## YAMAHA COMPUTER MSX Series <br> Technical Summary

Applicable models
AX-100 Series
YIS-503F Series
CX-5M Series

## CONTENTS

1. MSX AND COMPUTER - INTRODUCTION......................................................... 1
2. CPU BLOCK.................................................................................................... 9
3. MEMORY BLOCK............................................................................................. 27
4. THE SLOT CONCEPT AND THE MSX MEMORY CONTROLLER (MMC, YM5214) ......................................................... 43
5. I/O PORT BLOCK................................................................................................ 57
6. VIDEO BLOCK................................................................................................. 83
7. POWER CIRCUIT.............................................................................................. 99

## 1. MSX AND COMPUTER - INTRODUCTION

## 1-1 COMPUTER

Recently, CPUs (Central Processing Units) have been incorporated into various kinds of products. In products such as musical instruments or audio equipment, which are not equipped with any keyboards (unlike microcomputers), the CPU controls the entire system. This control function and the hardware are exactly the same as those of general microcomputers. Normally, the MSX Music Computer CX-5M is a microcomputer system with the BASIC LANGUAGE programming capability. However, various kinds of units or cartridges are connected to expansion connectors, called "SLOTS", and the CX-5M can be used and enjoyed as a music system (or a musical instrument) or a game machine.


Fig. 1-1-1 MSX (CX-5M) System Diagram

In this way, the MSX music computer CX-5M alone can play a role of various functions. Generally, in a large scale system (Main frame), the system is composed of 5 important elements of the computer, as shown in the following.


1. Input Unit . . . . . . . . . Keyboard etc.
2. Memory Unit . . . . . . .Internal Strage Unit
3. Control Unit
4. A.L.U . . . . . . . . . . . Arithmetic Logic Unit
5. Output Unit . . . . . . . .CRT, LPT etc.

Fig. 1-1-2 5 Important Elements of Computers


#### Abstract

In small microcomputer systems including the CX-5M, a LSI (8-bit CPU for CX-5M) is used. It is referred to as the CPU, and it combines the Control Unit and the A.L.U in the above figure. ROMs and RAMs are used as Internal Storage Units. The Input and Output Units in the above figure are called I/O (Input/Output) Ports or I/O Interfaces, and these control the I/O of data or codes to and from external units. In this way, the block diagram of small microcomputer systems can be classified into 3 blocks, CPU, Memory and I/O port blocks.




Fig. 1-1-3 Microcomputer System Block Diagram

$$
-3-
$$

## 1-2 INTRODUCTION TO CPU OPERATION

The operation of the CX-5M microcomputer seems to be very complicated. But it is nothing but the repetition of the following patterns: The CPU fetches and executes, one by one, Z-80A OP (Operation Codes) which are written in machine language and are stored in the ROM or RAM, in the exact order they have been programmed. At this time, there are only 4 basic operations of the CPU as in the following:

1. The Transfer of Codes and Data.
2. Logical Operation (AND, OR, EOR etc.)
3. Comparison
4. Programming, Jump (including IRQ processing)

The control unit within the CPU allows the CPU to execute the above 4 operations in the proper order and in the following manner.
The CPU:

1. Specifies the head address of the ROM or RAM (Memory Element) which has OP codes.
2. Reads the OP code stored at the address into the CPU.
3. Interprets the OP code
4. Following the OP code, accesses memories (ROM, RAM) and the specified addresses of I/O ports, or reads the data in the specified address into the CPU and performs arithmetic operations.

The above series of processes are always executed by the CPU of the CX-5M whatever role the CX-5M may play.

## 1-3 CX-5M SYSTEM BLOCK DIAGRAM

The block diagram of the CX-5M system is shown as in the following.


Fig. 1-3-1 CX-5M Block Diagram

The MMC chip ( 40 pin LSI) of the CPU Block (Fig. 1-3-1) is custom LSI and is called the MSX-Memory Controller. It performs the following functions:

1. Address control of the SLOT area (the expanded memory area) which is essential to the main feature of the MSX computer.
2. Memory control for the entire system.
3. I/O port address control.

Almost all of the I/O blocks, as shown in the Fig. 1-3-1, consist of LSis with 40 pins; especially, the LSI which is called the PPI chip which has $3 \times 8$ bits, parallel ports, and mainly functions as the I/O ports of the keyboard and the CX-5M SLOT control port.


| 1 PAGE : ROM2 | BASIC INTERPRETER PROGRAM |
| :---: | :---: |
| : A area | YRM101~103 or GAME ROM etc. (ROM Software Cartridge) |
| $\emptyset$ PAGE: ROM1 | MSX System Monitor Program (MSX BIOS) |
| : B area | FM Unit ROM (SEG-01 ROM) (Music System Monitor PROGRAM) (Music BIOS) |

Fig1-3-2 MSX (CX-5M) Memory area

The memory block in the Fig. 1-3-2 also shows the figure on the SLOT. This is one of the best features of the MSX (CX-5M) computer. Turning on the CX-5M will automatically allow the BASIC language to be available to the user. At this time, a program BIOS (BASIC INPUT/OUTPUT SYSTEM) called the "SYSTEM MONITOR" interfaces between the hardware and the software and the BASIC INTERPRETER program is stored within the ROM of the memory block in the Fig. 1-3-2. This ROM program will permit BASIC PROGRAMMING. Then, the use of the "CALL MUSIC" command will enable the CX-5M to be used as a music system. And the SFG-01 ROM in the Fig. 1-3-2 will function instead of ROM 1. Both the Music System Monitor and the Music System Control (Music BIOS) are stored within the SFG-01 ROM. This means that these programs will commence functioning instead of ROM 1, enabling the Music System to start its operation. Next, ROMs for the YRM102, 103 and 101 shown at SLOT 1 of the Fig. 1-3-2 can be operated instead of ROM 2 within the memory block of the Fig. 1-3-2.

The BASIC INTERPRETER program occupies all of ROM 2 and can not be operated when the Music System is activated. Thus, ROMs for the YRM102, 103 and 101 are inserted into this area to allow switch over into various kinds of systems.
The following chapters deal with the hardware of the CX-5M, centering upon the 3 important blocks of the microcomputer system.

In this chapter, explanations are given regarding the CPU and its peripheral circuits which make up the central part of the Music Computer CX-5M.

## 2. CPU BLOCK

2-1 CPU Peripheral Circuit Block Diagram


Fig2-1-1 CPU (Z-80A) BLOCK DIAGRAM

$$
-9-
$$



Fig. 2-1-2 ADDRESS DECODER AND MEMORY, I/O PORT BLOCK DIAGRAM


Fig. 2-1-3 CPU and MMC BLOCK DIAGRAM

## 2-2 System Clock Circuit

The MSX microcomputer system uses an 8 bit CPU which is equivalent to the Z-80A. The Z-80A is a high-speed version of the Z-80 and operates on system clock of 4 MHz . The system clock of the MSX computer system is 3.579545 MHz . The CX-5M computer system (hereon referred to as the CX-5M) enables this system clock to be generated by the clock card above the CPU circuit boards, and the system clock is supplied to the CPU and each LSI as shown in the Fig. 2-2-1.


Fig. 2-2-1 System Clock Supply Block Diagram


Fig. 2-2-2 Clock Timing Chart

This system clock is the reference clock for the entire operation of the CX-5M. Thus, it is crucial to check if the system clock is being supplied as it should be. By checking the waveforms of the system clock at the points $A$ and $B$ (Fig. 2-2-1), it can be judged if the failure occurs in the system clock generating circuit or the circuits which follow.

## 2-3 Z-80A CPU Control Bus Lines

As described in the preceding sections, the Z-80A is used as a CPU in the CX-5M. For operating this CPU, various kinds of lines are connected to it. In this section, explanations are given on various control lines called the "control bus," which are used for determining whether the CPU is functioning normally, or if it is in halt. If the CPU is in halt, it also must be determined if the CPU is in halt because of its breakdown or external failure.


Fig. 2-3-1 Z-80A Pin Assignments

As shown in the above Fig. 2-3-1, the 40 pins of the Z-80A can be classified into 3 groups and the Z-80A is equipped with a total of 13 control lines. In the MSX Computer System, the following 4 out of the above 13 control lines are not in use. These unused pins are treated in the following manner on the circuit board of the CX-5M CPU

1. $\overline{\mathrm{NMI}}$ (pin 17) Non-maskable Interrupt Request Input. This pin is pulled up to +5 V .
2. $\overline{\mathrm{HALT}}$ (pin 18)
3. $\overline{B U S A K}(\operatorname{pin} 23)$
4. $\overline{\text { BUSRQ }}$ (pin 25 Halt. This output shows that the CPU is in halt. This pin is unconnected.
Bus Acknowledge Output. This is used for DMA (Direct Memory Access) transfer. This pin is unconnected.
Bus Request Input. This is used for DMA transfer. This pin is pulled up to +5 V .


Fig. 2-3-2

As shown in the above, Non-Maskable Interrupt Request and data transier by DMA are not supported by the CX-5M.

## 2-4 CONTROL BUS

In the CX-5M, the CPU (Z-80A) transmits control information through 9 control lines. These 9 control lines can also be classified into the CPU Control Group and the System Control Group.


Fig. 2-4-1 CPU Z-80A Control Bus Terminal

## 2-4-1 System Control Bus Group

The System Control Bus Group is used for outputting control signals (of either high or low logic level) from the CPU to external devices including ROMs. RAMs, and I/O ports

## M1 (pin 27) < OUTPUT >

The M1 signal is sent out from the CPU to show that the CPU is currently reading (or fetching) an OP (Operation) code from the internal memory devices including ROMs or RAMs into the CPU.


Fig. 2-4-2 CPU Basic Timing (Machine Cycle) Cinart


Fig. 2-4-3 OP Code Fetch Cycle (M1 Cycle) Chart


## Note: Machine Cycle (M)

The following are the basic operation patterns used when the CPU executes programs. The repetition of the basic operation patterns will allow all programs (written by machine language) to be executed. The basic operation patterns are:

1. Fetching (Reading) Operation for OP codes.
2. Interpretation of OP codes.
3. READ/WRITE Operation of Memory Areas.
4. READ/WRITE Operation of I/O Ports.
5. Internal Operation of CPU.
6. IRQ (Interrupt Request) Process.

Normally, the basic operation patterns consist of a combination of the above 6 patterns. However, any basic operation pattern will start with both pattern 1 and 2 . The combination of the above patterns 1 and 2 is called the "M1 cycle." And the combination of patterns from 3 to 4 is called the " $M$ (Machine) cycle" and it is different from the M1 cycle. The M1 cycle is available for executing the following operations which can not be realized in other machine cycles.

1. Interpretation of OP Codes.
2. DRAM (Dynamic RAM) Refresh Control (See Fig. 2-4-3)

Note that the DRAM refresh control can be executed only in the M1 cycle.
Now, let's take a look at the machine cycles, using machine language programs.

3A312ø
LD A, (2031H) MEMORY


Fig. 2-4-5 LD A, (2031H) Program Sample

The program given in Fig. 2-4-6 allows data stored in the address 2031H (H = Hexadecimal) to be read and stored into the register A within the CPU. The following shows the machine cycle in that the CPU executes this OP code.


