HI-LO SYSTEMS

USER'S MANUAL
EPP-01A/04A/08A
PC-BASED
EPROM PROGRAMMER

USER'S MANUAL

EPP-01A/04A/08A

PC-based EPROM Programmer

Version V1.1
September 1992

HI-LO SYSTEMS

Model numbers mentioned in this manual are:

System Adapter Card : SAC-201
Programmer Module : EPP-01A, 1 socket
EPP-04A, 4 sockets
EPP-08A, 8 sockets

1. INTRODUCTION

2. INSTALLATION

3. USING THE PROGRAMMER

4. DETAILED FUNCTION DESCRIPTION

APPENDICES

1.1 Manual Contents

This manual describes the methods for installing and operating the EPROM Programmer with an IBM PC or compatible running MS-DOS or PC-DOS.

A user, who is familiar with general PC software installation problems, can easily prepare the equipment for operation.

This manual also contains information about the EPROM Programmer's usage and detailed functions.

1.2 Product Configuration

Before using this product, please carefully check that your package includes:

* PC System Adapter Card.
* 1-M. cable with two 25-pin, D-type connectors on each of the cable's ends.
* Programmer module.
* User's manual.
* One diskette with the following files:

README.DOC:	Software & hardware revision message.
EPP8Mx.EXE:	Main program for the programmer module.
EPP8Mx.DAT:	Initializing parameters which will be loaded upon main program execution. (x: 1 for one-socket programmer module, 4 for four-socket programmer module, 8 for eight-socket programmer module.)
DUMP.EXE:	Dump file to console in binary format.
HEXBIN.EXE:	HEX to BINARY file converter.
HEXBIN2.EXE:	EXTENDED HEX to BINARY file converter.
SPLIT2.EXE:	2-way file splitter.
SPLIT4.EXE:	4-way file splitter.
SHUFF2.EXE:	2-way file shuffler.
SHUFF4.EXE:	4-way file shuffler.

This chapter describes the method for installing an EPROM programmer on an IBM PC/XT/AT computer or compatible running MS-DOS or PC-DOS.

The installer can easily set up the programmer module for service by checking the system requirements and performing the following steps:

2.1 Host System Requirements:

* IBM PC/XT/AT or compatible PC.
* Max. PC speed is up to 25 MHz, zero memory wait state.
* Min. 640K bytes memory.
* Min. 1 floppy disk drive. A hard disk is preferable.
* Operating system: MS-DOS or PC-DOS, version 2.0 or later.

2.2 Software Installation Procedures:

2.2.1 The PC system has one hard disk drive and at least one floppy disk drive.

The hard disk installation procedure is very simple. Follow the listed steps to copy all the files on the supplied diskette to a subdirectory on the hard disk.

Steps as follows:

	Description
C:\>md EPP	To generate a programmer subdirectory (EPP)
C:\>cd EPP	To change directory to EPP
C:\>copy A:*.* C:	To copy all files in A: to the current directory in C:

Whenever a new updated diskette reaches you, by applying the above mentioned method, you can easily perform your own updating.

Then, proceed with the Hardware Installation Procedure.

2.2.2 The PC system has floppy disk drive only.

Proceed with step 2.3 Hardware Installation Procedures, because this configuration needs no Software Installation.

2.3 Hardware Installation Procedures:

Before starting the Installation Procedures, it is necessary for the user to make and use a work-copy disk of the original software diskette. Do not use the original software diskette! If you attempt to use the original diskette, your monitor will display a "disk write error" message whenever you terminate the main program. To install the EPROM Programmer and the software supplied, follow these steps:

Step 1:
Switch off your computer system, and carefully open the computer cover.

Step 2:
Check the DIP switches of the PC system adapter card,
For I/O address selection 2E0H (default):
SW1 : all off.
SW2 : position 7 on, others off.
For I/O wait state selection:
SW3 : position 2 on, 4 waits (default).

Step 3:
Gently insert the system adapter card into the PC slot, and fasten it to the PC frame with the slot cover screw.

Step 4:
Connect the programmer module to the system adapter card using the attached cable. The male cable end must be connected to the system adapter card; the female end must be connected to the programmer module.

CAUTION
======

Do not connect the programmer module to the system adapter card while the computer is turned on. Such an installation can put the module in an "unknown" state, and damage the DEVICE and/or the module.

Step 5:
Turn on the computer and check the LEDs on the programmer module.

ON LED must be ON.
BUSY LED must be OFF.
Other LEDs are in a random state.

If the LEDs are not in the correct state, please turn off the PC and check all connections between the system adapter card and the PC slot, and cable connections between the system adapter card and the programmer module. Then turn on the computer and again check the LEDs on the programmer module.

Step 6:
Boot DOS and then perform "Change I/O Base Address" function according to the following sequences:

Step 6-1
Boot DOS in disk drive a:
In response to the operating system prompt, replace the DOS diskette with a work copy and type the following command, if you are using the main program:

A>EPP8Mx <CR>

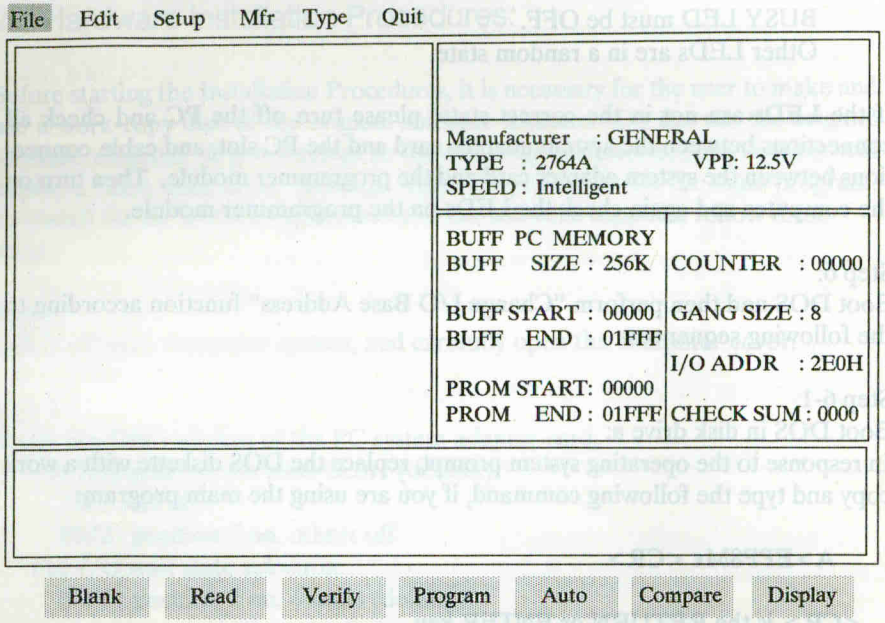
<CR> is the RETURN or ENTER key.

NOTE

x stands for socket number of the programmer. 1 for the one-socket programmer module, 4 for the four-socket programmer module and 8 for the eight-socket programmer module.

The main program will be executed, and will look for the parameter file EPP8Mx.DAT. If it exists on the default disk drive, the parameters of I/O address, Manufacturer, Type and Programming Speed will be loaded as your default state. Otherwise the values set before delivery, I/O address (2E0), Manufacturer (General EPROM), Type (2764A), Programming Speed (intelligent) will be considered as the default state.

At the same time, the main program will also check the identification circuit in the programmer module. Once it has been checked, the main program will display the following function menu on lthe screen and wait for the user's command.



Main menu

Step 6-2
If the function menu is not displayed on the screen, the work-copy diskette is possibly bad. Remake and use another work-copy diskette and then retry step 6-1.

NOTE

If this problem persists, please contact your local distributor for help.

Step 6-3
If the following error message appears on the screen,

Error identification on hardware!
Press "Q" to quit,
or press <CR> to continue...

The hardware I/O address of the system adapter card is mismatched with the software default I/O address, or conflicts with other add-on cards inserted in the PC. Press "Q" to quit to DOS prompt, turn off the computer and recheck all the connections between the system adapter card and the PC, and all cable connections between the system adapter card and the programmer module.

After rechecking all these connections, repeat the installation procedure from Step 1.

NOTE

Most problems are due to poor contacts between the system adapter card and the PC slot, and cable connections between the system adapter card and the programmer module.

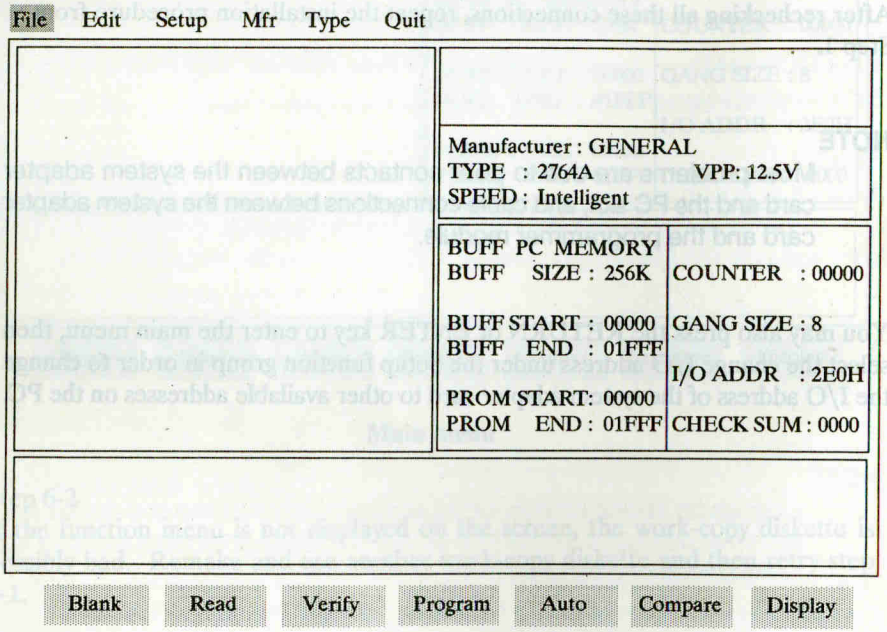
You may also press the RETURN or ENTER key to enter the main menu, then select the change I/O address under the Setup function group in order to change the I/O address of the system adapter card to other available addresses on the PC.

3. USING THE PROGRAMMER

In this chapter we will help the user step by step to become familiar with his new EPROM Programmer tools. The Installation Procedures described in the previous chapter must have already been performed.

3.1 Viewing the Main Menu

Studying the main menu before starting to operate the functions will help you familiarize yourself with the window menu features.



Main menu

3.1.1 Main Function Line

The top line of the main menu is called the main function line. It has six groups of functions: File, Edit, Setup, Mfr., Type, Quit.

3.1.2 Immediate Execution Line

The bottom line on the main menu is called the immediate execution line. From left to right, 7 functions for selection are BLANK, READ, VERIFY, PROGRAM, AUTO, COMPARE and DISPLAY; they can be immediately executed through the main program.

It is very simple to select an individual function by pressing the first letter of that function. After pressing the first letter of the desired function, the subwindow or description of that function will be prompted on the screen for the next instruction.

3.1.3 Status Field

On the right side of the main menu there are 4 windows with information on EPROM, from top to bottom, logo, version, current EPROM, Mfr., Type, VPP, Speed, working buffer and EPROM addresses, gang size, I/O address... This group of windows is called the status field.

Whenever you are going to use this programmer, check to make sure that the status in this field meets your requirements. Otherwise the EPROM will be destroyed or programmed to an unknown state.

3.2 Getting Started

This section presents a simple example to help you become familiar with the programmer module and some commonly used functions. Experiment with the suggested options. If you do not get the desired results, make sure all the cables are connected firmly.

Follow the steps to learn how to start and to exit the main menu and to use functions such as Mfr., TYPE, LOAD, BLANK and PROGRAM. The example here is to practise with a 2764A 8K EPROM; other 27 series EPROMs can also be used for exercise.

Now, let's practise with the 2764A EPROM you have at hand.

You can apply either of the following methods to begin the EPROM programming according the system you are using.

A. If you have copied the main program files into hard disk (see Section 2.2.1),

Change the current directory to the directory under which the main program has been copied.

Using EPP as an example, execute the main program under EPP directory as follows:

C:\EPP\EPP8Mx <ENTER>

NOTE

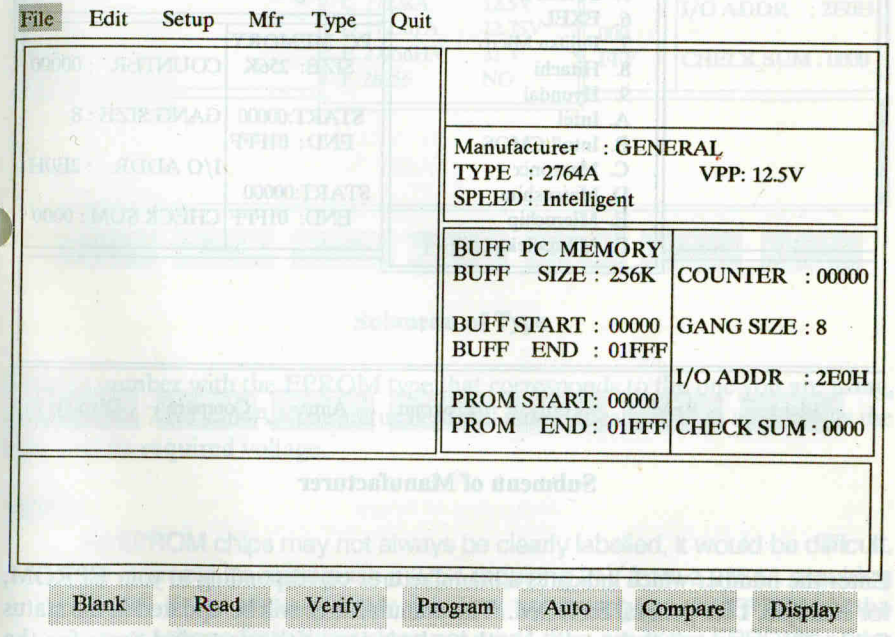
x stands for socket number of the programmer module. 1 for the one-socket programmer module, 4 for the four-socket programmer module and 8 for the eight-socket programmer module.

