HI-LO SYSTEMS

USER'S MANUAL
EML-ROM512
ROM/RAM EMULATOR

USER'S MANUAL

EML-ROM512

ROM/RAM EMULATOR

Version V2.0
February 1993

HI-LO SYSTEMS

Model numbers mentioned in this manual are

ROM/RAM Emulator	: EML-ROM512
Interface card	: SAC-201(A)

CONTENTS

1 . Introduction and Installation

1.1 Product Overview 1
1.2 Parts List 1
1.3 Included Software 2
1.4 System Requirements 2
1.5 Installation 2

2. Simple Instructions and Quick Reference 6

3. Software Functions 7

3.1 DOS Shell 8
3.2 Load BIN or HEX file to ROM/RAM 8
3.3 Save ROM/RAM data to disk 10
3.4 Edit ROM/RAM 11
3.5 Change I/O base address 13
3.6 Display Loaded File History 14
3.7 Display ROM/RAM content 14
3.8 Type Select 15
3.9 Disassemble 16
3.10 Output Signal Control(HALT and RESET) 16
3.11 QUIT 17

4. Using the Utility Files 18

4.1 HEX to BINARY Code Converters 18
4.2 Dump Binary File to Console 19
4.3 2-Way, 4-Way Binary File Splitter 20
4.4 2-WAY, 4-Way Binary File Shuffler 21

the factory is 2E0. You can select from 16 different addresses ranging from 200H to 2F0H. If you know another card in your PC uses 2E0 as its address please select another address on this card. The following list details the exact position for the I/O settings:

	DIP SW	POSITION	I/O ADDRES
	SW1	1	200H
	SW1	2	210H
	SW1	3	220H
	SW1	4	230H
	SW1	5	240H
	SW1	6	250H
	SW1	7	260H
	SW1	8	270H
	SW2	1	280H
	SW2	2	290H
	SW2	3	2A0H
	SW2	4	2B0H
	SW2	5	2C0H
	SW2	6	2D0H
(DEFAULT)	SW2	7	2E0H
	SW2	8	2F0H

If you are unsure about which addresses in your PC are being used already, a simple test during the software installation will help you make sure there are no address conflicts with our emulator.

SW3 on the interface card selects the I/O waiting state between the computer and the interface card. The default value is 4 wait states. There is generally no need to change this setting.

3. Remove the PC casing and insert the SAC-201(A) interface card into any available expansion slot in the PC.

4. Connect the interface card to the emulator unit with the supplied DB-25 cable.

5. The two 28 pin flat cables fit into the emulator unit connectors marked #1 ROM/RAM and #2 ROM/RAM. The pin assignment for these connectors is as follows:

6. Now turn on your PC. The software can be run off of a floppy drive or placed into a directory on the hard drive by simply copying a:*.* into the directory of your choice. If you are going to run from the floppy drive please make a work copy of the software. The original diskettes are write protected and will cause you problems when you exit the program. This is also a safety precaution in case one of your diskettes is damaged later on. Now execute the EML512.EXE file from the DOS A:> prompt or from the DOS prompt in the directory where you installed the software. If everything is set up correctly the emulator's main menu will appear on the screen with the company name and emulator model number appearing in the upper left hand corner. The emulator is now ready for use. If there is any problem with your existing setup the screen will display the following message:

Error Identification on hardware!
Press <Q> to Quit
or press <CR> to continue...

If this message appears you should:

A)Press <CR> to continue if you changed the I/O base address on the PC interface card to an address other than 2E0H and go to ** below.

B)Press <Q> to quit if you left the I/O base address on the PC interface card at location 2E0. Then proceed as follows:

* Shut the power switch off.
* Check if the DB-25 and the SAC-201(A) interface card are properly connected.
* Turn the PC power back on and execute the EML512.EXE software again. If the error message appears again, this time press <CR> to continue.

** Now press <5> to select the "Change I/O base address" function. Make sure that the current I/O base address in the software is the same as the address selected on the SAC-201(A) interface card. If the address selected in the software does not match the address selected on the interface card then the "--emulator does not exist--" message will appear at the bottom of the "Change I/O base address' window." When the hardware and software are set up correctly the "--Emulator exists--" message will appear at the bottom of this window.