1. The CPU will fetch the OP code $(3 A H)$ during the machine cycle $M 1$.
2. The OP code fetched in M1 is interpreted within the CPU, when the external dynamic RAM chips are refreshed.
3. The data necessary for the OP code interpreted in the machine cycle M1 is read in the machine cycles M2 and M3. The address data " 20 H " and " 31 H " stored in the memory after the OP code is read into the CPU and is set at the address counter within the CPU.
4. The CPU outputs the address data " 2031 H " to the address bus in the machine cycle M4. The data stored in the address " 2031 H " is read into the internal register A within the CPU. The CPUZ-80A thus performs these machine cycles and keeps executing machine language programs. The above example consists of 4 machine cycles but generally the machine cycle varies from 1 to 6 depending on each command. Also, each machine cycle is composed of several states. Each state has the same length: one $T$ period of the system clock Fig. 2-4-6. Namely, the M1 output of the CPU system control buss is the signal outputted from the CPU while the CPU is executing the M1 machine cycle, and this output is called the M1 output signal during the instruction fetch cycle. The CPU announces that the CPU is executing the OP code fetching operation by outputting the M1 signal.

Since the M1 signal shows that the CPU is currently executing an OP code fetching operation, a checking signal on pin 27 makes it possible to know if the CPU is functioning normally.
The M2 can also be outputted on the following occasion: when there is an interrupt from an external device to the CPU and the CPU accepts this interrupt signal, the CPU makes both the M1 and IORQ signals to be low (OV) to fetch OP codes or address vectors into the CPU from the external device which requested an interrupt to the CPU.

MREQ (pin 19) and IORQ (pin 20) <OUTPUT>
MREQ indicates that this cycle involves a READ/WRITE operation of memory while IORQ identifies an input cycle from an external I/O device. These two will not go low simultaneously.

(1) CPU (Z-80A) MEMORY and I/O PORT MAP DIAGRAM

(2) $\mathrm{CX}-5 \mathrm{M}$ I/O PORT DECODER by MMC chip

Fig. 2-4-7 CPU (Z-80A) Memory and I/O Area Block Diagram

As shown in Fig. 2-4-7, the CPU is provided with two areas: The Memory Area ( $0000 \mathrm{H} \sim$ FFFFH $=65536$ Bytes $=64$ Kbytes) I/O Area ( $00 \mathrm{H} \sim \mathrm{FFH}=256$ Bytes)

Address lines A0 ~A8 are connected to both the memory and I/O areas (address lines A9 ~A15 are also connected to the memory area). Thus, if the CPU sends address data, $00 \mathrm{H} \sim \mathrm{FFH}$, to the lowermost 8 bits ( $A_{0} \sim A_{7}$ ), both of the memory and I/O areas are selected simultaneously. To prevent this, the MREQ goes low ( $O \mathrm{~V}$ ) for selecting memory areas or the IORQ goes low ( $O \mathrm{~V}$ ) for selecting I/O areas.
In the MSX-BASIC language, POKE and PEEK commands are available for accessing memory areas and IN and OUT commands for I/O areas.


Fig. 2-4-8 Timing Chart for Accessing Memory Areas


The MREQ signal, along with the RFSH signal (pin 28), goes low ( 0 V ) for refreshing the dynamic RAMs. The IORQ signal, along with the M1 signal, is also used for responding to an interrupt request***

RD (pin 21) and WR (pin 22) < OUTPUT>
There are CPU commands regarding data transfer. There are two types of data transfer: data transfer from the CPU to external devices (memory and I/O areas) and from external devices to the CPU.

1. Data transfer from the CPU to external devices: WRITE Operation
2. Data transfer from external devices to the CPU:

READ Operation

On this occasion, the data is transferred in parallef via 8 data busses. The RD and WR system control busses determine if the CPU is attempting to perform READ operation or WRITE operation with the external devices that the CPU accesses.


Fig. 2-4-10 Read/Write Mode Operation

As shown in Fig. 2-4-10, the direction of the data movement within the 8 data busses (for 8 bits) for the READ operation is opposite to that of the WRITE operation. These READ and WRITE signals are used for controlling the direction of the bidirectional buffers for busses.

RFSH (pin 28) < OUTPUT >
Dynamic RAMs are used as main memories of the CX-5M. Without the refresh operation, these dynamic RAMS would lose data stored inside. The RFSH is an output signal from the CPU (Z-80A) for refreshing the dynamic RAMSs. The RFSH signal is outputted only in T3 and T4 states (OP code interpretation) of the M1 cycle (OP code fetch cycle) which was described before. And only at this time, can the dynamic RAMs be refreshed (Fig. 2-4-3).

$$
-22-
$$

The Row Address of the dynamic RAM, which is counted by the memory refresh register "R" within the CPU, appears on the address busses in Fig. 2-4-3. The address for performing refreshing is set by this operation. The RFSH signal goes low ( OV ) at the timing shown by Fig. 2-4-3 for performing refresh of the dynamic RAMs.

As shown in Fig. 2-4-3, the MREQ signal also goes low ( 0 V ) to show that the CPU is attempting to refresh memory areas. For maintenance purposes, observe the RFSH (pin 28) signal on the synchroscope to ensure that the CPU is outputting the refresh signal for dynamic RAMs. The operation of the CPU can also be checked simultaneously.

## 2-4-2 CPU Control Buss Group (RESET, WAIT, INT)

The CPU control busses are used for externally controlling the CPU operation. Thus, these busses are essential for checking the operation of CPU.
The signals other than the RESET signal are input signals to the CPU. These signals go low at certain a fixed timing for telling the CPU to perform WAIT or INTERRUPT operation. When faults result in circuits other than the CPU and either of these CPU control busses goes low ( 0 V ) accidentally, the operation of the CPU will not be guaranteed. Especially, when the RESET signal is being held low (OV), the CPU will not function at all. Also, holding the WAIT signal low ( $O \mathrm{~V}$ ) will disable the refreshing of the dynamic RAMs and hence result in faulty data within the dynamic RAMs. In this situation, the execution of programs is not possible.

RESET (pin 26) < INPUT>
In the CX-5M, the RESET operation is performed automatically when the power is turned on. At this time each device including LSIs can be reset.


Fig. 2-4-11 Reset Circuit for CX-5M

When the power is turned on, the signal on pin 3 of the PST- 518 reaches a constant voltage in a short time. But, the voltage at point A increases gradually depending on the time constant determined by R30 and C26. The time required for this signal to reach the threshold voltage level of the TTL 74LS14 is the reset time when the power is turned on.
The reset time is approximately 200 mS in the CX-5M.


Fig. 2-4-12 Reset Timing Chart

During the reset operation, the following operations occur within the CPU:

1. Interrupt is disabled.
2. Program counter, I register, and R register are all cleared.
3. Interrupt mode is set to 0 .

Namely, initial conditions are set within the CPU. Also, as shown in Fig. 2-4-11, the RESET signal is fed to other devices to reset them. Note that only PPI (LSI) can be reset by Hight ( +5 V ) signal. When the MMC (LSI) is reset, the ROMCS signal of SLOT 0 always appears due to internal operation.
When the RESET signal of the CPU is held low, the CPU will not function at all.

WAIT (pin 24) < INPUT >
The WAIT signal is used for requesting the CPU (Z-80A) to delay operation timing. When this signal goes low ( $O V$ ), the CPU inserts $T w$, one period of clock $\Phi$, within the machine cycle when this WAIT signal is received. During the period Tw, the CPU holds the status at the time the WAIT signal is received without executing the next operations.


A: " $L$ " of M1 during Instruction Fetch cycle 1 and 2 is fetched at the rise time and WAIT is made low.
B: WAIT (" $L$ ") is sent to the CPU. The CPU fetches this signal at the fall time. Wait time Tw is inserted.
C: Internally returned to " H ".

Fig. 2-4-13 WAIT Timing Chart for CX-5M

For example, when the CPU attempts to access to external devices (ROMs, RAMs, I/O ports etc.) at a slow operation speed and needs to synchronize with their speed, the CPU inserts the WAIT timing Tw by setting the WAIT signal low ( 0 V ).
Since it is prescribed in the MSX specification that one WAIT be inserted in the M1 cycle, the CX-5M allows the WAIT signal to be fed to the CPU.


Fig. 2-4-14 WAIT Signal Generation Circuit

The WAIT signal in Fig. 2-4-13 is used for securing time for reading programs or data from ROMs which are slow in access time and is used for SLOTs $1-3$ (described later in this manual).
In the CX-5M, the M1 signal which always appears at the instruction fetch cycle (M1 cycle) is fed to the MMC (LSI) where a timing is made as shown in Fig. 2-4-14. The timing made within the MMC is fed to the WAIT (pin 26) again

When the WAIT signal is kept at low (OV), the Tw clock is continuously inserted. This leads to loss of data which was stored in the dynamic RAMs because the CPU can no longer refresh the dynamic RAMS. Thus, observing the WAIT signal is exceedingly important for checking the operation of the CPU.

## 2-4-3 Checking CPU Operation

The summary on the pins of the CPU is given in the following:

| Source voltage | +5 V and $\pm 12 \mathrm{~V}$ for V-RAM |
| :---: | :---: |
| Block $\phi$ | When the CPU fetches an instruction, an "L" pulse is always transmitted. |
| CPU27 pin (MI) | When the CPU fetches an instruction, an " $L$ " pulse is always issued. |
| CPU24 pin (WAIT) | " L " pulse input of the almost same timing as that of M1. When this pin is at low level, the CPU enters in the wait state. |
| CPU28 pin (RFSH) | Refresh pulse for main D-RAM |
| CPU16 pin (INT) | "L" pulse input of 1/60 sec cycle (in the MSX-BASIC mode) |
| MMC25, 27 pin | " H " ouput in the MSX-BASIC mode No slot is selected. |
| MMC28 ~ 35 pin | "L" output (all pins) in the MSX BASIC mode (Slot 0, MSX-BASIC Select) |
| MMC17 ~ 20 pin | RAS, CAS2, CAS3, and MPX control outputs for main D-RAM. |
| MPX . . . . RAS, CAS2, and CAS3 timing switch signal |  |
| When an error occur | he above points mainly to find out the error point. |

Fig. 2-4-15 CPU Check Points

Careful observation of the signal on each pin will allow you to determine to some extent if the CPU itself is faulty or the circuits other than the CPU are faulty. In this regard, the checkpoints shown in the Table 2-4-15 are extremely essential.

## Side Slot (SLOT 3) Connector

The FM Sound Synthesizer Units are incorporated in the side slots of the CX-5M. The connector in use is a 60-pin edge connector and the assignment of pins $11 \sim 60$ is the same as that for the upper slot (SLOT 1).

## 3. MEMORY BLOCK

This section describes the CPU of the CX-5M microcomputer system and also the system memory map. A block diagram of the memory area appears in Fig. 3-1-1.


Fig. 3-1-1 Block Diagram of the CPU, MMC, ROM, and RAM, and I/O

## 3-1 CPU (Z-80A) and reset operation

As you turn on power to the CX-5M system, the reset circuit becomes active and causes the reset pins on the CPU and the MMC to go low (OV), thus resetting them.

Immediately after the CPU is reset, all 16 bits of the address bus A0 ~ A15 go low (OV). In the system now, read operation may start with address 0000 H in the memory area. This requires that the MSX-BASIC-ROM be mapped on the memory area with 0000 H as the starting address. The MSX-BASIC-ROM ( 32 K bytes comprising 62768 addresses) corresponds to addresses 0000 H to 7FFFF in the memory area. When the CPU reads from this area, it must select the area correctly. For this purpose, the ROMCS signal is provided; it is issued by the YM5214 (MMC) which operates as an address decoder. In addition, this IC decodes and issues the select signals for slots $0 \sim 3$ (Fig. 3-1-1). Immediately, the CX-5M is reset with power-on, the CPU reads the contents of address 0000 H in the slot 0 memory area in ROM (Fig. 3-1-1), that is, the MSX-BASIC-ROM. The slot select signal from the MMC must be set now to select the slot 0 memory area.

