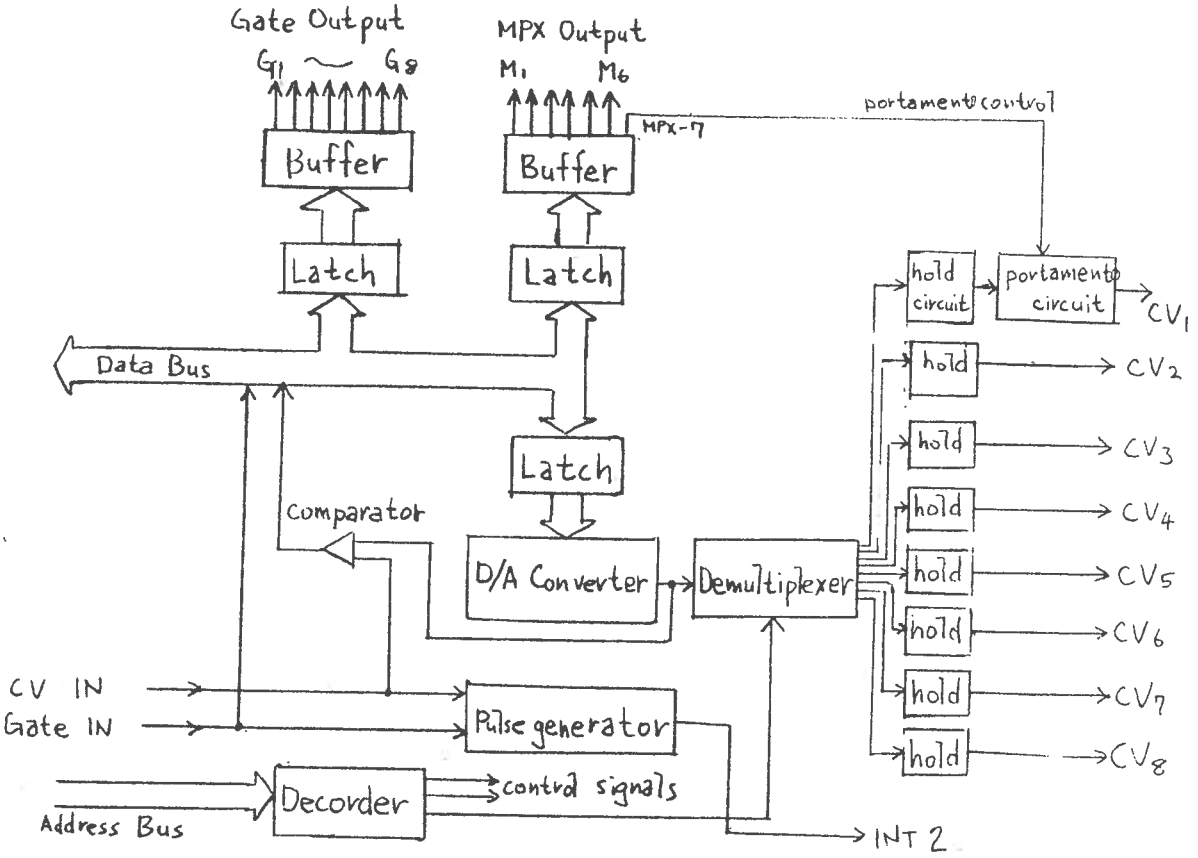


Fig 5-1 Interface Board Brock Diagram

5-1 Address Decoder

All the Gate, MPX, and CV data are output from the CPU onto the Data Bus. When the CPU sends them out, it denotes the kind of data it outputs through A_{15} - A_8 of the 8 high order bits of the Address Bus. A_3 - A_0, on the other hand, are fed to Demultiplexer to control demultiplexing of the CV data into 8 channels.

Decoding the address of A_{15} - A_8 is conducted at IC215, 216, 218 and 219. Shown below is their function table.

A_{15} A_{14} A_{13} A_{12} A_{11} A_{10} A_9 A_8	A_{15}~A_8 Hexadecimal	$\overline{WR}$	$\overline{DBIN}$	control signal
H H H L H L L L	E8	L	H	ⓐ
H H H L H L L H	E9	L	H	ⓑ
H H H L H L H L	EA	L	H	ⓒ
H H H L H L H H	EB	L	H	ⓒ (h)
H H H L H H L L	EC	H	L	(t)
H H H L H H L H	ED	H	L	ⓓ

Table 6.

ⓐ ⓑ etc. are the control signals, each being output at the address as shown above. (For instance, with the address E8xx, ⓐ becomes "H".) These are also indicated on the circuit diagram as marked with the same symbol upon the circuit line they appear.

ⓐ The strobe signal used to latch the Gate data on the Data Bus.

ⓑ The strobe signal used to latch the MPX data on the Data Bus.

ⓒ The strobe signal used to latch the CV data on the Data Bus.

(t) The signal to enable the Buffer in order to put on the Data Bus the comparator outputs of the EXT. Input of Gate, CV.

(d) not used now.

(h) The inhibit control signal of the de-multiplexer.

(See Fig.5-5)

5-2 Gate/MPX Output Circuit

a) Gate Output

When there should be any change on Gate 1 - 8, the CPU outputs the control signal (a) , together with placing the data correspond to the change of Gate 1 - 8 onto the Data Bus (for instance, when Gate 1 is "on", turning D_0 to "H").