* If the address selected in the software matches the address selected on the interface card already and you still receive the "--Emulator does not exist--" message, you probably have an address conflict with some other card installed on your PC. You should then go back to step one of the Installation instructions and choose a new address on the interface card and then execute EML512.EXE again and select the same address in the software using the "Change I/O base address" function and proceed as before.
* If all else fails call your local representative.

2. Simple Instructions and Quick Reference

This section is written for people already familiar with our programming products and also can be used as a quick reference if you have not used the emulator for some time and do not remember how to operate it correctly.

Execute EML512.EXE from the DOS A:> prompt or from the DOS prompt in the directory where the software is installed to enter the software's main menu.

Press <T> to enter the IC type menu. Now select which type of device that you want each ROM/RAM to emulate. Use <TAB> to choose which ROM/RAM you are controlling and press <CR> when done selecting the device types.

Before loading a file to the emulator's memory make sure that either the power on the target board that you are connected to is turned off or that the address and data bus of the target board are in the reset or halt state. This can be accomplished by using the output signal clips of the emulator if your target microprocessor supports the reset or halt functions. (Please see section 3.10: OUTPUT SIGNAL CONTROL)

Load your file to the emulator by pressing <2> for "Load BIN or HEX file to ROM/RAM" and then choose which ROM/RAM (1 or 2 for independent EPROMs or W for even odd operation) you want to load the file to and follow the prompts for file information.

After loading the file(s) that you want to test, turn the target board's power on if it was turned off, or release the RESET, HALT clip signals to begin emulating.

Please consult the following Software Functions section for more detailed operating instructions.

3. Software Functions

Execute EML512.EXE from the A:> prompt or from the DOS prompt in the directory that you loaded the software; the screen will display the emulator's main menu:

```
                                          # 1 TYPE: EPROM x512 CHECKSUM:014F
SRAM/EPROM X64 - X512 section V1.11
--------------- Main Menu ---------------
1. DOS SHELL                              #2 TYPE: EPROM x256 CHECKSUM : 7978
2. Load BIN or HEX file to ROM/RAM
3. Save ROM/RAM data to disk
4. Edit ROM/RAM                           ========= CURRENT STATUS =========
5. Change I/O base address                RESET(#1) ->LOW  |  HALT (#2) -> LOW
6. Display loaded file history
7. Display ROM/RAM content

T. Type select
O. Output signal control
D. Disassemble

Q. Quit
-----------------------------------------

Allocation buffer size : 128K bytes
Start address at 27A0:0000


Select function ? d
```

The available functions of the emulator are displayed on the main menu for your selection. You will also find the company name, model and version number in the upper left corner. Status windows for #1ROM/RAM and #2ROM/RAM and the Output Signal Clips are found in the upper right corner. To enter any of the main menu's functions simply press the number or letter in front of the function.

The following are descriptions of the emulator's functions in the order that they appear in the menu.

3.1 DOS Shell

Press <1> to temporarily go to DOS. The screen will display:

```
Type EXIT to return to the Emulator
Microsoft (R) MS-DOS (R) Version X.XX
          (C) Copyright Microsoft Corp. 1981-1987
C:\>>
```

You can operate under DOS as normally although you may be limited because EML512.EXE remains resident in your PC's RAM. To return to the emulator's main menu just type 'EXIT' at the DOS prompt.

3.2 Load BIN or HEX file to ROM/RAM

Note: Before using this function the power of the target board being emulated must be turned off or the address and data bus of the target system must be in the high impedance state. (See 3.10 Output Signal Control)

This function is used to load a binary or a hexadecimal file to the emulator from the hard disk or from a floppy disk. After entering this function you will be prompted, "ready to load (1/2/W/<CR>)?". Select '1' to load a file to the #1 ROM/RAM, '2' to load a file to the #2 ROM/RAM, or 'W' to load the even bytes of a file to the #1 ROM/RAM and the odd bytes of a file to the #2 ROM/RAM.