B. If you use a floppy disk only,

Your first step is to execute the main program directly from drive A: by typing:

A:>EPP8Mx <ENTER>

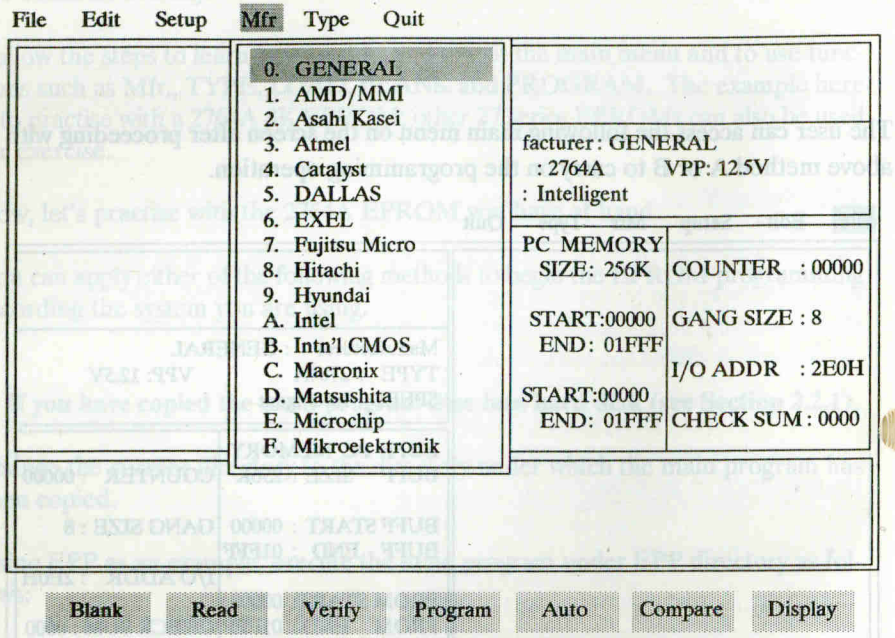
The user can access the following main menu on the screen after proceeding with above method A or B to carry on the programming operation.



Main menu

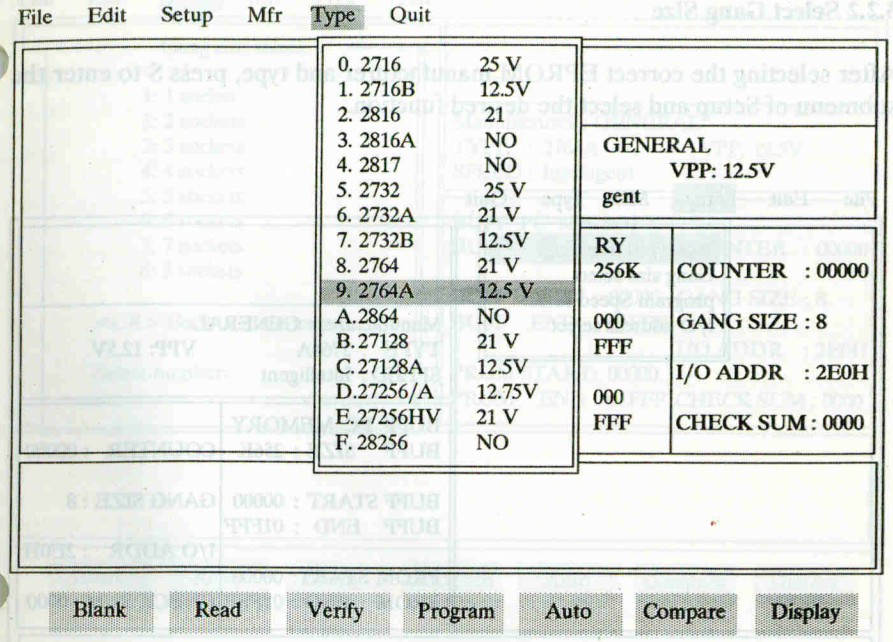
3.2.1 Select EPROM Type

Selecting the EPROM manufacturer, type and voltage should always be the first step. After typing M from the main menu, the screen will display the submenu under Mfr.



Submenu of Manufacturer

Enter the number which indicates a manufacturer corresponding to your EPROM, for example, 1 for general EPROM. The manufacturer will be updated in the status field. The screen will then display the subwindow with supported types for the selected manufacturer.



Submenu of Type

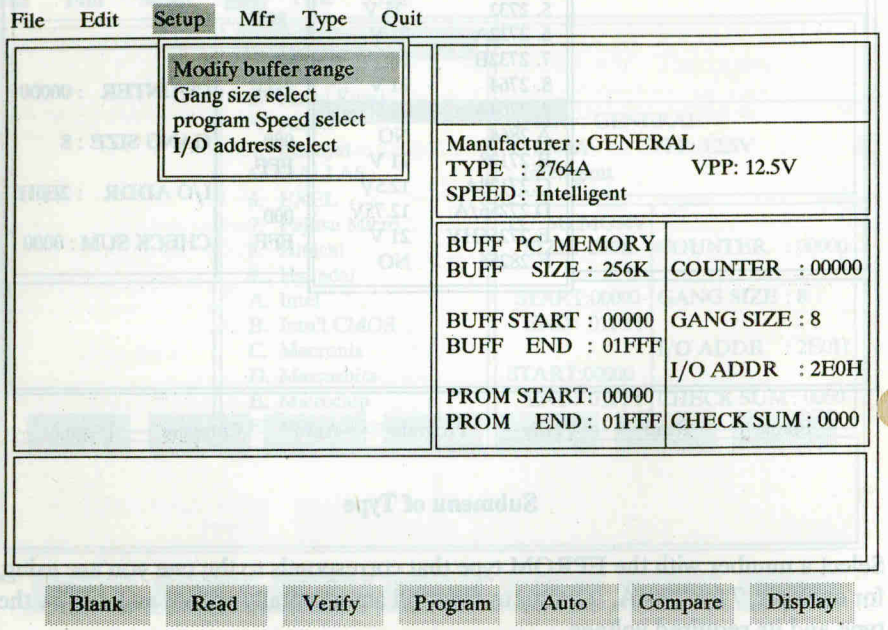
Select a number with the EPROM type that corresponds to the one you are using, for example, 7 for 2764A. The status field will automatically update and display the type and its required voltage.

NOTE

As EPROM chips may not always be clearly labelled, it would be difficult, for example, to distinguish a 2764 requiring 21V from a 2764A requiring 12.5V. If you encounter this problem, try to program the EPROM at the lower voltage first. If this does not work, then erase the EPROM and try the higher voltage. Please refer to Appendix B for more information about EPROM manufacturers, types and voltage selections.

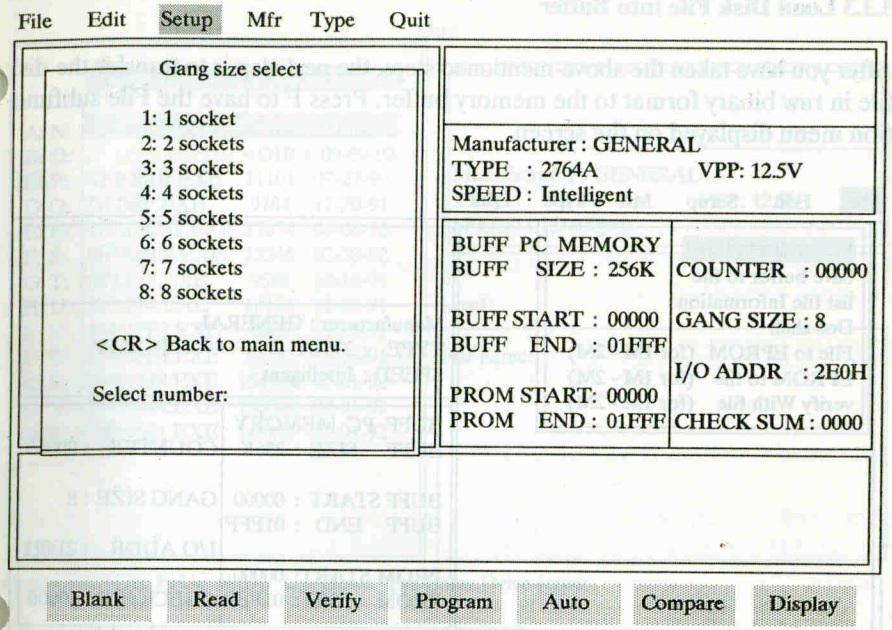
3.2.2 Select Gang Size

After selecting the correct EPROM manufacturer and type, press S to enter the submenu of Setup and select the desired function.



Submenu of Setup

Type G to obtain the submenu for gang size selection.



Gang size select

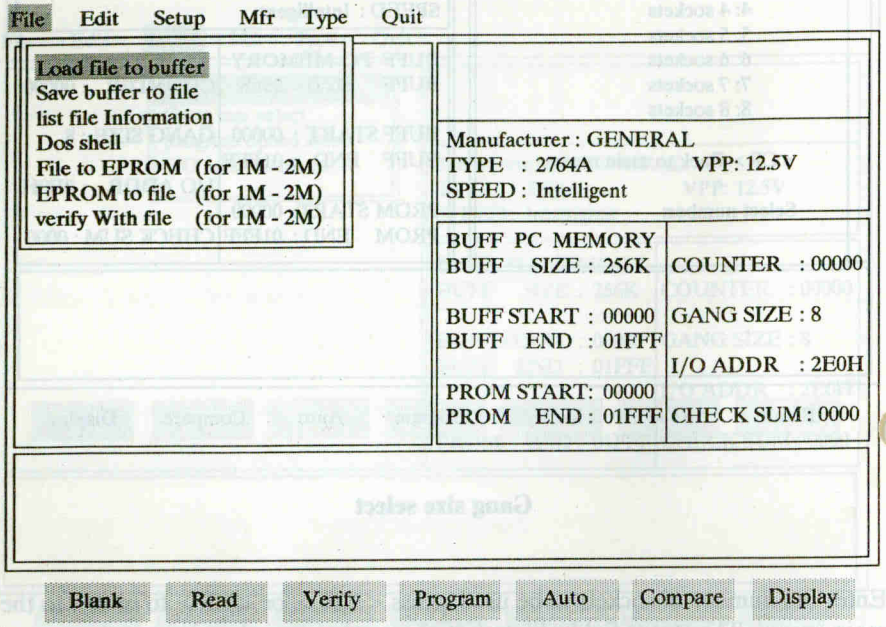
Enter the number of sockets to be used, press <ESC> or <CR> to return to the main menu. The status field will update the newly entered socket number.

NOTE

The one-socket programmer module dose not require gang size selection, because its gang size has already been fixed at 1.

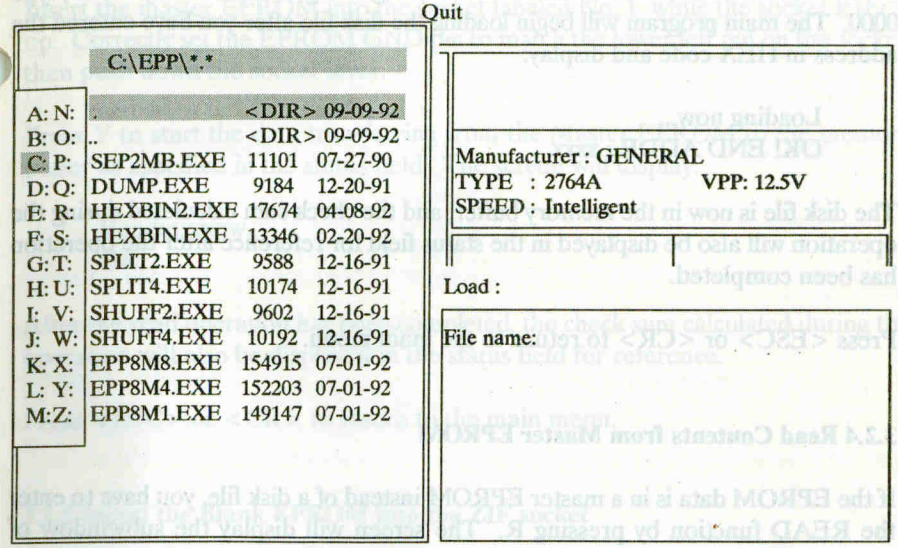
3.3.3 Load Disk File into Buffer

After you have taken the above-mentioned steps, the next step is to transfer the disk file in raw binary format to the memory buffer. Press F to have the File subfunction menu displayed on the screen.



File

Press L, or use the up/down arrow keys to select Load file to buffer function. Press <CR> to access a dialog window for the File loading on the screen.



Load file

When prompted, enter the complete file name including the drive letter and path names, if any, then press <CR>. Enter the buffer start address in the status field, and press <CR>. The file will be loaded to the specified buffer address.

An alternative way to enter in the file name is:

- Press the <TAB> key to change the active window from the load window to the directory window.
- Use the <UP> or <DOWN> keys to move the highlighted bar to the desired file or subdirectory.
- Press the <ENTER> key to select the desired file or subdirectory.
- The active window will be automatically changed back to the load window, and the main program will display the selected name on the "File Name" line.
- If the subdirectory name is selected, the directory window will list all the files of that subdirectory for the next selection.

After the file name has been entered, the main program will load the file into the specified buffer address.

If you wish to transfer the disk file to the very beginning of the memory buffer, type 0000. The main program will begin loading the disk file after you have entered the address in HEX code and display:

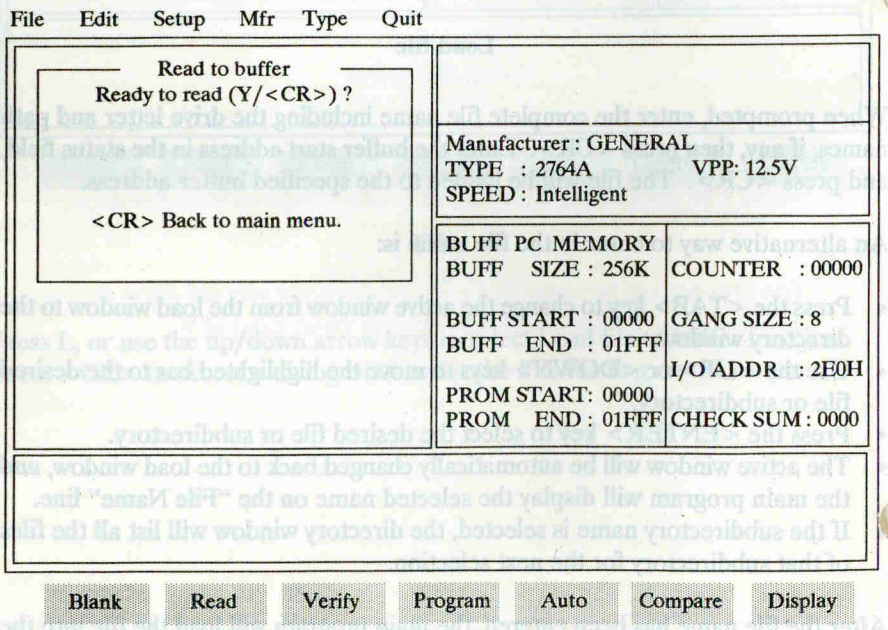
Loading now...
OK! END ADDR.: xxxx

The disk file is now in the memory buffer, and the check sum calculated during the operation will also be displayed in the status field for reference after the operation has been completed.