For this reason, at power-on-reset, the RESET signal is supplied to the MMC chip to allow selection of the slot 0 memory area. Following this, the program can select other slots.

## 3-2 CPU (Z-80A) memory area and CX-5M memory map

The following is a description of the CPU (Z-80A) memory area and the CX-5M memory map.

The 8-bit CPU (Z-80A) used in the CX-5M has a 16 -line address bus and an 8 -line data bus. The 16 address lines make it possible to select memory elements corresponding to $2^{16}$ addresses ( 65536 addresses). Each memory element may be conceptualized as a box that may store eight segments ( 8 bits), each corresponding to the 'L' or 'H' states.


Fig. 3-2-1 Z-80A (CPU) memory area

In the CX-5M system, the 64K byte memory space is controlled by splitting it up into four 16 K byte blocks. The memory may be extended by a 16 K -byte block at a time. To select a 16 K -byte block, you need bits A14 and A15 of the address line.

As the figure shows (Fig 3-2-2), fourteen addresses lines A0 ~ A13 allow you to specify any one of the 16384 addresses in any one of the four blocks. However, the 14 addresses lines alone do not make it possible for the system to ascertain which block (0page $\sim 3$ 3page) you are selecting the address from. To ascertain this, the system must decode the two bits A14 and A15. This makes it possible to select any one of the addresses in the 64 K byte (65536) space. (Fig. 3-2-2).


Fig. 3-2-2 CX-5M System memory area

Suppose that the CPU designates address A 500 H on the address bus. If ' A ' in hexadecimal is converted into binary, we get $A H=10=1010 B$. The bit pattern is ${ }^{A 15}$ (101001010000) A0. Since bit A15 $=1$ and bit $A 14=0$, the address is selected from block 2 page (RAM 2) (Fig. 3-2-2).

To decode the two most significant bits A15 and A14 and select the memory device in the areas $0,1,2$, and 3page, the CX-5M decodes and issues the corresponding chip select signal by means of the MMC chips.

## 3-3 CX-5M MSX-BASIC-ROM and RAM area

The 64K byte memory area in the CX-5M is referred to as slot 0 . The memory map of this area appears in Fig. 3-3-1.

ROM stores the following programs, MSX-BASIC being activated as the power to CX-5M is turned on.

1. MSX-OS system monitor (program controlling the entire CX-5M)
2. MSX-BASIX Interpreter (runs the MSX-OS and the BASIC program)

MSX-OS is referred to in general as the monitor program. It provides an interface between the external I/O units (TV, CRT, keyboard, CMT, etc.) and the CPU. Furthermore, it controls the entire CX-5M system. In this way, it is a program performing very important operations. As for the MSX-BASIC interpreter, it compiles the user coded BASIC programs into the machine language program that the CPU can understand. Again, MSX-OS may be considered as a group of subroutines written in the machine language. For a user running a monitor subroutine, a uniform access method is offered for the part of the programs called MSX-BIOS. This assures compatibility among the software products.

Let us now see the hardware feature related to the mapping of the ROM and D-RAM into the memory space and access to the ROM and D-RAM.


Fig. 3-3-1 Slot 0 Memory Area

## 3-3-1 ROM area (MSX-BASIC ROM 0000H ~ 7FFFF 32K byte)

In the MSX ROM, chips are selected by setting CS and OE low. As Fig. 3-3-1 shows, signal ROMCS is entered in the ROM.

Let us imagine that the CPU has issued address data to select 6 A 55 H and EA55H on the address bus.


Fig. 3-3-2

The MSX-BASIC-ROM is provided with fifteen address lines input pins (AO ~A14), making it possible to select 215 ( $=32768 ; 32768: 1024=32 \mathrm{~K}$ bytes) memory areas.

As in the above example, if the CPU selects addresses 6 A55H or EA55H, address 6 A55H alone will be selected in either instance within the ROM because address lines A15 are not connected within ROM. In order to avoid this, address A15 is used as dip select signal


|  | ROM |
| :---: | :---: |
| 0 |  |
|  | 0000 |

$\uparrow$
A15: ADDRESS LINE
Fig. 3-3-3

- 33 -

In other words, as Fig. 3-3-3 shows, the ROM may be selected according to the state (high or low) of A15. This,


Fig. 3-3-4

Since A15 is low("0") in the interval 0000 H and 7FFFF and high (" 1 ") in the interval 8000 H and FFFFH, input of A15 in ROMCS makes it possible to select one of the addresses in the 64 K byte space. Have a look at the block diagram of the MXS memory are(Fig. 3-3-1). You will notice here that ROMCS entering MSX-BASIC-ROM is not from address line A15 but from ROMCS of MMC. This is because the following conditions apply simultaneously when the CPU selects MSX-BASIC ROM:

1. Slot 0 is selected
2. An access to the memory
3. ROM contents are valid only in the Read mode (when the data and the program are read).

In the above conditions, the program running at the time causes input of the following control signals from the CPU to MMC:

1. 8-bit signal from PPI port $A$ to select slot 0 (see later)
2. Control bus MREQ goes low when the CPU selects a memory area.
3. RD is low when the CPU is in the Read mode. These control signals and the state of address line A15 are entered in the MMC where these are decoded. These may be issued from the ROMCS pin of the MMC if the prevailing conditions permit.


Fig. 3-3-5

## 3-3-2 RAM AREA

Capacity of the memory area available to the user is indicated at Left top of the screen as soon as the CX-5M is connected to power and MSX-BASIC is activated. The data appearing here will show that the space available is smaller than 32 K bytes. This is because the area available is a temporary one, forming part of RAM, being given to the user when the CPU (Z-80A) is on. This particular RAM area is referred to as the system work area, occurring from F380H to FFFFH in the CX-5M. If you use BASIC, you need not concern yourself about this area.


Fig. 3-3-6 Block Diagram of the RAM Area

Read/write operations in the RAM area of CPU does not differ basically from ROM read operations.

In the CX-5M, the RAM used is a D-RAM. For this reason the read/write operation here calls for the following:

1. Periodic refreshing
2. Provision of addresses and data in two rounds (instead of one)
3. There are read and write modes for these operations.


Fig. 3-3-7 RAM Peripheral Circuit

As explained earlier, the ROM (MSX-BASIC-ROM) used is a 32 K byte maskROM. It occurs in the memory from 0000 H to 7 FFFH . As for the RAM, it consists of two 16 K byte blocks, adding up to a 32 K byte area. It is assigned the memory area extending from 8000 H to FFFFH.

DATABUS


Fig. 3-3-8 D-RAM Memory Block Diagram

Have a look at the D-RAM (Fig. 3-3-g). First take the point depending on whether the signal entering the SEL pin is low or high, either the group-A pins $(1 A \sim 4 A)$ or the group-B pins $(1 B \sim 4 B)$ are led to output pins IY $\sim 4 Y$. At the SEL pin, input of the MPX signal takes place from the MMCLSI chips. Accordingly, address and data are switched between $A 0 \sim A 7$ and $A 8 \sim A 13$ To the address bus selector, 14 address lines ( $A_{0}$ $\sim A_{13}$ ) are connected. This indicates that a memory space of $2^{14}$ or 16384 bytes, that is 16 K bytes, can be selected within D-RAM by using address lines $A_{0} \sim A_{13}$. However, only eight address inputs $\left(A_{0} \sim A_{7}\right)$ are possible for D-RAM as the terminals show. This is because D-RAM receives the address data at two timings: RAS timing and CAS timing. The address data is divided into two timing schemes by the TTL chips (74LS157) in the address line selector.


Fig. 3-3-9 Address Line Selector

Let us now turn to the operations of the TTL chips.


Fig. 3-3-10 TTL 74LS157 Internal Block

Next have a look at the internal block diagram for the U-KAM chips (MB81416-12).


Fig. 3-3-11 MB81416-12 Internal Block

The following explains the D-RAM chip operations with reference to the timing chart. (CPU READ MODE)

1. The CPU (Z-80A) issues addresses data (A0 ~ A15) to the address bus (A in Fig. 3-3-12).
2. At the trailing edge of the RAS signal from the MMC chip ( $B$ in Fig. 3-3-12), the contents of address A0 $\sim$ A7 are latched in decoder R (Fig. 3-3-11).
3. From MMC clip, the MPX signal is sent to the TTL chips (74LS157), and address A8 $\sim$ A13 will appear at address pins AO ~A7 of the D-RAM (C in Fig. 3-3-12).
4. When CAS enters D-RAM from the MMC chips the contents of address A8 $\sim$ A13 are latched in decoder C (Fig. 3-3-11), (D in Fig. 3-3-12).
5. When CAS is entered in the D-RAM chip, data stored in the memory cell is output to data output pins D0 to D4. (Ein Fig. 3-3-12)


Fig. 3-3-12 D-RAM Timing Chart

As Fig. 3-3-11 shows, the D-RAM chips (MB81416-12) process 4 bits of data simultaneously. For this reason, the D-RAM chips are paired into two or four, respectively, an 8-bit, 16K-byte memory configuration (Fig. 3-3-8).

The D-RAM pairs are used to form a 32 K byte memory in the CX-5M.
Next, we will take up address decoding for the D-RAM chips.

As we have already seen, the internal addresses in the D-RAM chips are selected by means of 14 address lines A0 ~A13. However, these 14 address lines cannot determine where in the 64 K byte memory (from 0000 H to FFFFH ) the respective $16-\mathrm{K}$ byte memories can be allotted. The two address lines A14 and A15 are required to determine this.


Fig. 3-3-13 MMC circuit diagram of the memory (D-RAM)

| A15 | A14 | A13 |  | $\curvearrowright$ |  |  |  |  |  |  |  |  | AØ |  |  | ADDRESS | RAM3 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FFFF |  |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | C0.0 |  |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | BFFF |  |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | $8 \emptyset 0 \emptyset$ |  |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7FFF |  |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4000 | MSX |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3FFF | ROM |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 |  |

Fig. 3-3-14 Memory allocation
As Fig. 3-3-14 shows, the MMC chips allot the two D-RAM chips to the memory.
Table ... Memory allocation

If we take into consideration address data $A 0 \sim$ A13 only as in Fig. it becomes evident that when the CPU sends address data as output to the 14 address lines, addresses get simultaneously selected in MSX-BA-SIC-ROM, RAM2, and RAM3. Now, if two more bits (A15, A14) are added to the 14 address lines, then we get a total of 16 address lines; only one address can be selected in the 64 K bytes ( 65536 addresses) are from 0000 H to FFFFH .

In other words, the 2-bit address lines A15 and A14 serve to assign the MSX-BASIC-ROM chips ( 32 K bytes), RAM2 ( 16 K bytes), and RAM3 (16K bytes) correctly to the 64 K byte memory area.

Figure 3-3-13 shows that those address lines A15 and A14 are decoded in the MMC chips after their entry from the CPU (Z-80A). This generates the following signals, the respective memory chips being assigned to the memory.


In the first instance above, RAM3 will be selected. In 2, RAM2 will be selected and in 3 the selection will be for MSX-BASIC-ROM.

## 4. THE SLOT CONCEPT AND THE MSX MEMORY CONTROLLER (MMC, YM5214)

Over here we shall offer a simple description of the Slots, a very important characteristic feature of the CX-5M microcomputer. At the same time, we shall have a look at the MMC (YM-5214) controlling the entire memory area in this system.