These on the Data Bus are then latched on IC204, 205, and going through the Exclusive OR Gate

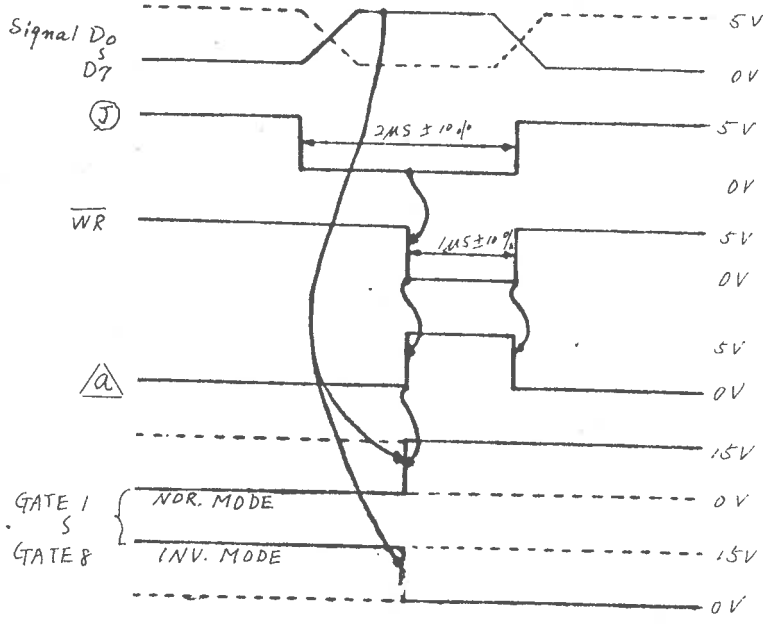


Fig. 5-2 GATE 1 ~ 8 Timing Diagram.

(IC201, 206) and Open-collector Inverter (IC202, 203), are sent out as Gate output. En route, at the Exclusive OR Gate, the polarity of the Gate signal can be inverted in either way (NOR - INV).

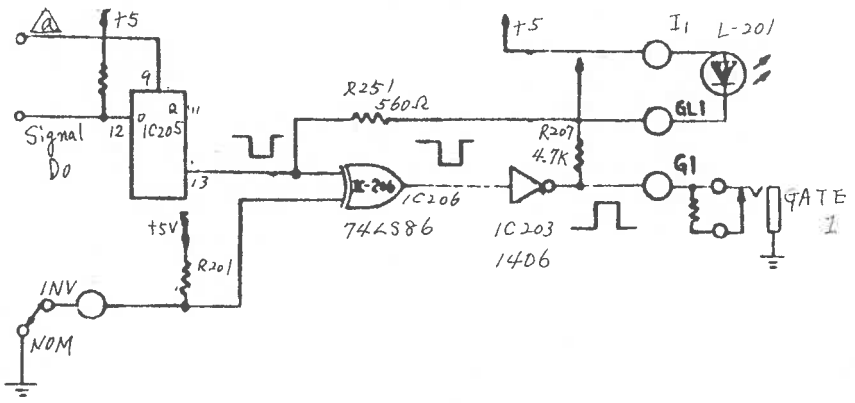


Fig. 5-3 GATE Circuit

When it is NOR, the Gate output is 15V at the time of "on" and OV when "off". When INV, these are inverted.

b) MPX Output

The strobe signal used for latching the MPX output is ⚠b . The process thereafter is just about the same with the Gate output as above. On MPX circuit, however, there is no polarity inversion circuit such as used on Exclusive OR Gate.

5-3 CV Output Circuit

a) D/A Converter

For D/A converter, with which to convert the CV digital data on the Data Bus into analog voltage for output, MC-8 employs the 12 bit D/A Converter DAC-80-CBI-V (IC 224).

With each INT 3, the CPU outputs the data of CV 1 - 8 (of 1 byte each) in time sequence onto the Data Bus D_0 - D_7. It also sends out with each output of one data the control signal $\triangle c$.

Although the CV data are stored in the form of 7 bits (127 steps, or 2^7 - 1), they are handled as 8 bits at the time they are being output, with D_0 being added holding at "L". The data then are latched on IC222, 223 with $\triangle c$ as the strobe signal.

The CV data latched there are then sent out to DAC through its 8 high order bits. Since the inputs of DAC are 12 bits, the low order 4 bits are hardware-tacked to "L". DAC 80 then outputs D/A converted analog voltage (0V - 10V) in accordance with the digital input data (of 12 bits). Offset in D/A converter, if any, can be corrected by adjusting VR201, and for width with VR202.

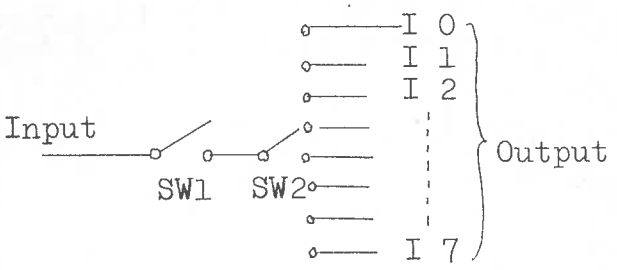
The outputs of D/A converter are sent out to input to the demultiplexer IC228 (TC4051 or MC14051).

b) Demultiplexer

The CV data are output from the CPU in time sequence from CV 8 to CV 1 with one byte each at a time, and with each one byte output, the control signal $\triangle c$ is also output. The content of the signal that are latched on IC222, 223, are therefore to change sequencially from the data of CV 8 to the data of CV 7, ... and correspondingly, the outputs of the D/A converter are from CV 8 to CV 7 ..... (See Fig.5-5)

Now, it needs here to allocate to each separate channel these data in serial output from the D/A converter, and that is the function of the demultiplexer.