Press <ESC> or <CR> to return to the main menu.

3.2.4 Read Contents from Master EPROM

If the EPROM data is in a master EPROM instead of a disk file, you have to enter the READ function by pressing R. The screen will display the subwindow of READ.



Subwindow of Read

Insert the master EPROM into the socket labeled No. 1 while the socket lever is up. Correctly set the EPROM GND pin to match the lower left pin on the socket, then push down the socket lever.

Press Y to start the data transferring from the Master EPROM to the memory buffer as specified in the status field. The screen will display:

Reading now...
OK!

After the read operation has been completed, the check sum calculated during the operation will also be displayed in the status field for reference.

Press <ESC> or <CR> to return to the main menu.

3.2.5 Insert the Blank EPROM into the ZIF socket

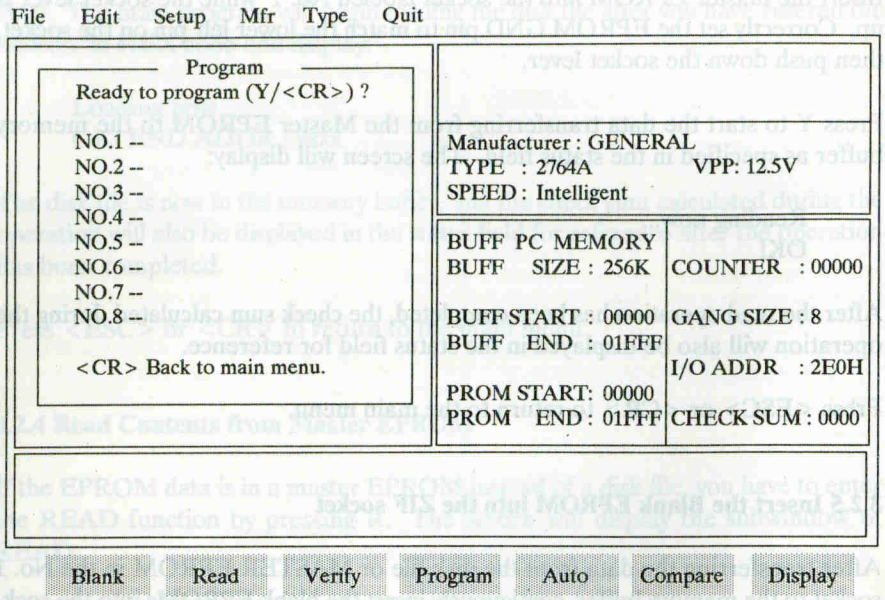
After transferring the data from the disk file or MASTER EPROM in the No. 1 socket to the memory buffer, contiguously insert the blank EPROMs into the sockets labeled from 2 to 8 in the same manner as the master EPROM was inserted into its socket.

CAUTION
= = = = = =

The EPROM must be inserted with the notched end towards the end of the socket; it will otherwise be destroyed or programmed to an unknown state.

3.2.6 Program Buffer Contents to EPROM

After loading the disk file or reading the master EPROM data into the memory buffer, and inserting the blank EPROMs, press P to program the EPROMs. The subwindow of PROGRAM will appear on the screen.



Subwindow of Program

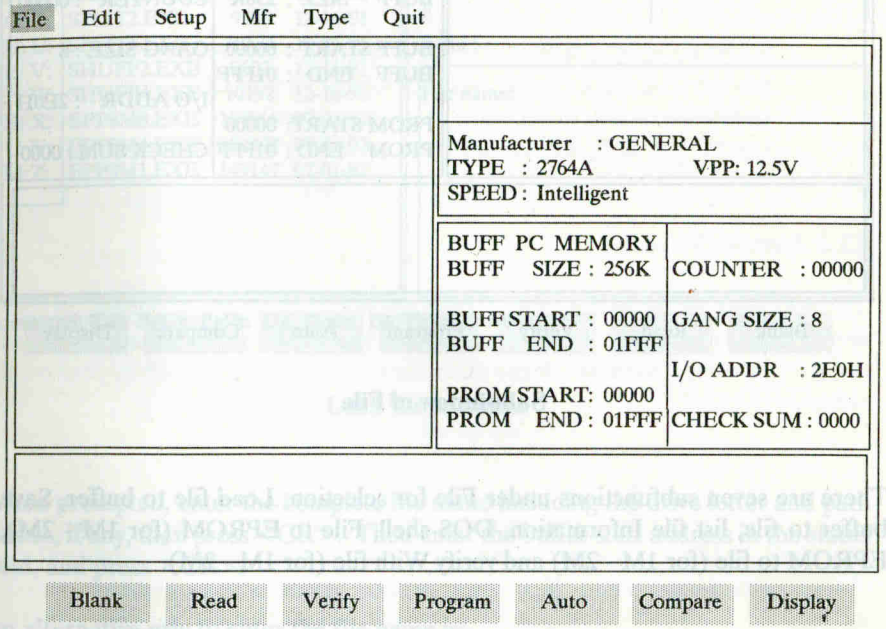
Press Y to activate the programming function. By the end of the programming process, the comparison between the EPROM contents and memory buffer will be executed automatically. If there are discrepancies, they will be displayed as follows:

#1. OK!
#2. ERROR AT XXXX
#3. OK!
#4. OK!
#5. OK!
#6. OK!
#7. OK!
#8. OK!

This completes the whole programming process. To program other EPROMs, wait till the BUSY LED turns off, then replace the EPROMs and type Y again.

Press <ESC> or <CR> to return to the main menu and exit the programming process.

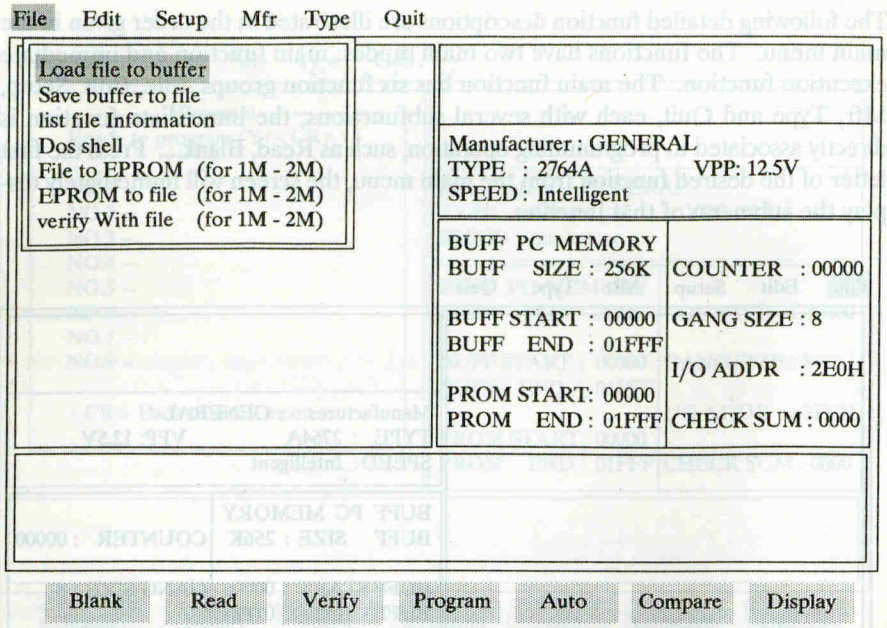
The following detailed function descriptions are illustrated in the order given in the main menu. The functions have two main modes: main function and immediate execution function. The main function has six function groups, File, Edit, Setup, Mfr, Type and Quit, each with several subfunctions; the immediate function is directly associated to programming operation, such as Read, Blank.... Press the first letter of the desired function from the main menu, the screen will immediately display the submenu of that function.



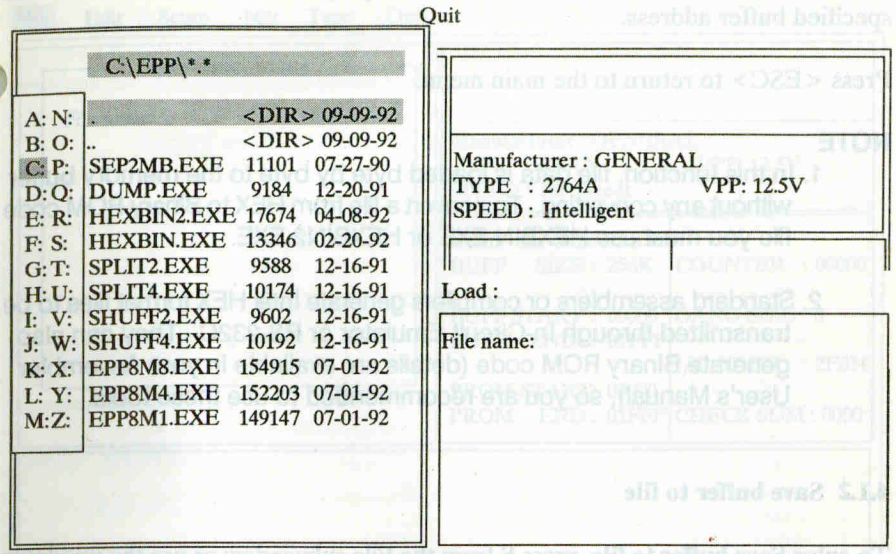
Main menu

4.1 File

To enter **File**, press **F** or use the left/right arrows keys to move the highlighted bar to the desired function and then press <CR>. The File subwindow will appear on the screen.



Subwindow of File



Load file to buffer

There are seven subfunctions under File for selection: Load file to buffer, Save buffer to file, list file Information, DOS shell, File to EPROM (for 1M - 2M), EPROM to file (for 1M - 2M) and verify With file (for 1M - 2M).

4.1.1 Load file to buffer

To enter **Load file to buffer**, press L from the File subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. A dialog window for the File loading will appear on the screen.

When prompted, enter the complete file name including the drive letter and path names, if any, then press <CR>. Then enter the buffer start address in the status field, and press <CR>.

An alternative way to enter the file name is:

- Press the <TAB> key to change the active window from the load window to the directory window.
- Use the <UP> or <DOWN> keys to move the highlighted bar to the desired file or subdirectory.
- Press the <ENTER> key to select the desired file or subdirectory.
- The active window will be automatically changed back to the load window, and the main program will display the selected name on the "File Name" line.
- If a subdirectory name is selected, the directory window will list all the files of that subdirectory for the next selection.

After the file name has been entered, the main program will load the file into the specified buffer address.

Press <ESC> to return to the main menu.

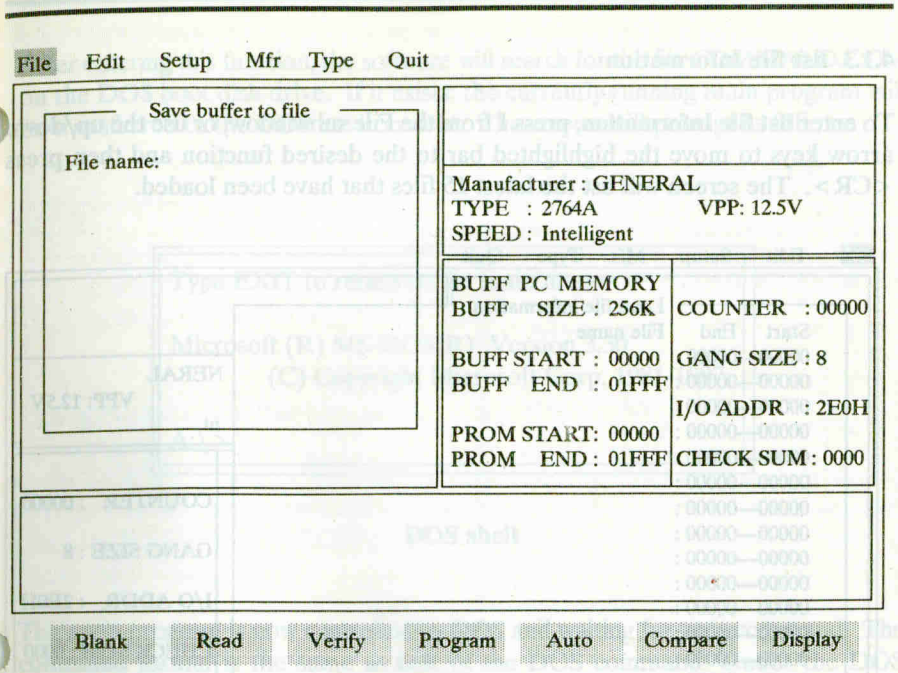
NOTE

1. In this function, file data is loaded byte by byte to the memory buffer without any converting. To convert a file from HEX to Binary ROM code file you must use HEXBIN.EXE or HEXBIN2.EXE.

2. Standard assemblers or compilers generate Intel HEX format files to be transmitted through In-Circuit Emulator or RS-232C. They can also generate Binary ROM code (details are available in your Assembler User's Manual), so you are recommended to use these tools.

4.1.2 Save buffer to file

To enter **Save buffer to file**, press S from the File subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. A dialog window for the Save file to buffer function will appear on the screen.



Save buffer to file

When prompted, enter the complete file name under which the file to be saved, including the drive letter and path names, if any, and then press <CR>. Enter the buffer start address of your desired portion, press <CR>, then enter its corresponding buffer end address and press <CR>.

When the screen displays the "OK!" message, the specified buffer contents have already been saved.