Fig. 4-1-1 CX-5M Slots Area Block Diagram

## 4-1 CX-5M memory area

The CPU (Z-80A) described so far uses a 64 K byte memory area. In the CX-5M, this memory area is allotted to the area referred to as slot 0 . It is used as ROM and D-RAM.

The following will give an idea about the 64 K byte memory area.


Fig. 4-1-2 Memory Block Diagram: 64 K Bytes in the Mainframe (CX-5M)

1. In the CX-5M, 16 K bytes each are blocked together to form units to map the memory by
2. Each block is mapped as follows, using address bits A14 and A15.

| A15 | A14 | MEMORY EREA | PAGE |
| :---: | :---: | :---: | :---: |
| 1 | 1 | $\mathrm{C} \emptyset \emptyset \emptyset \mathrm{H} \sim$ FFFFH | 3 PAGE |
| 1 | $\emptyset$ | $8 \emptyset \emptyset \emptyset \mathrm{H} \sim$ BFFFH | 2 PAGE |
| $\emptyset$ | 1 | $4 \emptyset \emptyset \emptyset \mathrm{H} \sim 7 \mathrm{FFFH}$ | 1 PAGE |
| $\emptyset$ | $\emptyset$ | $\emptyset \emptyset \emptyset \emptyset \mathrm{H} \sim 3 F F F H$ | $\emptyset$ PAGE |

Fig. 4-1-3 Mapping Table
3. Each 16 K bytes block forms a page as shown in the figure.
4. As shown in Fig. 4-1-4 if the CPU accesses pages 0 and 1 (MSX-BASIC-ROM: 32K bytes), $2^{15}$ or 32768 bytes of addresses may be selected from. These relate to 15 lines A0-A14 directly entering the ROM. These are arranged by ROMCS in memory area $0000 \mathrm{H}-7 \mathrm{FFFH}$.


Fig. 4-1-4
5. As shown in Fig. 4-1-4 for pages $2 \sim 3$, (D-RAM: 16K bytes $\times 2$ blocks) direct selection of $2^{14}=16384$ byte addresses is possible by address lines AO ~ A13. Also, by using the CAS2 signal, a 16 K byte D-RAM page 2 is mapped in the area $8000 \mathrm{H}-\mathrm{BFFFH}$ while by using $\overline{\text { CAS3 }}$, the 16 K byte D-RAM page 3 is mapped in C 000 H -FFFFH. Slot 0 (memory area in the CX-5M mainframe) and slots $1-3$ are related as snown below.


Fig. 4-ו-b Slots Map

The CX-5M can be provided with a 64 K byte memory area each for slots 1-3 also (Fig. 4-1-5). Furthermore, the MSX specifications allow three slots to be added for each of the existing slots. Thus, there is provision for altogether 16 slots ( 1 M bytes) in the memory area.
This manual covers only the configuration normally used (Fig. 4-1-5).

In the CX-5M, the individual slots are referred to as follows:

$$
\begin{aligned}
& \text { SLOT } 1=\text { UPPER SLOT (GAME SLOT) } \\
& \text { SLOT } 2=\text { REAR SLOT } \\
& \text { SLOT } 3=\text { SIDE SLOT }
\end{aligned}
$$

All these are external slots.

## 4-2 MACHINE LANGUAGE PROGRAM EXTENSION

In the following, we shall assume that the machine language program in Fig, 4-2-1 is extended into page 1 in slot 1 (with the program cartridge inserted into the upper slot).

Normally, if the MSX-BASIC language is used, the entire 64 K byte area from 0000 H to FFFFH in slot 0 can be accessed for (read/write) operations, using the address data (A0-A15) from the CPU. If, however, the program ROM is extended to the page 1 in slot 1 , then the process followed is an described below. As a result, all accesses to the page 1 block of slot 1 for read operation are switched to slot 1 .

1. When power is on, all pages are set to be selected from slot 0 .
2. Starting from slot 1 , a check is made to find out if programs are extended in pages 1 and 2 of the respective slots.
3. If a program is found to be extended in some area, a check is made to ascertain the program type and the data is written in the specific work area provided for the purpose in the mainframe (D-RAM).
4. The LSIs and other circuit components in the CX-5M are initialized.
5. Control is transferred to the program extension. Or, the MX-BASIC language mode is set up.

When the CPU has issued as output addresses 0000 H -FFFFH, the following will take place.

1. $0000 \mathrm{H}-3 F F F H$ (page 0 ) $\qquad$ SLOT 0
2. $4000 \mathrm{H}-7 \mathrm{FFFH}$ (page 1)
SLOT 1
3. $8000 \mathrm{H}-$ FFFFH (page 2,3 ) SLOT 0


Fig. 4-2-1 Memory Selection

In this way, a memory area with a total capacity of 64 K bytes is built up.

Normally, programs are extended in pages 1 and 2 of slots 1-3. The MMC chips issue the following signals to select (chip select) memory areas:

1. CS 1 $\qquad$ 4000H-7FFFH
2. CS 2 8000H-BFFFH
3. CS1,2 $\qquad$ $4000 \mathrm{H}-\mathrm{BFFFH}$

## 4-3 Slot select signal

The following are the essential considerations underlying slot selection. As described so far, the slot select signals (SLOT 1, SLOT 2, and SLOT 3) determine the slot from which one of the pages ( 16 K bytes each) is to be selected.

For example, if we consider page 0 alone as in Fig. 4-3-1, we have


Fig. 4-3-1 0 Page Block Map

If page 0 is selected from siot 1 (Fig. 4-3-1), and the CPU issues an address from 0000 H to $3 F F F H$, then SLOT 1 signal along goes low while the other signals all go high.

This is what occurs in the other pages also.

Next, let us see how the slot select signal is generated.


Fig. 4-3-2 Slot Signals Generated by the PPI and MMC

As Fig. 4-3-2 shows, the PPI and MMC chips generate the slot select signals.

The 8-bit output from port A of the PPI enter the MMC as they are. The respective bits of port A are as described below:
PA0 and PA1: Determine the slot from which to select the page 0 area
PA2 and PA3: Determine the slot from which to select the page 1 area
PA4 and PA5: Same as above except that page 2 or 3 is
PA6 and PA7: Substituted for page 0 or 1.
A slot is selected as follows according to the state of the two bits of PPI port A:

| PA0 | PA2 | PA4 | PA6 | PA1 | PA3 | PA5 | PA7 |  |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| SLOT0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | SLOT1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
|  | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| SLOT2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1

If all the pages are selected from slot 0 , port $A$ will be as follows:

| PA0 | PA1 | PA2 | PA3 | PA4 | PA5 | PA6 | PA7 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



Fig. 4-3-3

If (1), (6), (3), (4) must be selected in Fig. 4-3-3:

| PA0 | PA1 | PA2 | PA3 | PA4 | PA5 | PA6 | PA7 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\ldots$ |  |  |  |  |  |  |  |$\ldots 04 \mathrm{H}$

If (13), (6), (3), (4) must be selected in Fig. 4-3-3:

| PA0 | PA1 | PA2 | PA3 | PA4 | PA5 | PA6 | PA7 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\ldots$ |  |  |  |  |  |  |  |$\ldots .07 \mathrm{H}$

In this way, slot data to be selected for the respective pages are set in port A I/O Address A8H of the PPI and entered in the MMC. As a result, the respective slot select signals are issued by the decode circuit shown in Fig. 4-3-4.


| LS153 |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| INPUT |  |  | OUTPUT |  |
| G | SELECT |  |  |  |
|  | B | A | 1Y | $2 Y$ |
| H | X | X | L | L |
| L | L | L | 1C0 | 2C0 |
| L | L | H | 1C1 | 2 C 1 |
| L | H | L | 1 C 2 | 2C2 |
| L | H | H | 1 C 3 | 2 C 3 |


| LS139 |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| INPUT |  |  |  | OUTPUT |  |  |
| G | SELECT |  |  |  |  |  |
|  | B | A | $Y \emptyset$ | Y1 | Y2 | Y3 |
| H | X | X | H | H | H | H |
| L | L | L | L | H | H | H |
| L | L | H | H | L | H | H |
| L | H | L | H | H | L | H |
| L | H | H | H | H | H | L |


| SELECT PAGE |  |  |  |
| :--- | :---: | :---: | :---: |
| A15 A14 PAGE PPI PORTA  <br> L L ØPAGE $0000 \mathrm{H} \sim 3$ FFFH CS $\varnothing$ L, H <br> L H 1PAGE $4000 \mathrm{H} \sim 7$ FFFH CS1L, H <br> H L 2PAGE $8000 \mathrm{H} \sim$ BFFFH CS2L, H <br> H H 3PAGE $\mathrm{COOOH} \sim$ FFFFH CS3L, H |  |  |  |

SELECT SLOT

| SLOT | CSL, | CSH |
| :---: | :---: | :---: |
| $\emptyset$ | $\emptyset$ | $\emptyset$ |
| 1 | 1 | $\emptyset$ |
| 2 | $\emptyset$ | 1 |
| 3 | 1 | 1 |



| PPI Reg ADR | $A 7 \sim A 2$ | A1 | A0 | $R D=$ " $L$ ", WR = ' H "' | $R D=$ ' $H^{\prime \prime}$, WR = ' L " |
| :---: | :---: | :---: | :---: | :---: | :---: |
| A8H (PORT A) | 10101X | 0 | 0 | IN | OUT |
| A9H (PORT B) | 10101X | 0 | 1 | (IN) | OUT |
| AAH (PORT C) | 10101X | 1 | 0 | IN | OUT |
| COMMAND Reg | 10101X | 1 | 1 |  | COMMAND Reg WRITE |

Fig. 4-3-4, 4-3-5 Slot-Select Signal (MSX-Specification)

The MMC chips used in CX-5M perform the following control operations:

1. Memory select for slot 0


Fig. 4-3-7
2. Slot select signal for slots $1-3$
3. Memory select for slots $1-3$


Fig. 4-3-8
4. Address select for the I/O port.

The MMC chips used in CX-5M perform the following control operations:

1. Memory select for slot 0


Fig. 4-3-7
2. Slot select signal for slots $1-3$
3. Memory select for slots $1-3$


Fig. 4-3-8
4. Address select for the I/O port.


Fig. 4-3-9

## 5. I/O PORT BLOCK

The CPU and memory blocks discussed so far meet the minimum requirement for microcomputer system operations. However, the operator cannot enter from outside data or command for CPU control, nor receive or display different types of control data items or results of operations performed by the CPU.

This raises the need for devices which receive data from outside and output results to outside.


Fig. 5-1-1 $/ 10$ Devices Block Diagram

The special areas used for connection with such devices are referred to as the input/output port area or I/O interface.


Fig. 5-1-2 Block Diagram of $/ 1 O$ Devices Ports

The I/O port allows read/write (input/output) operation just as the memory area in RAM chips do.

The CPU Z-80A used in the CX-5M has a main memory area ( 64 K bytes) and an I/O area ( 256 bytes).


Fig. 5-1-3 CX-5M I/O Port Memory Map

Data from the CPU is output to the devices on ports $A$ and $B$ through one address line (A0) (Fig. 5-1-4 (1)).
In the case of Fig. 5-1-4 (2), data is output to the four ports designated by two address lines (A0, A1).


Fig. 5-1-4 (1)


Fig. 5-1-4 (2)

- 59 -

In this way, the Z-80A CPU is connected through eight addresses lines to $256 \mathrm{I} / \mathrm{O}$ port addresses.

The CX-5M I/O port block diagram appears below.