The demultiplexer TC4051(MC14051) employed here can be regarded as an analog switch as shown with an equivalent circuit of Fig.5-4. It is controlled with the inputs of C,B, A, Inhibit. By altering these C,B,A, inputs in synchronous with the changes on CV channels the data of CV1-8 can be taken out independently. The C,B,A control signals are output from the CPU thru A_2-A_0 , then latched by ⓒ .



SW1: OFF when INHIBIT is at H
ON when INHIBIT pin is at L level

SW2: Channel converting Control signals C,B,A into octal digit makes contact.

Fig. 5-4 Demultiplexer

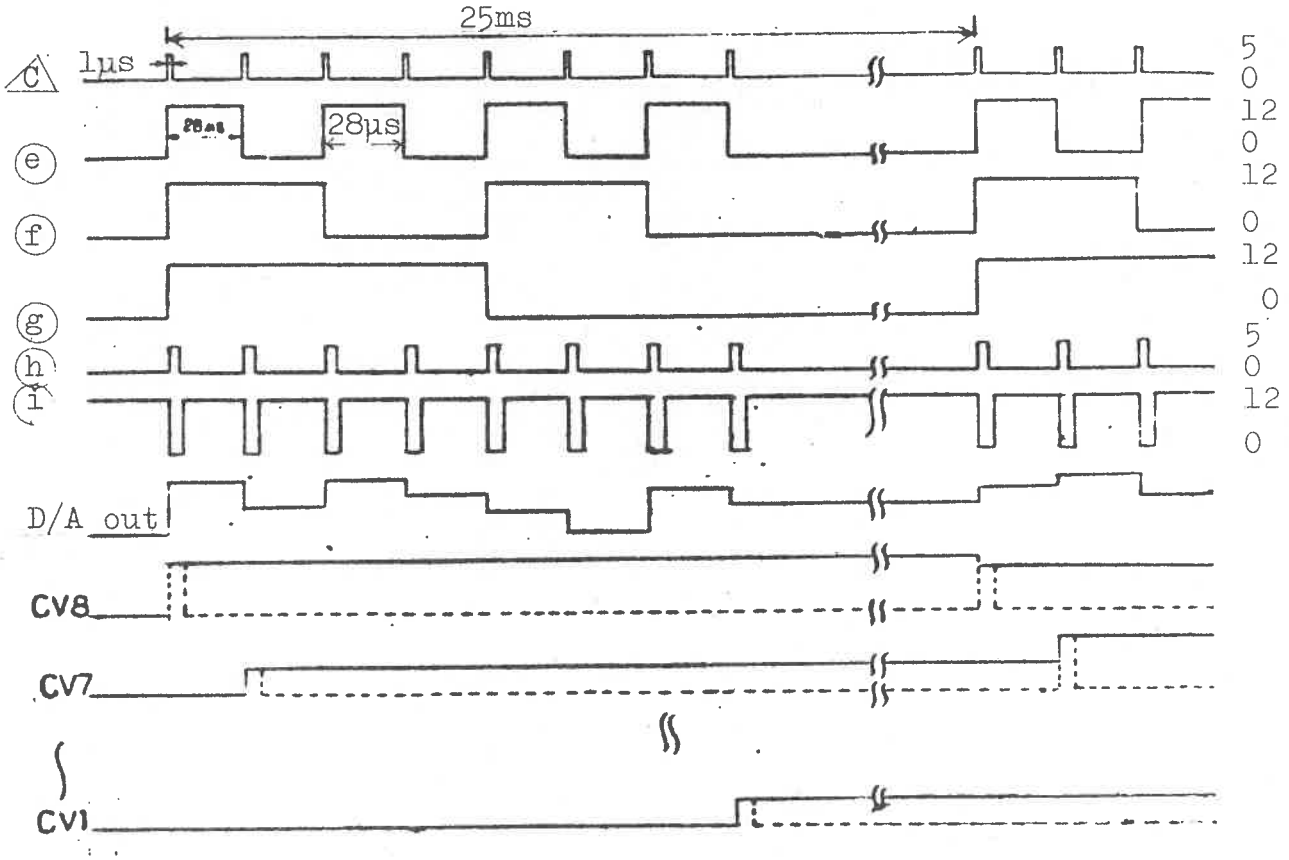


Fig. 5-5 CV Output Timing Diagram

Although the demultiplexer is said to separate the CV to each channel, its output is in fact a short pulse which appears only momentarily as shown by dotted line on Fig.5-5. To make use of it, it must be "held on" for some duration. For this, a hold circuit is provided on the path to each channel, consisting of a dual gate FET (E412) and OP amp (μPD1458 or μPC4558). CV 1 alone is further added with a portamento circuit for providing special effect on outputs. It can be switched on-off either manually or through MPX 7.

5-4 External Input

MC-8 is facilitated with the loading capability of musical data through externally played keyboard. (See "Mico-Composer Instruction Manual")

When in this mode, externally input data of CV, Gate are to be converted into digital form of CV, Step time and Gate time to suit for storing in the memory. To conduct this, however, the CPU at the same time has to know when and how these input CV or Gate are being changed.

a) Pulse Generation Circuit (INT 2)

This is the circuit where the pulse INT 2 is generated to inform the CPU of the change which has taken place on CV or Gate input.