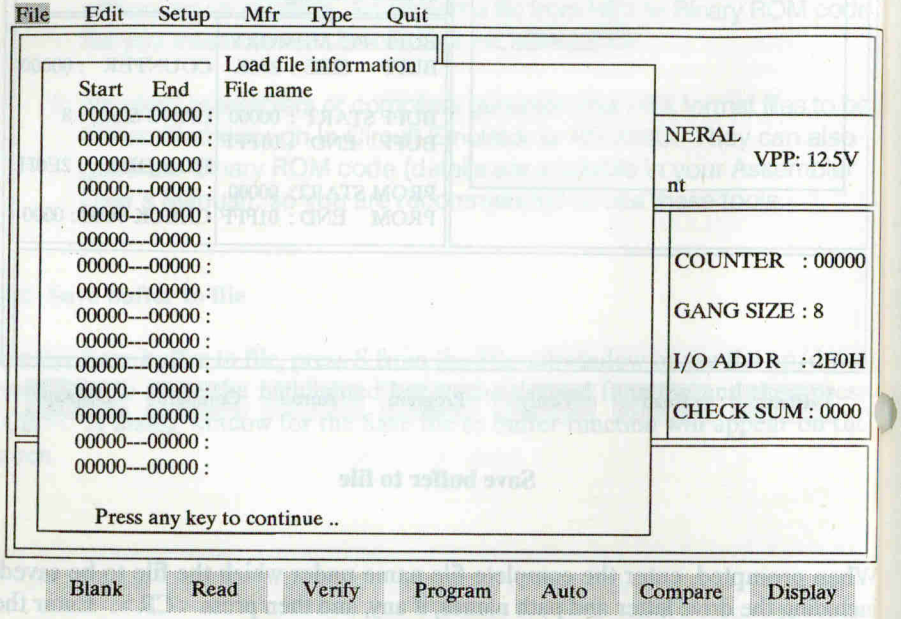
Press <ESC> to return to the main menu.

NOTE

Files saved under this function are exact Binary file without a convert.

4.1.3 list file Information

To enter **list file Information**, press **I** from the File subwindow, or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. The screen will list the latest 15 files that have been loaded.



list file Information

Press <ESC> to return to the main menu.

4.1.4 DOS shell

To enter **DOS shell**, press **D** from the File subwindow or use the up/down arrow keys to to move the highlighted bar to the desired function and then press <CR>.

After entering this function, the software will search for the file COMMAND.COM on the DOS boot disk drive. If it exists, the currently running main program will be transferred to the DOS environment. The screen will prompt as follows:

```
Type EXIT to return to the main menu

Microsoft (R) MS-DOS(R)  Version 3.30
          (C) Copyright Microsoft Corp. 1981-1987

A:\>
```

DOS shell

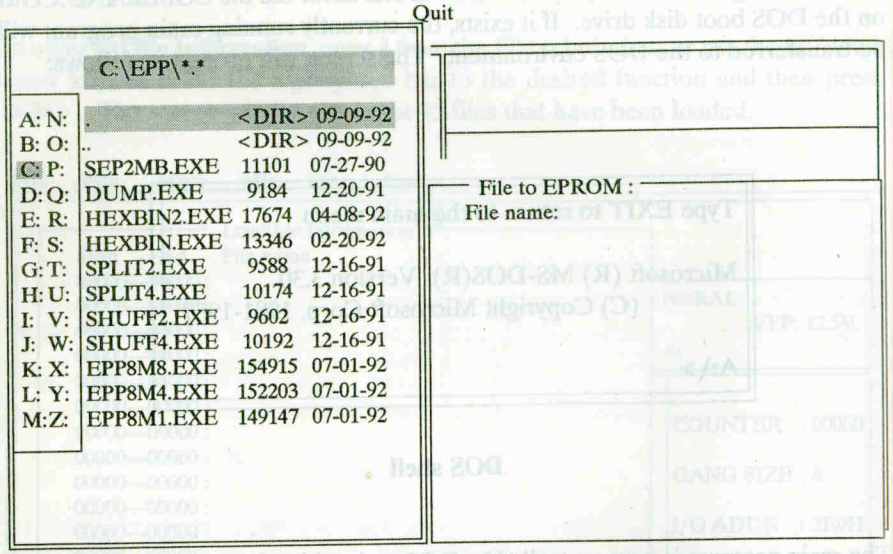
The main program is now controlled by DOS and waiting for your command. The command format is the same as that of the DOS command. Under the DOS command prompt, you can type EXIT and press <CR> to return to the main menu.

NOTE

This function will invoke the COMMAND.COM. The user has to prepare this file on the DOS boot disk drive, otherwise the function will not work.

4.1.5 File to EPROM (for 1M - 2M)

To enter **File to EPROM** (for 1M - 2M), press **F** from the File subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. The File to EPROM subwindow will appear on the screen.



File to EPROM

Enter the complete file name including the drive letter and path names, if any, from the File to EPROM subwindow.

An alternative way to enter the file name is:

- Press the <TAB> key to change the active window from the File to EPROM window to the directory window.
- Use the <UP> or <DOWN> keys to move the highlighted bar to the desired file or subdirectory.
- Press the <ENTER> key to select the desired file or subdirectory.
- The active window will be automatically changed back to the File to EPROM window, and the main program will display the selected name on the "File Name" line.
- If the subdirectory name is selected, the directory window will list all the files of that subdirectory for the next selection.

After the file name has been entered, the state of the 8 sockets will be displayed on the screen and the main program is ready to take further instruction. Press Y, and the file will be programmed to EPROMs on the sockets.

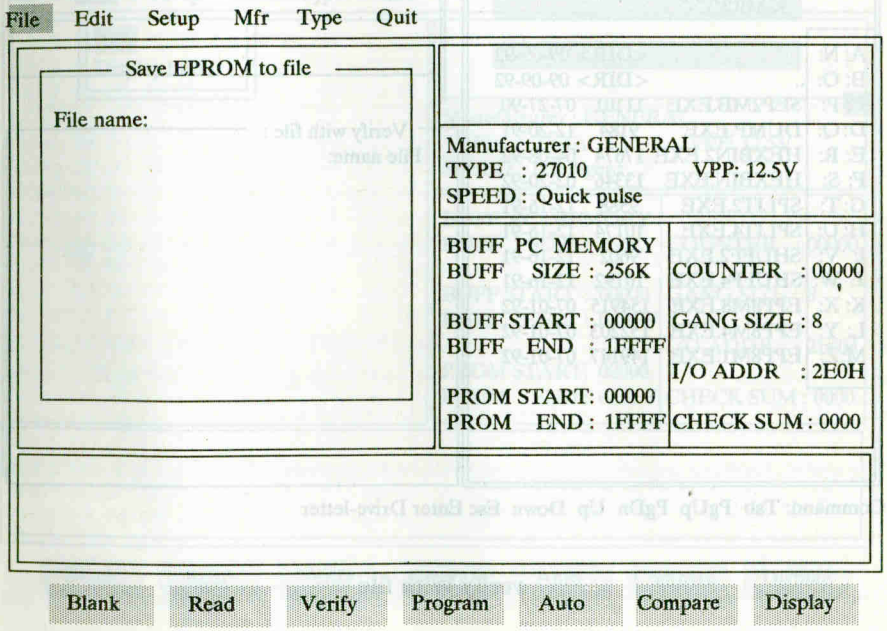
Press <CR> to return to the main menu.

NOTE

This function is similar to the Load file function. It enables the user to conveniently program to EPROM a file from 1M to 2M in size without going into the limited memory buffer.

4.1.6 EPROM to file (for 1M - 2M)

Enter **EPROM to file** (for 1M - 2M), press **E** from the File subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. A Save EPROM to file subwindow will appear on the left column of the main menu.



EPROM to file

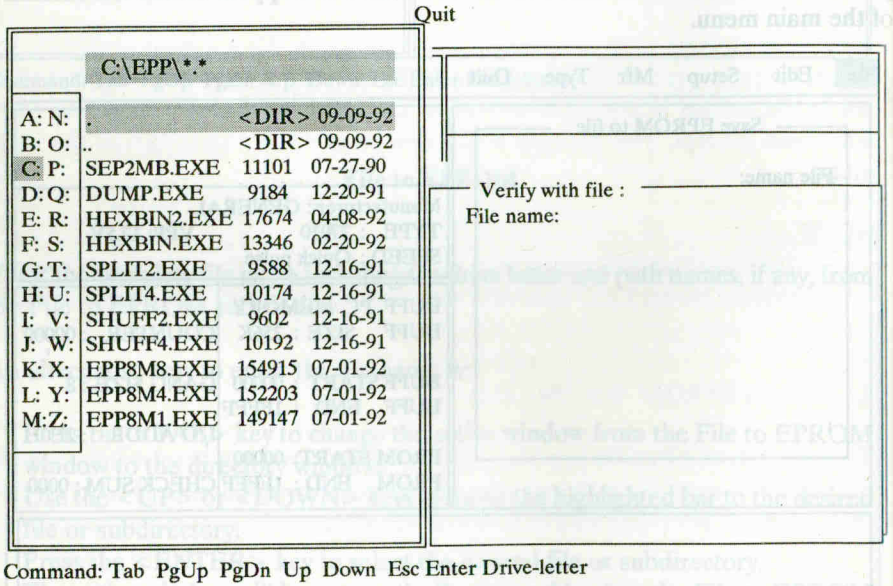
Key in the file name to be saved from EPROM to file and its end address. The EPROM data will be saved from address 00000 in the file.

NOTE

This is a function similar to Save buffer to file function. It enables the user to directly read the EPROM from 1M to 2M in size, and then save it to a file without going into the limited buffer.

4.1.7 verify With file (for 1M - 2M)

Enter **verify With file** (for 1M - 2M), press **w** from the File subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. A verify With file subwindow will appear on the left column of the main menu.



verify With file

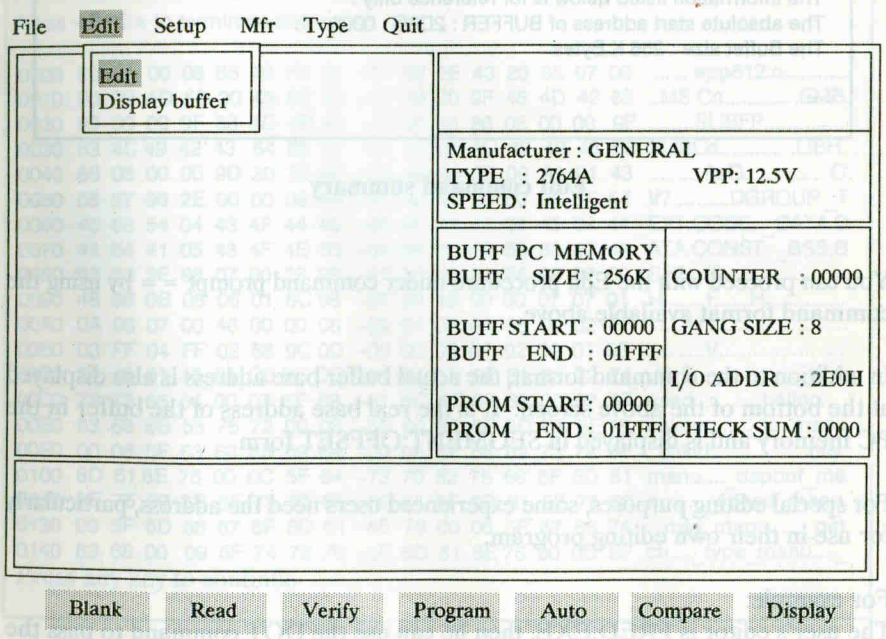
Key in the intended file name, the EPROM data will be verified with the file. The alternative way to enter the file name is the same as that mentioned in Section 4.1.5.

NOTE

This function is similar to the Load file function. It enables the user to directly verify EPROM data from 1M to 2M in size with the file without going into the limited buffer.

4.2 Edit

To enter **Edit** function group, press **E** from the main menu or use the left/right arrow keys to move the highlighted bar to Edit and then press <CR>. The Edit submenu will appear on the screen.



Subwindow of Edit

After entering the subfunction, select E again; The screen will display the Edit command summary.

```
                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 cahracters
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 : 2D8C : 0000
  The Buffer size : 256 K Bytes

= =
```

Edit command summary

You can proceed with the Edit procedure under command prompt = = by using the command format available above.

In addition to the command format, the actual buffer base address is also displayed at the bottom of the above screen. It is the real base address of the buffer in the PC memory and is displayed in SEGMENT:OFFSET form.

For special editing purposes, some experienced users need the address, particularly for use in their own editing program.

For example:
The user's editor is FRED.EXE, then he can use the DOT command to pass the base address:

- FRED SEGMENT: OFFSET <CR>

The user may move this parameter from the command line to his own program in the same manner as executing his program under DOS command prompt:

A>FRED SEGMENT: OFFSET <CR>

Type Q and <CR> to return to the main menu.