Fig. 5-1-5 I/O Port Block Diagram

## 5-1 I/O PORT CONTROL BY THE MMC CHIPS

Except the LPT (printer) port, all the I/O ports are implemented on the 40-pin LSIs (Fig. 5-1-6 (1)). The MMC chips control all these I/O ports.


|  |  | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| PPI | ABH | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
|  | AAH | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
|  | A9H | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
|  | A8H | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
|  | A2H | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
|  | A1H | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
|  | A0H | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| VDP | $99 H$ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
|  | $98 H$ | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|  | $91 H$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
|  | $90 H$ | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Fig. 5-1-6 (1) and (2) Block Diagram of IIO Ports and the MMC

IORQ entering the MMC chips is a system control signal indicating that the Z-80A CPU has selected the I/O port area.

Address lines A3 ~ A7 are decoded inside the MMC chips on the basis of which I/O ports are assigned to the respective areas $00 \mathrm{H}-\mathrm{FFH}$.


Fig. 5-1-7

- 62 -


## 5-2 I/O Ports and Their Peripherals

## CX-5M I/O port address map

According to the MSX specifications, the 256 byte Z-80A I/O port area from 00 H to FFH is reserved for the standard MSX system devices as follows:


Fig. 5-2-1

Addresses 80 H to FFH are set as above as $\mathrm{I} / \mathrm{O}$ port addresses. Addresses 00 H to 7 F can be used freely. Basically, however, for I/O devices not specified above, I/O port addresses must be set in the memory space by applying the memory mapped $/ / O$ method. This is because the system will not operate if the I/O port address space (which is not subject to slot selection) is used simultaneously with some other device.

## 5-3 I/O Peripheral Circuit

The following devices are interfaced to the Z-80A peripheral circuit (microcomputer system) and the I/O peripheral circuit using the I/O port addresses described in the earlier section.

## 5-3-1 I/O Devices Controlled by PPI ( $\mu$ PD8255)



Fig. 5-3-1 External I/O Configuration Based on PPI and SSG I/O Ports

$$
-64-
$$

## 5-3-2 SSG (YM-2149) Controlled I/O Devices

Pins to be controlled by PPI ( $\mu$ PD8255)

| Keyboard control | Port B | PB0 $\sim$ PB7 | Scanning-Data-In X0 $\sim$ X7 |
| :--- | :---: | :--- | :--- |
|  | Port C | PC0 $\sim$ PC3 | Scanning-Data-Out Y0 $\sim$ Y9 |
| CMT control (on/off) | Port C | PC4 | Recorder-Motor-On/Off-Signal-Out |
| CMT data output | Port C | PC5 | Recorder-Data-Out |
| CAP LED control | Port C | PC6 | CAP-Lock-LED-Control-Signal-Out |
| Alarm sound output | Port C | PC7 | Beep-Sound-Out |
| Slot select output | Port A | PA0 $\sim$ PA7 | Slot-Select-Signal-Out |

Pins to be controlled by SSG (YM-2149)

| Software controlled sound output | CHA |  | $\begin{array}{l}8 \text { octaves, tri-chord sound plus noise } \\ \text { Envelopes are controlled collectively } \\ \text { (not individually). }\end{array}$ |
| :--- | :--- | :--- | :--- |
|  | CHB |  |  |$)$

Fig. 5-3-2 (A)(B)

## 5-3-3 VDP (TMS9929A, TMS9918A) Video Display Processor

Performs all control operations for the CRT. For details, refer to the material provided separately.

## 5-3-4 Printer control

## Eight-bit parallel

Hand-shake based on Busy and Strobe signals.

## Level TTL output:

Conforms to JIS. Anphenol 14-pin connector.

## 5-4 Outline of PPI (mPD8255A) operation

## Basic I/O port control operation



Fig. 5-4-1 Internal Block Diagram in PPI Control Mode 0
Shown above is the internal mode diagram (mode 0) of the PPI. In the MSX system, the PPI (mPD8255A) is set with I/O port addresses A8H-ABH. The CPU selects and controls the RD, WR, CS, A1, and AO lines, and thereby controis the command registers and the R/W control block Fig. 5-4-1. In this way, it controls the operations of, respectively, ports PA, PB, and PC.

In the above figure, all the 24 pins (PA-PC) appears as I/O port terminals. Depending on the way you use port PC, there are three modes:

| MODE 0: | PA-PC mode of the I/O ports selected when MSX-BASIC is activated. |
| :--- | :--- |
| MODE 1: | 8-bit handshake mode (independent of each other ports PA and PB) |
| MODE 2: | Bidirectional handshake mode (") |

## 5-4-1 Mode setting from command registers

When MSX-BASIC is activated, the PPI is set in Mode 0 . All the discussions from now on will center on Mode 0.

As already shown, the MSX system uses the four I/O port addresses A8H-ABH (4 bytes) and selects the following internal addresses Fig. 5-4-2:

| A1 | A0 | 1/O Address | R - NAME | R/W |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | A 8 H | PA Register | R/W | Internal | data latch address |
| 0 | 1 | A 9 H | PB Register | R/W | " | " " |
| 1 | 0 | AAH | PC Register | R/W | " | " ${ }^{\text {" }}$ |
| 1 | 1 | ABH | Control register | W | $D 7=1$ | Command register |
|  |  |  |  |  | D7 $=0$ | For PC register bit control |

Fig. 5-4-2

These I/O ports are selected and controlled by the five signal lines fed to the command registers and the R/W control block.

| RD: | Reads the contents of the selected register when low, and sends them to the CPU |
| :--- | :--- |
| WR: | Writes from the CPU into the selected register when low. |
| CS: | Chip-selects the PPI (mPD8255A) when low. |
| A1~A0: | Directly connected to the address bus. This signal selects a register inside the |

The following conditions are required when the PA-PCI/O states must be controlled by the command registers.


$$
\begin{aligned}
& \text { RD }=\text { "H" } \\
& W R=" L " \\
& C S=" L "
\end{aligned}
$$

The MSB (D7) of the data to be written is always ' 1 '. *When $\mathrm{D} 7=0$, the PC port is controlled bitwise.

The above setting latches in the command registers shown in the figure the data determining whether input or output will be performed at the respective ports (PA-PC). Till the data is re-written, ports PA-PC will be used for input or output as decided above.

When MSX-BASIC is activated, the following I/O status will prevail:
PA port: For output to the MMC
PB port: For input of keyboard data, RETURN, and DATA
PC port: For output of keyboard data, cassette data, cassette control, and CAPS-LED

The format of the data written in the command registers is as shown below (Mode 0: write only):

| D7 | $\mathrm{D} 7=1 \ldots$. . for addressing a command register by $\mathrm{D} 6 \sim \mathrm{D} 0$. D7 $=0 \ldots$. . for the Bit control of the PC port. |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| D6 D5 | Sets the PA group mode <br> D6 and D5 are both set to zero in the MSX system. <br> $\mathrm{D} 6=0, \mathrm{D} 5=0:$ MODE $0 \quad \mathrm{D} 6=0, \mathrm{D} 5=1:$ MODE $1 \mathrm{D} 6=1, \mathrm{D} 5=1,0:$ MODE 1 |  |  |  |
| D4 | Sets the I/O direction of the PA port. | D4 = 1: INPUT | D4 $=0$ : | OUTPUT |
| D3 | Sets the I/O direction of the PC port bits $4 \sim 7$. | D3 = 1 : INPUT | D4 $=0$ : | OUTPUT |
| D2 | Sets the PB group mode. | D2 = 1 : MODE 1 | D2 $=0$ : | MODEO |
| D1 | Sets the I/O direction of the PB port. | D1 = 1: INPUT | D1 $=0$ : | OUTPUT |
| D0 | Sets the I/O direction of the PC port bits 0 $\sim 3$. | D0 = 1: INPUT | D1 $=0$ : | OUTPUT |

Fig. 5-4-3

D7-D0 will be set as shown below when MSX-BASIC is activated.

| D7 | 1 | Command Register selection |
| :--- | :--- | :--- |
| D6 | 0 |  |
| D5 | 0 | Sets the PA port to mode 0. |
| D4 | 0 | Sets the PA port to the output port. |
| D3 | 0 | Sets the 4 to 7 bits of the PC port to the output port. |
| D2 | 0 | Sets the PB port to mode 0. |$\quad$| This setting is performed by |
| :--- |
| writing " $10000010^{\prime \prime}(=" 82 "$ |
| in hexadecimal) in the ABH |
| address (I/O address). |

Set D7-D0 by writing $10000010=82 \mathrm{H}$ in I/O address ABH .

Fig. 5-4-4

Input/output for the ports PA-PC is set as above. In the MSX system, ports PA and PC are set to output and port $P B$ to input.

When the data in port PA is given out as output, the process that follows is as shown below:
$\mathrm{A} 1=\mathrm{A} 0=" 0 " . .$. PA register select $\ldots . . \mathrm{A} 8 \mathrm{H}$
The output data is written in the selected PA register.

The above will start data output at port PA.

Outline of the PPI MSX system operation
As indicated in Table 3 and Fig. 16, ports PA-PC of PPI are set in the MSX system. The circuits (interface circuits) connected to the respective ports are as shown below:


Fig. 5-4-5
If an 8 -bit CPU is used, only a 64 K byte area can be addressed normally. However, in case that the system can be upgraded in the future, the MSX system allows a memory area extension of 64K bytes (besides the main 64 K byte area) referred to as slots.

This 64 K -byte area is divided into four blocks, 16 K bytes each, so that memory selection can be made within slots.

| A15 A14 |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| FFFFH <br> COOOH | $\begin{gathered} \text { 16K BYTE } \\ \text { RAM } \end{gathered}$ | 1 | D | H | L |
|  | 16K BYTE <br> RAM | 1 | C | G | K |
|  | 32K BYTE | 0 | B | F | J |
|  | ROM | 0 | A | E | 1 |
| 0000H |  |  |  |  |  |
|  | SLOTO |  | LO | LO |  |

Slots $0 \sim 3$ are selected by means of the respective slot select signals in Fig. 5-4-5. The 16 K byte blocks are selected basically by A14 and A15.

Fig. 5-4-6

Output data from port PA of the PPI generates the following select signals.

| PA Port |  |  |
| :---: | :---: | :---: |
| CS3 | H | Signal specifying the slot that selects the area $\mathrm{COOOH} \sim$ FFFFFH. |
|  | L |  |
| CS2 | H | Signal specifying the slot that selects the area $8000 \mathrm{H} \sim$ BFFFH. |
|  | L |  |
| CS1 | H | Signal specifying the slot that selects the area 4000H $\sim$ 7FFFH. |
|  | L |  |
| CSO | H | Signal specifying the slot that selects the area $0000 \mathrm{H} \sim 3 \mathrm{FFFH}$. |
|  | L |  |


| $H / L$ | $0 / 0$ | $0 / 1$ | $1 / 0$ | $1 / 1$ |
| :---: | :---: | :---: | :---: | :---: |
| Slot | Slot 0 | Slot 1 | Slot 2 | Slot 3 |

Fig. 5-4-7

Example: When MSX-BASIC starts: All the 16 -byte blocks are assigned to slot 0
Data output from port PA of the PPI: $00000000: 00 \mathrm{H}$

Example: When the 32-K byte RAM is selected and the B and the I areas are used:
Data output from port PA of the PPI: 000001 11: 07H

When CSOL $=$ " H ", CSOH $=$ " H ", area $0000 \mathrm{H}-3 F F F H$ in slot 3 is selected.

When CS1L $=$ " $H$ ", CS1H $=$ " $L$ " area 4000 H -7FFFH in slot 1 is selected.