You will then be prompted for the file name, you should:

A) Enter the complete file name that you want to load including the path name if the file is not in the same directory as EML512.EXE file.

or

B) Press the <TAB> key to switch control to the window on the left and use the following commands:

* <TAB>: switches control between the left and the right windows.

* <UP ARROW> & <DOWN ARROW>: used to move the highlighted bar to a file or a subdirectory where a file is located.
* <CR or Enter>: selects the file located under the highlighted bar to be loaded to the emulator.
* <ESC>: Exits the file loading menu.
* <DRIVE NAME>: Selects which drive's files will be shown in the selection window on the left.

NOTE: These commands will only work when the TAB key is pressed and the left window is active/highlighted.

```
C:\WORK\EML512\NES\*.*                      # 1 TYPE: EPROM x512 CHECKSUM:014F

A: N : .              <DIR>    12-28-92
B: O : ..             <DIR>    12-28-92     #2 TYPE: EPROM x256 CHECKSUM : 7978
C: P : EML.NCV          478    12-29-92
D: Q : EML_TRI.EXE    47277    08-21-92     ==== CURRENT STATUS ====
E: R : EML              516    08-21-92     RESET(#1) ->LOW  I  HALT (#2) -> LOW
F: S : EML.C          16319    01-11-93
G: T : DCL.H           7301    01-11-93       LOAD :
H: U : EMLB.C         16375    08-21-92
I : V : EMLD.C         4238    07-03-92     Ready to LOAD(1/2/W/CR>)?  1
J: W : EMLA.ASM       13210    12-29-92     File name: ICTEST.BIN
K: X : EMLC.C         87410    01-06-93     <B>in,<I>ntel HEX, <M>otorola S HEX: I
L: Y : ICTEST.BIN     32768    07-08-92     File start seg.(0000) :
M: Z : EML.EXE        48124    01-11-93     Unused bytes will be   <1>00,  <2>FF
       EML_H.EXE      47420    12-29-92                            <3>Don't care  :
       CURRENT.STS      399    01-11-93
       EML_T.EXE      47420    12-29-92     Press <CR> to main menu or <1>, <2>
                                            loads #1, #2 or <W> loads WORD wide;
                                            #1 is EVEN, #2 is ODD
Command: TAB  Esc  Enter
```

Once a file has been selected to be downloaded to the emulator you will be prompted "<B>in, <I>ntel HEX, <M>otorola S HEX:_". You should select 'B' if the file you are loading is a binary file, 'I' if the file you are loading is an Intel HEX file, or 'M' if the file is in one of the Motorola S Record formats. If your assembler or compiler has the option to output binary files, it is best to use this option. Files already in binary format load quicker to the emulator. All HEX files are actually converted to Binary by our software before being loaded to the emulator.

If 'B' is chosen you will be prompted "Load Address(00000):_", at this point you should press <CR or enter> on the keyboard if you want the file loaded at address 0 of the emulator's memory or enter in the HEX number of the address where you would like the file to be loaded. The file will now be loaded to the emulator.

If 'I' is chosen you will be prompted for "File start seg.(0000):_", here the file start segment address is required. All data lying between the start segment X 16 and the end of the file will be converted to binary and loaded to the emulator's memory. The segment range is from 0 to F000.

If 'M' is chosen you will be prompted for "File start addr.(000000):_", here the actual start address of the file is required. All data lying between the start address that is given and the end of the file will be converted to binary and loaded to the emulator. The address range is from 0 to FFFFFF.

If 'I' or 'M' was selected above you will be prompted, "Unused bytes will be <1>00 <2>FF <3>don't care:_". In a typical application when a HEX file is loaded to an EPROM only a portion of the EPROM's available memory is actually filled with your program or data after being converted to binary. The rest of the unused portions of the EPROM may be filled with all 00's or all FF's depending on your preference. To do this choose '1' or '2' at the prompt. If you would like to load more than one HEX file to one ROM, you should choose '3' at the prompt. This means that the portions of the EPROM not used by the first HEX file loaded to the EPROM can be filled with whatever other data you want by loading other HEX files.