4.2.2 Display buffer

To enter **Display buffer**, press **D** from the Edit submenu or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. Once you have entered the function, the buffer contents between the range from buffer start to buffer end address shown in the status field will be displayed on the screen, in binary format, as follows:

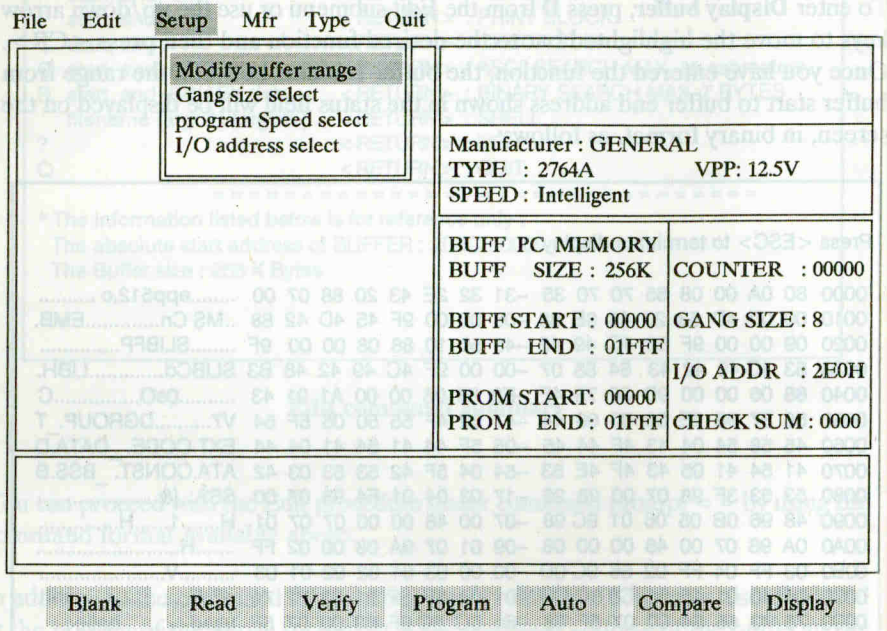
```
Press <ESC> to terminate display

0000  80 0A 00 08 65 70 70 35 --31 32 2E 43 20 88 07 00  ....epp512.c ....
0010  00 00 4D 53 20 43 6E 88 --05 00 00 9F 45 4D 42 88  ..MS Cn.....EMB.
0020  09 00 00 9F 53 4C 49 42 --46 50 10 88 08 00 00 9F  ....SLIBFP.....
0030  53 4C 49 42 43  64 88 07 --00 00 9F 4C 49 42 48 B3 SLIBCd.....LIBH.
0040  88 06 00 00 9D 30 73 4F --E3 88 06 00 00 A1 01 43  .....0sO.....C
0050  56 37 96 2E 00 00 06 44 --47 52 4F 55 50 05 5F 54  V7.....DGROUP._T
0060  45 58 54 04 43 4F 44 45 --05 5F 44 41 54 41 04 44  EXT.CODE._DATA.D
0070  41 54 41 05 43 4F 4E 53 --54 04 5F 42 53 53 03 42  ATA.CONST._BSS.B
0080  53 53 3F 98 07 00 28 26 --17 03 04 01 F4 98 07 00  SS?...(&........
0090  48 96 0B 05 06 01 6C 98 --07 00 48 00 00 07 07 01  H.....1.....H.....
00A0  0A 98 07 00 48 00 00 08 --09 01 07 9A 08 00 02 FF  .....H.....
00B0  03 FF 04 FF 02 56 9C 0D --00 00 03 01 02 02 01 03  .....V.....
00C0  04 40 01 45 01 C0 8C CC --01 0A 5F 5F 61 63 72 74  .@.E.....__acrt
00D0  75 73 65 64 00 07 5F 68 --61 6C 6C 6F 63 00 07 5F  used....._halloc._
00E0  63 68 6B 63 75 72 00 08 --5F 69 6F 5F 6D 61 6E 75  chkcur....._io_manu
00F0  00 06 5F 63 68 6B 69 64 --00 0A 5F 68 69 73 74 5F  ..chkid.....hist
0100  6D 61 6E 75 00 0C 5F 64 --73 70 62 75 66 5F 6D 61  manu...._dspbuf_ma
0110  6E 75 00 0B 5F 73 70 65 --65 64 5F 6D 61 6E 75 00  nu....._speed_manu.
0130  09 5F 6D 66 67 5F 6D 61 --6E 75 00 06 5F 67 65 74  ._mfg_manu....._get
0140  63 68 00  09 5F 74 79 70 --5F 6D 61 6E 75 00 0D 5F  ch...._type_manu...._
Press any key to continue
```

Display buffer

Press ^S to hold the display or press <ESC> to terminate the display. Then press any key to return to the main menu.

4.3 Setup

To enter the **Setup group function**, press **S** from the main menu or use the left/right arrow keys to move the highlighted bar to the desired function and then press <CR>. The Setup subwindow will appear on the screen.

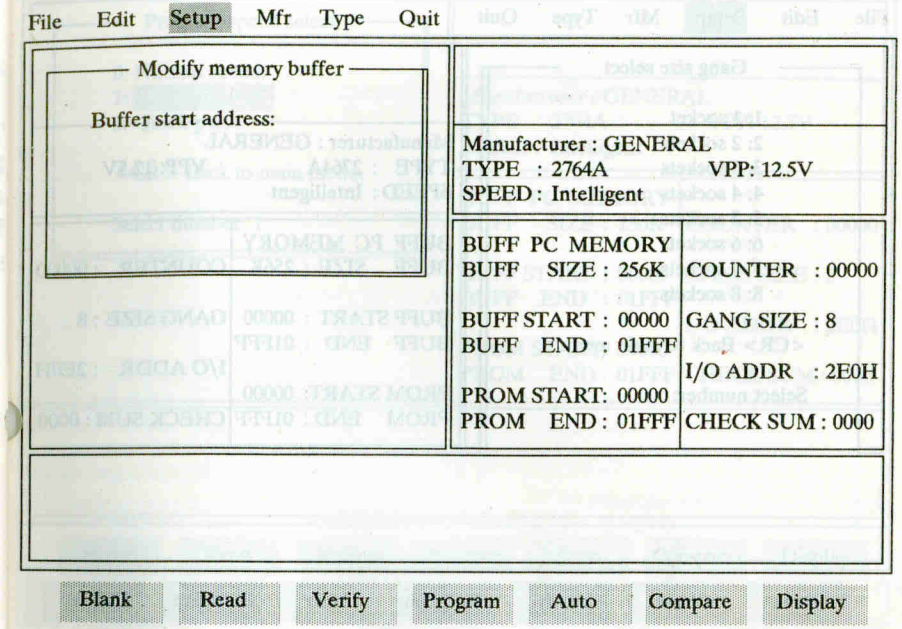


Subwindow of Setup

The 4 subfunctions under Setup are Modify buffer, Gang size, Speed select and I/O address.

4.3.1 Modify buffer

To enter **Modify buffer**, press **M** from the Setup subwindow or use the up/down keys to move the highlighted bar to the desired function and then press <CR>. A dialog window for Modify buffer will appear on the screen.



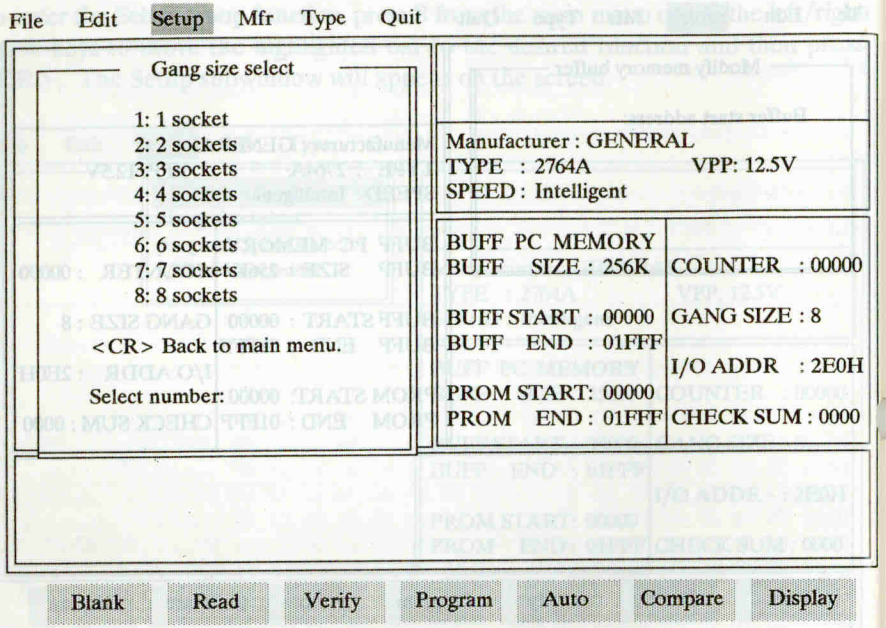
Modify buffer

The prompted screen requires the user to enter the revised buffer start address and press <CR>. Then enter the revised buffer end address and press <CR>. Finally enter the revised EPROM start address and press <CR>. All the values entered in HEX code will be updated in the status field for programming reference.

Press any key to return to the main menu.

4.3.2 Gang size

To enter **Gang size**, press **G** from the Setup subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. The gang size selection window will appear on the screen.



Gang size

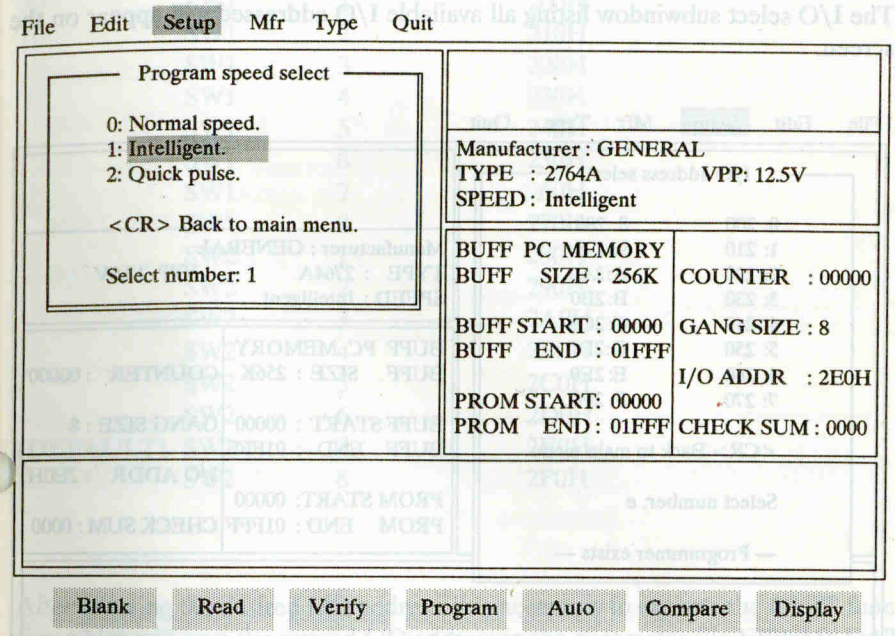
As the subwindow prompted, type the number of sockets to be worked on. This number will also be updated in the status field for programming reference.

Press <ESC> or <CR> to return to the main menu.

NOTE

The user does not have to select the gang size for the one-socket programmer module, because its gang size has already been fixed at 1.

To enter **Speed select**, press **S** from the Setup subwindow or or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. The Speed select subwindow will appear on the screen.



Speed select

When prompted, enter the number which precedes the speed you wish to work with. The selected speed will be updated in the status field for programming reference.

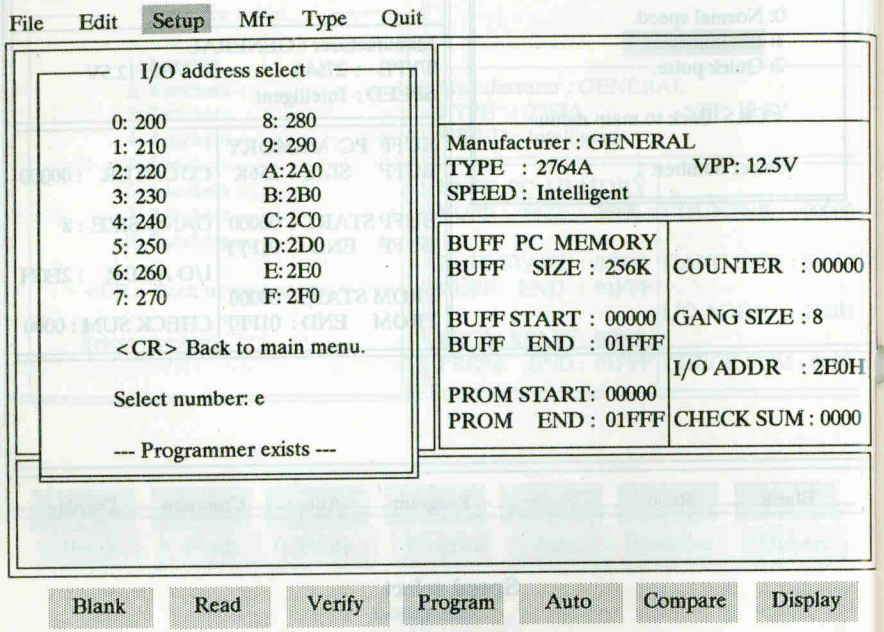
Press <ESC> or <CR> to return to the main menu.

NOTE

Most EPROMs have to apply the programming speed (Algorithm) as listed in the data sheet. Generally, they are grouped into Normal speed, Intelligent and quick pulse.

4.3.4 I/O select

To enter **I/O select**, press **I** from the Setup subwindow or use the up/down arrow keys to move the highlighted bar to the desired function and then press <CR>. The I/O select subwindow listing all available I/O addressed will appear on the screen.



I/O select

To select the I/O address set one of the DIP switches on the adaptor card in the on position. Only one position on the adapter card should be selected and turned on. The DIP switch positions correspond to I/O addresses as follows:

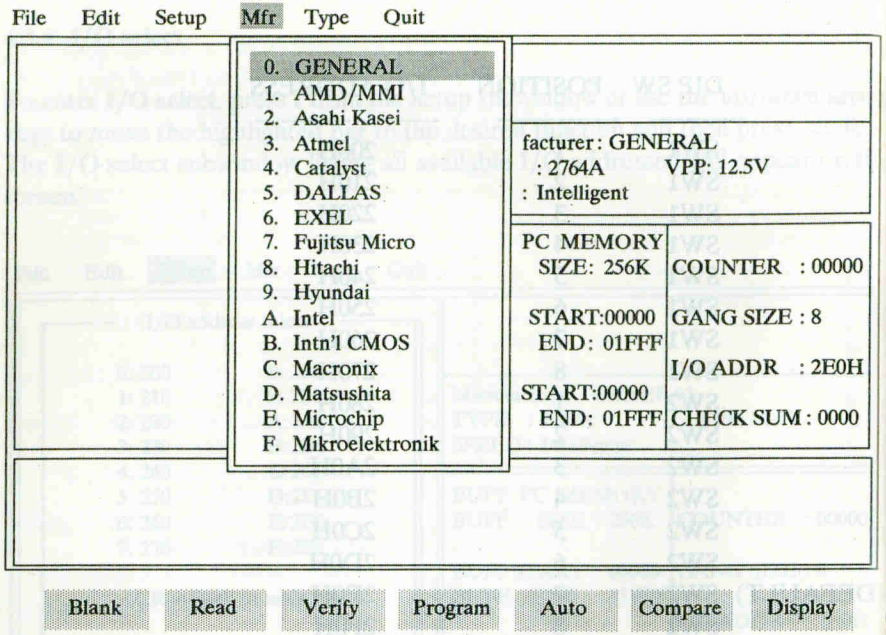
	DIP SW	POSITION	I/O ADDRESS
	SW1	1	200H
	SW1	2	210H
	SW1	3	220H
	SW1	4	230H
	SW1	5	240H
	SW1	6	250H
	SW1	7	260H
	SW1	8	270H
	SW2	1	280H
	SW2	2	290H
	SW2	3	2A0H
	SW2	4	2B0H
	SW2	5	2C0H
	SW2	6	2D0H
(DEFAULT)	SW2	7	2E0H
	SW2	8	2F0H

After entering the desired I/O address, it is necessary to execute the QUIT function which will save the revised I/O address in the parameter file EPP8Mx.DAT for later use.