When CS3L, $H=$ " $L$ ", CS2L, $H=$ " $L$ ", area 8000 H -FFFFH in slot 0 is selected.

Thus, by sending data to the MMC, port PA selects the slots.
(2) Keyboard scan circuit with PBO $\sim$ PB7 and PCO $\sim$ PC3:

As Fig. 20 shows, the keyboard is built up of a switch matrix only. Basically, a $10 \times 8$ or 80 key matrix is possible only (Fig. 20) covering Y0 $\sim Y 9$ (10) and $X 0 \sim X 7$ (8). In actuality, however, only a $9 \times 8$ or 72 -key matrix is available covering $Y 0 \sim Y 8(9)$. In the circuit, the eight ports PBO $\sim$ PB7 serve as input terminals for the set key data, and the four ports PCO $\sim$ PC3 send sense signals to YO-Y8 in response to the sensing. The sense signals send a low signal to $Y 0 \sim Y 8$ at a timing. If any one of the keys is pressed, one of the pins $X 0 \sim X 7$ will go low. This will constitute the key data (keyed in data), and will be sent to the CPU. Once $X 0 \sim X 7$ are pulled up high, they remain in that state as long as none of the keys is pressed.


Fig. 5-4-8

LS145 in Fig. 20 is an O.C.BCD to decimal decoder, pulling down bits $0-9$ in the output to low in response to the 4 -bit input data. When $A=B=C=D=$ low, the terminal will go low, and if $A=$ high while $B=0=C=D=$ low, the terminal 1 will go low. Again, in the decimal mode, when $A=$ high, $B=C=$ low, but $D=$ high, and the number exceeds 9 , then all the output terminals will go high.


Fig. 5-4-9

The keyboard circuit is basically the one shown in Fig. 21. It issues the sense data to set Y0 $\sim$ Y9 low in Fig. 20. If any of the keys is pressed now, only those of the terminais in the range X0~X7 that correspond to the keys pressed will go low. At the same time, this will be set as 8 bits of parallel data (PB0 $\sim$ PB7) in the PPI.
(3) CMT control circuit using PC4 and PC5


Fig. 5-4-10

The PPI controls the CMT for its output. The PC terminal of port PC in the PPI is used for turning the CMT motor on/off. The PC5 terminal has the circuit shown in Fig. 22. It is used as the output port for data to CMT.

## CMT motor on/off control

When PC4 = "L", point $(A)$ in fig. $22=$ "H" $\mathrm{Tr}=\mathrm{ON}$; Relay OFF
When PC4 $=$ " H ", point $(\mathrm{A})$ in fig. $22=$ "L" $\mathrm{Tr}=$ OFF; Relay ON (motor on

## Circuit for data output to the CMT

The MSX system communicates with the CMT as specified below:

| Input | Connected to the earphone jack of the CMT (Cassette) unit. |
| :---: | :---: |
| Output | Connected to the microphone jack of the CMT (Cassette) unit. |
| Synchronization method | Start-stop (asynchronom) synchronization. |
| Transmission rate | 1200 Baud - Default value <br> DATA " 0 " . . . . . . 1200 Hz - 1 Cycle <br> DATA " 1 " . . . . . . 2400 Hz - 2 Cycles <br> 2400 Baud - Software selected <br> DATA " 0 " . . . . . . 2400 Hz - 1 Cycle <br> DATA" 1 " . . . . . . 4800 Hz - 2 Cycles |


| Synchronization method | FSK (Frequency Shift Keying) Kansas City Standard Waveform. |
| :--- | :--- |
| Connector | DIN $45326(8 \mathrm{pin})$ |



Fig. 5-4-12 PC Port PC5 Terminal Data Output Timing

The PC terminal gives a square wave output. The waveform is blunted by means of a CR filter consisting of C31 ~ C32 and R39 ~ R41. CR (Fig. 20) in consideration of the recording connection to CRM (cassette).
(4)

PC6 and PC7 functions


Fig. 5-4-13

As Fig. 5-4-13 shows, PC6 controls the CAPS-LED on the keyboard.

LED on: Uppercase alphabetic characters
LED off: Lowercase alphabetic characters PC7 sends a 1-bit signal for beeping

## 5-5 Outline of SSG (YM2149) operations

Software compatible with PSG (AY-3-8910)


Fig. 5-5-1 SSG Internal Block Diagram

$$
\text { - } 77 \text { - }
$$

## Software sound generator

Sound generators A, B, and C in SSG (Fig. 5-5-1) can independently determine the sound intervals. The respective intervals (frequencies) are set by using R0 $\sim$ R5 each of which is a 2-byte register.

The sound generated by the tone generator is emitted as output corresponding to $A, B$, or $C$. The 3-channel mixer (internal SW) sets in mixer register R7 which of the sounds, $A, B$, or $C$ is to be emitted.

Finally, VCA A, B, and C determine the respective sound volumes which are sent as output outside the IC.

Here, a noise generator is provided. Here also, a selection is made by a 3-channel mixer to determine the output channel for the sound (CHA $\sim \mathrm{CHB}$ ). The sound, mixed with the interval signals, is given as output.
(1) Tone generator ( $A, B$, and $C$ ) interval (frequency) determination

Tone generator $A, B$, and $C$ have a 2-byte register each. Here the data obtained by using a specific mathetical expression is written to set for $A, B$, and $C$ independent intervals (frequencies). R0, R2, R4, and R5 contain fine adjustment data, and R1 and R3 contain fine adjustment data.
(2) Sound source selection using 3-channel mixer

Outputs from CHA, CHB, and CHC can be made independent of each other. However, the 3-channel mixer (SW) determines whether the tone generator ( $\mathrm{A}, \mathrm{B}$, or C ) or noise generator output can be sent to CHA-CHB. Sound sources are controlled according to the bits in mixer register R7 as follows:

Register R7

| Bit-No. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Function | $10 B$ | 10 A | CCH | BCH | ACH | CCH | BCH | GCH |
|  | $\mathrm{L}:$ Input <br> $\mathrm{H}:$ Output | Noise | Noise | Noise | Tone | Tone | Tone |  |

Fig. 5-5-2 Register R7

## Bits 0-5 are the mixer SW ON/OFF bits.

## $H$ : No sound emission

L: Sound emission

If the bits are as follows

(3) Sound volume control with VCA's Finally, the sound volumes are determined tor the three independent outputs from VCA A, B, and C. Data is written in bits $0-3$ of the registers R10 $\sim$ R12 to control sound volume in 16 stages.
Sound volumes in the respective VCA's can be set independently also.
To each VCA, control signals are fed from the sound volume envelope generator. The operation is as follows:

Register 15: Sets a sound volume envelope waveform Registers R13 and R14: Set sound volume waveform period EP.

If the sound volume envelope controls VCA A-C, bit 4 of registers R10~R12 goes high. Sound volume control by the sound volume envelope does not apply independently to $\mathrm{ACH} \sim \mathrm{CCH}$, but is common to all channels.


Fig. 5-5-4 R15 Waveform Setting

## SSG control

| Octal number | REG | BIT | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | R0 | Channel A Tone Period | 8-BIT Fine Tune A |  |  |  |  |  |  |  |
|  | R1 |  |  |  |  |  |  |  |  |  |
|  | R2 | Channel B Tone Period | 8-BIT Fine Tune B |  |  |  |  |  |  |  |
|  | R3 |  |  |  |  |  |  |  |  |  |
|  | R4 | Channel C Tone Period | 8-BIT Fine Tune C |  |  |  |  |  |  |  |
|  | R5 |  | 4-BIT Coarse TuneC |  |  |  |  |  |  |  |
|  | R6 | Noise Period | 5-BIT Period Control |  |  |  |  |  |  |  |
|  | R7 | $\overline{\text { Enable }}$ | $\overline{\text { IN/OUT }}$ |  | Noise |  |  | Tone |  |  |
|  | R7 |  |  |  |  | B | A | C | B | A |
|  | R10 | Channel A Amplitude |  |  |  | M | L3 | L2 | L1 | LO |
|  | RK1 | Channel B Amplitude |  |  |  | M | L3 | L2 | L1 | L0 |
|  | R12 | Channel C Amplitude |  |  |  | M | L3 | L2 | L1 | LO |
|  | R13 | Envelope Period | 8-BIT Fine Tune E |  |  |  |  |  |  |  |
|  | R14 |  |  |  |  |  |  |  |  |  |
|  | RT5 | Envelope Shape/Cycle |  |  |  |  |  |  |  |  |
|  | R16 | I/O Port A Data Store | 8-BIT PARALLEL I/O on Port A |  |  |  |  |  |  |  |
|  | R17 | I/O Port B Data Store | 8-BIT PARALLEL I/O Port B |  |  |  |  |  |  |  |

Fig. 5-5-5 Internal Registers

## SSG control lines



Fig. 5-5-6

| A1 | A0 | BDIR | BC1 | BC2 | XXX | I/O addres |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 1 | 1 | 1 | $X X X$ | A0H |
| 0 | 1 | 1 | 0 | 1 | $X X X$ | $A 1 H$ |
| 1 | 0 | 0 | 1 | 1 | $X X X$ | $A 2 H$ |
| 1 | 1 | 0 | 0 | 1 | $X X X$ |  |

Fig. 5-5-7 SSG Control

## SSG control by the CPU

(1) Figure 5-5-1 shown the internal connections of SSG. As the figure shows, it has 18 registers. The functions of the respective registers are specified by initially setting in them the corresponding data.

Thus, the register No. to be selected is set at address latch ( $\mathrm{A} O H$ ). Next, the respective control data is set in data write (A1H).

## SSG PORT (IOA and IOB) operations

(1) General port control circuit based on IOAO $\sim 5$ and IOBO $\sim 6$ selects when 1 A is low and 1 B is high.


Fig. 5-5-8 General Port (JOY1) Circuit

As the above diagram shows, IOA operates as the input and IOB as the output port. There are two general ports, these being selected as follows by the IOB6 bit:

IOB6: L: JOY1
IOB6: H: JOY2

IOB4 issues the slot signal (sense signal) when the bit hit externally by, say, the joystick is fed to 1A $\sim 4 A(1 B \sim 4 B)$ of LS157. IOB0 and IOB1 can set TRGAI and TRGBI to high or low initially.
(2) IOA7 CMT data input

Data from the CMT (data recorder) enters the IOA7 terminal. In actuality, however, the input is shaped into a square wave in the waveform shaping circuit as shown in the figure.


