SHARP

RAM FILE BOARD

MODEL **MZ-1R18**

INSTRUCTION MANUAL

INTRODUCTION

Thank you for buying the MZ-1R18 SHARP RAM File Board. Before using the RAM File Board, please read this manual carefully to assure correct operation.

The contents of this manual are subject to change without notice.

If the product is defective, please contact the store where you bought it.

SHARP is not responsible for damage incurred during, or as a result of, operation.

CAUTION

1. This unit consists of precision parts, such as LSI circuits, that can be affected by the operating environment. Do not use the unit in places subject to direct sunlight, extreme change of temperature or high humidity or dust.
2. Do not bump or drop this unit.
3. Do not touch the bottom of this board or the ICs, because static electricity might damage the ICs.
   Note that the aluminum foil lapping this board is to protect the ICs from static electricity. Be sure, therefore, to remove the foil immediately before installation.

CONTENTS

1. General 3
2. Installing the RAM File 3
3. Using the RAM File 3
4. Block Diagram 6

1. General

The RAM File is an external storage device which has a capacity of 64K bytes. It can be accessed through the BASIC command such as RUN, SAVE, or LOAD, and also controlled by the machine language programs.

Memory capacity : 64K bytes

2. Installing the RAM File

The procedures for RAM File installation are described in the MZ-800 OWNER'S MANUAL.

3. Using the RAM File

Note 1) When the power switch of the computer unit is turned off, all the contents stored in the RAM File are cleared. If you have any programs or data that you want to preserve, be sure to store them in the cassette tape or floppy disk and then turn off the switch.

2) As for the sequential data file, it is possible to be read from or written to the RAM file, while as for the random data file, it is not.

- When using the RAM File by BASIC

Using the command such as SAVE or LOAD, programs or data can be read or written by file.
For example, you can save a BASIC program into the RAM File by entering the following command.

SAVE "**RAM**: filename"
↑
<device name>

By specifying "**RAM:**"to the <device name>, the command such as SAVE or LOAD is executed to the RAM File.

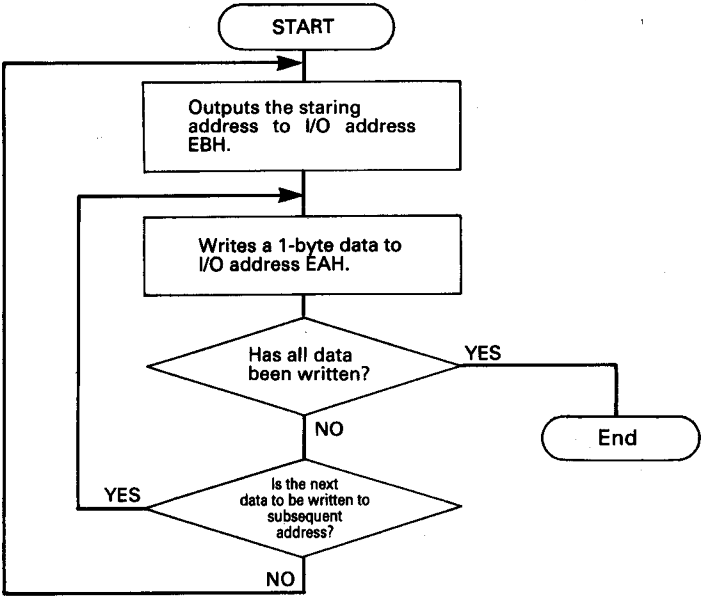
IF the INIT command is used, the RAM File can be used as a buffer (in which data can be stored temporarily) for printer, and the parallel processing of the printer and computer will be possible.

Entering more programs or data than the memory capacity of the RAM File (which changes by use of the INIT command) causes an error.
In this case, delete the unnecessary file by the DELETE command, and then write the programs or data.

- When using the RAM File by machine language

  When using the RAM File by the machine language program, you must first output the starting address of the RAM File area for writing or reading data to I/O address EBH, then you can write or read a 1-byte data through I/O address EAH.

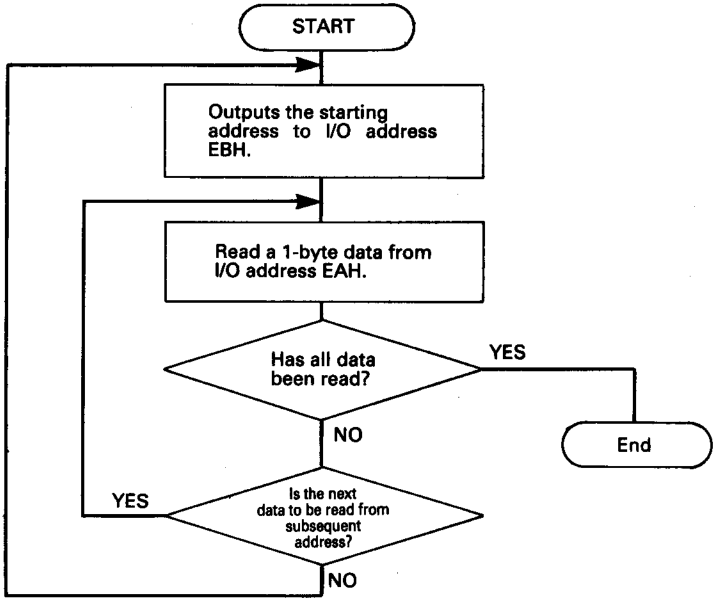
  (An algorism for writing data)