Press <ESC> or <CR> to return to the main menu.

4.4 Mfr.

To enter **Mfr.** (Manufacturer), press **M** from the main menu or use the left/right arrow keys to move the highlighted bar to the desired function and then press <CR>. The Mfr. subwindow with currently supported manufacturers will appear on the screen.

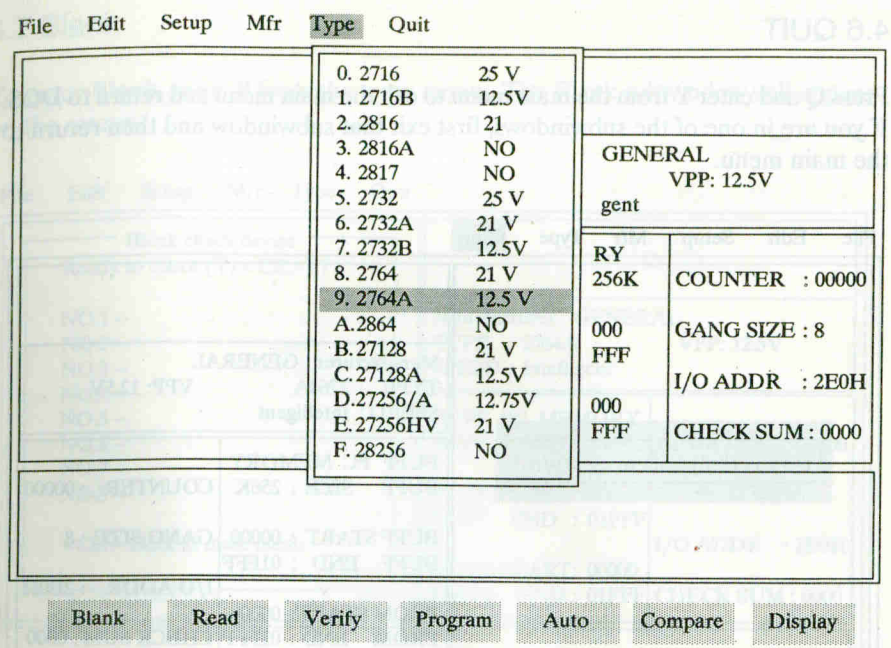


Subwindow of Manufacturer

Enter the number or letter which indicates a manufacturer corresponding to that of your EPROM. The selected manufacturer will then be updated in the status field for programming reference and a subwindow (as illustrated in Section 4.5) under function type will be displayed on the screen.

4.5 Type

To enter **Type**, press **T** from the main menu or use the left/right arrow keys to move the highlighted bar to Type and then press <CR>. The Type subwindow will appear on the screen.



Subwindow of Type

Select the number or letter which precedes the type corresponding to your EPROM type. The selected type and voltage will be automatically updated in the status field for programming reference.

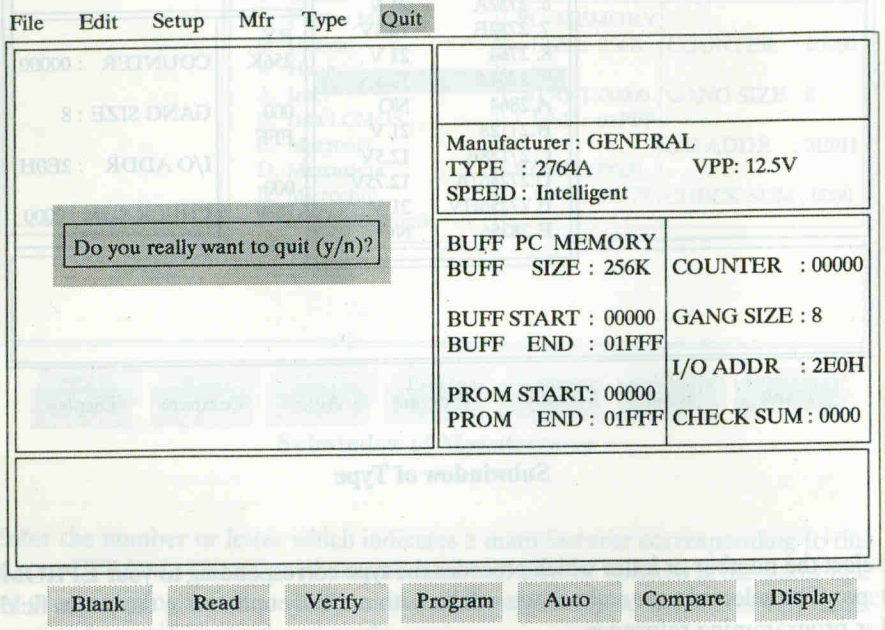
Press <ESC> or <CR> to return to the main menu.

NOTE

If the EPROM is not clearly labeled, it may be difficult to distinguish a EPROM 2732 requiring 25 V from a EPROM 2732A reguiring 21 V. In this case, the user should try to program the EPROM with the lower voltage first. If it does not work, then erase the EPROM and try the higher voltage.

4.6 QUIT

Press **Q** and enter **Y** from the main menu to exit the main menu and return to DOS. If you are in one of the subwindows, first exit that subwindow and then return to the main menu.

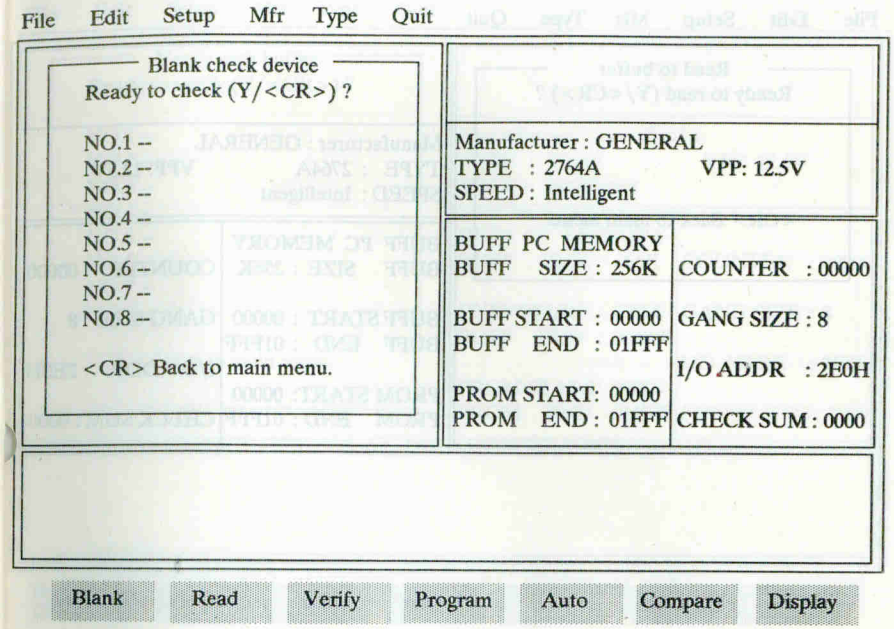


Subwindow of Quit

Before leaving for DOS, the main program will save all parameters including manufacturer, type, I/O address and programming speed in the parameter file EPP8Mx.DAT for later use.

4.7 Blank

To enter **Blank**, press **B** from the main menu. The Blank subwindow will appear on the screen.

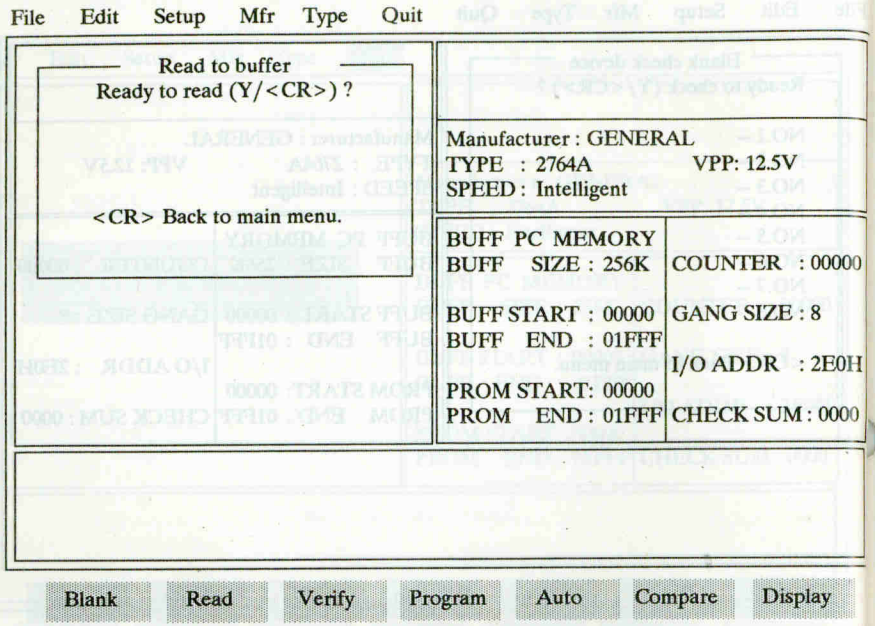


Subwindow of Blank

Press Y to start the blank check, or press <CR> to return to the main menu. If a chip fails the test, the first error address will be displayed, and the programmer module will continue to test the rest. If all pass, a "Blank check OK!" message will be displayed on the screen.

4.8 Read

To enter **Read**, press **R** from the main menu. The Read subwindow will appea on the screen.



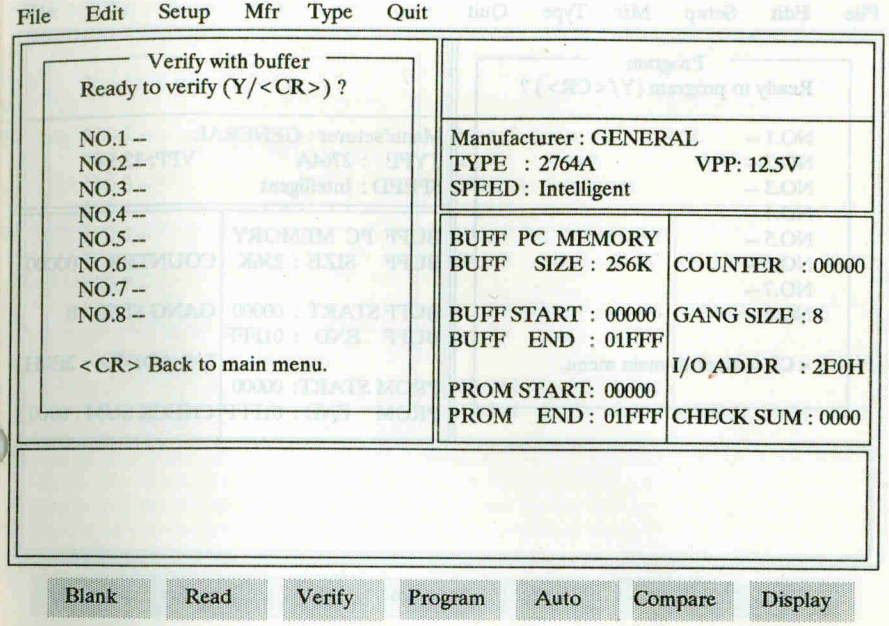
Subwindow of Read

Press Y to activate the function of reading data from the master EPROM to th buffer shown on the status field, or press <CR> to return to the main menu. Th subwindow will show a "Reading now..." message during processing. After it h been completed, the "Read OK!" message will appear.

The check sum calculated during the processing will also be displayed in the stat field after device reading.

4.9 Verify

To enter **Verify**, press **V** from the main menu. The Verify subwindow will appear on the screen.



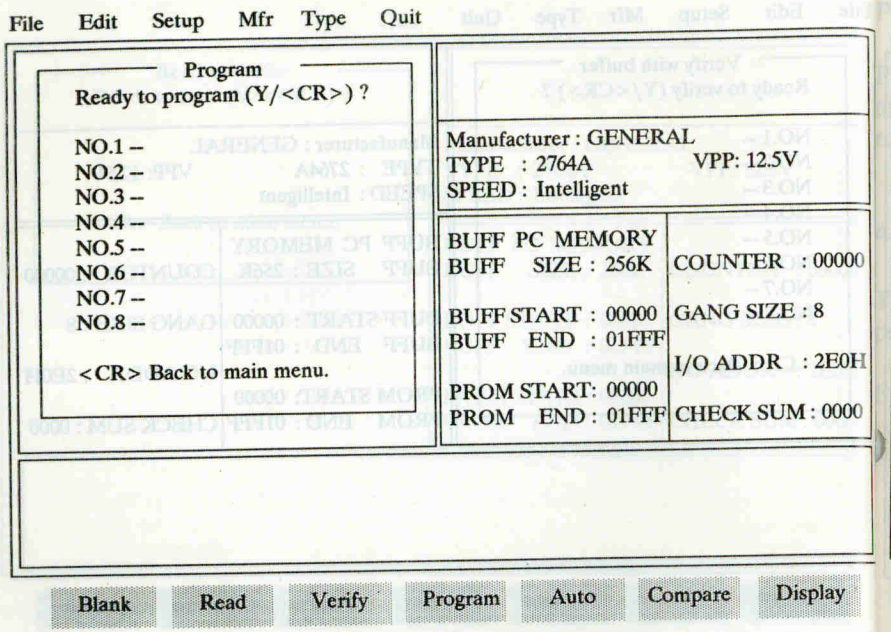
Subwindow of Verify

Press Y to verify the EPROMs with the buffer shown on the status field, or press <CR> to return to the main menu. During the operation, a "Verifying now" message will appear on the screen. Once it has been completed, the "Verify OK!" result will appear next to its associated socket number.

The Verifying routine will be terminated if the operation is error free.

4.10 Program

To enter **Program**, press **P** from the main menu. The Program subwindow wi appear on the screen.