3.3 Save ROM/RAM data to disk

This function saves the data/contents of the ROM/RAM Emulator's memory to disk in binary format. After entering this function you will be prompted: "Ready to save (1/2/W/<CR>)?_". Pressing '1' will save the #1 ROM/RAM's data/contents to disk, pressing '2' will save the #2 ROM/RAM's data to disk, pressing 'W' will save the #1 ROM/RAM as the even bytes of the file and the #2 ROM/RAM as the odd bytes of the file. Pressing <ESC> or <CR> will return you to the main menu.

```
SRAM/EPROM X64 - X512 section V1.11
------------------ Main Menu ------------------
1. DOS SHELL
2. Load BIN or HEX file to ROM/RAM
3. Save ROM/RAM data to disk
4. Edit ROM/RAM
5. Change I/O base address
6. Display loaded file history
7. Display ROM/RAM content

T. Type select
O. Output signal control
D. Disassemble

Q. Quit
-----------------------------------------------

Allocation buffer size : 128K bytes
Start address at 27A0:0000

Select function ? 3
```

```
# 1 TYPE: EPROM x512 CHECKSUM:01 4F

#2 TYPE: EPROM x256 CHECKSUM : 7978

CURRENT STATUS
RESET(#1) ->LOW    | HALT (#2) -> LOW

SAVE:
Ready to save (1/2/W/CR>)? 1
File name: TEST.BIN
Start address: 0
End address: FFFF
```

After choosing '1', '2', or 'W' you will be prompted for a file name and you should include all drive and path information. You will also be prompted for "start address:" and "end address:", these are the addresses in the emulator's memory that you want saved to disk as a file.

3.4 Edit ROM/RAM

Note: Before using this function the power of the target board being emulated must be turned off or the address and data bus of the target system must be in the high impedance state. (See 3.10 Output Signal Control)

When this function is selected our full screen editor's menu will appear after selecting '1', '2', or 'W' as before. Choosing '1' or '2' allows editing of the binary file in the #1 or #2 ROM/RAM's. Selecting 'W' means that you are editing the word wide contents of the emulator with #1 containing the even bytes and #2 containing the odd bytes. Using this menu the ROM/RAM of the emulator can be directly edited. The menu for the editor function is as follows:

```
                        EDITING COMMAND SUMMARY
 ==========================================================
 D [start], [end]                   <RETURN>:  DUMP
 E start                            <RETURN>:  EDIT
 M start, end, destination          <RETURN>:  MOVE BLOCK
 F start, end data                  <RETURN>:  FILL BLOCK
 P start, end                       <RETURN>:  PRINT BLOCK
 C start, end                       <RETURN>:  CHECK SUM
 S start, end, ASCII data           <RETURN>:  ASCII SEARCH MAX. 15 characters
 B start, end, BINARY data          <RETURN>:  BINARY SEARCH MAX. 7 Bytes
 . filename [argu1] [argu2] ...     <RETURN>:  SHELL
 ?                                  <RETURN>:  HELP
 Q                                  <RETURN>:  QUIT
 ==========================================================
 * The information listed below is for reference only :
   The absolute start address of BUFFER : 27A0 : 0000
   The Buffer size  : 128 Kbytes
```

Under the == command prompt you can use all of the commands listed above. The following is a brief description of the editor's commands:

The DUMP command displays all of the data between the specified start and end addresses to the screen.

The 'E', or EDIT, command allows you to directly edit the ROM/RAM Emulator's contents, using the keyboard, starting at 0000 if no address is specified. <page up> and <page down> keys and the arrow keys may be used to scroll through the memory in this mode.

The MOVE BLOCK command actually copies the block of data between the specified start and end addresses to the specified destination address.

The FILL BLOCK command will fill all the bytes between the specified start and end addresses with the specified data that is entered in the command.

The PRINT BLOCK command will send a block of data to a printer to produce a hard copy of the data.

The CHECKSUM command will add up all of the bytes in a specified block of data and return a four digit HEX checksum for that data.