In this circuit, CV inputs pass through the voltage follower (IC227), and after differentiated, are full-wave rectified at IC208. Therefore, if any change is occured on CV input, there appears a pulse signal at the point (o) . (Fig.5-6)

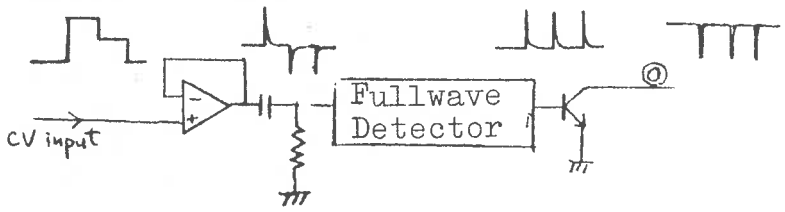


Fig 5-6

Gate input, too, is AC coupled to IC213 of the full-wave rectifier after passing through the buffer. (to be inverted when it is under Gate INV mode.) When the input Gate signal has any change (on-off or off-on) a pulse signal will appear on (o) .

If, by any chance, there are chattering on Gate input or any time lag between the CV and Gate inputs, unnecessary pulse would be produced at (o) . To eliminate this, a one shot multivabrator (IC214) is employed here in the production of INT 2. (See Fig. 5-7)

With input of INT 2, the CPU sends out the control signal (c) and reads in the data of CV and Gate. For this, the Gate signal can go straightly onto D_7 through the buffer, but as for the CV data, it has to be A/D converted into digital data before putting on the Bus.

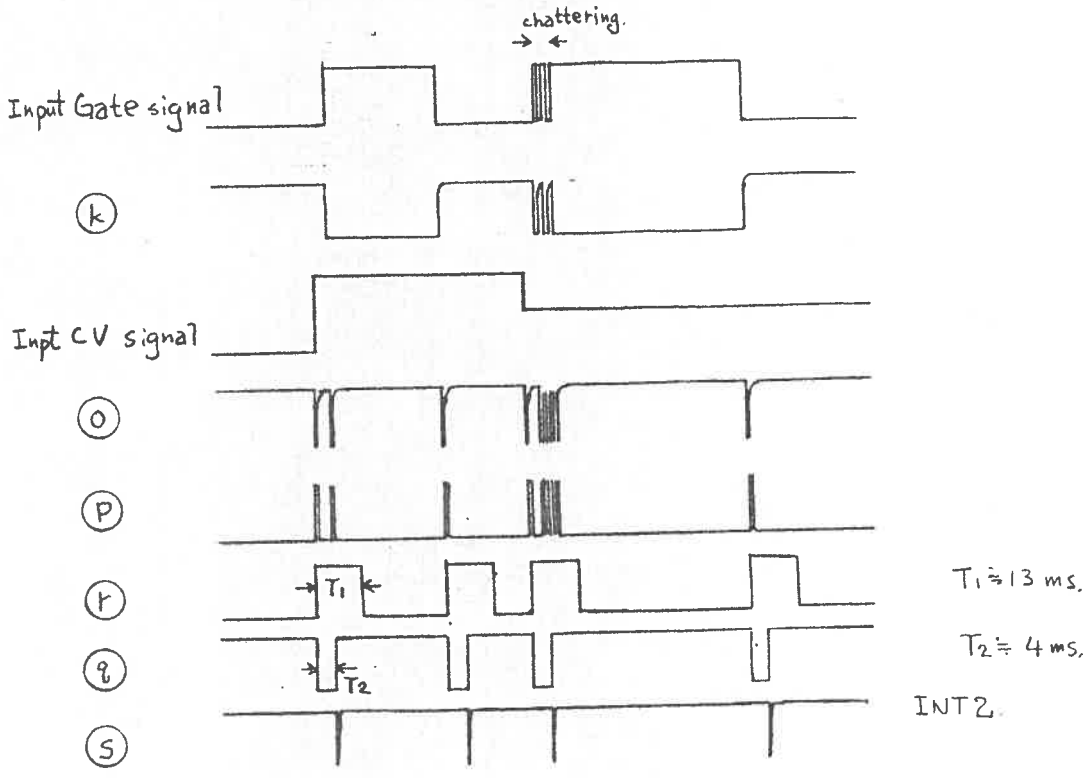


Fig. 5-7 Ext. Input Timing Diagram

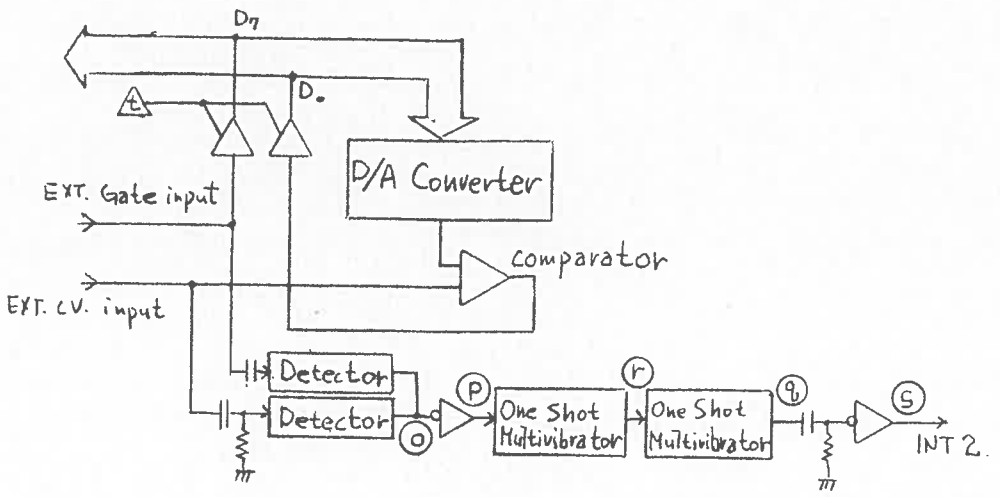


Fig. 5-8 EXT. Input

b) A/D Conversion

The input CV on EXT Input mode must firstly be converted into digital form. This A/D conversion employed on MC-8 is called the successive approximation method which uses the same DAC 80. To say briefly, the way can be explained as follows:

The CPU outputs the CV data (1000 0000) for the first comparison, to correspond to half a value of the CV output in its full scale. It is D/A converted (to become about 5.33V) and put into the comparator for comparison with EXT Input CV. The comparator outputs the result onto D_0 which is then read by the CPU.