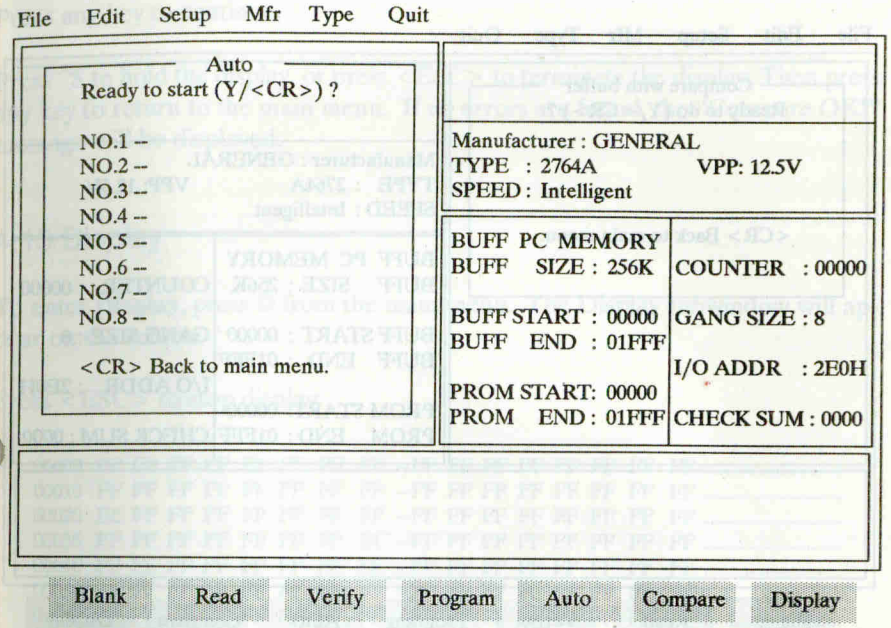
Subwindow of Program

Enter Y to activate the programming from the memory buffer to the EPROMs, press <CR> to return to the main menu. A programming message as well as scrolling counter number will simultaneously appear on the status field. If t transfer has been completed, a "Program OK!" message will appear next to i associated socket number.

Auto verify the programed data after the function has been completed.

4.11 Auto

To enter **Auto**, press **A** from the main menu. The Auto subwindow will appear on the screen.

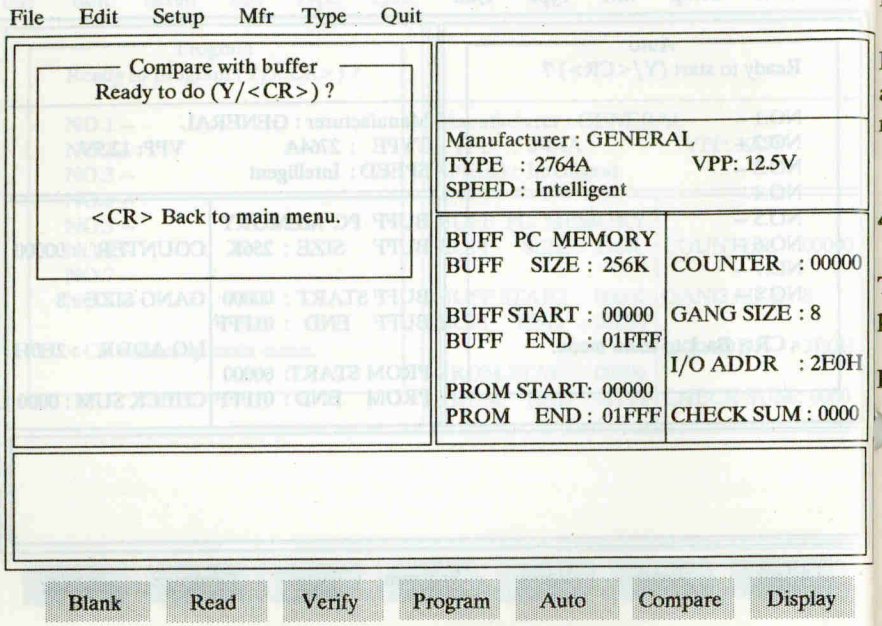


Subwindow of Auto

Press Y to activate the Auto function, or press <CR> to return to the main menu. This function is similar to the program function but it automatically blank checks before programming and verifies afterwards.

4.12 Compare

To enter **Compare**, press **C** from the main menu. The Compare subwindow will appear on the screen.



Subwindow of Compare

The programmer module will compare the buffer shown on the status field with the device in the master socket.

Press Y to compare the data in the device with that in the memory buffer, or press <CR> to return to the main menu. The screen will display the differences in the following format:

Error at:
Press <ESC> to terminate display

00000:DA - (00000:21) , 00001:1F - (00001:05)

Press any key to continue.

Press ^S to hold the display, or press <ESC> to terminate the display. Then press any key to return to the main menu. If no errors are found, the "Compare OK!" message will be displayed.

4.13 Display

To enter **Display**, press **D** from the main menu. The Display subwindow will appear on the screen.

Press <ESC> to stop display

```
00000 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00010 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00020 EC FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00030 FF FF FF FF FF FF FF EC -FF FF FF FF FF FF FF FF ................
00040 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00050 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF EC ................
00060 FF FF FF EC FF FF FF FF -FF FF FF FF EC FF FF FF ................
00070 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00080 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00090 FF FF FF FF FF FF FF FF -FF FF FF FF EC FF FF FF ................
000A0 EC FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
000B0 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
000C0 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
000D0 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
000E0 EC FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
000F0 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00100 FF FF FF FF FF FF FF EC -FF FF FF FF FF FF FF FF ................
00110 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00120 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00130 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
00140 FF FF FF FF FF EC FF FF -FF FF FF FF FF FF FF FF ................
00150 FF FF FF FF FF FF FF FF -FF FF FF FF FF FF FF FF ................
```

Subwindow of Display

The screen displays the address and current contents of the master EPROM in the No. 1 socket. Press <ESC> to exit the display and press any key to return to the main menu.

APPENDIX A. UTILITY PROGRAM

A.1 HEX to BINARY Code Converter

The converter can convert a HEX format ASCII file into a RAW BINARY file. Some Assemblers or Compilers can generate a HEX format file from a user's source program, then transmit it to a stand-alone programmer or an In-Circuit Emulator through a RS-232C hardware interface.

This programmer module is not stand-alone. It is controlled by the CPU on the PC using a direct I/O control technique; it also shares the large memory resources of the PC. This programmer software will load all types of file without any converting during the file loading. It continuously loads the file content byte by byte into the software buffer for programming use.

This HEX converter will convert your HEX format file to an executable ROM code that is recognizable to your target CPU.

We assume that the user understands the differences between the HEX format code and the ROM format file.

There are 3 types of HEX format files that can be converted to BINARY files by HEXBIN.EXE with a maximum conversion size of 64K bytes.

1. INTEL HEX format
2. MOTOROLA S HEX format
3. TEKTRONIX HEX format (seldom used)

HEXBIN2.EXE can convert INTEL and MOTOROLA extended HEX format files to binary files. Its maximum conversion size is 256K bytes.

The starting address may be specified on HEXBIN2, and the leading garbage will be skipped to maintain the small size of the output binary file.

The input command under DOS command prompt is:

A>HEXBIN2 [HEX FILE NAME] [BIN FILE NAME] [HEX FORMAT]
[start address] <CR>

[ ] : Option <CR> : return key or enter key.

HEX FILE NAME and BIN FILE NAME are standard file names specified by DOS.

HEX FORMAT: **I** for INTEL HEX
M for MOTOROLA S HEX
T for TEKTRONIX HEX

Start address: HEXADECIMAL digit, this parameter does not need to be assigned in the HEXBIN converter.

1) In the INTEL extended HEX format, the start address represents the start segment address. All data located between the start segment x 16 and the end of the file will be picked up. The segment range is from 0 to F000.

2) In the MOTOROLA S HEX format, the start address represents the actual start address. All data located between the start address and the end of the file will be picked up. The address range is from 0 to FFFFFFFF.

For example:

A>HEXBIN2 DEMO.HEX DEMO.TSK I 1000 <CR>

HEXBIN2 will convert the HEX file DEMO.HEX to the BINARY file DEMO.TSK, using the INTEL HEX converting technique, and only data located after 10000H will be converted.

This program also 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>INTEL <M>MOTOROLA <T>TEKTRONIX) : I
SEGMENT ADDRESS : 1000

A.2 Dump BINARY File to Console

Most ROM code BINARY files cannot be displayed on the screen by a DOS TYPE command. DUMP.EXE can convert a BINARY file into HEXADECIMAL code and display the file on the console or printer. Even though it is meaningless, a designer may want or need to keep a paper copy for later use.

The input command under DOS command prompt is:

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

[ ] : Option <CR> : Return key or Enter key
[^P] : Ctrl + P

It will connect the PC with the printer; the data displayed on the screen will be output to the printer.

FILENAME: standard file name that is specified by DOS.

Start address: HEXADECIMAL digits start dumping from this address, range from 0 to FFFFF.

A.3 2-way, 4-way BINARY File Splitter

SPLIT2.EXE can split a 16-bit source file into two 8-bit files. One is the collection of data located on the LOW byte of a 16-bit file. The other is the collection of data located on the HI byte of a source file. This utility is 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 a 32-bit source file into four 8-bit files. The first file is the collection of data located on the 1st byte of a 32-bit file. The second file is the collection of data located on the 2nd byte of a 32-bit file. The third file is the

collection of data located on the 3rd byte of a 32-bit file. The fourth file is the col
lection of the data located on the 4th byte of a 32-bit file.

The input command under DOS command prompt is:

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

[ ] : Option <CR> : Return key or Enter key

Input file, output EVEN file, output ODD file, output 1st file, output 2nd file, outpu
3rd file and output 4th file are standard file names specified by DOS.

A.4 2-way, 4-way BINARY File Shuffler

SHUFF2.EXE can shuffle two 8-bit source files into a 16-bit file. The first 8-bit fil
will be collected to the LOW byte of the 16-bit file, and the second 8-bit file will b
collected to the HI byte of a 16-bit file.

SHUFF4.EXE can shuffle four 8-bit source files into a 32-bit file. The first 8-bit fil
will be collected to the 1st byte of the 32-bit file. The second 8-bit file will b
collected to the 2nd byte of the 32-bit file. The third 8-bit file will be collected t
the 3rd byte of the 32-bit file. The fourth 8-bit file will be collected to the 4th byt
of the 32-bit 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] <CR>

[ ] : Option <CR> : Return key or Enter key

Output file, input EVEN file, input ODD file, input 1st file, input 2nd file, input 3r
file and input 4th file are standard file names specified by DOS.

APPENDIX B. LIST OF SUPPORTED DEVICES

GENERAL

0. 2716	25 V	8. 2764	21 V	G. 27512	12.75V
1. 2716B	12.5V	9. 2764A	12.5V	H. 27512HV	21 V
2. 2816	21 V	A. 2864	NO	I. 27010	12.75V
3. 2816A	NO	B. 27128	21 V	J. 27020	12.75V
4. 2817	NO	C. 27128A	12.5V		
5. 2732	25 V	D. 27256/A	12.75V		
6. 2732A	21 V	E. 27256HV	21 V		
7. 2732B	12.5V	F. 28256	NO		

AMD/MMI

0. 2716	25 V	C. 2864A/AE	NO	O. 27C513	12.5V
1. 2716B	12.5V	D. 2864B/BE	NO	P. 28F512	12 V
2. 2817A	NO	E. 27128	21 V	Q. 27C010	12.75V
3. 2732	25 V	F. 27128A/AP	12.75V	R. 27H010	12.75V
4. 2732A	21 V	G. 27C128	12.75V	S. 27HB010	12.75V
5. 2732B	12.5V	H. 27256/P	12.75V	T. 27C100	12.75V
6. 9732	21 V	I. 27C256/P	12.75V	U. 28F010	12 V
7. 9732B	12.5V	J. 27H256	12.75V	V. 27C020	12.75V
8. 2764	21 V	K. 28C256	NO	W. 28F020	12 V
9. 2764A/AP	12.75V	L. 28F256	12 V		
A. 27C64	12.75V	M. 27512/P	12.5V		
B. 9764	21 V	N. 27C512/L/P	12.75V		

Asahi Kasei

0. 28C64	NO

Atmel

0. 28C16	NO	9. 27256	12.5V	I. 27C512R	13 V
1. 28HC16/L	NO	A. 27C256	12.5V	J. 27C513	12.5V
2. 28C17	NO	B. 27C256R	12 V	K. 27C513R	13 V
3. 28HC17	NO	C. 27HC256/L	12.5V	L. 27C010/L	13 V
4. 27HC64/L	12.5V	D. 28C256	NO	M. 27C011	13 V
5. 28C64	NO	E. 28HC256/L	NO	N. 29C010	NO
6. 28HC64/L	NO	F. 29C256	NO		
7. 28PC64	NO	G. 29C257	NO		
8. 27C128	12.5V	H. 27C512	12.5V		

Catalyst

0. 28C16A	NO	4. 27128A	12.5V	8. 27512	12.75V
1. 28C17A	NO	5. 27256	12.5V	9. 27010	12.75V
2. 2764A	12.5V	6. 27HC256	12.75V	A. 27HC010	12.75V
3. 28C64A	NO	7. 28C256	NO	B. 28F010	12 V

DALLAS

0. DS1220AB/AD	NO	2. DS1230Y/AB	NO
1. DS1225D/E	NO		

EXEL

0. 2816A	NO	2. 2817A	NO	4. 2865A	NO
1. 28C16A	NO	3. 2864A	NO		

Fujitsu Micro

0. 8516	25 V	7. 28C65	NO	E. 27C512	12.75V
1. 2732	25 V	8. 27128	21 V	F. 27C1000	12.5V
2. 2732A	21 V	9. 27C128	21 V	G. 27C1001	12.5V
3. 8532	25 V	A. 27256	12.5V	H. 27C1001PG	12.5V
4. 2764	21 V	B. 27C256	21 V	I. 27C2001	12.5V
5. 27C64	21 V	C. 27C256A	12.5V		
6. 28C64	NO	D. 27C256H	12.5V		

Hitachi

0. 462716	25 V	8. 4827128	21 V	G. 27C512G	12.75V
1. 462732	25 V	9. 27256	12.5V	H. 27C101/G	12.5V
2. 482732A	21 V	A. 27C256/H	12.5V	I. 27C101AG	12.5V
3. 27C64	21 V	B. 27C256G	12.75V	J. 27C101ATG	12.5V
4. 482764	21 V	C. 27C256AG	12.5V	K. 27C301G	12.5V
5. 58064	NO	D. 27C256HG	12.5V	L. 27C301AG	12.5V
6. 58C65	NO	E. 58C256	NO		
7.27128A/AG	12.5V	F. 27512G	12.75V		