Fig. 5-5-9

## 6. VIDEO BLOCK

Data is transferred through an 8 bit data bus between CPU and VDP. Timing is taken by the VDP signal from NMI and the RD and WR signals from CPU.
As shown in Figure 6-1-1, CPU writes or reads (status only) to or from 98 H (data read/write) and 99 H (instruction to VDP/VDP status) which are I/O addresses. The address decoder for 98 H and 99 H is enclosed in the MMC (MSX Memory Controller). When 98 H and 99 H of I/O addresses are accessed by CPU, VDP of the 12th pin of MMC becomes LOW, and AD of the address bus selects the address through LOW (98H) or HIGHT (99H). Data read and write to the video RAM is performed through VDP (above I/O addresses).
VRAM uses an MB8116 ( $16 \mathrm{~K} \times 1 \mathrm{bit}$ ) dynamic RAM. Through timing of $\overline{\text { RAS }}$ and CAS from VDP, VRAM receives 14 bit addresses by dividing them into low order 7 bit and high order 7 bit.
AD0 through AD7 of the VDP are also connected to the DI (data input) pins of VRAM. After the address from VDP is received by dividing into the high and low orders, the write data is sent. (Figure 6-1-1). The read data from VRAM is output to RD0 through RD7 of VDP through the DO (data output) pins, then it is output from VDP to the data bus.
The INT pin of VDP generates an interrupt signal once for one frame ( 1 screen) (approximately every $1 / 60$ sec. in the NSTC method and every $1 / 50 \mathrm{sec}$. in the PAL, SECAM method), and the signal is transferred to the CPU INT pin. Through such an interrupt, CPU performs various interrupt processes.
(1) The composite video signal from the 36 th pin of VDP (TMS0918A) is impedance converted into 75 ohm by the buffer in the clock card, and the signal is output from the video connector to the display (in the NTSC mode).
(2) In PAL, the color differential signal from the 35th, 36th and 38th pins of VDP (TMS9929A) is converted into the PAL composite video signal through the color encoder, and the signal is output from the video output device. At the same time, the signal is converted into the channel frequency and output from the RF output device.
(3) In the case of SECAM, the color differential signal from the 35th, the 36th and 38th pins of VDP (TMS9929A) are put into the RGB matrix sheet. The $Y$ signal is mixed with the $B-Y$ signal and $R-Y$ signal, then the $R$ signal, $G$ signal, $B$ signal and $Y$ signal are output.

## 6-1 VDP

VDP (Video Display Processor) is a special-purpose LSI used to control the display data from CPU and to display images on family-use color TVs and color monitor TVs.

The display memory in general microcomputers has a dual port memory mode which enables access from both the microprocessor and CRT controller. In some cases, the display memory and microprocessor memory co-exist so that a memory element will be used for both purposes, that is, the memory cycles of the CRT controller and microprocessor are time-shared and the memory can be equally accessed by each (interleave mode).
In the dual port memory mode, the memory has not always been efficiently used because the memory cycle is divided for the microprocessor and CRT controller. VDP has a special-purpose bus, separated from the microprocessor bus, for the display DRAM, because VDP frequently refers to the display memory for the purpose of handling animation called sprites and for taking display dot pattern images into LSI through the memory indirect designation. Through this bus, DRAM reading from VDP can be efficiently performed, and a moving picture can be displayed.


VDP
Fig. 6-1-1 VDP and Video-RAM

## Features of VDP

(1) It performs read/write and refresh to VRAM (dynamic RAM).
(2) There are 4 types of still picture modes, so graphics, characters and multi-color patterns can be displayed.
(3) It has 32-sheet sprite (graphic pattern for animations) display functions, so moving an image is easy and the 3-dimensional effect can be attained for games and animations.
(4) Sixteen colors including white, black and transparent can be displayed.
(5) The color differential signal in the PAL mode is output (TMS9929A only)

## 6-2 VDP Pins and Functions

Figure 6-2-1 shows functions of VDP pins.


Fig. 6-2-1 VDP Pins and Functions

## 6-3 VDP Block

The internal block diagram of VDP is as shown in Figure 6-3-1. The lower half area of the block diagram is similar to the general CRT controller configuration, but the lower half area has been added with the sprite control hardware. There are the shift register that stores 4 sprite dots, the sprite color register, and the counter which designates the sprite display start position.
For each horizontal scan, information for 4 sets is loaded from the sprite with priority (small sprite No.) and which coincides with the raster position of the vertical scanning line.


## 6-4 VDP Display Mode

There are four modes for VDP display as shown in Table 6-4-1.
Each mode is selected through the VDP register.

Table 6-4-1 VDP Display Modes

| VDP name | MSX BASIC name | BASIC command |
| :---: | :---: | :---: |
| Graphic I | Mode 1 $32 \times 24$ text | Screen 1 |
| Graphic II | Mode 2 $\quad$ High-resolution graphic | Screen 2 |
| Multi-color | Mode 3 $\quad$ Multi-color | Screen 3 |
| Text | Mode 0 $40 \times 24$ text | Screen 0 |

As one of the features of VDP display, a table is provided on the display RAM, and data in the table indirectly designates the memory to be used as the display pattern. This method is similar to the character display method using the character generator.
An example of the graphic I memory designation is shown in Figure 6-4-2. In this mode, pattern positions on the screen are assigned from 0 through 767.
The pattern name table designated by the base address corresponds to the area assigned on the screen. Data written on the N -th table means the screen pattern position N .
Through 256 types of data written on the N-th table, the dot pattern in the pattern generator area and the color information in this pattern color table area are input to VDP and a still graphic image is made.


Some information will be needed, in addition, for sprite display. Therefore, VDP reads the following data into the display area:
(1) Pattern name
(2) Pattern generator
(3) Pattern color code
(4) Any other sprite information

Furthermore, DRAM read/write from CPU enters the above cycle, so the memory bus is very busy.
The total number of horizontal dots in graphic I (mode 1), II (mode 2) and multi-color mode (mode 3) is 342, and 256 dots of these make a valid display area. The memory cycle of DRAM is 372 ns . Therefore, there are 171 memory cycles in which VDP can access DRAM for one horizontal scan. (Fig. 6-4-3)


Fig. 6-4-3 VDP Memory Cycle (in NTSC)

## 6-5 VDP Register

VDP contains 9 registers; eight are write-only resisters and one is a read-only register, they perform VDP operation mode setting and VRAM address setting.

| $\begin{aligned} & \text { Register } 0- \\ & \text { Register } 1 \end{aligned}$ | Designates the function and display mode. |
| :---: | :---: |
| Register 2 | Inserts the value obtained by dividing the pattern name table address by 400 H . |
| Register 3 | Inserts the value obtained by dividing the color table address by 40 H . |
| Register 4 | Inserts the value obtained by dividing the pattern generator table address by 800 H . |
| Register 5 | Inserts the value obtained by dividing the sprite attribute table address by 80 H . |
| Register 6 | Inserts the value obtained by dividing the sprite generator table address by 800H. |
| Register 7 | Designates the text color/back drop color. |

* Registers 2, 3 and 4 are used for display of the character pattern plane, while registers 5 and 6 are used for display of the sprite.
- VDP register operations
(1) Register 0
(2) Register 1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 | 0 | 0 | M3 | EV |

M3: Refer to Table 6-5-1.
EV: $\quad 0 \rightarrow$ External video signal input not permissible
$1 \rightarrow$ External video signal input permissible

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $4 / 16 K$ | BLANK | IE | M1 | M2 | 0 | SIZE | MAG |

4/16K: $\quad 0 \rightarrow 4 \mathrm{~K}$ bit DRAM used for VRAM
$1 \rightarrow 4$ or 16 K DRAM used for VRAM

BLANK: $\quad 0 \rightarrow$ Screen display stop
$1 \rightarrow$ Ordinary screen display

IE: $\quad 0 \rightarrow$ Interrupt to CPU inhibited
$1 \rightarrow$ Interrupt to CPU permitted

M1, M2: Refer to Table 6-5-1.

SIZE: $\quad 0 \rightarrow$ Sets the sprite size to $8 \times 8$.
$1 \rightarrow$ Sets the sprite size to $16 \times 16$.
MAG: $\quad 0 \rightarrow$ Displays the sprite in its normal size.
$1 \rightarrow$ Displays an enlarged sprite.

Table 6-5-1 M1 through M3 Bit and Screen Modes

| Screen mode | M1 | M2 | M3 |  |
| :---: | :---: | :---: | :---: | :---: |
| $40 \times 24$ characters text mode <br> (mode 0) | 14 |  |  |  |
| $32 \times 24$ characters text mode <br> (mode 1) | 0 | 0 |  |  |
| High-resolution mode <br> (mode 2) | 0 | 0 | 1 |  |
| Multi-color mode <br> (mode 3) | 0 | 0 | 0 | 0 |

(Register 2 value) $\times \& \mathrm{H} 400$ is the actual base address.
(4) Register 3

(Register 3 value) $\times \& H 40$ is the actual base address.
(5) Register 4

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 0 | 0 | 0 |  | PGTBA |  |

(PGTBA) $\times \& H 800$ is the pattern generator table base address.
(6) Register 5

(7) Register 6

(SPTBA value) $\times \& H 800$ is the sprite pattern table base address.

Note: Table 6-5-2 applies in the case of a high-resolution mode.

$$
-90-
$$

Table 6-5-2 VDP Registers 3 and 4 in Mode 2

| Register name | Relating table name | Register value | Base address |
| :---: | :---: | :---: | :---: |
| Register 3 | Color table | $\& \mathrm{H} 7 \mathrm{~F}$ | 0 |
|  |  | $\& \mathrm{HFF}$ | $\& \mathrm{H} 2000$ |
| Register 4 | Pattern generator table | 3 | 0 |
|  |  | 7 | $\& H 2000$ |

(8)
Register 7

| 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Character color |  | Background color |  |  |  |

For color codes, see below.

| Decimal code | Hexadecimal code | Color | Decimal code | Hexadecimal code | Color |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | Transparent | 8 | 8 | Red |
| 1 | 1 | Black | 9 | 9 | Bright red |
| 2 | 2 | Green | 10 | A | Dark yellow |
| 3 | 3 | Bright green | 11 | B | Bright yellow |
| 4 | 4 | Dark blue | 12 | C | Dark green |
| 5 | 5 | Bright blue | 13 | D | Magenta |
| 6 | 6 | Dark red | 14 | E | Gray |
| 7 | 7 | Cyan | 15 | F | White |

(9) Register 8

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $F$ | $5 S$ | C |  | $5 S N$ |  |  |  |

F: Becomes zero when picture writing for a screen is over. It does not function unless the register 1 IE bit is 1 .
$5 \mathrm{~S}, 5 \mathrm{SN}: \quad 5 \mathrm{~S}$ bit becomes 1 when 5 or more sprites are on a horizontal scanning line and $F$ bit is 0 . Concurrently, the 5 th sprite number is set as the 5 SN value.

C: $\quad$ Becomes 1 when two or more sprites are overlapping.


## 6-6 Sprite

Sprite is a function which overlaps and displays characters consisting of $8 \times 8$ dots or $16 \times 16$ dots with the under-screen. (Figure 6-6-2). There are 5 types of sprites attributes provided; shape, color, display position, size ( $8 \times 8$ or $16 \times 16$ ) and enlarged display (two-folds each in vertical and horizontal direction). The shape is determined by the sprite pattern table written on the VRAM sprite pattern table. The size and enlarged display are selected by the VDP register 1 . The number of sprites that can be displayed on the screen is 32 . There are 256 sprite patterns that can be defined. The sprite attribute table consists of 4 byte data as shown in Figure 6-6-1.


Fig. 6-6-1 Sprite Attribute Table Configuration

The first 2 bytes express the coordinates of the left-upper corner point of the sprite. When the vertical coordinate value of the first byte is 208 or more, all sprite planes having larger numbers are not displayed.

(Here the sprite pattern becomes smaller because it is a maximum $32 \times 32$ dots.)


Fig. 6-6-2 Display Screen Configuration

The third byte is the displayed pattern number.
Bit 7 in the 4th byte is called the EC (Early Clock) bit. When the EC bit is 1 , the sprite is shifted to the left on the screen by 32 picture elements. This enables the sprite to smoothly pass the screens left border. Four bits (bit 0 through 3 ) express sprite colors.