When on comparison the input EXT CV is found to be higher than the D/A output from the CPU, the comparator outputs "H", with which the CPU is to decide the data to be "0" for the highest bit.

On the following step, the CPU firstly outputs again a trial data of half a value of the next (0100 0000), then follows the same comparison to decide the data (0 or 1) for the next highest bit. It then goes on repeating the same until all data for 7 bits become to be decided one by one.

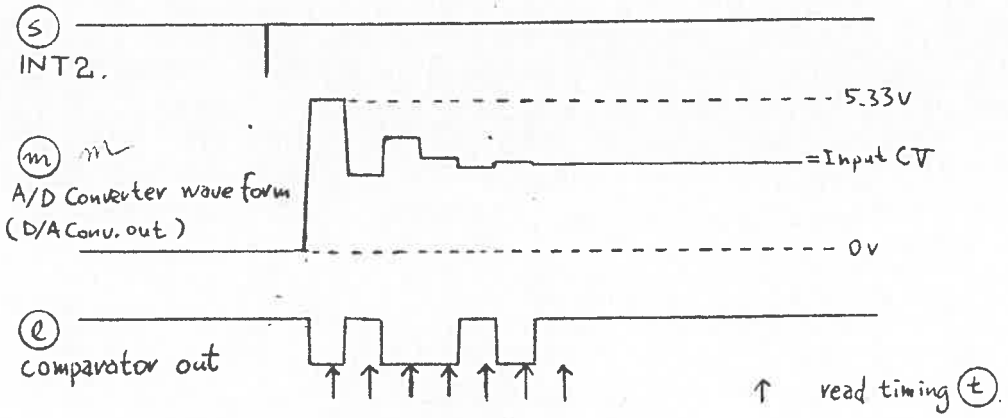


Fig 5-9 A/D Convertor

6. OPTIONAL MEMORY BOARD (OP-77)

OM-8 (OP-77) is the Optional RAM Board for MC-8. It had been in fact an optional equipment at first, but is now made to be included into the standard equipment.

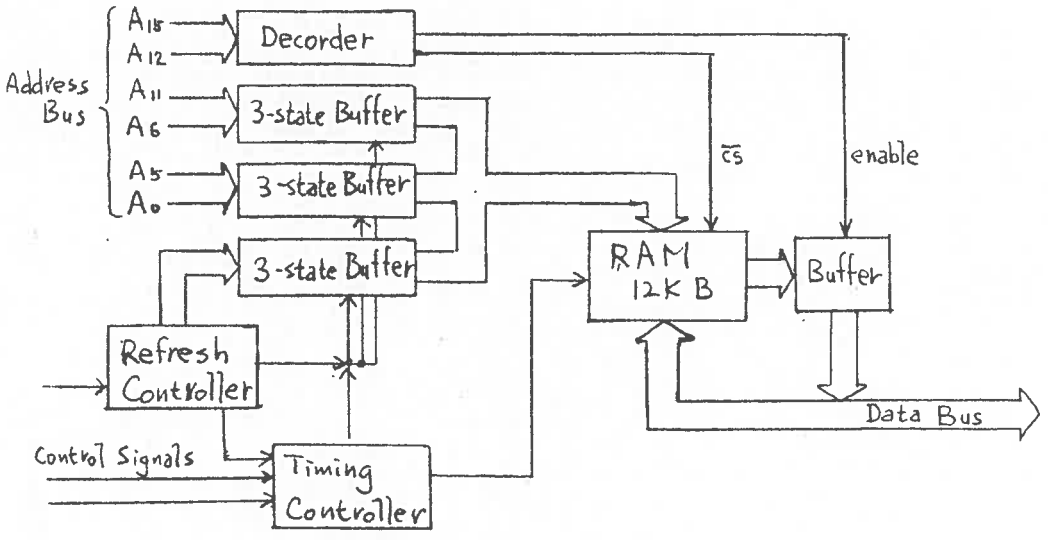


Fig 6-1. Brock Diagram of OM-8

6-1 RAM

μPD414D (or, TMM415P-4) is used on OM-8 as its memory chip, which is a 4096 word x 1 bit dynamic RAM (for dynamic, see 2-3). When used 8 chips in parallel, 4 kB (4096 word x 8 bit) memory capacity can be obtained. 24 chips are used on OM-8 securing 12 kB memory addresses alloted to 5000 - 7FFFF.

Although 12 bits are necessary for the address lines of μPD414D (or, TMM415P-4), this is achieved by dividing them in two groups of 6 bits each when making addressing, due to its limited pin numbers.

The low order 6 bits (A_5 - A_0) are used to input to RAM in synchronous with the Row Address Strobe (RAS), while the other high order 6 bits (A_{11} - A_6) are with the Column Address Strobe (CAS) signal.