Hyundai

0. 27C64A	12.5V

Intel

0. 2716	25 V	C. (P)87C64	12.75V	O. 27(C)513	12.75V
1. 2816	NO	D. 27128	21 V	P. 28F512	12 V
2. 2816A	NO	E. 27128A/B	12.5V	Q. 27010	12.75V
3. 2817A	NO	F. 27C128	12.75V	R. 27C010/A	12.75V
4. 2732	25 V	G. P27128A	12.5V	S. 27(C)011	12.75V
5. (P)2732A	21 V	H. 27256	12.75V	T. 27C100	12.75V
6. 2732B	12.5V	I. 27C256/A	12.75V	U. 28F010	12 V
7. 2764	21 V	J. P27(C)256	12.75V	V. 27C020	12.75V
8. 2764A	12.5V	K. 28F256/A	12 V	W. 28F020	12 V
9. (P)27C64	12.75V	L. 87C256	12.75V		
A.P2764A	12.75V	M. 87C/68C257	12.75V		
B. 2864A	NO	N. 27(C)512	12.75V		

Intn'l CMOS

0. 27CX256	12.75V	1. 27CX010	12.75V

Macronix

0. 27C256	12.75V

Matsushita

0. 2764	21 V	1. 27128	21 V

Microchip

0. 28C16/A	NO	6. 28C64/A	NO	C. 27LV256	13 V
1. 28HC16	NO	7. 28CP64	NO	D. 28C256	NO
2. 28C17/A	NO	8. 27C128	13 V	E. 28CP256	NO
3. 28HC17	NO	9. 27256	12.75V	F. 27C512	13 V
4. 27C64	13 V	A. 27C256	13 V	G. 27LV512	13 V
5. 27HC64	13 V	B. 27HC256	13 V	H. 27C513/R	12.75V

Microelektronik

0. 2716	25 V	1. 2732	25 V

Mitsubishi Elec.

0. 2716	25 V	6. 27C128	21 V	C. 27C101	12.75V
1. 2732	25 V	7. 27256	12.5V	D. 28F101	12 V
2. 2732A	21 V	8. 27C256/A	12.75V	E. 27C201	12.75V
3. 2764	21 V	9. 27512	12.75V		
4. 28C64A	NO	A. 27C512A	12.75V		
5. 27128	21 V	B. 27C100	12.75V		

MXIC

0. 27C256	12.75V	1. 27C512	12.75V	2. 27C1000	12.75V

National

0. NMC2716	25 V	I. NMC27CP128	13 V
1. NMC27C16	25 V	J. NMC27C128C	12.75V
2. NMC27C16B	13 V	K. NM/NMC27C256	13 V
3. NMC27C16H	25 V	L. NMC27C256B	12.75V
4. NMC2816	NO	M. NMC27C256C	12.75V
5. NMC9816A	NO	N. NMC27CP256	12.5V
6. NMC9817/A	NO	O. NMC87C257	12.75V
7. NMC2732	25 V	P. NM27LC256	12.5V
8. NMC27C32	25 V	Q. NM/NMC27C512	13 V
9. NMC27C32B	12.75V	R. NMC27C512A	12.75V
A. NMC27C32H	25 V	S. NM27P512	12.75V
B. NMC27C64	13 V	T. NM27LC/LV512	12.75V
C. NMC27C64B	12.75V	U. NM/NMC27C010	12.75V
D. NMC27CP64	12.5V	V. NM27LV010	12.75V
E. NM27LC64	13 V	W. NMC271023	12.5V
F. NMC98C64	NO	X. NMC27C020	12.75V
G. NM/NMC27C128	13 V		
H. NMC27C128B	12.75V		

NEC

0. 2716	25 V	7. 27256	21 V	E. 27C1000/A	12.5V
1. 2732	25 V	8. 27256A	12.5V	F. 27C1001/A	12.5V
2. 2732A	21 V	9. 27C256	21 V	G. 27C2001	12.5V
3. 2764	21 V	A. 27C256A	12.5V		
4. 27C64	21 V	B. 28C256	NO		
5. 28C64	NO	C. 27512	12.5V		
6. 27128	21 V	D. 27C512	12.5V		

Oki

0. 2716	25 V	6. 2764A	12.5V	C. 27256	12.5V
1. 2816A	NO	7. 27C64	21 V	D. 27C256H	12.5V
2. 28C16A	NO	8. 28C64A	NO	E. 27512	12.5V
3. 2732	25 V	9. 27128	21 V	F. 271000	12.5V
4. 2732A	21 V	A. 27128A	12.5V	G. 27C1000	12.5V
5. 2764	21 V	B. 27C128	21 V	H. 27C2000	12.5V

Panasonic

0. 27C64A	12.75V	1. 27C256	12.75V	2. 27C512	12.75V

Ricoh

0. 27C32	21 V	2. 27C64	21 V	3. 27C256	12.75V
1. 5H32	21 V				

Rockwell

0. 2816A	NO	1. 87C32	21 V	2. 87C64	21 V

Samsung

0. 2816A	NO	4. 2864A	NO	8. 2865AH	NO
1. 28C16	NO	5. 2864AH	NO	9. 28C65	NO
2. 2817A	NO	6. 28C64	NO	A. 28C256	NO
3. 28C17	NO	7. 2865A	NO		

SEEQ

0. 2816A/AH	NO	7. 2764	21 V	E. 5143	21 V
1. 5516A/AH	NO	8. 5133	21 V	F. 27256	12.5V
2. 2817A/AH	NO	9. 2864/H	NO	G. 27C256	12.5V
3. 5517A/AH	NO	A. 28C64	NO	H. 28256	NO
4. 38C16	NO	B. 28C65	NO	I. 28C256/A	NO
5. 38C32	NO	C. 52B33/H	NO		
6. 52B13/H	NO	D. 27128	21 V		

SGS-Thomson

0. 2716	25 V	7. 27C64	12.5V	E. 87C257	12.5V
1. 27C16	25 V	8. 27C64A	12.5V	F. 27512	12.5V
2. 2732	25 V	9. 27128A	12.5V	G. 27C512	12.5V
3. 2732A	21 V	A. 27256	12.5V	H. 27C1000	12.5V
4. 27C32	25 V	B. 27C256	12.5V	I. 27C1001	12.5V
5. 2764	21 V	C. 27C256B	12.5V	J. 27C2001	12.5V
6. 2764A	12.5V	D. 28F256	12 V		

Sharp

0. 5762	12.5V	3. 57126	12.5V	6. 57254	12.5V
1. 5763	12.5V	4. 57127	12.5V	7. 57255	12.5V
2. 5764	12.5V	5. 57128	12.5V	8. 57256	12.5V

Signetics

0. 27C64	21 V	3. 27C256	12.5V	6. 27C010	12.5V
1. 27C64A	12.5V	4. 27C256F	12.5V		
2. 27C64AF	12.5V	5. 27C512	12.5V		

SMOS

0. 27C64H	21 V	1. 27128H	21 V	2. 27C256H	12.5V

SONY

0. CXK27C256DQ	12.75V	2. CXK27C1000DQ	12.75V
1. CXK27C512DQ	12.75V	3. CXK27C1001DQ	12.75V

TI

0. 2732	25 V	B. 27C128	12.5V	M. 27PC512	12.5V
1. 2732A	21 V	C. 27PC128	12.5V	N. 29F512	NO
2. 27P32A	21 V	D. 27256	12.5V	O. 27C010A	12.5V
3. 27C32	12.5V	E. 27C256	12.5V	P. 27PC010A	12.5V
4. 27PC32	12.5V	F. 27PC256	12.5V	Q. 28F010	12 V
5. 2764	21 V	G. 29F256	NO	R. 29F010	NO
6. 27P64	21 V	H. 29F258	NO	S. 27C020	12.5V
7. 27C64	12.5V	I. 29F259	NO		
8. 27PC64	12.5V	J. 87C257	12.75V		
9. 28C64	NO	K. 27512	12.5V		
A. 27128/A	21 V	L. 27C512	12.5V		

Toshiba

0. 2732A	21 V	C. 24256A	12.5V	O. 27512A	12.5V
1. 2732D	25 V	D. 27256A	12.5V	P. 57512A	12.5V
2. 2464	21 V	E. 27256B	12.5V	Q. 541000	12.5V
3. 2464A	12.5V	F. 27256(D)	21 V	R. 571000/A	12.5V
4. 2764A	12.5V	G. 54256	21 V	S. 57H1000A	12.5V
5. 2764D	21 V	H. 54256A	12.5V	T. 541001	12.5V
6. 24128	21 V	I. 57256	21 V	U. 571001/A	12.5V
7. 27128	21 V	J. 57256A	12.5V	V. 57H1001A	12.5V
8. 24128A	12.5V	K. 57H256	12.5V		
9. 27128A	12.5V	L. 24512	12.5V		
A. 27128D	21 V	M. 24512A	12.5V		
B. 24256	12.5V	N. 27512	12.5V		

VLSI

0. 27C64	12.5V	2. 27C128	12.5V	4. 27C512	12.5V
1. 28H64	NO	3. 27C256	12.5V		

Waferscale

0. 27C64	12.5V	5. 57C128F	12.5V	A. 57C256F	12.5V
1. 27C64F	12.5V	6. 57C128FB	12.5V	B. 27C512L	12.5V
2. 57C64F	12.5V	7. 27C256	12.5V	C. 27C512F	12.5V
3. 27C128F	12.5V	8. 27C256F	12.5V	D. 27C010L	12.5V
4. 57C128	12.5V	9. 27C256L	12.5V	E. 27C010R	12.5V

Xicor

0. 2816A	NO	4. 2864B	NO	8. 28256	NO
1. 2816B	NO	5. 2864H	NO	9. 28C256	NO
2. 2816C	NO	6. 28C64	NO	A. 28C256B	NO
3. 2864A	NO	7. 28C64B	NO		

APPENDIX C. TROUBLE SHOOTING

We have provided the following troubleshooting guide to help you overcome some of the more commonly experienced problems. This guide, however, is not intended to be a repair manual. If you encounter problems other than those described here, please contact your dealer or our Sales Department.

INSTALLATION PROBLEMS

PROBLEM 1 :
When I turn my computer on, I get no beeps, the fan doesn't spin, nothing happens!

RESOLUTION 1 :
1-1 The power cord may be disconnected from the computer or the wall. Check the power cable.

1-2 You may not have a chip correctly inserted in the ZIF socket. Make sure your chip is correctly installed and the handle is down.

1-3 Your power supply may not have sufficient power to drive both your system and the system adapter card.

PROBLEM 2 :
When I try to use a programmer module, I get communication error messages!

RESOLUTION 2 :
2-1 You may not have the I/O port set correctly for your programmer. Double check the I/O port assignment.

2-2 You may not have a chip correctly inserted in the ZIF socket. Make sure your chip is correctly installed and the handle is down.

2-3 There may not be good connection between the system adapter card and the programmer module. Double check the cable connection.

2-4 Your system may be running too fast. Try slowing your system down as much as possible, or try using an IBM AT-8 MHz or compatible.

2-5 The bus speed on your system may be too fast. The system adapter card will not run with bus speeds greater than 486 33 MHz.

PROBLEM 3 :
When I install the system adapter card, some of my other peripherals start behaving strangely!

RESOLUTION 3 :
3-1 You are probably experiencing an I/O port conflict. Double check the I/O port assignments on all your peripherals, including the system adapter card.

10 Things to do Before Calling Your Dealer

1. Reboot the computer and try again.
2. If you change switches or jumpers, write down the original settings.
3. Repeat all the steps, following the instructions in this manual.
4. Make sure all cards and cables are firmly attached.
5. Remove any memory resident programs from memory.
6. See if your problem is listed in the Trouble-Shooting section.
7. Try it on another system.
8. Compare system requirements with your configuration.
9. Ask your in-house "guru" (every office has one).
10. Ask whoever installed the product.

GENERAL TROUBLESHOOTING CHECKLIST

If your problem is not described above, check the following :

1. Is the system adapter card fully seated in its slot?
2. Are all cable connections securely fastened?
3. Does the system adapter card jumper setting match the I/O address displayed by the programmer software?
4. Does any other card on the bus have the same I/O address as the system adapter card?

APPENDIX D. GLOSSARY

The most commonly used terminologies are explained here so that the user may refer to them whenever necessary.

*** Memory Buffer**
The buffer is a block area of PC memory allocated by the main program through DOS. This buffer is used by the main program as an intermediate storage.

The main program can read the DEVICE contents to the buffer and save it onto a disk file or perform the reverse operation, that is, to load a disk file to the buffer and program it to the DEVICE. You have the advantage of manipulating the buffer contents at will. The modified buffer can always be saved to a disk file for future reference.

The minimum allocated size of the buffer is 64K bytes, and the maximum size is the maximum available memory in the PC.

Since the memory buffer is dynamically allocated through DOS, the actual base address on the PC may vary from system to system, software to software. The user need not refer to the actual base address of the memory buffer.

*** Buffer Start and Buffer End Addresses**
The buffer start and end addresses are offset addresses specified from the base address of the MEMORY BUFFER. This is the specified portion where its information can be programmed to the DEVICE and the DEVICE contents can be read onto it.

*** Device Start and Device End Addresses**
The start and end addresses are offset addresses specified for the DEVICE contents.

*** Check Sum**
This value is the sum of all data contents between buffer start and end addresses. This value will be calculated during the DEVICE reading, file loading, type changing or after buffer editing.

*** I/O Address**
This is the I/O base address of the system adapter card. Each I/O interface card added into a PC slot will occupy one or more I/O addresses. The default I/O base address of the system adapter card is 2E0 and occupies 4 contiguous spaces (2E0 to 2E3).

*** Counter**
This is the programming address counter. During the DEVICE programming, the counter value will be increasingly displayed on the screen.

*** Mfr, Type, VPP, Speed**
Every DEVICE has its own manufacturer (MFR), type number (TYPE), programming voltage (VPP) and programming speed (SPEED or algorithm). Please refer to APPENDIX B for the details for each EPROM supported.

MMAUEPP01A.P9/1992