## 6-7 CPU, VDP and VRAM Timing

Data transfer timing between CPU, VDP and VRAM is as described below.
(1) Two byte data is transferred from CPU to VDP; the first byte is data to be written in the register, the second byte is the register to be written, both are specified with low-order 3 bits.

|  | MSB |  | Bit |  |  |  | LSB |  | Control signal |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\overline{\text { CSW }}$ | $\overline{\text { CSR }}$ | MODE |
| First byte: Data | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | Low | High | High |
| Second byte: <br> Register selection | 1 | 0 | 0 | 0 | 0 | RS1 | RS2 | RS3 | Low | High | High |

Timing is as shown below.

mode $X \times X$


DATA

(2) Read of the status register is performed by one byte.

|  | MSB |  | Bit |  |  |  | LSB |  | Control signal |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\overline{\text { CSW }}$ | $\overline{\text { CSR }}$ | MODE |
| Status register data | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | High | Low | High |

Timing is as shown below.

(3) CPU executes write to VRAM through VDP. The first byte sets the low-order 8 bits of the VRAM address, the second byte sets the high-order 6 bits, and the third byte sets the data.

|  | MSB |  | Bit |  |  |  | LSB |  | Control signal |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\overline{\text { CSW }}$ | $\overline{\text { CSR }}$ | MODE |
| First byte: Address set-up | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | Low | High | High |
| Second byte: <br> Address set-up | 0 | 1 | AO | A1 | A2 | A3 | A4 | A5 | Low | High | High |
| Third byte: Data | DO | D1 | D2 | D3 | D4 | D5 | D6 | D7 | Low | High | Low |


$\overline{\mathrm{CAS}}$

(4) Three bytes are needed when CPU reads data from VRAM, similar to write.

|  | MSB |  | Bit |  |  |  | LSB |  | Control signal |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $\overline{\text { CSW }}$ | $\overline{\text { CSR }}$ | MODE |
| First byte: Address set-up | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | Low | High | High |
| Second byte: <br> Address set-up | 0 | 1 | A0 | A1 | A2 | A3 | A4 | A5 | Low | High | High |
| Third byte: Data | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | High | Low | Low |



## 6-8 Encoder and Clock Circuit

Models using PAL and RGB matrix (except NTSC mode) convert the color differential signal and luminance signal into the video signal through the encoder board or the RF signal through the RF converter. (Fig. 6-8-1) In models of NTSC mode, the video signal is output from VDP (TMS9918), so the video signal is output through the buffer amplifier in the clock card. (Figure 6-8-2)


Fig. 6-8-1 PAL/RGB Matrix Encoder


Fig. 6-8-2 NTSC Video Signal

## (1) PAL encoder board

This board converts $\mathrm{Y}, \mathrm{R}-\mathrm{Y}$, and $\mathrm{B}-\mathrm{Y}$ signals into a video signal by the encoder IC (LVA510). It further converts this video signal and the audio signal into RF signal by the RF modulator to output it (see Fig. 6-8-3)
The synchronous components included in the $Y$ signal are taken out ( H sync and V sync) by the synchronous separation circuit $\left(Q_{2}\right)$ and sent to the No. 10 and 11 pins of the encoder IC. The PAL color subcarrier frequency $(4,433,619 \mathrm{MHz})$ oscillated in the IC1 is added to the B-Y modulator through the No. 2 pin to modulate the $B-Y$ signal. It is also modulated by the $R-Y$ modulator through the $90^{\circ}$ phase circuit, together with the synchronous signal.
The chroma signal from the B-Y and R-Y modulators passes the mixing circuit and is output as a video signal through the buffer amplifier, together with the luminance signal which is inversed 180 degrees in the phase inversion circuit ( $\mathrm{Q}_{1}$ ) (No. 7 pin).
The video signal for the RF modulator is output from the No. 6 pin and converted into the predetermined channel frequency in the RF modulator, together with the audio signal, to output from the RF OUT. (The RF modulator differs depending on the PAL-I, PAL-B, and PAL-G.)

$$
-95-
$$

Fig. 6-8-4 shows a block diagram of the RF modulator. The carrier oscillating circuit oscillates the channel frequency (see Fig. 6-8-5) specified by each RF modulator to send it to the AM modulation circuit. (The RF modulator provided with the channel selector switch changes the oscillating frequency by turning the switch.) The audio signal is converted into the FM signal in the FM modulation circuit and sent to the AM modulation circuit to amplitude and modulate the carrier frequency, together with the video signal from the video $\operatorname{IN}$.
This AM modulation circuit is a kind of balanced modulator which uses two diodes. It suppresses the carrier frequency and outputs the side band component. This output signal adjusts the output impedance to $75 \Omega$ in the filter and the impedance conversion circuit. It is output as the RF signal.


Fig. 6-8-3 PAL Encoder Block


Fig. 6-8-4 RF Modulator Block Diagram


Fig. 6-8-5 Specifications for PAL/NTSC Systems

## (2) RGB matrix board

The RGB matrix board generates the R.G.B. signal and synchronous signal based on the R-Y, B-Y, and Y signals from the VDP (9929A).
The synchronous component of the $Y$ signal is removed in the sync clip circuit. This $Y$ signal is added to the $R-Y$ and $B-Y$ signals through mixing resistance to generate $R$ and $B$ signals, respectively. They are output to $R$ out and $B$ out through the amplifiers.
The burst clip circuit removes the burst signal in the $B-Y$ signal.
The $Y$ signal passes the level shift circuit after its synchronous component is removed in the sync clip circuit. It is added to the -R and -B signals obtained, by converting the R and B signals in the inversion circuit to generate the G signal. The audio signal is directly output to Audio out.
The AVC terminal and Ys terminal are set to +5 V and +2 V , respectively.
Each output signal is output to the monitor television of the RGB linear input type.


Fig. 6-8-6 RGB Matrix Board Block Diagram

## 7. POWER CIRCUIT

The power circuit system used in the CX-5M computer system is called the RCC (ringing chalk converter) system. Fig. 7-1-1 shows the basic circuit of the RCC system. $\operatorname{Tr} 1$ is a switching transistor (main transistor). When this transistor is ON, energy is stored in the inductance L1 of the transistor $T$, and when OFF, the energy is released on the L3 side. Thus, the repetition of the transistor's switching operation generates output. RB is the base current limiting resistance. R0 is the activation resistance. When the resistance is low, Tr1 is readily activated. Transformer T is the oscillating transformer to insulate the primary side and secondary side. The energy stored in the transformer T (inductance) is obtained by the following equation.

$$
\begin{aligned}
& \mathrm{U}=\left(\mathrm{V}_{\text {in }}{ }^{2} \times \mathrm{T}_{\mathrm{on}}{ }^{2}\right) / 2 \mathrm{~L} 1 \\
& \mathrm{U}: \text { Stored energy } \\
& \mathrm{L} 1: \text { Self-inductance } \\
& \mathrm{V}_{\text {in }}: \text { Input voltage } \\
& \mathrm{T}_{\mathrm{on}}: \text { Transistor ON time }
\end{aligned}
$$



Fig. 7-1-1 RCC Basic Circuit Board

The circuit operation of the RCC system is as follows.

1. In Fig. 7-1-1, when the switch is ON, current starts to flow at the base of $\operatorname{Tr} 1$ through Ro. Current flows in the coil L1, and induced current flows in L2.
2. On L2 side, voltage at A point rises to promote the flow of current of Tr1. For this reason, Tr1 is saturated instantaneously, the current flowing in L1 does not change even if time passes, and the induced current does not flow in L2. At this time, counterelectromotive force (stored energy) is generated, and this energy makes the induced current flow on L3 side. The base of Tr 1 is reverse biased by the counterelectromotive force of L2, and Tr1 is turned OFF.
3. Next, when all the stored energy in L 1 is released, the base current flows in $\operatorname{Tr} 1$ through Ro, and Tr1 starts its operation again.

In this way, the stored energy in L1 is output by the switching operation of $\operatorname{Tr} 1$ on L3 side to provide the required voltage.

## 7. POWER CIRCUIT

The power circuit system used in the CX-5M computer system is called the RCC (ringing chalk converter) system. Fig. 7-1-1 shows the basic circuit of the RCC system. $\operatorname{Tr} 1$ is a switching transistor (main transistor). When this transistor is ON, energy is stored in the inductance L1 of the transistor $T$, and when OFF, the energy is released on the L3 side. Thus, the repetition of the transistor's switching operation generates output. RB is the base current limiting resistance. R0 is the activation resistance. When the resistance is low, Tr1 is readily activated. Transformer T is the oscillating transformer to insulate the primary side and secondary side. The energy stored in the transformer T (inductance) is obtained by the following equation.

$$
\begin{aligned}
& \mathrm{U}=\left(\mathrm{V}_{\text {in }}{ }^{2} \times \mathrm{T}_{\mathrm{on}}{ }^{2}\right) / 2 \mathrm{~L} 1 \\
& \mathrm{U}: \text { Stored energy } \\
& \mathrm{L} 1: \text { Self-inductance } \\
& \mathrm{V}_{\text {in }}: \text { Input voltage } \\
& \mathrm{T}_{\mathrm{on}}: \text { Transistor ON time }
\end{aligned}
$$



Fig. 7-1-1 RCC Basic Circuit Board

The circuit operation of the RCC system is as follows.

1. In Fig. 7-1-1, when the switch is ON, current starts to flow at the base of $\operatorname{Tr} 1$ through Ro. Current flows in the coil L1, and induced current flows in L2.
2. On L2 side, voltage at A point rises to promote the flow of current of Tr1. For this reason, Tr1 is saturated instantaneously, the current flowing in L1 does not change even if time passes, and the induced current does not flow in L2. At this time, counterelectromotive force (stored energy) is generated, and this energy makes the induced current flow on L3 side. The base of Tr 1 is reverse biased by the counterelectromotive force of L2, and Tr1 is turned OFF.
3. Next, when all the stored energy in L 1 is released, the base current flows in $\operatorname{Tr} 1$ through Ro, and Tr1 starts its operation again.

In this way, the stored energy in L1 is output by the switching operation of $\operatorname{Tr} 1$ on L3 side to provide the required voltage.

The following drawing shows the block diagram of the actual circuit.


The following is the description and operation of the circuit.

1. In Fig. 7-1-2, D1, C1, and RB are the base drive circuits, and the base current of $\operatorname{Tr} 1$ is determined by RB.
2. Tr 2 operates as a voltage controller and cartridge protector.
3. The photocoupler (PC1) feeds the voltage fluctuation of +5 V back to the control circuit through the error detection circuit at all times. The control circuit controls output, based on the information fed back, by increasing and decreasing the base current of Tr2 and changing the oscillating frequency of Tr1.
4. VR1 connected to the base of Tr3 in the error detection circuit adjusts the output voltage $(+5 \mathrm{~V})$.
5. SW1 connected to Tr4 in the cartridge protector circuit is ON at all times. Therefore, $\operatorname{Tr} 4$ remains OFF normally.
6. When power is ON and the ROM cartridge is set in the upper-slot erroneously, SW1 is turned OFF. Next, Tr4 is turned ON, and the current flowing in the photocoupler (PC1) increases. Then, the photocoupler (PC1) on the control circuit is turned ON to turn ON Tr2. The oscillating frequency of Tr1 increases, the energy stored in L1 decreases, and output voltage lowers. When the ROM cartridge is properly set afterwards, A and B points in Fig. 2 are short-circuited, and voltage increases again.
7. Overcurrent protection resistance RP When excess current flows on the load side, current in proportion to it flows in Tr1. At this time, RE voltage also increases, which increases the base current of Tr2 through RP. Consequently, the oscillating frequency of $\operatorname{Tr} 1$ increases and output voltage is decreased.

The oscillating frequency of Tr 1 is about 45 KHz .