Since μPD414D (or, TMM415P-4) is a dynamic RAM, it needs to be refreshed periodically in order to maintain data not to volatilize by leakage. This is done by activating $\overline{RAS}$ to "L" once in less than 2ms through 64 possible address selections with the bits A5-A0.

6-2 Control Signals for RAM

a) Address Decoder

On OM-8, the control signals of Chip Select ($\overline{CS}$) of RAM and of Buffer Enable for data output are produced by decoding the 4 high order address bits (A_{15} - A_{12}). Shown below (Table 7) is such decoding functions.

Address Bus				A15-A12	IC $\overline{CS}$			Buffer Enable	
A15	A14	A13	A12	Hexadecimal	620-627	630-637	640-647	$\overline{DBIN}$ H	$\overline{DBIN}$ L
L	H	L	H	5	L	H	H	H	L
L	H	H	L	6	H	L	H	H	L
L	H	H	H	7	H	H	L	H	L
others				-4, 8-	H	H	H	H	H

Table 7 Decoder function

As seen from the above, when the address is in the order of 5000_H (5000 - $5FFF_H$), RAMs of IC620 - 627 are to be selected. And just as the same, when it is in 6000_H (6000 - $6FFF_H$), RAMs of IC630 - 637; in 7000_H (7000 - $7FFF_H$) to IC640 - 647. And, if $\overline{DBIN}$ becomes "L" at the same time with any of these addresses from 5000 - $7FFF_H$, the Data Buffer will be opened for the data to be placed and be read by the CPU.

b) Read, Write Timing

For reading and writing, μPD414D (or, TMM415P-4) has to fulfil the conditions for timing as described on Table 8.

PARAMETER		μPD414D		TMM415P-4		units
		min	max	min	max	
Cycle Time	t_{RC}	500		480		ns
Refresh Time	t_{REF}		2		2	ms
RAS-CAS Delay Time	t_{RCL}	90	135	35	85	ns
Address Hold Time	t_{AH}	80		35		ns

Table 8 Read/Write Timing

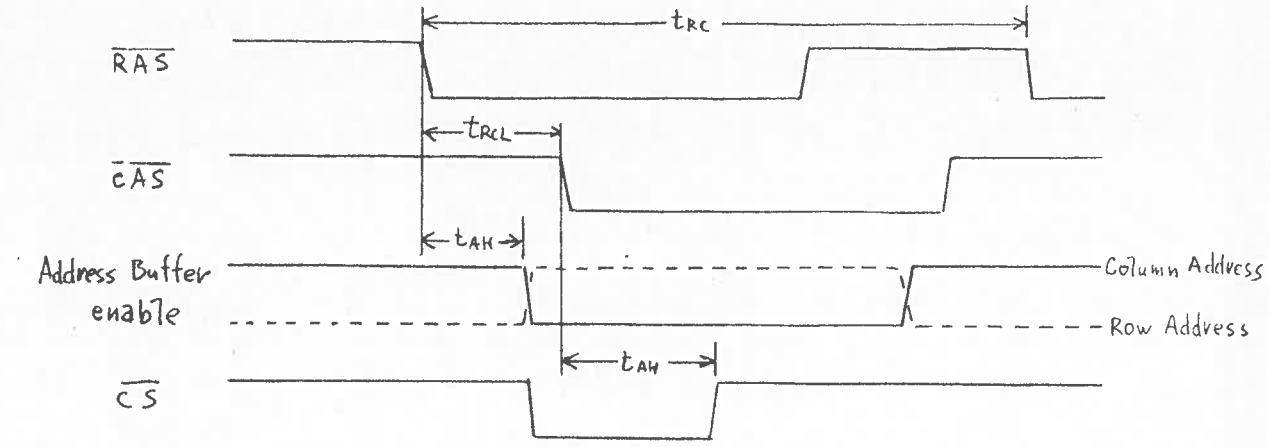


Fig. 6-2

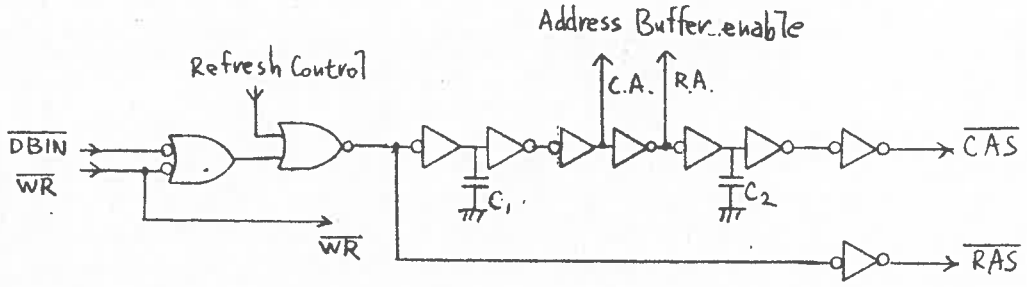


Fig-6-3

In order to fulfil the conditions of Table 8 and to get on proper timing for read, write and refreshing, the circuits as shown in Fig.6-3 are implemented to provide the signals with a time delay.

However, if there is any variation on C1, C2 capacitance and device electrical characteristics, the operation may become disturbed as such that, writing is unable, or stored data become changed, etc.
The correct operation can be reestablished through timing adjustment by trial and error replacement of C1, C2 with other values.