The 'S' and 'B' search functions allow you to search through a block of data for some specific ASCII or binary information.

Some advanced users use the " . " command to pass the absolute start address of the buffer(given in SEGMENT : OFFSET form under the editor menu) to their own special purpose editing programs.

Example:

.SPECIAL_EDIT.EXE SEGMENT : OFFSET <CR>

The '?' command displays the above menu commands for reference.

The 'Q' command returns you to the ROM/RAM Emulator's main menu.

3.5 Change I/O Base Address

This function is used to select the address the host PC will communicate with the Emulator at. Our interface card is designed so that there are 16 choices of addresses ranging from 200H to 2F0H. After entering this function simply press the number or letter in front of the address that you would like to select. The current software base address is displayed towards the bottom of the dialogue window. The current base address must match the address that was selected on the SAC-201(A) interface card during the hardware installation in order for the emulator to operate correctly. The"--Emulator Exists--" message will be displayed at the bottom of the function's window if everything is set up correctly.

3.6 Display Loaded File History

This function will display the start and end addresses of the last 20 files loaded to the emulator in the following format:

```
Start        End          #1  or  #2  or  W   File name
00000   ---  07FFF   :        #1              ICTEST.BIN
00000   ---  07FFF   :        #2              INTEL_2.HEX
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :
00000   ---  00000   :

Press any key to continue ....
```

3.7 Display ROM/RAM Content

This function will display the ROM/RAM data in the #1 ROM/RAM, the #2 ROM/RAM, or will display the #1 data as the even bytes and the #2 data as the odd bytes when 'W' is chosen at the "ready to display (1/2/W/<CR>)?" prompt. Press the <ESC> key to stop the display. Please note, all information in the emulator's memory is in binary format, the data appears as hexadecimal numbers for display purposes only.

3.8 Type Select

This function is used to select which type of device each port will emulate. Use the <TAB> key to switch the cursor between the #1 and #2 ROM/RAM's. Press the number in front of the device that you want to emulate to select that type.

```
SRAM/EPROM X64 - X512 section V1.11
------------------- Main Menu -------------------
1. DOS SHELL
2. Load BIN or HEX file to ROM/RAM
3. Save ROM/RAM data to disk
4. Edit ROM/RAM
5. Change I/O base address
6. Display loaded file history
7. Display ROM/RAM content

T. Type select
O. Output signal control
D. Disassemble

Q. Quit
-------------------------------------------------------

Allocation buffer size : 128K bytes
Start address at 27A0:0000



Select function ? t
```

```
# 1 TYPE: EPROM x512 CHECKSUM:014F

#2 TYPE: EPROM x256 CHECKSUM : 7978

=========== CURRENT STATUS ===========
RESET(#1) ->LOW   |  HALT (#2) -> LOW

------------- TYPE SELECT -------------
   #1                        #2
0. EPROM x64              0. EPROM x64
1. EPROM x 128            1. EPROM x 128
2. EPROM x 256            2. EPROM x 256
3. EPROM x 512            3. EPROM x 512
4. SRAM x 64              4. SRAM x 64
5. SRAM x 256             5. SRAM x 256
6. EPROM x 010 (need POD-1M32)
7. EPROM x1024(need POD-1M40)

Select which ?

<TAB> #1,#2 select/ <CR> to main menu_
```

Choices 6 and 7 can only be used if you have the corresponding adapter boards. To emulate a 27(C)010 device the POD-1M32 32 pin adapter is required. To emulate a 27(C)1024 device the POD-1M40 40 pin adapter is required. To use these adapters simply insert the #1 ROM/RAM plug into the adapter's U1 socket and the #2 ROM/RAM plug into the adapter's U2 socket. The rest of the operation is the same as before.