START
Outputs the staring address to I/O address EBH.
Writes a 1-byte data to I/O address EAH.
Has all data been written?
YES
End
NO
Is the next data to be written to subsequent address?
YES
NO

The following example writes data 41H to address D3C5H in the RAM File.

```
LD   A, C5H·············· Lower byte of the starting address
LD   B, D3H·············· Higher byte of the starting address
LD   C, EBH}              Outputs the starting address with an
OUT (C) , A} ···········  indirect OUT instruction.
LD   A, 41H     }
OUT (EAH) , A   } ······ Outputs 1 byte (41H).
```

(An algorism for reading data)


START
Outputs the starting address to I/O address EBH.
Read a 1-byte data from I/O address EAH.
Has all data been read?
YES
End
NO
YES
Is the next data to be read from subsequent address?
NO

The following example reads a 1-byte data from address CD01H in the RAM File.

```
LD   A, 01H··········Lower byte of the starting address
LD   B, CDH··········Higher byte of the starting address
LD   C, EBH   }  ......Sets the starting address with an indirect
OUT   (C) , A }       OUT instruction.
IN   A, (EAH) ······Inputs 1 byte.
```

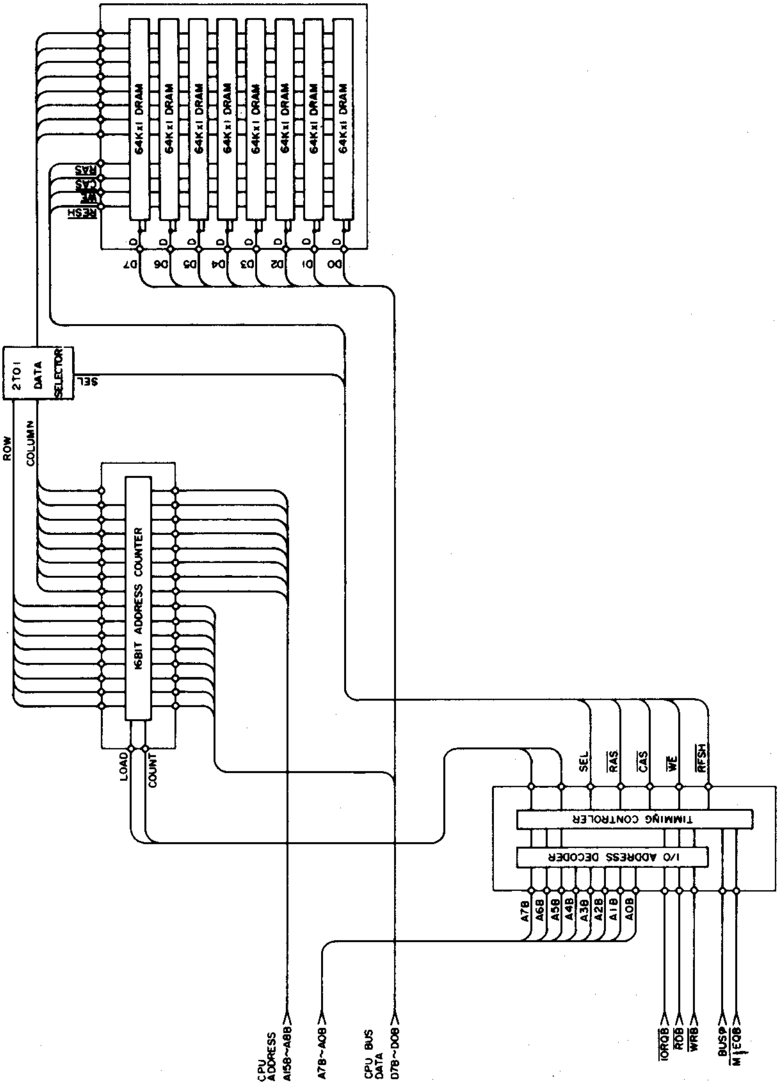
Address Auto-increment Function

Once a starting address is set, the address is automatically incremented each time a 1-byte data is written or read.

4.Block Diagram

I/O address decorder

This section determines the I/O addresses. For the RAM File they are set to EAH and EBH ($EA and $EB).



Direct/indirect I/O instructions

There are two types of I/O instructions in the Z-80 instruction set.

① Direct I/O instructions	IN A, (n)	OUT (n),A
② Indirect I/O instructions	IN r, (C)	OUT (C), r

n: Hexadecimal number

r: Register

In direct I/O instructions, an I/O address is directly indicated by a 1-byte hexadecimal number; in indirect I/O instructions, an I/O address is indirectly indicated by the number held in register C.

When a direct I/O instruction is executed, address value n is output to address bus A0 to A7 and the contents of register A is output to address bus A8 to A15. Then, an Input/Output operation is carried out between the specified I/O port and register A.

When an indirect I/O instruction is executed, the contents of register C is output to A0 to A7 and the contents of register B is output to A8 to A15. Then, an Input/Output operation is carried out between the specified I/O port and register r.

SHARP CORPORATION
OSAKA, JAPAN

Printed in Japan
Gedruckt in Japan
Imprimé au Japon
Stampato in Giappone

4H 1.65-I(TINSE1223ACZZ)