6-3 Refresh

At M_1 during the machine cycle (see 2-2, Status Strobe), the memory cells on the OM-8 are refreshed.
M_1 is the instruction fetch cycle where read/write is not conducted. M_1 also is the cycle that must appear in every instruction cycles (2-8μs). Therefore, if M_1 is taken with refresh address being changed every time, refreshing all the cells can be achieved within 2ms interval through all 64 address selections.

The control signal for refreshing is produced by latching $\overline{\text{DBIN}}$ when M_1 is turned to "H". (The refresh control signal is called "RCS".)

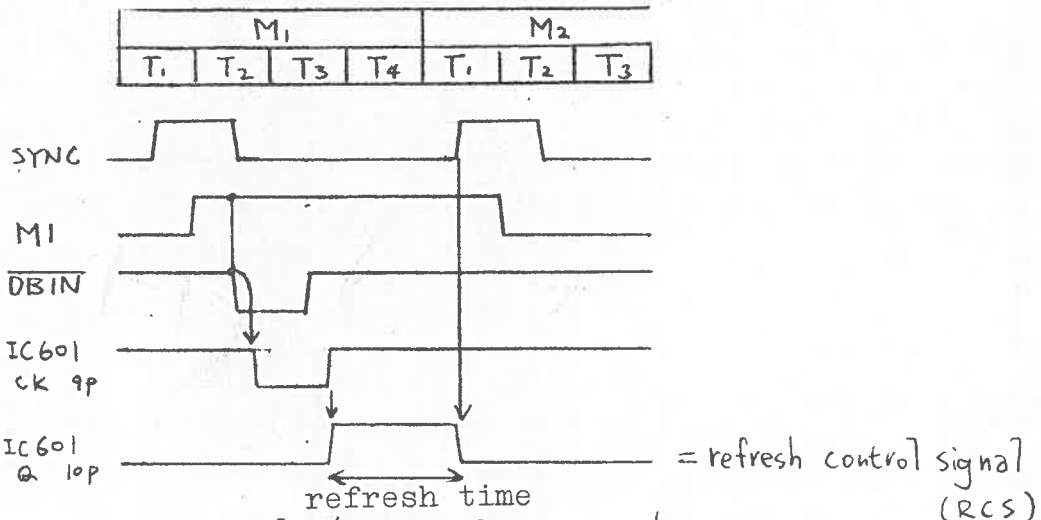


Fig 6-4 refresh control timing diagram.

With this control signal, the RAM's address inputs become isolated from the Address Bus and in their place the output of IC609 (74LS393) is connected through IC610. This IC609 is a binary counter to count the pulses of RCS in cycling through 6-bit binary numbers (full scale to $2^6 = 64$). That is, every time the CPU completes one instruction, the output of the counter IC609 is incremented by 1, which in

turn input to the RAM.

In the meantime, the RAS (Row Address Strobe) is also output from RCS (see Fig.6-3). If, then, the RCS should come 64 times within every 2 ms, the addressing of IC609 is also 64 times in every 2 ms. Since RCS's output is in every 2 - 8 μsec as CPU executes every instruction, the condition can be fulfiled. As long as the power supply is connected, the data on memory can be kept on secured.

6-4 Miscelleneous

a) Noise Protection

Since all the RAM on OM-8 work simultaneously at the time of refreshment, a current surge on that moment would occur, producing a pulse like noise on the power source circuit line. To eliminate this, following countermeasures have been incorporated.

1) A 0.1mfd ceramic capacitor is connected accross each RAM DC pin and ground for decuppling.

2) On +12V line especially, where the power consumption is most, a π (pi) type LC noise filter is inserted on the connection to the external DC supply.

7. TROUBLE SHOOTING

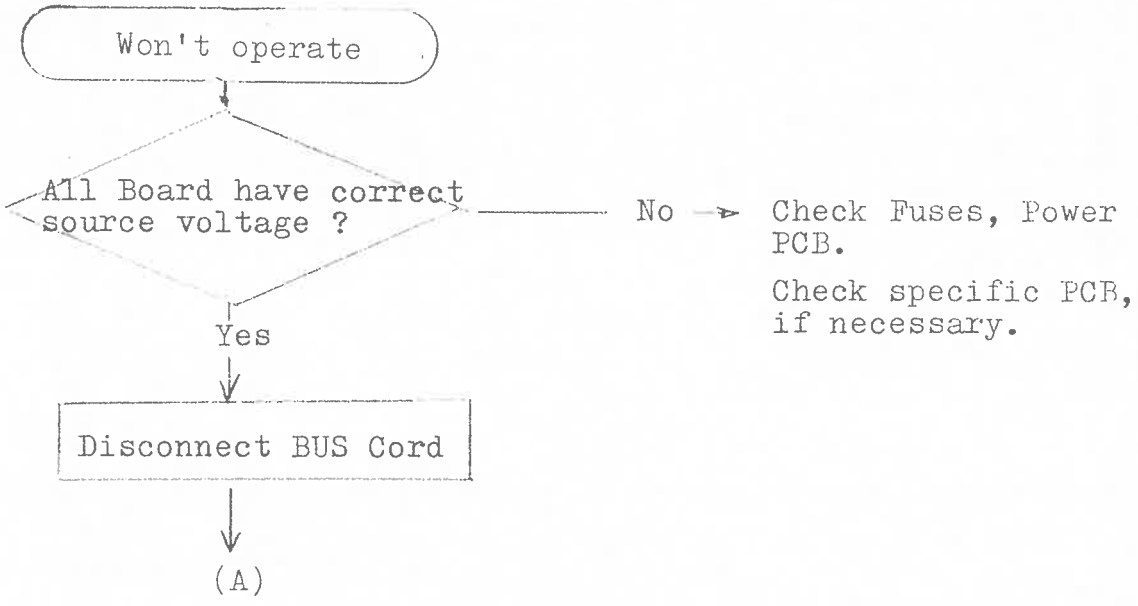
(A Process Example to isolate suspicious device)