3.9 Disassemble

We offer several different disassemblers for use with this product.(Please contact your local representative for more information.) This function can only be used if one of these disassemblers is installed in the same directory or is on the same floppy diskette as the EML512.EXE file. The '1', '2', or 'W' modes can be selected as before and then the data will be disassembled from binary information into assembly code.

```
SRAM/EPROM X64 - X512 section V1.11
---------------------Main Menu ------------------
1. DOS SHELL
2. Load BIN or HEX file to ROM/RAM
3. Save ROM/RAM data to disk
4. Edit ROM/RAM
5. Change I/O base address
6. Display loaded file history
7. Display ROM/RAM content

T. Type select
O. Output signal control
D. Disassemble

Q. Quit
--------------------------------------------------------

Allocation buffer size : 128K bytes
Start address at 27A0:0000


Select function ? d
```

```
# 1 TYPE: EPROM x512 CHECKSUM:014F

#2 TYPE: EPROM x256 CHECKSUM : 7978

CURRENT STATUS
RESET(#1) ->LOW  |  HALT (#2) -> LOW

DISASSEMBLE :
Ready to disassemble (1/2/W/CR>)? 1
0. dasmz80.exe
1. dasm85.exe
2. dasm48.exe
3. dasm51.exe
4. dasm09.exe
5. dasm11.exe

Select which ?
Press <CR> to main menu or <1>, <2>
disassembles #1, #2 or <W> disas-
sembles WORD; #1 is EVEN, #2 is ODD
```

3.10 Output Signal Control

This function is used to control the output signals from the two clips marked "RESET" and "HALT" that are attached to the two 28 pin flat cables.

The #1 ROM/RAM flat cable has two clips to attach to the target board, the GND clip should be connected to the Ground located on the target board and the RESET clip should be connected to the RESET pin on the microcontroller if available.

The #2 ROM/RAM flat cable also has two clips that should be attached to the target board. The GND clip should be connected to the Ground of the target board and the HALT clip should be connected to the HALT pin of the microcontroller(if one is present).

When this function is selected the Output Signal Status window expands to show the status of the output signals on these clips as well as some simple instructions. Pressing <R> will toggle the value of the RESET output signal and pressing <H> will toggle the value of the HALT output signal.

Use the HALT or RESET clips to reset or halt the target CPU before attempting to load a file to the emulator's memory. This will leave the data and address buses on the target board in the HI-impedance state for file loading. If this is not done the target CPU and the emulator will fight for the signals on the bus and your data may be corrupted.

If there are no RESET or HALT pins on your target CPU then the target board's power must be turned off before a file is loaded, and turned back on after the file is loaded.

3.11 Quit

When <Q> is pressed the I/O base address and the types of devices being emulated will be saved to the current working disk and the cursor will return to DOS. The next time that you enter into the EML512.EXE software the I/O base address and the types of devices being emulated will be the same as the last time when you QUIT the main menu.

4.1 HEX TO BINARY code converters

Before any information is loaded to our emulator it is first converted to binary format. These HEX converters will convert your HEX format files to executable ROM code that is recognizable to your target CPU. Even though our emulator can load HEX files directly, some users may like to use this converter to convert all their HEX files to binary format so that they will be downloaded at a faster rate to the emulator. If, for example, a user has a 2Mbit or larger file that needs to be split into several files these HEX to BIN converters will be useful to produce files that our other Split and Shuffle utilities can work on. (SPLITX.EXE and SHUFFX.EXE only accept binary files as input.)

Two HEX file converters are included with your software. The first is HEXBIN.EXE which has a maximum conversion size of 64K bytes. This converter can convert the following file formats to binary(BIN) format files:

INTEL HEX format
MOTOROLA S HEX Formats
TEKTRONIX HEX format

HEXBIN2.EXE can convert INTEL, MOTOROLA, and TEKTRONIX extended HEX format files to binary files with a maximum conversion size of 256 Kbyte. Another difference between HEXBIN2.EXE and HEXBIN.EXE is that HEXBIN2.EXE allows you to specify a starting address, the leading "garbage" of the file will be skipped to maintain a small sized output binary file.

The HEX to BIN converters can be run in the command line mode or in the prompt mode. In the command line mode the converters can be operated as follows at the DOS prompt:

```
> HEXBIN2.EXE    [HEX file name] [BIN file name] [HEX format]
                 [start address] <CR>
```

All entries in brackets are options.

HEX file name and BIN file name are the standard file names specified in DOS. The HEX file name is the input file and the BIN file name is the resulting output binary file.

HEX format is: I for INTEL HEX
M for MOTOROLA S HEX(S records)
T for TEKTRONIX HEX

Start address is a Hexadecimal digit.

Note: In the INTEL extended HEX format, the start address represents the start segment address. All data lying between the start segment X 16 and the end of the file will be converted to binary format.

In the MOTOROLA S HEX format the start address represents the actual file start address. All data lying between the start address and the end of the file will be converted to binary format.

For example:

```
A:> HEXBIN2 DEMO.HEX DEMO.BIN I 1000 <CR>
```

HEXBIN2 will convert the HEX file DEMO.HEX to the binary file DEMO.BIN using the INTEL HEX converting technique, and only data lying after 1000OH will be converted.

These programs can be used in the prompt mode as follows:

```
A:> HEXBIN2 <CR>
```

```
HEX FILE NAME   : DEMO.HEX
BIN FILE NAME   : DEMO.BIN
HEX FORMAT (<I>NTEL <M>OTOROLA <T>EKTRONIX) : I
```

4.2 Dump Binary File To Console

Executable ROM code BINARY files can not be displayed on the screen using a DOS TYPE command. DUMP.EXE can convert a BINARY file to HEXADECIMAL characters that can be displayed on the screen or by a printer.

The input command under the DOS command prompt is:

A:> [^P] DUMP.EXE FILENAME [start address] <CR>

All entries in brackets [ ] are options.

^P : <CTRL> + P   <CR> : return or Enter key

The ^P option can be used to connect the PC with the printer, the data displayed on the screen will also be output to the printer.

FILENAME is a standard filename as specified in DOS.

Start address is a hexadecimal digit that specifies what address the display will start at.

4.3 2-way, 4-way Binary File Splitter

SPLIT2.EXE can split a 16-bit binary source file into two 8-bit binary files. One file is the collection of data lying on the LOW byte of the 16-bit words in the source file. The other file is the collection of data lying on the HIGH byte of the 16 bit words in the source file. The utility is generally used to split a 16 bit file into two 8 bit files that can be programmed to an EVEN EPROM and an ODD EPROM.

SPLIT4.EXE can split 32-bit binary format source files into four 8-bit binary files. The first file is the collection of data lying on the 1st byte of the 32-bit words in the source file. The second file is the collection of data lying on the 2nd byte of the 32-bit words in the source file. The third file is the collection of data lying on the 3rd byte of a 32-bit words in the source file. The fourth file is the collection of data lying on the 4th byte of a 32-bit word in the source file.

The input command under the DOS command prompt is:

A:>SPLIT2 [input file] [output EVEN file] [output ODD file name] <CR>
A:>SPLIT4 [input file] [output 1st file] [output 2nd file]
[output 3rd file] [output 4th file] <CR>

All items in brackets[ ] are standard filenames specified under DOS.

4.4 2-way, 4-way Binary File Shuffler

SHUFF2.EXE can shuffle two 8 bit binary source files into a 16-bit binary file. The first 8-bit file will be collected to the LOW bytes of the 16 bit words in the output file. The second 8-bit file will be collected to the HIGH bytes of the 16-bit words in the output file.

SHUFF4.EXE can shuffle four 8-bit binary format source files into one 32-bit binary file. The first 8-bit file will be collected to the 1st byte of the 32 bit output file. The second 8-bit file will be collected to the 2nd byte of the 32 bit output file. The third file will be collected to the 3rd byte of the 32 bit output file. The fourth file will be collected to the 4th byte of the 32 bit output file.

The input command under the DOS command prompt is :

```
A:>SHUFF2     [output file] [input EVEN file] [input ODD file] <CR>
A:>SHUFF4     [output file] [input 1st file] [input 2nd file]
              [input 3rd file] [input 4th file]
```

Again, all terms in brackets are standard DOS file names.

MMAUEMLROM512.P 2/1993