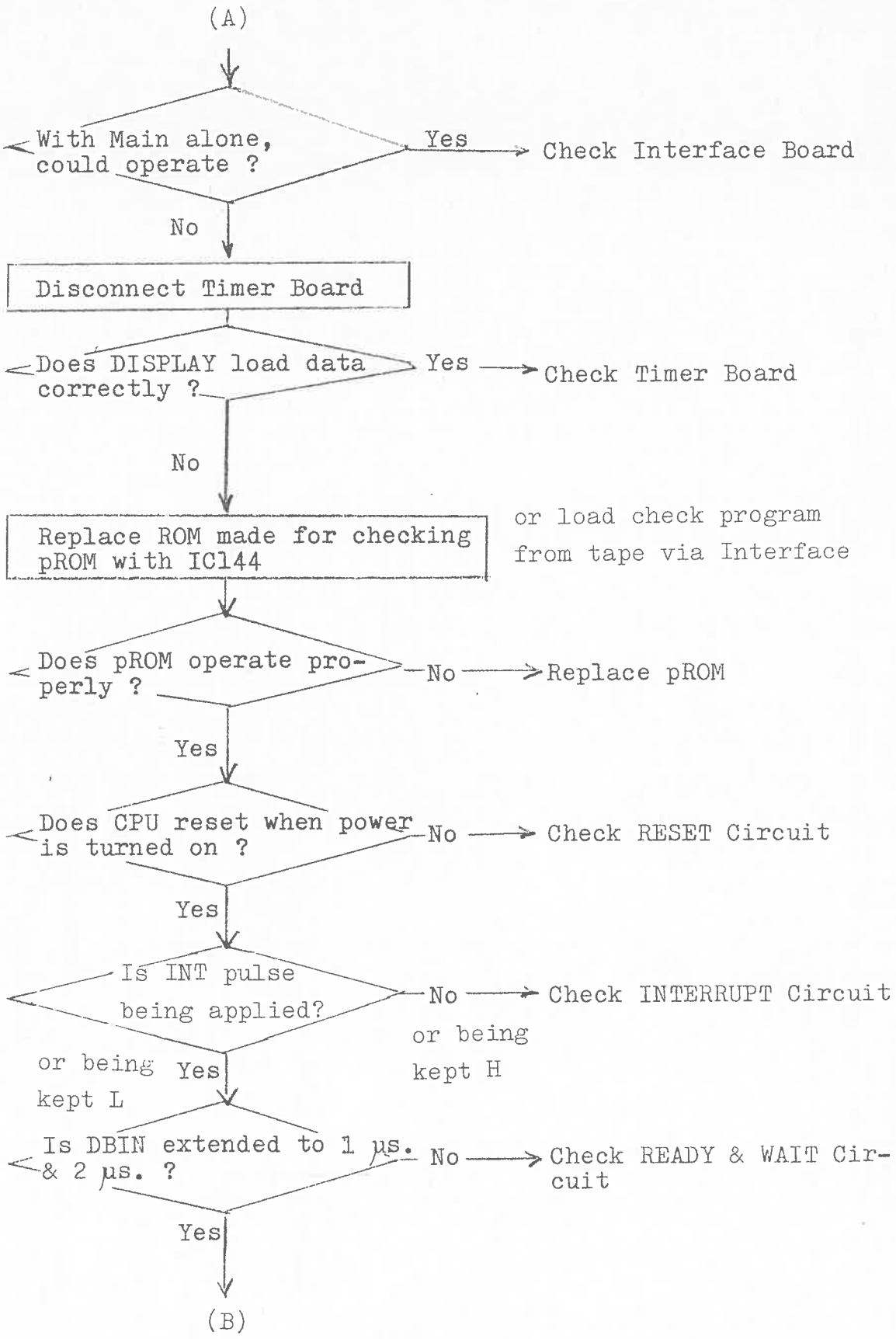
Troubles on MC-8 may be that a troubled place appears quite clear or that it won't operate and no clue can be find.

The former will be that, CV_5 does not output, or data could not be loaded from the external tape, or OM-8 does not work, etc. In these occasions, we can see the place to check such as, for CV5, the wiring from demultiplexer - to Hold Circuit - to Jack, or, otherwise, among within the Tape Interface Board, or on OM-8, etc.

But, with troubles on the CPU or its vicinity, they would so often cause no operation at all; that is, even when the Power switch is turned on, there is no display or, if it appear, would only show a meaningless random display alone; or, a phenomenon provided is only to show no loading at all with pressing keys.

We are discussing below on how we can proceed with checking on such occasions:




(A)
With Main alone, could operate ?
Yes
Check Interface Board
No
Disconnect Timer Board
Does DISPLAY load data correctly ?
Yes
Check Timer Board
No
Replace ROM made for checking pROM with IC144
or load check program from tape via Interface
Does pROM operate properly ?
No
Replace pROM
Yes
Does CPU reset when power is turned on ?
No
Check RESET Circuit
Yes
Is INT pulse being applied?
No
Check INTERRUPT Circuit
or being kept H
or being kept L
Yes
Is DBIN extended to 1 µs. & 2 µs. ?
No
Check READY & WAIT Circuit
Yes
(B)